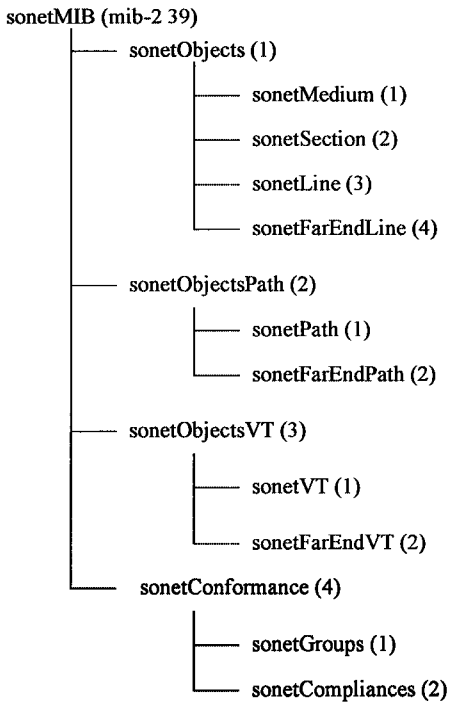


Figure 10-6
SONET/SDH MIB (RFC 1595).

- *sonetMediumCircuitIdentifier:* This indicates the transmission vendor's circuit identifier.

- *sonetSection:* These objects contain data on the SONET/SDH section layer. sonetSection includes the SONET/SDH current table and interval tables. The measurements given here for the sonet current entry table refer to the current 15-min interval. Their explanations are as follows:
  - *sonetSectionCurrentStatus:* This indicates the status of the SONET/SDH interface. Current status represents the cumulative value of defects. sonetSectionDefect has a value of 1, sonetSectionLOS has a value of 2, and sonetSectionLOF has a value of 4. LOS means loss of signal defect and LOF means loss of frame defect.
  - *sonetSectionCurrentESs:* This is a counter that refers to the number of errored seconds.
  - *sonetSectionSESs:* This indicates the number of severely errored seconds.
  - *sonetSectionCurrentSEFs:* This refers to the number of severely errored framing seconds.

  sonetSectionIntervalTable includes measurements taken during previous 15-min intervals. The table can have previous data from 4 to a

maximum of 96 completed intervals. The default number of intervals is 32. The parameters of this table are as follows:

- *sonetSectionIntervalNumber:* This is an integer from 1 to 96.
- *sonetSectionIntervalESs:* This indicates the number of errored seconds.
- *sonetSectionIntervalSESs:* This indicates the number of severely errored seconds.
- *sonetSectionIntervalSEFs:* This indicates the severely errored framing seconds.
- *sonetSectionIntervalCVs:* This counts the coding violations.

- *sonetLine:* This is similar to the sonetSection. The data for the SONET line layer are collected here. It also contains a current table and an interval table. The measurements are done in 15-min intervals. The parameters included in the sonetLineCurrentTable are as follows:
  - *sonetLineCurrentStatus:* This is a sum of all the defects in the line.
  - *sonetLineCurrentESs:* This indicates the number of errored seconds.
  - *sonetLineCurrentSESs:* This indicates the number of severely errored seconds.
  - *sonetLineCurrentCVs:* This indicates the number of coding violations.
  - *sonetLineCurrentUASs:* This refers to the number of unavailable seconds.

  sonetLineIntervalTable includes the previous 15-min intervals for which measurements are available. The parameters of the sonetLineIntervalEntry are sonetLineIntervalNumber, sonetIntervalESs, sonetLineIntervalSESs, sonetLineIntervalCVs, and sonetLineIntervalUASs. As the explanations of these terms are similar to those in the sonetLineCurrentEntry, we will not repeat them here.

- *sonetFarEndLine:* This is valid only for far-end block error at the SONET/SDH line layer. sonetFarEndLine includes current and interval tables. The parameters used in the current and interval tables are similar to those used in sonetLineCurrentEntry and sonetLineIntervalTable.

- *sonetPath:* This also has current and interval tables. These tables contain measurements on the SONET/SDH path layers. The values are for 15-min intervals. The parameters included in the current entry table are as follows:
  - *sonetPathCurrentWidth:* This indicates the type of SONET/SDH path. For SONET the values of N are 1, 3, 12, 24, and 48. For SDH the values are 1, 4, and 16.
  - *sonetPathCurrentStatus:* This is the sum of the defects in the interface.
  - *sonetPathCurrentESs:* This indicates the number of errored seconds.
  - *sonetPathCurrentSESs:* This includes measurements on the severely errored seconds.

- *sonetPathCurrentCVs:* This contains measurements on coding violations.
- *sonetPathCurrentUASs:* This indicates the number of unavailable seconds.

Sonet interval table parameters are similar to the current table parameters, and they include the values taken in the previous 15-min intervals.

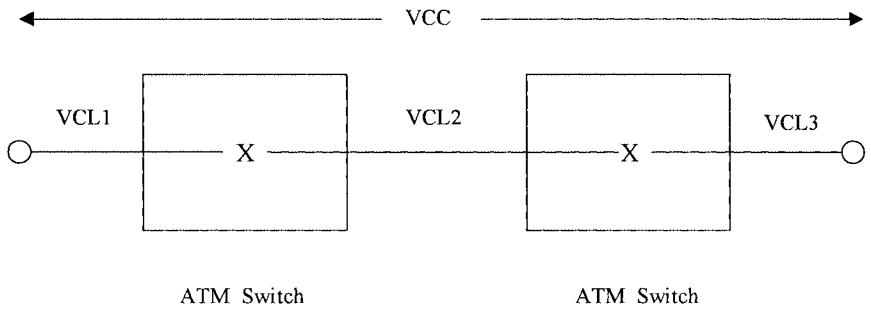
- *sonetFarEndPath:* This is similar to sonetFarEndLine. It is valid for far-end block error code. sonetFarEndPath has also current and interval tables.
- *sonetVT:* This includes statistics on virtual tributaries for SONET and virtual channels for SDH. There are also current and interval tables. Statistics on errored seconds, severely errored seconds, code violations, and unavailable seconds are included.
- *sonetFarEndVT:* This includes statistics collected for the far-end block error code. There are also current and interval tables for statistics.
- *sonetConformance:* This includes sonetGroups and sonetCompliances groups. These groups are collections of objects that include information on SONET/SDH virtual tributaries, section, line, and path interfaces, and far-end information where applicable. This information is used as a benchmark for checking whether the SONET/SDH interfaces meet the requirements.

The managed objects defined in RFC 1595 can be used by SNMPprotocols.

10.5 Operation and Maintenance (OAM)

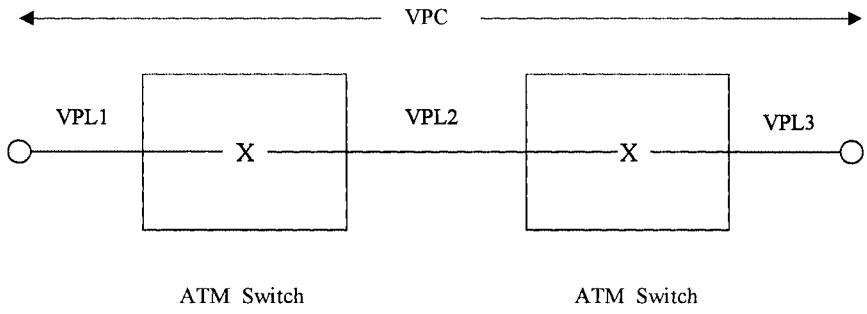
Before we look into the OAM aspects of the ATM layer and the physical layer, it is essential to introduce the commonly used concepts of *virtual channel connection* (VCC) and *virtual path connection* (VPC). A VCC is a connection that forms a path between two end points. The VCC is the result of the concatenation of virtual channel links (VCLs) and this concatenation occurs at an ATM switch, as shown in Figure 10-7. The VCC is used for information transfer between user to user, user to network, and network to network. A virtual channel identifier (VCI) is unique across a VC link and identifies a VC link. Just like VCCs, VPCs are formed over virtual path links (VPLs). A virtual path identifier (VPI) identifies a VP link.

Figure 10-7
ATM virtual channel and virtual path connections.



X - ATM Cross-Connect Point

Virtual Channel Links and Virtual Channel Connection



X - ATM Cross-Connect Point

Virtual Path Links and Virtual Path Connection

ITU-T Recommendation I.610 (Reference 10.2) describes the operation and maintenance (OAM) functions at the physical and ATM layers. OAM functions are performed at five hierarchical levels, namely F1, F2, F3, F4, and F5 (Figure 10-8), and they result in corresponding bidirectional information flows. Levels F1, F2, and F3 are associated with the physical layer, and levels F4 and F5 are for ATM layer. OAM flows at the ATM and physical layers are shown in Figure 10-9. Explanations of the five information flows are as follows:

- *F5:* Information flow extends between the NEs responsible for virtual channel connection functions. There are two types of F5 flows: *end-to-end* F5 flow is used for end-to-end VCC operations; *segment* F5 flow communicates OAM information within one VCC link or interconnected VCC links. A VCC may have one or more OAM segments. F5 is used for continuity checks on the active VCCs per interface at

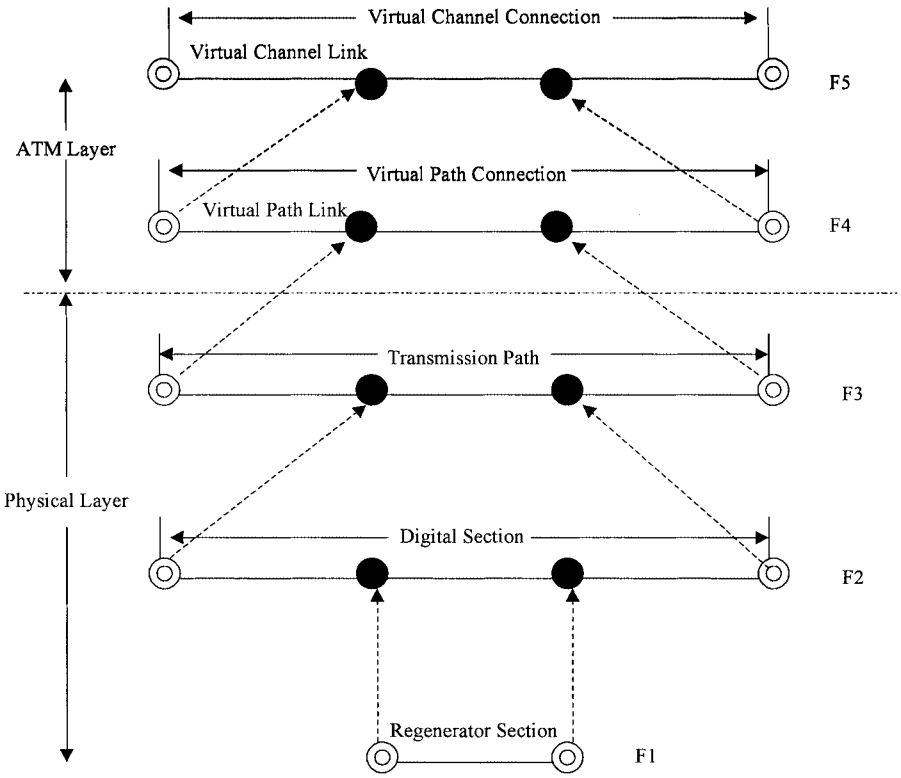


Figure 10-8
Relationship between OAM levels and ATM layers.

the end-to-end or segment level. F5 flows are also used for fault and performance management of VCCs.

- *F4:* This refers to the bidirectional information flow containing OAM cells at the virtual path level. Here also, the F4 information flow can be end-to-end or be restricted to one segment. F4 information flow can be used for fault and performance management of VPCs. F4 information flows are useful for detecting degradation of VP performance, loss or misinsertion of cells, and late-arriving cells.
- *F3:* Information flows are between the NEs that perform payload assembling and disassembling, cell delineation, and header error control (HEC) functions.
- *F2:* Information flows are between section end points.
- *F1:* Information flows are between regenerator sections. A regenerator section is a portion of a digital section.

The OAM flows depend upon the type of transmission used. As we have seen, ATM can use either SDH or SONET at the physical layer. F1, F2,

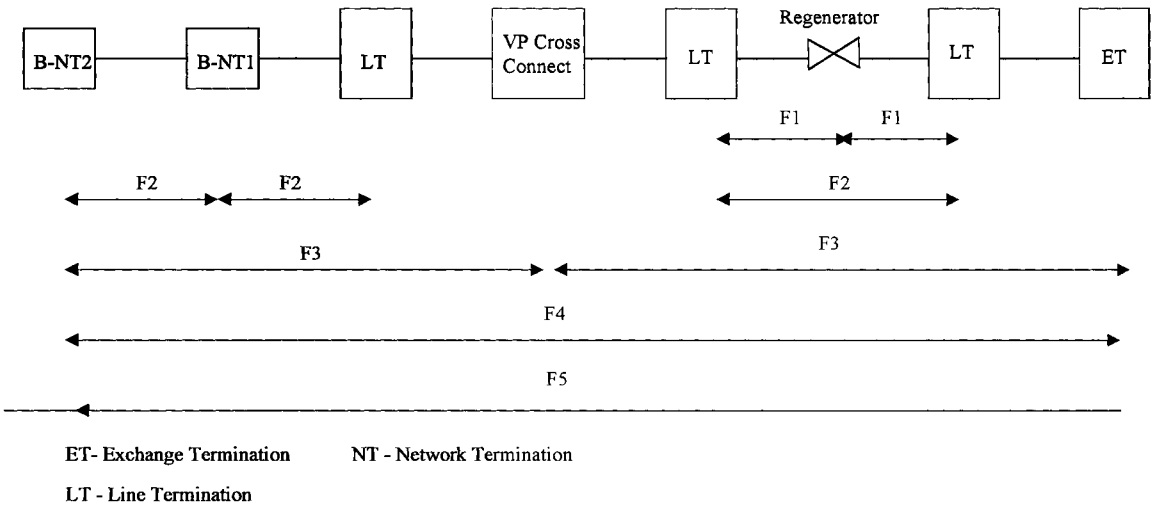


Figure 10-9
OAM flows at the physical and ATM layers.

and F3 flows help detect and report unavailability states, carry defect information on the affected end points, and help monitor and report equipment failures. These flows also aid in error monitoring and reporting at the regenerator section, multiplex section, and transmission path levels for SDH and SONET. Some of the errors reported can be loss of signal, loss of frames, degraded performance, loss of pointer or path, loss of cells, defective insertion, or suppression of cells.

Refer to Section 10.2 for more details on OAM for the physical and ATM layers.

10.6 ATM Network Management

ATM network management follows the standards developed by the ATM Forum and ITU-T. ITU-T Recommendation I.751 (Reference 10.3) describes management of the ATM NE view. In this recommendation, MOCs are grouped into ensembles for convenience and these ensembles perform certain specific management functions. ATM transport-specific ensembles are the following:

- *ATM Layer Management:* This includes configuration and fault management for transport path and virtual path adaptation, for the vir-

tual path layer, and for the virtual channel layer. Configuration of an ATM interface, allocation of bandwidth and VPI/VCI ranges associated with ATM interfaces, and detection and reporting of alarms within the ATM layer are also part of the layer management.

- *ATM VP/VC Connection Management:* This covers such functions as establishing and releasing VP and VC channel connections, allocating the virtual path identifier (VPI) and virtual channel identifier (VCI) for these connections, and managing the VP and VC channel connections.
- *ATM Performance Management:* This covers the monitoring of performance parameters such as gauges and counters for the VP and VC layers and transport paths.

10.6.1 ATM Forum Network Management

The ATM Forum network management model is specific to ATM networks. Primarily, ATM Forum defines a set of protocol-independent requirements for management services and MIBs for managing the ATM networks. The management protocols can be either CMIP or SNMP. The ATM Forum network management reference model (Figure 10-10) is similar to the TMN physical architecture. The reference model defines network management interfaces for managing ATM devices, private networks, and public networks. The ATM forum management interfaces are as follows:

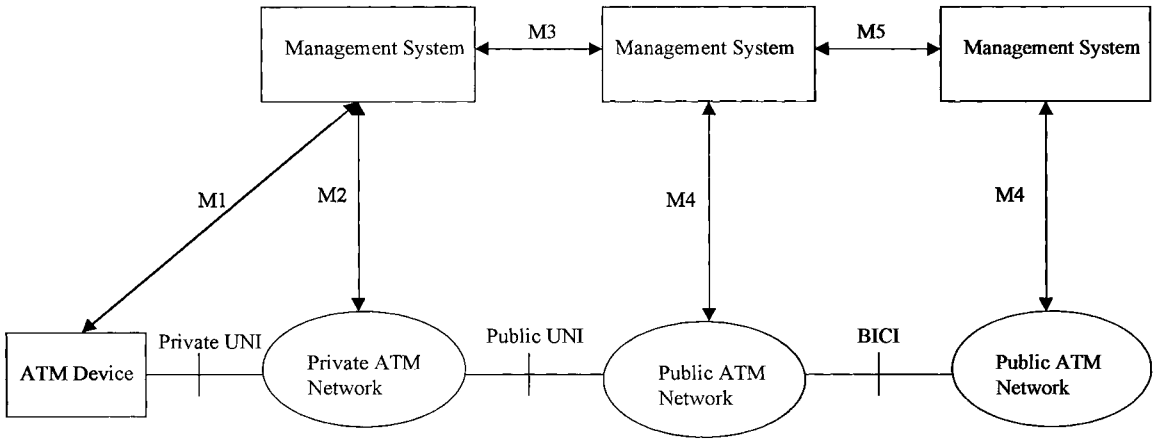


Figure 10-10
ATM Forum network management interfaces.

- *M1:* This is the interface for managing ATM terminal devices.
- *M2:* This is the interface for managing an ATM private network.
- *M3:* This is the customer network management interface. It provides management functions for the customer portion of an ATM network.
- *M4:* The OS or the managers manage individual network elements using the M4 interface. Network elements refer to ATM switches, ATM cross-connects, ATM concentrators, remote ATM switches, and distributed ATM switching systems. The M4 interface also supports element management and service management functions.
- *M5:* This is the network management interface between two network providers.

10.6.1.1 M3 Interface. The M3 interface (Figure 10-11) includes two classes of functions that users can access (Reference 10-13). Class 1 functions include monitoring configuration, fault, and performance management on the customer's portion of an ATM public network. Customer network management (CNM) must be able to retrieve the following:

- Details on the CNM agent and the protocols used for customers' user network interfaces (UNIs).
- Performance information on the ATM layers and physical layers of customers' UNIs.

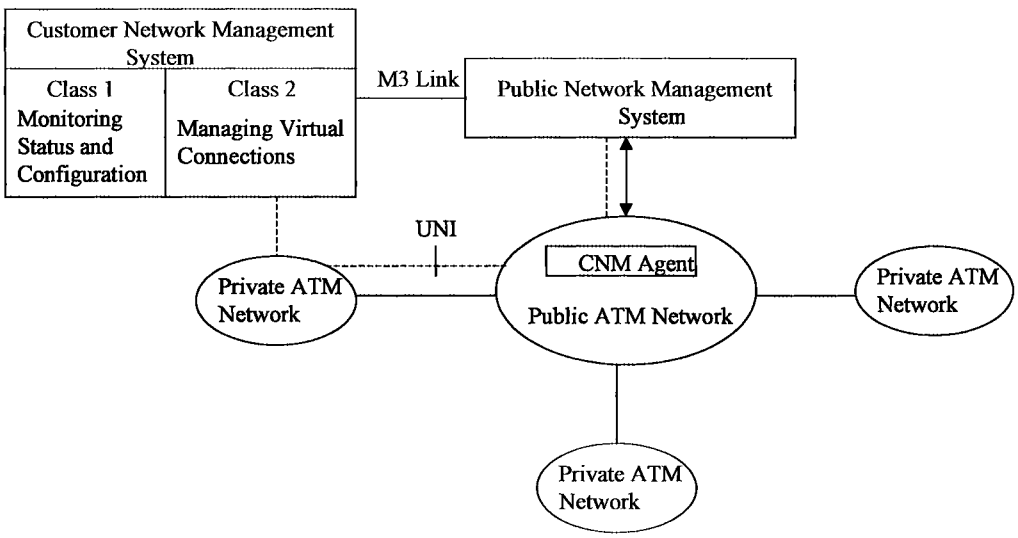


Figure 10-11
Customer network management for private and public networks.

- Information on how the ATM UNIs are configured and the status of UNIs at the cell level. This is useful for dynamically updating the configuration information.
- Configuration information at the UNI physical layer and the alarm status of the physical line.
- Configuration and status information on ATM VPLs and VCLs of customers' UNIs.
- Configuration and status information on VPCs and VCCs associated with customers' UNIs.
- Alarms and alarm-related information associated with the status of customers' UNIs.
- Descriptors associated with customers' UNIs on preconfigured traffic.

Class II functions basically perform addition, deletion, or modification of virtual connections and subscription information on an ATM public network. CNM customers must have the ability to add, delete, or modify the following:

- ATM layer configuration information
- VPL and VCL configuration and status information
- VPC and VCC configuration and status information
- ATM traffic descriptors and information objects for virtual path and virtual channel connections

Each segment of an ATM PVC network is modeled as an ATM public network. In each ATM public network, a CNM agent supports the MIB required to manage M3 interfaces. If there are many ATM PVC networks, element managers use multiple CNM agents to collect the information required for end-to-end management.

Figure 10-11 shows CNM functions as a subset of the management functions provided by the public network management system of a public network provider. The M3 link can be a dedicated link (private line), as shown in the figure. The CNM can access the public network management system using ATM UNI, as shown by dotted lines in the figure. The M3 link uses the SNMP protocol for managing class 1 and class II functions. If UNI is used, then AAL5 is used. Instead of defining new managed objects, managed objects defined in other MIBs are reused for the M3 interface. The M3 interface is fully explained in Reference 10.13.

10.6.1.2 M4 Interface. Network elements use the protocol-independent MIB for the M4 interface. Protocol-specific MIBs are developed from the

protocol-independent MIBs. These protocol-specific MIBs can be used with CMIP or SNMP. The M4 interfaces specified are for PVCs. The M4 interface is fully described in Reference 10.14.

Protocol-independent MIBs are described by managed entities. Managed entities are defined by the purpose of the entity, the attributes of the entity, the managed operations performed on the entity, the notifications emitted by the entity, and the relationship with other entities. Note that there can be one or more managed entities within an ATM NE.

ATM Forum network management uses five logical layers (management) of M.3010. These layers represent functional components and are not physical systems. These functional components can be implemented in different ways. For example, functions of NML and EML can be combined into a single physical implementation.

The basic functions of SMFAs, such as configuration management, fault management, performance management, and security management, are similar in ATM. However, there are some ATM-specific requirements that need to be implemented in ATM.

We have already looked into configuration management, fault management, performance management, and security management in Chapter 3. In this section, we discuss the ATM-specific issues related to these functional areas.

10.6.1.3 Configuration Management. ATM NE configuration identification and change reporting are important for the effective management of the ATM network by the manager or OS. Information on initialization, installation, changes in the externally manageable physical and logical components of ATM NE, and the relationships between these components must be available for effective systems management. Some of the physical and logical components are circuit packs, equipment, physical path termination points, and so on. The views of these physical and logical components presented by the NEs have to be current.

To present the current configuration view, it is essential that configuration changes are automatically reported to the managing systems. Notifications of configuration changes to the operational states of managed entities are sent to the ATM NEs. The managed entities that support the configuration identification and change reporting function (defined in Reference 10.14) are ATM NE cross-connection control, ATM NE, attribute value change record, equipment, equipment holder, event forwarding discriminator, latest occurrence log, log, managed entity creation log record, managed entity deletion log record, physical path termination point,

plug-in unit, software, state change record, TC adaptor, TC adaptor PM current data, and threshold data.

A managing system must be able, if required, to configure and reconfigure physical path terminations on an ATM NE as either a user network interface (UNI), a broadband interswitching system interface (BISSI), or a broadband intercarrier interface (BICI). The managing system must also be able to retrieve configuration data associated with UNI, BISSI, or BICI.

In addition to configuring ATM parameters, a managing system must be able to establish VPL-to-VPL cross-connections and VCL-to-VCL cross-connections in an ATM NE. Once VPL and VCL cross-connections are made, the managing system should be able to bring down the existing VPL and VCL cross-connections in an ATM NE and release the associated resources. The managing system must be able to retrieve the configuration data associated with the previously configured VPL and VCL termination points for analysis purposes. The configuration of VPL/VCL termination points and cross-connections must include the capability to establish and tear down multipoint VPL and VCL cross-connections in the ATM NE, and add or remove VPL and VCL termination points to and from existing multipoint cross-connections.

Managing systems must be able to configure and reconfigure active VPL and VCL termination points as either segment (Figure 10-12) or non-segment end points. However, VPC and VCC segments across UNI and

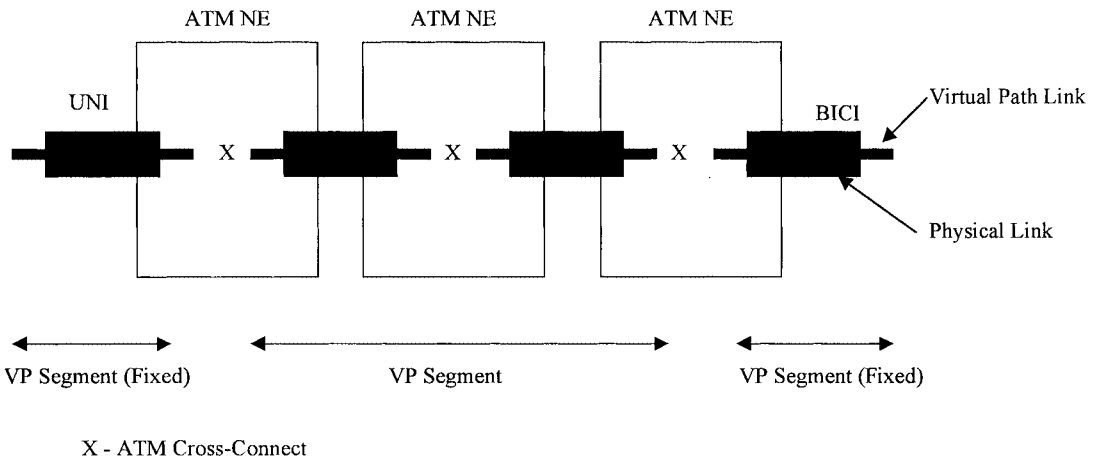


Figure 10-12
Example of a VP segment.

BICI are automatically configured as segment end points when VPLs and VCLs are configured.

Event flow control includes the event-forwarding discriminator function in ATM NEs. NEs should be able to forward alarms, configuration updates, and threshold crossing alerts to managing systems. It should also be possible to suppress the forwarding of selected notifications. The suppression of notifications must be based on notification type, on specific details of notification type, on the type of managed entity reporting the notification, or on some specific aspects of a managed entity.

10.6.1.4 Fault Management. ATM NEs must be able to send notifications of detected failures to an OS via the M4 interface. A managing system must be able to retrieve the alarm notifications log from an ATM NE. A managing system must also be able to perform loopback testing on VPC/VCC operations, administration, and maintenance cells. The results of the loopback tests are returned as pass or fail.

10.6.1.5 Performance Management. ATM-specific performance management consists of monitoring performance, managing traffic, monitoring user parameter control (UPC) or network parameter control (NPC) disagreement, and collecting performance management control and network data.

ATM performance monitoring consists of monitoring physical layer performance and cell level protocol. Physical layer performance monitoring must support ITU-T Recommendation G.774.01 (Reference 10.9) and ANSI T1.231 (Reference 10.10) for monitoring SDH transport performance.

Cell level protocol monitoring must be supported by the M4 interface. Cell level performance monitoring involves collecting threshold data counts to detect protocol abnormalities at the transmission convergence sublayer and the ATM layer. Managing systems must be able to retrieve the detailed log information stored in ATM NEs to detect, statistically analyze, and rectify cell-processing defects.

Two scenarios affect VPC/VCC performance. In one case, cells may be discarded due to transmission errors and problems in the network. In the other case, incoming cells may be discarded because incoming cells do not conform to the prenegotiated cell specifications. Managing systems must be able to distinguish between these two scenarios while collecting data on cell abnormalities.

10.6.1.6 Security Management. Security management includes verifying a session requester's unique user identifiers. There must be a record

of the user identifiers requested by session requesters that can be used to detect any security violations. Authentication involves verifying whether a session requester's user identifiers are authorized.

Security management also involves ensuring that only authorized users access resources in ATM NEs. To maintain data and system integrity, data and resources in ATM NEs must be created, modified, and deleted by authorized users. There must be a security audit record of all the security-related activities performed that can be used to detect and recover from intrusions and disruptions. Security administration develops and maintains these security-related aspects in the ATM NEs.

10.6.1.7 Integrated Local Management Interface. Integrated local management interface (ILMI) was supposed to be an interim solution when it was published. However, it has continued as a permanent one. ILMI (Reference 10.15) furnishes a link-specific view of the configuration of an ATM interface and status and control information about its physical and ATM layer parameters. Here, an ATM device such as a switch or an end system supports one or more ATM interfaces. Each ATM interface includes a set of managed objects and ATM interface ILMI attributes for achieving ILMI functions.

An interface management entity (IME) is associated with each ATM interface in an ATM device, as shown in Figure 10-13. The ATM end system has one interface, so there is one IME. The private ATM switch has two IMEs as there are two ATM interfaces. Either SNMP or AAL5 protocol is used between IMEs. An IME can access ATM interface MIB information associated with an adjacent IME. Each IME contains an SNMP agent and a management application. Adjacent IMEs must include the same MIB. SNMP for ILMI does not use UDP and IP addressing. For ILMI, SNMP uses a well-known VPI/VCI value.

ILMI MIB has the following characteristics:

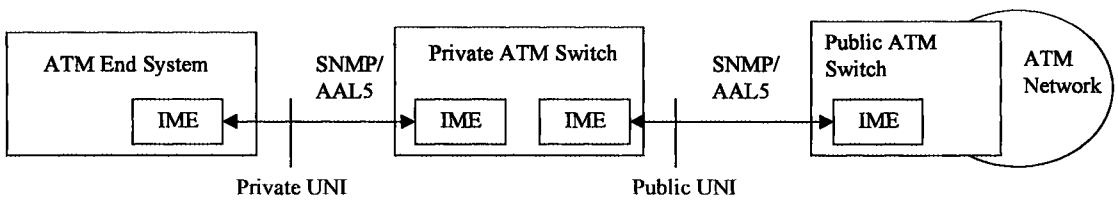


Figure 10-13
Integrated local management interface.

- *Textual conventions MIB:* This includes common textual conventions and object identifiers. These definitions are included in a single module such that other MIB modules can import the definitions. Some of the definitions are borrowed from other RFCs, and some object identifiers have been defined exclusively for the ATM interface.
- *Link management MIB:* This includes descriptions of the objects, procedures on how these objects can be used, and the actual definitions of the objects required for link management of ATM interfaces.
- *Address registration MIB:* This is used for address registration at the UNI. Address registration procedures include capabilities such as dynamic addition/deletion of additional network prefixes and user parts, and deregistration of addresses with the loss of ILMI connectivity.
- *Service registry MIB:* includes a service registry. This registry is useful for locating ATM network services.

ATM interface MIB groups for ILMI are shown in Figure 10-14. The details of MIB object groups are as follows:

- *Physical layer group:* This provides details of the physical interface over a physical link or the virtual interface over a virtual link.
- *ATM layer:* This includes objects for the ATM layer.
- *Virtual path connection:* This contains objects that provide details on the VPC, such as VPI value, VPC status, and QOS parameters at the VPC local end point.
- *Virtual channel connection:* This includes objects that provide details on the VCC, such as VCI and VPI values, VCC status, and QOS parameters at the VCC local end point, just like VPC. When ILMI communication takes place over a physical link, the VPI and VCI values identify a VCC. However, when ILMI communication is done using a virtual link, the VPI value is set to zero.
- *Network prefix:* This permits switches to automatically configure network prefixes in end systems.

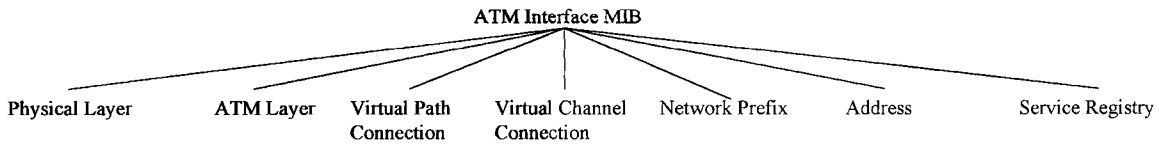


Figure 10-14
ATM ILMI MIB groups.

- *Address:* This mechanism permits end systems to automatically configure the ATM address for ATM interfaces on switches.
- *Service registry:* This is provided to locate network services.

Customers can use ILMI to retrieve UNI-related information. For this, ILMI must be part of UNI and customer access must be done through ILMI.

Figure 10-14 explains how network management functions such as discovery for configuration, fault isolation, and troubleshooting can be done. A network management system can request data from RFC 1695 MIB, as shown in the firm lines for MIB data on ATM device B. If management data is required from ATM device A, the proxy relays the SNMP commands via the IME in ATM device B, and traps to and from the IME in ATM device A.

10.6.1.8 ATM MIB. ATM uses different types of MIBs at different interfaces. Internet ATM MIB, defined in RFC 1695 (Reference 10.12), is used between switches. ILMI MIB is used at the UNI interface. ATM MIB is primarily used to manage permanent virtual circuits (PVCs). Additional objects are required to manage SVCs. ATM MIB can be used with the SNMP protocol.

In addition to the managed objects defined in RFC 1695, managed objects such as those defined in RFC 1213 (MIB-II), RFC 1407 [managed objects for digital signal 3 (DS3)/E3 interface], RFC 1595 (SONET MIB),

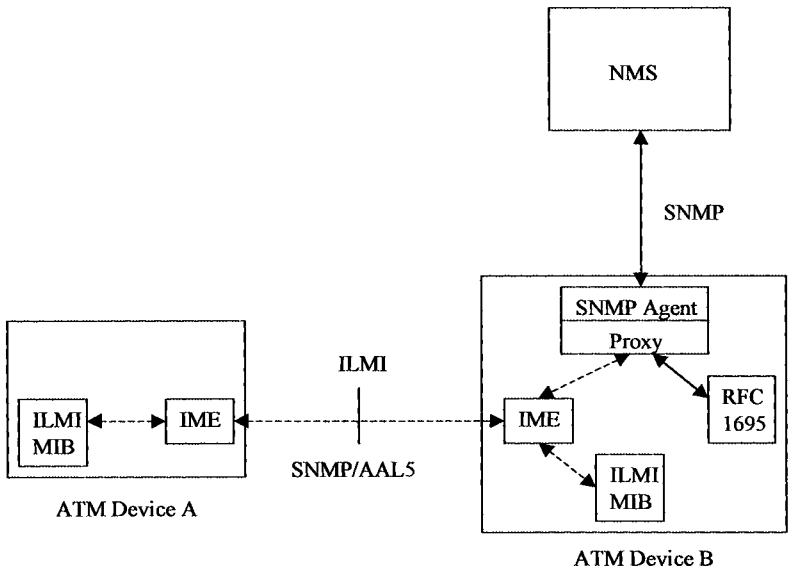


Figure 10-15
ATM network management, SNMP, and ILMI.

and RFC 1694 (SMDS) may also be required. The Main object groups of ATM MIB are shown in Figure 10-16. Explanations of the main ATM MIB object groups are as follows:

- *Configuration group:* This consists of objects for ATM cell layer configuration and local ATM interfaces, which are not supported by the ifTable. Managed objects supported in this group provide the following details:
  - Maximum number of VPCs and VCCs supported at the ATM interface
  - Number of VPCs and VCCs configured at this interface
  - Maximum number of active VPI and VCI bits
  - VPI and VCI values supporting the ILMI at the ATM interface
  - The type of ATM address at the ATM interface, such as native, private, E.164 or other
  - IP address and textual name of the neighbor to which a NMS can send SNMP messages
- *DS3 physical layer convergence protocol (PLCP) group:* This provides for the configuration and state parameters of the DS3 PLCP sublayer. DS3 PLCP is used to carry ATM cells over DS3 transmission paths. Some of the details available from this group are the following:
  - Number of severely errored framing seconds.
  - Indication of whether there is an alarm for DS3 PLCP. The conditions for alarm are an incoming yellow signal and an incoming loss of frame.

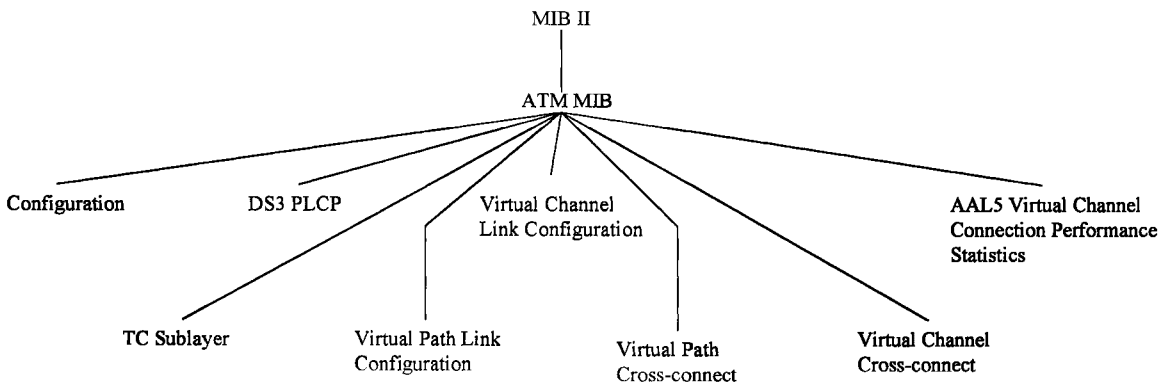


Figure 10-16
ATM MIB object groups (RFC 1695).

- *Transmission convergence (TC) sublayer group:* This provides objects for configuration and state parameters for the TC sublayer. The TC sublayer is used to carry ATM cells over SONET or DS3. Some of the details that can be gathered from the objects in the group are the following:
  - The number of times that out-of-cell delineation events occur.
  - Indication of whether an alarm condition is present for the TC sublayer. The alarm condition is available for loss of cell delineation; otherwise, there is no alarm condition.
- *VPL configuration group:* This contains information on the configuration and state information of a VPL. The objects in this group can also be used to create, delete, or modify a VPL that terminates in an ATM host or switch or that is cross-connected to another VPL. ATM VPL is implemented in an ATM host, ATM switch, and ATM network. VPL traffic parameters are manipulated by setting the ATM receive and transmit traffic descriptor indices in VPL tables. The receive and transmit traffic descriptor indices are associated with receive and transmit VPLs (Figure 10-17). There are many objects defined in this group that can be used to control the functioning of a VPL.
- *VCL configuration group:* This contains information on the configuration and state of a VCL at an ATM interface. Managed objects defined for this group can also be used to create, delete, or modify a VCL that terminates in an ATM host and ATM switch or that is cross-connected to another VCL.
- *VP cross-connect group:* This contains information on the configuration and state of VP cross-connects. The cross-connects can be point-to-point, point-to-multipoint, or multipoint-to-multipoint. A cross-connect index identifies the VPLs that are cross-connected to each other. The managed objects in this group can also be used to create, retire, and reconfigure VP cross-connects.
- *VC cross-connect group:* This contains information on the configuration and state of point-to-point, point-to-multipoint, or multipoint-to-multipoint VC cross-connects. VC cross-connect managed objects are functionally similar to VP cross-connect group managed objects.

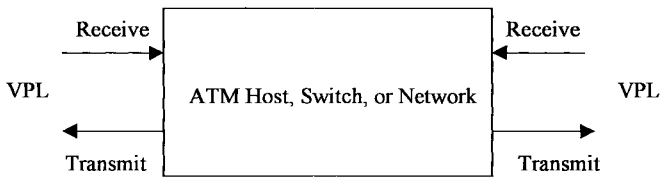
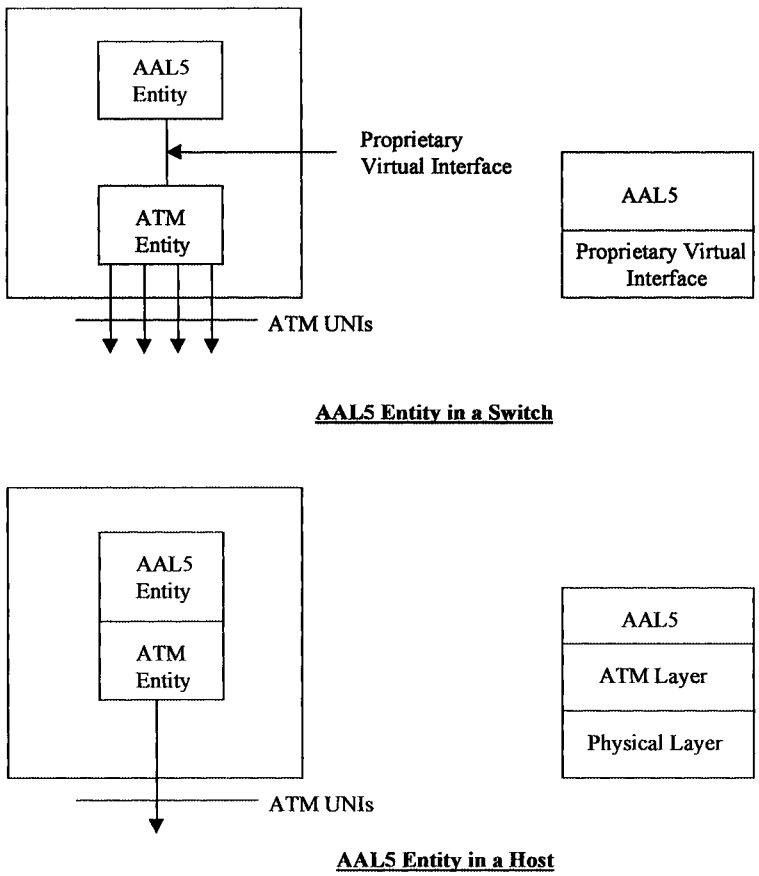


Figure 10-17
VPL Bidirectional traffic flows.

- *AAL5 VCC performance group:* There are two parts associated with the performance management of AAL5. In one part, ifTable is used to collect performance statistics on an AAL5 entity in a switch or a host (Figure 10-18). In the other part, managed objects defined in the AAL5 VCC performance group are used to gather AAL5 performance statistics per VCC. The VCC interface with an AAL5 entity in an ATM switch is done via a proprietary virtual interface, as shown in Figure 10-18. The managed objects in the AAL5 VCC performance group contain the VPI and VCI values of the AAL5 VCC identified by the ifIndex interface. The objects defined in this group can also be used to make available a number of Cyclic Redundancy Check (CRC)-32 errors, partially reassembled AAL5 PDUs, and AAL5 PDUs discarded due to large size.

Figure 10-18
Managing AAL5 in a switch and a host.



10.7 Important Issues in Broadband Network Management

Broadband network management is currently limited to the management of PVCs, and there is a broad consensus on managing PVCs. However, the same cannot be said for the management of SVCs. Additional modeling effort is required to manage SVCs to extract signaling information from ATM VCs, the analysis and routing of calls, and the administration of customers' numbers and services. These issues are treated in detail in Reference 10.17.

We have specifically discussed network management approaches that are covered by ITU-T and the ATM Forum, and we have also included the IETF MIBs for SONET/SDH and ATM. In addition, organizations such as ETSI, TINA-C, ANSI, and SIF are also actively involved in different aspects of broadband network management, and OMG is involved in different aspects of distributed network management. ITU-T has also defined an open distributed management architecture for distributed network management. TMN models are mostly static, covering limited regions and centralized network management; these static models must be integrated in a distributed manner to cover the different management protocols and transport technologies of global telecommunications network management. We will examine distributed network management in detail in Chapter 11.

Integrating broadband and distributed network management is one of the most important challenges in implementing broadband network management solutions. The issues involved are highlighted in Reference 10.18. For a good, sound design, designers should be well aware of the issues involved with different broadband network management architectures and the impact of distributed network management on the implementation of broadband network management.

10.8 Summary

The chapter begins with a brief overview of the B-ISDN reference model. We also introduce some of the important concepts and terms associated with ATM, SONET, and SDH, and we examine the network management of SONET and SDH. As the ATM Forum has made significant contribu-

tions to ATM implementation and network management, we describe ATM Forum network management in detail. We also discuss the Internet MIBs defined for SONET and ATM, and briefly highlight some of the important issues involved in broadband network management.

10.9 References

10.1. ITU-T Recommendation I.321, B-ISDN Protocol Reference Model and Its Application, 1991.

10.2. ITU-T Recommendation I.610, B-ISDN Operation and Maintenance Principles and Functions, 1995.

10.3. ITU-T Recommendation I.751, Asynchronous Transfer Mode Management of the Network Element View, 1996.

10.4. ITU-T Recommendation G.803, Architectures of Transport Networks Based on the Synchronous Digital Hierarchy (SDH), 1993.

10.5. ITU-T Recommendation G.805, Generic Functional Architecture of Transport Networks, 1995.

10.6. ITU-T Recommendation G.831, Management Capabilities of Transport Networks Based on the Synchronous Digital Hierarchy (SDH), 1996.

10.7. ITU-T Recommendation G.784, Synchronous Digital Hierarchy (SDH) Management, 1994.

10.8. ITU-T Recommendation G.774, Synchronous Digital Hierarchy (SDH) Management Information Model for the Network Element View, 1992.

10.9. ITU-T Recommendation G.774.01, Synchronous Digital Hierarchy (SDH) Performance Monitoring for the Network Element View, 1994.

10.10. ANSI T1.231, *Layer 1 In-Service Digital Transmission Performance Monitoring*, New York: American National Standards Institute, 1993.

10.11. Brown, T., and Tesink, K., Definitions of Managed Objects for the SONET/SDH Interface Types, RFC 1595, 1994.

10.12. Ahmed, M., and Tesink, K., Definitions of Managed Objects for ATM Management Version 8.0 using SMIv2, RFC 1695, 1994.

10.13. ATM Forum, Customer Network Management (CNM) for ATM Public Network Service (M3 Specification), 1994.

10.14. ATM Forum, M4 Interface Requirements and Logical MIB, 1994.

10.15. ATM Forum, Integrated Local Management Interface (ILMI) Specification Version 4.0, 1996.

10.16. SIF Architecture Group, SIF SONET TMN Architecture E-OS and NE, 1994.

10.17. Gillespie, A., Broadband Management after Permanent Connections, *IEEE Communications*, vol. 35, no. 10, pp. 54—59, 1997.

10.18. Manley, A., and C. Thomas, Evolution of TMN Network Object Models for Broadband Management, *IEEE Communications*, vol. 35, no. 10, pp. 60—65, 1997.

10.19. Black, U., *ATM Foundation for Broadband Networks*, Englewood Cliffs, N.J.: Prentice Hall, 1995.

10.20. Stallings, W., *ISDN and Broadband ISDN with Frame Relay and ATM* (3d ed.), Englewood Cliffs, N.J.: Prentice Hall, 1995.

10.21. Kumar, B., *Broadband Communications* (signature ed.), New York: McGraw-Hill, 1998.

This page intentionally left blank.

CHAPTER 11

Recent Trends: Distributed Network Management, CORBA, Java, Web, and TMN

11.1 Introduction

Telecommunications and computer networks are expanding in size and complexity because of the liberalization and globalization of the telecommunications industry. At the same time, new services are being offered by the telecommunications industry. These add to the complexity of the telecommunications networks. To reliably manage these complex networks, the TMN workload and functionality required have also increased manyfold. One of the means of coping with the increased workload is partitioning the TMN workload and functions into smaller, manageable subsets. Here distributed network management comes into the picture.

For distributing the TMN functions, ITU-T and ISO are working on open distributed management architecture (ODMA). At the same time, common object request broker architecture (CORBA) is getting increased acceptance as a means to distribute the TMN workload.

In addition to the impact of CORBA on TMN, Java (Sun Microsystems) has contributed significantly in the TMN arena as a platform-independent simple programming language. Java has impacted the ways in which TMN services are provided on different platforms.

Similarly, recent advances in Web technology have made their impact on TMN. Web technology is yet another technology that is available for managing TMN functions. Also, some of the Web technologies help the users of TMN by facilitating easy customer interfaces and service provisioning.

When we have many useful technologies impacting TMN, it is also important that these new technologies be integrated in a meaningful and useful manner to provide the maximum advantages. From this perspective, we will study different technologies in this chapter. As it is not our intent to dwell on each one, only a brief overview of each technology is presented. The focus is primarily on how these technologies can be used in TMN. Therefore, those who are interested in learning more about these technologies should refer to the references at the end of the chapter.

11.2 Distributed Processing

Distributed processing forms the basis for distributed network management. Therefore we will introduce the topic of distributed processing and subsequently examine distributed network management architecture and issues. Some of the important distributed processing architectures are:

- *RM-ODP:* The details of the reference model for open distributed processing (RM-ODP) are furnished in ITU-T Recommendations X.901 through X-904. RM-ODP includes the concepts and characteristics of distributed systems such as different levels of abstraction, modeling approach, and distributed-processing-related transparencies. Implementation details are not included in these documents.
- *CORBA:* This is an architecture for distributed computing published by OMG. OMG is devoted to promoting the object-oriented theory and practice for software development. OMG publishes industry guidelines and object management specifications for application development.
- *COM/DCOM:* Component Object Model (COM) and Distributed Component Object Model (DCOM) are the components of Microsoft's distributed computing architecture. While COM is used for interprocess communication within a system, DCOM is used for communication between clients and servers in different processes across an intranet or Internet.
- *SOM/DSOM:* System Object Model (SOM) and Distributed System Object Model (DSOM) are the components of IBM's distributed computing architecture. SOM and DSOM functions are similar to those of COM/DCOM.

11.3 Open Distributed Processing

Of the above prominent architectures, we will limit ourselves to the RM-ODP and CORBA. ITU-T and ISO are working on standardization for open distributed processing (ODP). The ODP standards consist of the following ITU-T recommendations:

- *X.901 (Reference 11.6):* Explains the overview of ODP and contains details on key concepts, the outline of ODP architecture, and how ODP is to be interpreted and applied.
- *X.902 (Reference 11.7):* Defines the concepts and framework for distributed processing systems.
- *X.903 (Reference 11.8):* Referred to as the architecture document; contains specifications for open distributed processing.
- *X.904 (Reference 11.9):* Contains interpretations of concepts using formal description techniques.

Distributed processing systems are required to solve the problems of scalability of single architecture systems, heterogeneous computing environments, and legacy systems. Though there are many requirements of distributed processing systems, we will look into only some of them. The important requirements of distributed systems are:

- *Distribution transparency:* The details and problems and differences due to the distribution of functionality and workload in different components of a distributed system must not be visible to the users. Also, the implementation details of combining heterogeneous components must be hidden.
- *Security:* As the systems are remote and data are distributed in many systems, these systems must be protected against unauthorized access and users.
- *Fault tolerance:* When the distributed systems are large and complex, the possibility exists of some component failing. Failure of one or more components should not affect the operation of the other components of the distributed system.
- *Federation:* It should be possible to coordinate the activities of various components in the distributed systems belonging to different administrative and technical domains.
- *Modularity:* The different components of the distributed systems have to be autonomous and, at the same time, must be able to be coordinated to work in a cooperative manner.

Standardization of distributed processing has four stages: system specification in terms of object modeling, interrelation of system specifications to viewpoint specifications, system infrastructure definition using relevant distribution transparency categories, and establishment of a framework for distributed system conformance.

Object modeling has to cover all the resources in the distributed systems and must apply to all viewpoints, provide tools for the requirements and design of specification languages, and address structuring issues in the distributed systems.

To simplify and categorize the specifications of a whole system, different abstraction levels have been identified; these abstractions are known as viewpoints. The viewpoints considered are:

- *Enterprise:* Focuses on the purpose, scope, and policies of a system within an organization and its environment.
- *Information:* Relates to the semantics of information handled by a distributed system, the constraints on the information, and the interpretation of the information.

- *Computation:* Covers the functional decomposition of a distributed system resource into a set of objects that interact with interfaces.
- *Engineering:* Deals with the infrastructure required to support interaction between objects of distributed systems.
- *Technology:* Covers implementation details such as design methodology, programming languages and details, details on databases, and so forth.

As we have seen, distribution transparency (hiding the implementation of distributed systems from the users of the systems) is an important requirement. Note that it is not necessary to implement all the distribution transparency categories given below. The following distribution transparencies have been defined in ITU-T Recommendation X.901:

- *Access transparency:* Enables distributed components to work together irrespective of differences in architectures, data representation, and programming languages.
- *Failure transparency:* Is required to hide failure and recovery of other objects and itself. Necessary for providing a fault-tolerant distributed system.
- *Location transparency:* Specifies that location of an object in a distributed system is not important during interaction with other objects.
- *Migration transparency:* States that even if an object has moved, the distributed system will not be affected.
- *Persistence transparency:* Ensures that activation and deactivation of an object do not impede the sharing of resources.
- *Relocation transparency:* Provides availability of the interfaces associated with an object despite changing location.
- *Replication transparency:* Allows an object to be replicated while still maintaining a single interface.
- *Transaction transparency:* Is required to coordinate transactions involved in scheduling, monitoring, and recovery of multiple objects to maintain data consistency.

When a distributed system is built with different components, with the possibility of some of the components being procured from vendors, it is necessary to establish well-set rules governing the behaviors of each of the components of a distributed system. These rules have to be elaborately laid out with reference to the different viewpoints. The rules also must be covered in conformance specifications and must later be validated by conformance testing.

To construct distributed systems, a set of ODP functions has been defined. These ODP functions are divided into the following groups:

- *Management function:* Consists of node, object, cluster, and capsule management functions. A *node management function* controls the processing, storage, and communication functions within a node. An *object management function* provides for the checkpointing and deletion of an object. A *cluster* is a combination of engineering objects for the purposes of deactivation, checkpointing, reactivation, recovery, and migration. A *cluster manager* manages basic engineering objects in a cluster. A *capsule* is a combination of objects for the purpose of encapsulation of processing and storage; a *capsule manager* manages the engineering objects in a capsule. Capsule managers and cluster managers provide the cluster management functions.
- *Coordination function:* Consists of event notification, checkpoint and recovery, deactivation and recovery, group, replication, migration, transaction, and engineering interface reference tracking functions. Here the group function coordinates interactions of objects in a multiparty binding.
- *Repository function:* Includes storage, information organization, relocation, type repository, and storage functions. A *trading function* supports importation of service offers by service users or clients and exportation of service offers by service providers or servers.
- *Security function:* Consists of access control, security audit, authentication, integrity, confidentiality, nonrepudiation, and key management functions.

We have only included basic concepts of ODP here; for more details, consult References 11.6, 11.7, 11.8, and 11.9.

11.4 Distributed Network Management

Earlier, centralized network management systems were the norm. Network management systems on mainframes were used to control the network management functions. These systems present a scalability problem when the NEs or agents to be managed grow in numbers. This increase in workload leads to performance bottlenecks. In addition to the

increased workload, the network management functions performed were relatively simple, such as detection of warning conditions, and many problem modifications were done manually. However, these scenarios are changing with the ever increasing demands from the network management systems, increasing intelligence in NEs, and, sometimes, a need to solve the problems in a nearly real-time environment.

To alleviate some of these problems, distributed network management is gaining popularity. The trend is also toward Windows NT—based server solutions on Intel processors. This reduces the cost and permits the growth of network management systems by adding more and more computing systems. Distributed network management permits the incremental growth of network management systems, enabling legacy systems to coexist with new network management systems. There are also other advantages to distributed network management, such as load balancing and better reliability through redundancy.

Distributed network management solutions come with a price and are rather complex because of the various technological issues involved. Some of the important issues to be addressed are: how to handle security of EMs and NEs in different domains, time synchronization, distribution transparency, data integrity, appropriate distribution of workload between EMs and NEs, coordination between EMs, management of failures and failure policies, and so on. The adoption of distributed systems architecture to network management has been studied in detail by the Telecommunications Information Networking Architecture Consortium (TINA-C), OMG, ITU, and ETSI.

11.5 Open Distributed Management Architecture (ODMA)

ODMA is outlined in X.703 (Reference 11.5). ODMA provides the architecture for distributed network management as well as management of open distributed applications and distributed resources. It uses the principles of RM-ODP and extends them to the OSI systems management. The ODMA document describes the enterprise, information, computational, and engineering viewpoints.

Open distributed management includes support for management of resources, coordination of distributed management activities, management of systems of different sizes and complexities, distribution

transparencies, portability of management applications, modularity of different components used, seamless integration with legacy systems, and access transparency to support different communication protocols.

No specific notational technique is suggested for explaining enterprise viewpoint specifications. An enterprise specification should describe the relationship between objects with respect to the managing and managed roles.

The information viewpoint contains invariant, static, and dynamic schema. An *invariant schema* contains relationships between information objects; these relationships are always valid. A *static schema* refers to the assertions that are true at a given point in time. A *dynamic schema* includes the assertions that are applicable when a system operates. Object Modeling Technique (OMT), developed by Rumbaugh, is used for describing the information viewpoint.

The computational viewpoint is described using GDMO and GRM. A computational management template specification consists of computational interfaces, a behavior specification, and an environment contract specification. There are three types of computational management interfaces: management-operation, notification, and linked replies interfaces. Interface signatures definitions may be done either using interface definition language (IDL) or TMN CMIS services.

The engineering viewpoint includes functionality of objects that support distribution transparencies. While the computational viewpoint focuses on when and why objects interact, the engineering viewpoint is concerned with how the objects interact. The engineering viewpoint also includes how communications protocols are used between objects.

Some specific functions have been defined for distributed management. They are:

- *Operation dispatching function:* Controls the binding of management operation client interface and management operation server interfaces. It also aids in adjusting to the dynamic changes in management operation server interfaces.
- *Notification dispatching function:* Facilitates binding between notification server interfaces and notification clients.
- *Policy enforcing function:* Ensures that management policies are enforced and that any violations of management policies are reported to the appropriate components.

We have introduced the basic concepts involved in ODMA. For more details on ODMA, refer to Reference 11.5.

11.6 CORBA

It is important to note that CORBA is a distributed computing architecture. CORBA is being slowly accepted as a solution for distributing the TMN workload among different element managers. It can also be used to distribute TMN workload in the network element management layer, service management layer, and business management layers as well. CORBA can be used in the following scenarios:

- *Functional distribution:* CORBA can be used when one network element wishes to distribute the workload to one or more element managers. As an example, alarm monitoring can be done in one element manager and performance management can be done in another element manager. In this case, the agent receives management-related commands and transmits responses and M-EVENT-REPORTs related to alarm monitoring to one element manager. The same agent, however, receives management-related commands and transmits responses and M-EVENT-REPORTs related to performance management to another element manager.
- *Geographical distribution:* When the number of network elements reporting to an element manager increases, the performance of the EM application is affected. One solution is to partition the workload of an EM by adding another EM. This can be based on the geographical distribution. As an example, all the network elements in the United States can report to one EM in the US, and all the network elements in Japan can report to another EM in Japan. This concept of TMN workload based on geographical distribution is similar to that of domains and subdomains.

11.6.1 Overview of CORBA Architecture

The latest CORBA document is The Common Object Request Broker: Architecture and Revision 2.2, dated February 1998 (Reference 11.10). Figure 11-1 illustrates the basic CORBA architecture. As per CORBA architecture, there are four primary components that interact with each other using a communication bus. These primary CORBA components are:

- *Application objects:* Are not standardized by OMG; these are products developed by vendor groups and do not use standard interfaces. Application objects use the other CORBA services.

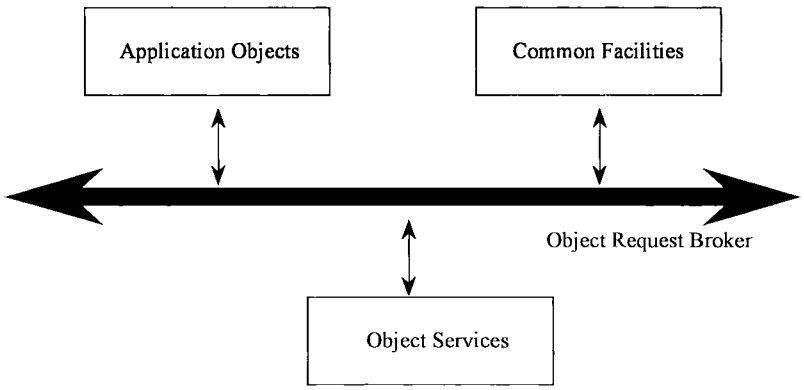


Figure 11-1
Basic CORBA architecture.

- *Common facilities:* Contain high-level application services and do not provide basic services such as object services. Some of the examples of the common facilities are: telecommunications, finance, healthcare, network/systems management, graphical user interfaces, and business-related objects.
- *Object services:* Provide the important services required for CORBA and are used by common facilities and application objects. Object services are responsible for providing basic services that can be independently used by applications, for example naming service, trading service, concurrency, persistency, security, event service, transaction services, and others.
- *Object request broker (ORB):* Is the communication bus used by CORBA applications in a distributed environment. It also enables communication and interoperability between applications in different environments.

For communication between CORBA applications, general interface ORB protocol (GIOP) is used. GIOP is a connection-oriented transport protocol that defines seven message formats that cover all the ORB request and reply semantics. Internet inter-ORB protocol (IIOP) specifies how GIOP messages are exchanged over TCP/IP connections. IIOP is positioned in TCP/IP stack as shown in Figure 11-2. IIOP has become a standard protocol for linking objects across a TCP/IP network.

11.6.2 CORBA Services

CORBA architecture provides many additional services that enhance the functionality of CORBA.

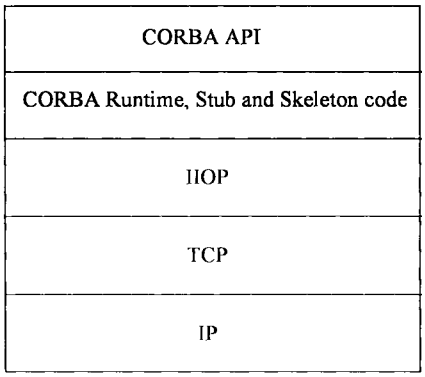


Figure 11-2
IIOP relationship to TCP/IP.

The CORBA implementations require a naming service. The CORBA objects are registered in the naming service. The naming service takes the name of an object and returns the object reference in the server. CORBA implementations have naming service implementations. However, if the number of objects is large, this creates a large load on the naming service. Partitioning the naming service or using a multitiered lookup strategy are some of the solutions to this problem (Reference 11.3).

CORBA has a flexible event service. An event service primarily consists of one or more consumers, suppliers, and an *event channel* (the transmission medium between consumers and suppliers). The supplier places the message to be transmitted to a consumer on the event channel. Depending upon whether the channel operates under a push or a pull model, the message is pushed to the consumer or the consumer retrieves the message from the event channel. The event service is very useful for forwarding notifications and alarms to element managers.

The life cycle service provides services to create, delete, move, and copy objects. It controls the life cycle of an object.

The CORBA trader service accepts and stores service offers from servers. A client requests information from the trader service and receives the information requested from trader service if it is available.

The CORBA level 1 security service provides authentication, authorization, encryption, delegation, auditing, and logging. CORBA level 1 security over DCE/Kerberos is also provided by some vendors. Similarly, security over the socket layer and IIOP are also being worked out. With these security implementations, security, which can be a major issue in a distributed environment, is no longer an issue in CORBA.

For more details on CORBA services, consult Reference 11.11.

11.6.3 How CORBA Applications Work

Many vendors have CORBA implementations. Some of these are: Orbix (IONA Technologies), PowerBroker (ExperSoft), VisiBroker (VisiGenic), and others. CORBA object classes are defined using interface definition language (IDL). An IDL interface definition defines a CORBA object class. A CORBA interface definition specifies the services provided by the CORBA object class, the exceptions that can be generated by that class, and the attributes of that class.

The services provided by and the exceptions generated by a CORBA object class are furnished by the *operation declarations*. An operation declaration contains input parameters, output parameters, the operation type, and the exceptions that can be raised. While mapping IDL to C++, each operation is mapped to a C++ member function. An IDL attribute has information on the state of an object, and the attribute declaration states whether the attribute value can be set to one or more values, whether the attribute value or values can be retrieved, or whether both operations can be performed. Here also, while mapping IDL to C++, each attribute is mapped to a set of C++ member functions. IDL compilers do this IDL to C++ mapping.

As the focus of the chapter is on CORBA and TMN, we will not look further into IDL. For more information on IDL, refer to one of the many books on the subject.

The applications written using IDL must be compiled using an IDL compiler. The IDL compiler produces the mappings to C++, Java, Smalltalk, or C. The results of IDL compilation are client and server stubs. The client and server codes, along with the client and server stubs, are normally implemented as libraries (Figure 11-3).

CORBA needs an activation component to start the server. The activation component forms the initial connection between a client and server. Usually, a client requests a connection with the server. The server contains the executable code, which an ORB can activate to create a server process. The server code has to populate initial objects and transfer control to ORB to receive incoming calls from one or more clients.

CORBA uses an implementation repository and an interface repository. The implementation repository contains information on where the server executables reside. The activation component queries the implementation repository to look for the server and gets back the address of the server. The interface repository is involved in run time type checking.

One IDL interface is required for client-managed applications, and another for server applications.

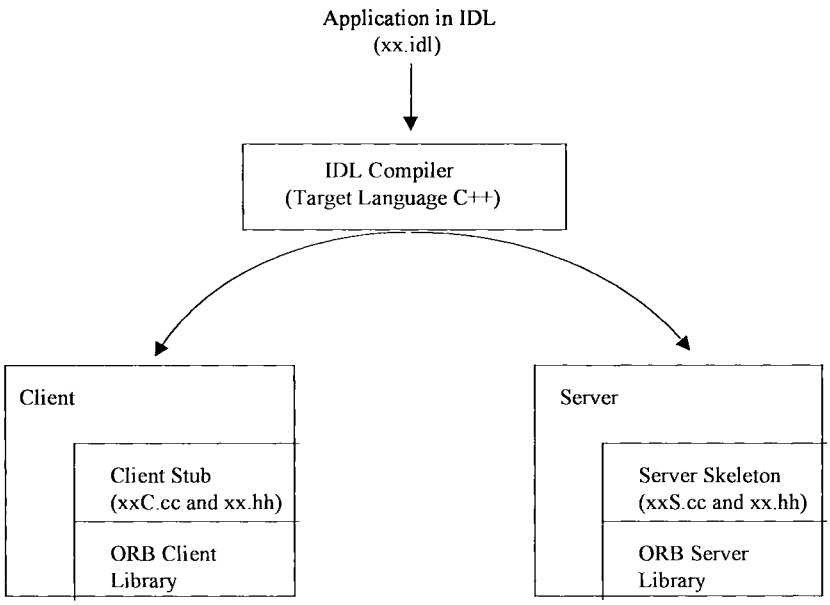


Figure 11-3
Building a CORBA application.

11.6.4 CORBA-based TMN

TMN is well established in the telecommunications arena. It includes four standard components: distributed management functions, layered TMN architecture, communication protocol (CMIP), and information models using GDMO/ASN.1. The primary focus of CORBA is in distributed computing environments. CORBA is appealing for TMN implementations as a solution to some of the scalability problems in TMN. Also, pure TMN solutions require a large investment of money and development effort. CORBA, however, requires less development effort and expense. As a result, there is an ongoing effort in the telecommunications industry to implement TMN architecture using CORBA.

Both TMN and CORBA attempt interoperability at syntactical and semantic levels, but the approaches are slightly different. TMN provides communication interoperability. CORBA targets interoperability at the application process level and heterogeneous programming environments, providing the flexibility to decompose system components with standard interfaces. The ability to decompose systems facilitates easy integration in heterogeneous programming environments and legacy systems.

To implement the TMN architecture with CORBA, the basic CORBA services would need to provide standard management functions used in TMN. IIOP can be used instead of CMIP. However, the generic nature of

CORBA services does not allow for powerful domain-specific systems management function support as specified by TMN standards. There is a good discussion on OSI and CORBA in Reference 11.13.

CORBA standards for TMN are being looked into by industry consortiums such as TeleManagement Forum and OMG. In an attempt to preserve the great value of the TMN information model, Joint Inter-Domain Management (JIDM) of TeleManagement Forum is developing formal translation from GDMO/ASN.1 models into IDL and vice versa. However, as with other standard body activities, there is a good deal of politics involved in preparing these standards. The practical and political considerations are going to decide some of the issues of how CORBA will penetrate TMN arena.

There are some fundamental differences between CORBA and TMN (Reference 11.12). In TMN, message transfer between an EM and NEs is asynchronous. In the case of CORBA, the member operation between a client and a server is synchronous. A member operation has one result. However, in the case of CMIS M-GET, there can be multiple replies. Also, in the case of CORBA event service, there is no confirmation of a delivery from a supplier to a consumer. However, the notifications in TMN carried in an M-EVENT-REPORT can be confirmed or nonconfirmed.

There are also some differences between CORBA and TMN in the information model. In TMN an object is identified by object identifiers. In TMN, containment hierarchy provides the relationship between object classes and can be used for uniquely naming an object. Also, scoping and filtering can be used to perform operations on object instances on a selective basis. In the OMG model, an object cannot be identified by object reference, as there can be one or more references to the same object. As a result, the notion of object identity is weak in the OMG model. In TMN, object classes can have conditional packages. This facilitates object instances of a class to have different properties. This feature is not available in interface definitions in IDL.

ASN.1 has more complex data types than CORBA, such as tag values, subrange types, and compound types. But IDL allows only primitive type constants such as boolean, integer, floating point, character, and string. As a result, in mapping GDMO/ASN.1 to IDL, some information can be lost.

Many TMN applications using CORBA are available. CORBA is useful for service management layer applications. Some of the service management applications that can use CORBA are service order entry, provisioning, and billing. There are some CORBA-based applications in the areas of fault and performance management as well. Some of the motivations for using CORBA in TMN are:

- In TMN, resources, including networks and equipment of different types, are defined using GDMO/ASN1. The information models used in TMN use the object-oriented design and implementation concepts. CORBA IDL supports the concepts of managed object classes and managed objects. In CORBA, IDL can accommodate information models in TMN by mapping of GDMO/ASN1 data types to CORBA data types.
- It is necessary to distribute the TMN workload among different element managers. These element managers may be in different workstations. As an example, alarm summary and alarm history may be displayed on one workstation, while the traffic management—related data may be on another manager on another workstation. Figure 11-4 shows this type of architecture.

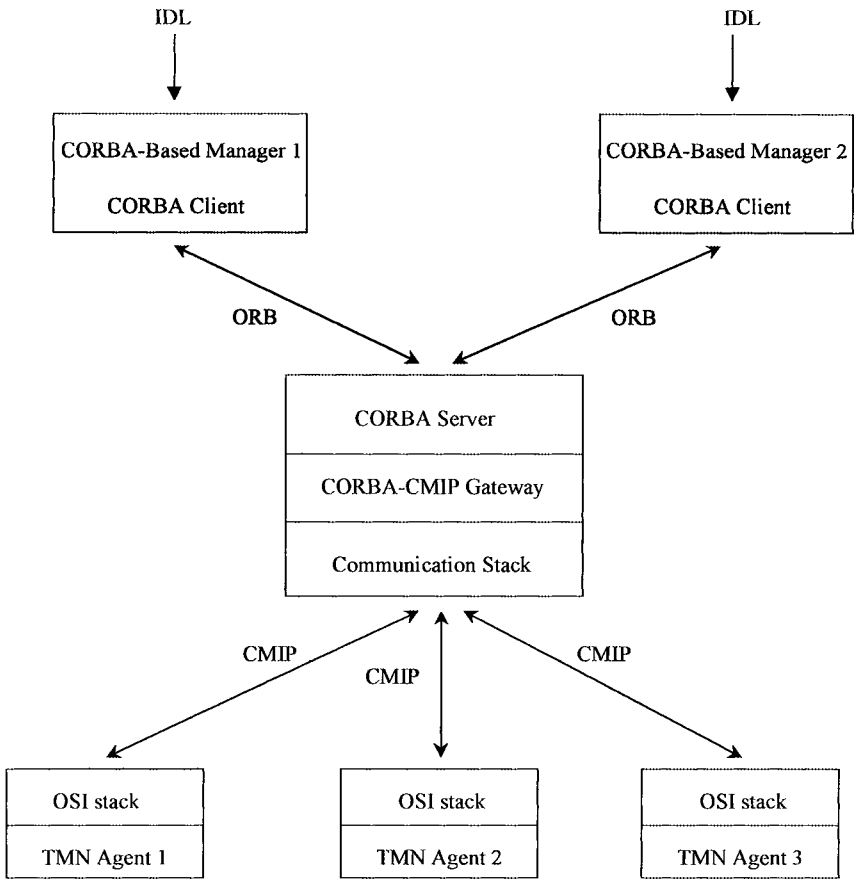


Figure 11-4
Example of CORBA and TMN integration.

- CMIP requires the BER encoding and decoding to transfer CMIP commands and replies. This key paradigm is available in IDL without the encoding and decoding requirements of GDMO.
- Simple mappings from IDL to C++ and other languages are available. This also makes IDL attractive as a means to define the resources and integrate them with existing management applications without rewriting.
- CORBA services such as naming, event service, transactions, security, trading, and so on, are being continuously added to the already existing CORBA implementations. These additional services make it easy for developers and implementers to concentrate on building new applications and to extend the existing CORBA-based applications.
- The ability to dynamically discover objects at run time using a dynamic invocation interface (DII) provides for the separation of management applications and the underlying technology.

11.6.5 TMN and CORBA Integration

There are different approaches to organizing managers and agents in CORBA clients and servers. In one approach, the manager code libraries reside in clients' workstations or desktops. These agent workstations or desktops need to have OSI seven-layer stacks. In the workstation, which has a CORBA server, we have the server code, the CORBA-CMIP gateway, and the communication stack. The CORBA-CMIP gateway primarily converts the CORBA member operations to CMIP commands downstream and converts responses from the CMIP commands back to CORBA member operations. The communication stack consists of the OSI seven-layer protocol stack with ACSE to provide the association establishment and release between the server and the agent. This scenario is shown in Figure 11-4.

With this approach, it is possible for one or more managers to exchange management information with one or more agents. As an example, TMN agent 1, TMN agent 2, and TMN agent 3 may send fault management data to CORBA-based manager 1. On the other hand, TMN agent 1, TMN agent 2, and TMN agent 3 may send performance management data to CORBA-based manager 2. This is an example of functional distribution of management information.

Instead of the preceding scenario, TMN agent 1 and TMN agent 2 may be responsible for collecting management information in a particular geo-

graphical region. These data may be consolidated in CORBA-based manager 1. Similarly, TMN agent 3 may refer to another geographical region. The management information of this region may be sent to CORBA-based manager 2. This distribution of management information is an example of the geographical distribution of management information.

Let us go into more detail on how the CMIP and CORBA flows work. In Figure 11-4, CORBA-based manager 1 sends the CMIP commands such as M-GET, M-SET, M-CANCEL-GET, M-CREATE, and M-DELETE using IDL member operations to the server. The server, which has a CORBA-CMIP gateway, converts the member operations to the CMIP commands. These CMIP commands will be sent to the TMN agent 1 using the communication stack in the CORBA server workstation. The responses to these CMIP commands will be sent from the TMN agent 1 to the CORBA server workstation using the communication stack in the agent. In the CORBA server workstation, CORBA-CMIP gateway converts the CMIP responses from TMN agent 1 to IDL member operations and the responses are sent to CORBA-based manager 1 as a function call.

However, notifications in the M-EVENT-REPORT come in the opposite direction of the CMIP commands such as M-CREATE, M-DELETE and others. These are sent from the TMN agents to the server, and the CORBA-CMIP gateway converts the M-EVENT-REPORT to IDL invocations, which are sent to the managers. These managers must be able to interpret the IDL methods.

Alternatively, we can use another approach to exchange management information between managers and agents (Figure 11-5). In this case, the manager is TMN based and the agents reside in CORBA clients. The communication between agents in the CORBA clients and CORBA server is done using ORB. In the CORBA server, the CORBA-CMIP gateway converts the IDL member operations and attributes to CMIP commands and transmits the management information to the TMN managers. Here also, the server needs to have an OSI communication stack and ACSE component.

The responses to the CMIP commands from the TMN-based managers or to M-EVENT-REPORTS from the CORBA-based agents are in the form of IDL member operations and are received by the CORBA server. In the CORBA server, the CORBA-CMIP gateway converts IDL member operations to CMIP-based responses or M-EVENT-REPORTS and dispatches the CMIP responses or M-EVENT-REPORTS to the TMN managers over the OSI communication stack. Reference 11.2 has an example of how OSI and CORBA framework can be integrated.

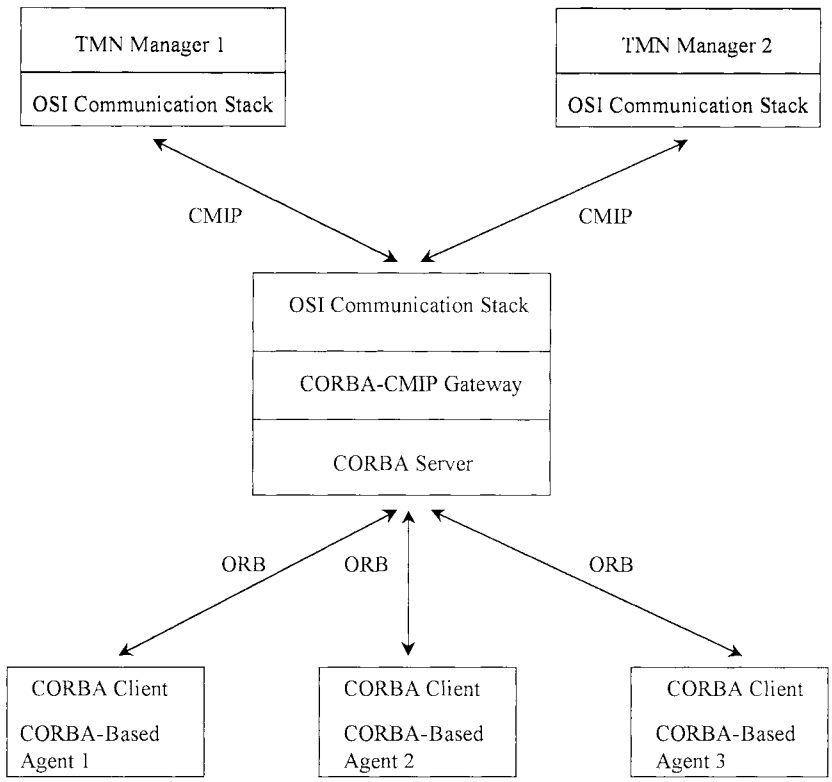


Figure 11-5
Example of CORBA and TMN integration.

11.6.6 Joint Interdomain Management (JIDM)

Many TMN applications use CMIP as management protocol and GDMO and ASN.1 as information models. Similarly, there are a large number of NMSs and NEs that use SNMPv1 or SNMPv2 for management protocols and structure of management information (SMI)-based MIBs as information models. For distributed network management, the CORBA-based managers and CORBA-based clients use GIOP or IIOP for transport and CORBA IDLs for object class definitions.

In the telecommunications and computer industries, the trend is initially for these different protocols to coexist and interoperate because of the cost and development effort involved. After the initial coexistence stage, the general practice is to migrate network management toward stable management protocols and information models. Therefore there is a demand for CMIP, SNMP, and CORBA to be able to work together. Figure 11-6 shows how CORBA managers, OSI managers, and SNMP managers can coexist.

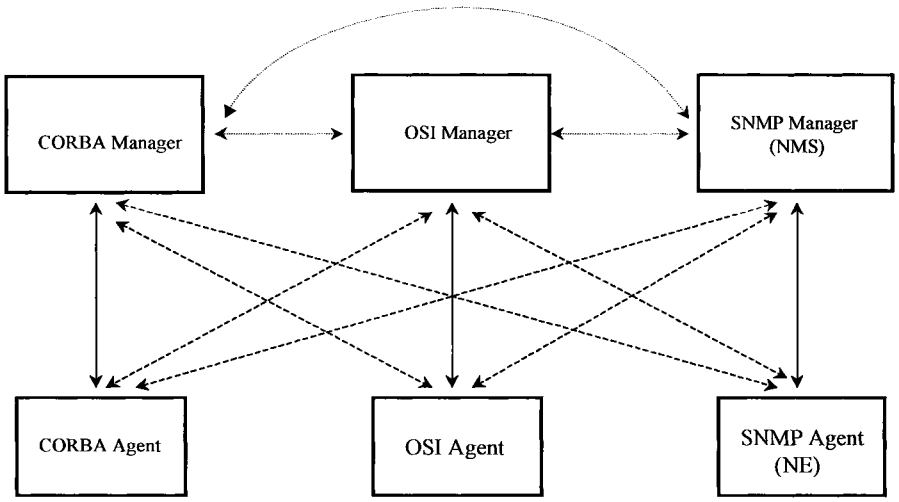


Figure 11-6
Different management domains.

Some of the issues involved in interoperation between GDMO/ASN.1 and CORBA, and between CMIP and SNMP and CORBA, are:

- The application interfaces to CORBA managers and clients are done using IDL declarations. However, in TMN, resources are defined using GDMO and ASN.1. So if management protocols using IDL are to be interpreted, then object definitions in IDL have to be translated to GDMO and ASN.1. Similarly, the applications interfacing with managers need to understand the responses and notifications from the network elements in IDL form.
- Similarly, for TCP/IP-based Internet network management, information models for the network elements are in the form of SMI-based MIBs. Thus the transfer of management information from NEs and NMSs involves the conversion of the information model from IDL to SMI-based MIBs and vice versa.
- In addition to the differences in the information models, there are differences in management protocols. In TMN, CMIP protocols are used, and in Internet network management, SNMP protocols are used. In CORBA there are no management protocols; only the transport protocols are defined. As a first step for CORBA to be used as a management protocol, CORBA needs to have management protocols.
- As can be observed from Figure 11-6, for coexistence of managers based on different management protocols, there is also a need to have CORBA management protocols for communication of management

information between the CORBA managers and TMN managers or SNMP managers. Work is not being done on this issue.

Issues such as information and management protocol translations, which are required for CMIP, SNMP, and CORBA to be able to work together, have been worked out by the Joint Inter Domain Management Group (JIDM). JIDM is a joint task force of X/Open and TeleManagement Forum.

The static translation of GDMO/ASN.1 to IDL, IDL to GDMO, and SNMP SMI-based MIBs to IDL is addressed by specification translation. The algorithms for specification translation are furnished in detail in Reference 11.1. Some commercial translators are currently available to perform the specification translation.

The other major issue of interoperability is that of differences in management protocols. As an example, CORBA uses GIOP and IIOP protocols for communication between clients and servers. Note that GIOP and IIOP are generic transport protocols. In TMN, communication of management information between OSI managers and OSI agents is done using CMIP protocols. Similarly, in the case of Internet network management, network management stations and network elements use SNMP protocols for communicating management information.

Another major issue is the communication of management information between managers in CORBA, CMIP, or SNMP domains. This management information communication requires protocol translations between domains. This protocol translation is handled by the interaction translation. Some of the challenges involved in mapping between CORBA and CMIP for conveying management information are nicely highlighted in Reference 11.1.

The interaction translation can be done by a gateway. This gateway has to map between CORBA IDL and CMIP PDU and vice versa. While CMIP PDU and IDL are mapped, the GDMO identifiers have to be mapped to IDL method invocations. GDMO is case sensitive, and this aspect has to be taken into consideration too. Besides, the CMIP scoping and filtering have to be mapped to a sequence of IDL invocations and vice versa.

For coexistence between CORBA and TMN, the gateway will require the use of many CORBA services. The OSI managers and agents require AE titles or distinguished names to form associations. This will require the use of CORBA naming service. Life cycle services to create new object instances and event services for forwarding notifications are also required.

11.6.7 CORBA Implementation Notes

In software development, object-oriented design and analysis are becoming popular in TMN. OMT object modeling is one of the popular tools for developing the designs. Commercial tools are available to translate OMT object models to IDL interface definitions. After the objects are declared in IDL, IDL compilers can be used to map the IDL interfaces to C++ application stubs and skeletons.

There are many vendors with CORBA implementations. These vendors are adding new CORBA services to cater to the increasing requirements of TMN and the computer industry. So, instead of building CORBA services from scratch, it is better to use the services provided by CORBA implementations available from vendors. IONA is one of the popular CORBA vendors, and many telecommunications equipment and service providers are using IONA's Orbix.

Note that for sending notifications, event services are required. Multithreading is required for better performances, especially in cases where lengthy responses are involved. Linked replies from M-GET constitute one such example.

For many TMN applications, Windows 95/98 and Windows NT operating systems on Intel-based platforms are being selected because of the relative less cost when compared to the Unix systems. Alternatively, CORBA clients can be on Windows 95/98 and the CORBA servers on Windows NT or different Unix platforms. Similarly, many TMN applications on Windows NT are also becoming popular for managers. So the availability of CORBA applications on Windows 95/98 and Windows NT must be carefully examined when selecting CORBA vendors.

Many TMN applications use traditional relational databases and, in some cases, object-oriented databases for providing data persistence. It is very important to check whether, along with the CORBA implementations, the operating systems provide interface to the right type of databases. Also, some C++ applications use standard template libraries (STL) extensively. Along with the language used for implementations, issues such as whether C++ STL support is available also have to be considered.

The naming service is quite an important component. When a large number of objects are being used, performance has to be considered. This aspect has to be looked into when selecting CORBA vendors.

These days, many TMN applications such as fault, performance, and service management using CORBA are also available. So, depending upon

the functionality required, it is better to go for applications already available in the market instead of reinventing the applications all over.

11.7 Web-based TMN

Web browsers such as Netscape Navigator and Microsoft Internet Explorer are available in most of the computer platforms. These Web browsers can be used as TMN application end user interfaces. Web applications use hypertext markup language (HTML) to display a Web page in a browser. HTML is a formatting language for creating Web pages.

Hypertext transfer protocol (HTTP) is a connection-oriented application-level protocol for transferring files to Web pages. There are many versions of HTTP. The first, referred to as HTTP/0.9, was a simple protocol for raw data transfer across the Internet. HTTP/1.0 was an improvement over HTTP/0.9 and is defined in RFC 1945. The improved version permitted Multipurpose Internet Mail Extensions (MIME)-like messages (References 11.14, 11.15, and 11.16). These MIME-like messages, in addition to the data, contain information about the data transferred and modifiers on the requests and responses. HTTP/1.1, which is a proposed standard, is yet another improvement on HTTP/1.0. HTTP/1.1 contains more stringent implementation requirements and permits hierarchical proxies, caching, persistent connections, and virtual hosts.

Most of the Internet documents go through frequent revisions, so HTTP/1.1 is not the end of the story. There will be more revisions, clarifications, and improvements. As already mentioned in Chapter 8, readers must use the latest documents before implementing any RFCs.

HTTP is basically a request/response protocol. A client or a user agent sends a request to a server over a TCP/IP connection. A client that initiates a request to a server is known as a *user agent*. The request contains the server location, protocol version, MIME-like message containing the type of request, client information, and, if required, the body content. The response from the server contains the protocol version, status of the request, and MIME-like message, which itself contains the server response, entity-related information, and entity body. The status of the request indicates whether or not the request was successful. If the request was not successful, then the status will contain an error code.

In HTTP, persistent connection is the default behavior in HTTP/1.1. The connection between a client and server is always maintained and is closed only after the closing of the TCP connection is signaled by either

the client or the server. This is an improvement where a separate TCP connection is established to retrieve URL from a server. A persistent connection allows a client to make multiple requests to a server without waiting for responses.

HTTP permits access authentication whereby a server can challenge a client request and a client may be asked to provide authentication information. Digest authentication for HTTP as per RFC 2069 is also allowed (Reference 11.17).

A connection to a server from a client or user agent can be of three types, as shown in Figure 11-7. In a *proxy*, the request is rewritten and is forwarded to a server, whereas in the case of a *gateway*, the request is translated to the server's protocol and sent to the server. A proxy can be either a server or client for requests from other clients; a gateway is an intermediary server for some other servers. In the last type of connection, the *tunnel* is simply a relay mechanism that forwards the messages to the other connection.

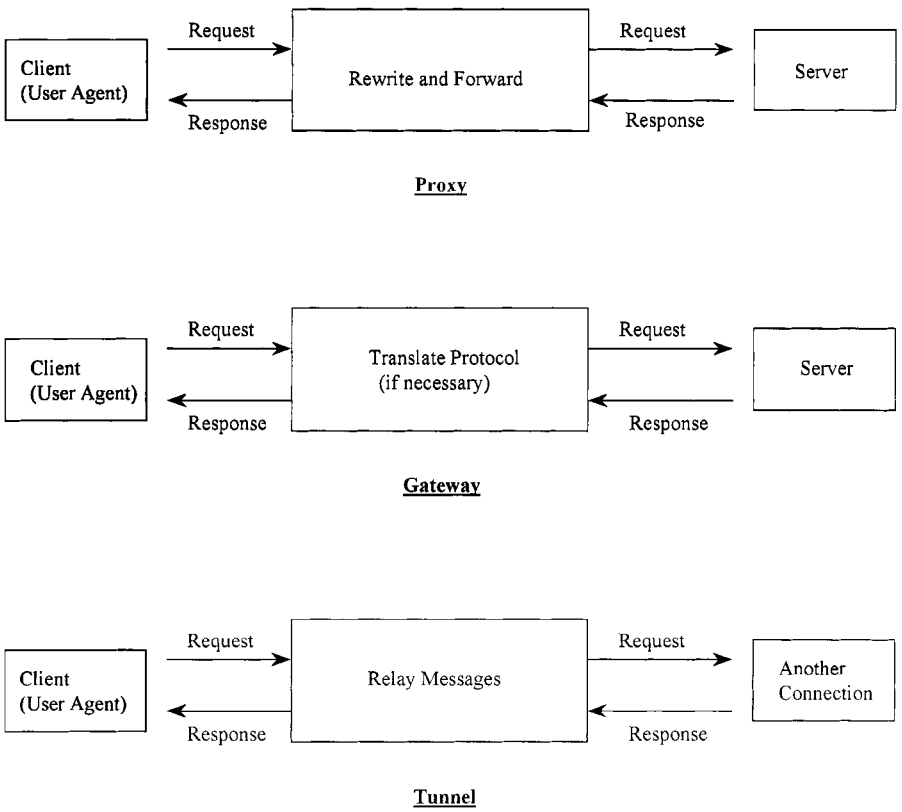


Figure 11-7 Different HTTP connections.

A Web browser (Figure 11-8), which is a client or a user agent, can use HTTP to request information from a Web server, which runs an HTTP daemon. A common gateway interface (CGI) reads the HTTP data, parses the data, and processes the data in the HTTP server. The CGI can be written in C, C++, Fortran, PERL, TCL, any Unix shell, or any language that can be executed on a system. Data can be transferred to a CGI process using a method. There are many methods available to a CGI. In response to a request from a Web browser, a Web server can retrieve HTML documents or any type of file. Files can be viewed in the Web browser.

One of the advantages of using Web browsers is that TMN applications can be viewed from anywhere. Also, it is easy to make changes, as it is only necessary to perform modifications or updates in the Web server once. These changes will be available to all the Web browsers connected to the Web servers.

Due to security considerations, it is not prudent to access all management information from a Web. For Web security, firewalls can be used to restrict the information to viewers. Some of the TMN applications that are suitable for viewing and performing changes from Web browsers are as follows:

- A customer must be able to access different telephone or Internet service providers using Web browsers. The customer must be able to compare the rates and types of services offered by different service providers. These services may be used while requesting a new telephone service or changing telephone service from one telephone service provider to another.

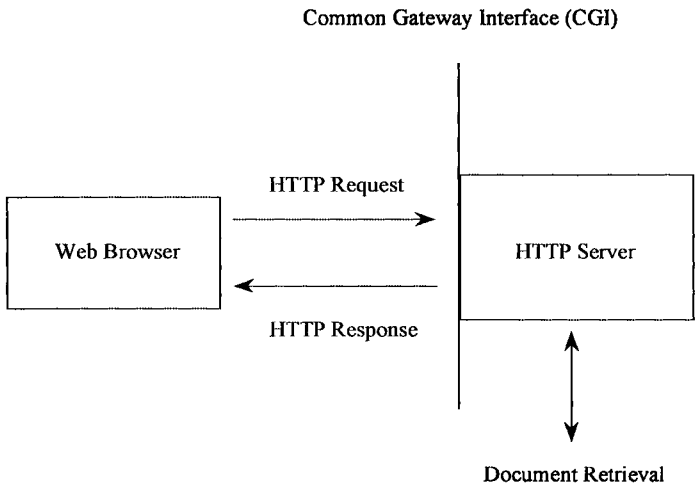


Figure 11-8
Document retrieval by Web browser.

- Once a customer shortlists the service providers, the customer must be able to negotiate the rates and types of services to be received.
- Once a telephone service provider is selected, the customer must be able to order telephone services or Internet services using a Web browser. The service provider checks these requirements and validates customer credit history. Here, the service provider provides the necessary security to protect vital customer data. The service provider also checks whether the required services can be provided on the due date and time and internally informs concerned departments and the customer about the customer order.
- Small individual customers must be able to view performance data from the Web browsers whenever required.
- In another scenario, the topology, fault, and performance data must be available to customers via Web browsers. These are useful for customers who have their own networks and use service providers to access other networks.
- A telephone customer must be able to change the types of services, such as number of telephone connections, line speeds, and so on, using the Web browsers.

As HTTP is a static protocol, it is not possible to dynamically refresh the Web pages in cases such as an alarm being received. This is where Java is useful. Java applets can be executed in different platforms and provide the dynamic capability required to update the network management applications dynamically.

There are many CORBA implementations that have integrated Java and the Web. IONA's OrbixWeb is one such implementation. Though there are many considerations to keep in mind when selecting vendor applications, the integration of CORBA and Java can be an important factor.

11.8 Web-based Enterprise Management

Some of the computer industry leaders, such as BMC Software, Cisco Systems, Compaq, Intel, and Microsoft, have joined hands to propose a Web-based architecture to manage varied types of systems, networks, and applications. This architecture was proposed in 1996. The result of this work is Web-Based Enterprise Management (WBEM). WBEM is a collec-

tion of technologies to manage enterprises and is touted as independent of vendor, protocol, or management standards.

A note of caution is required here. Especially in the computer industry, there is a tendency to form different groupings of vendors to produce "standards" for different technologies. Over the years, many groupings and bodies were formed, many meetings were held, and then the groups and bodies dissolved and the people in the computer industry forgot about them.

In a similar manner, leaders in the computer industry support WBEM. However, in the TMN arena, where mostly ITU-T, ANSI, and others guide the standards, computer industry—based technologies such as WBEM have limited impact and appeal. They cannot have a major impact in the telecommunications industry without the endorsement of industry service providers such as AT&T, BT, NTT, MCI, and others, and equipment manufacturers such as Lucent, NEC, Nortel, Siemens, Erricson, and others. In addition, the investments and development efforts required to implement the technologies in the telecommunications industry are quite large. Therefore, at best the solution for the TMN industry can be the coexistence of TMN and WBEM with some features of WBEM being utilized. Keeping in mind this note of caution, we will briefly look into the important aspects of WBEM.

The primary objectives of WBEM are:

- To manage devices and applications using Web browsers
- To support all devices and applications using existing standards and protocols
- To unify management schemas
- To provide for accessing of management information

Figure 11-9 shows the architecture for WBEM. The Internet browser is used to browse different management information. Between the Internet browser and the HyperMedia Object Manager (HMOM), HyperMedia Management Protocol (HMMP) is used. The HyperMedia Management Schema (HMMS) provides the management information. HyperMedia Object Manager and the devices or applications, communicate management information using management protocols such as SNMP, Desktop Management Interface (DMI), CMIP, HMMP, and others.

Let us look in some more detail into the different pieces of the puzzle. HMMP is the management protocol to access management information. HMMP is a request/response protocol. A process acting as a client sends management requests to a process acting in the role of a server. The server performs the requested task and sends a response to the client. HMMP

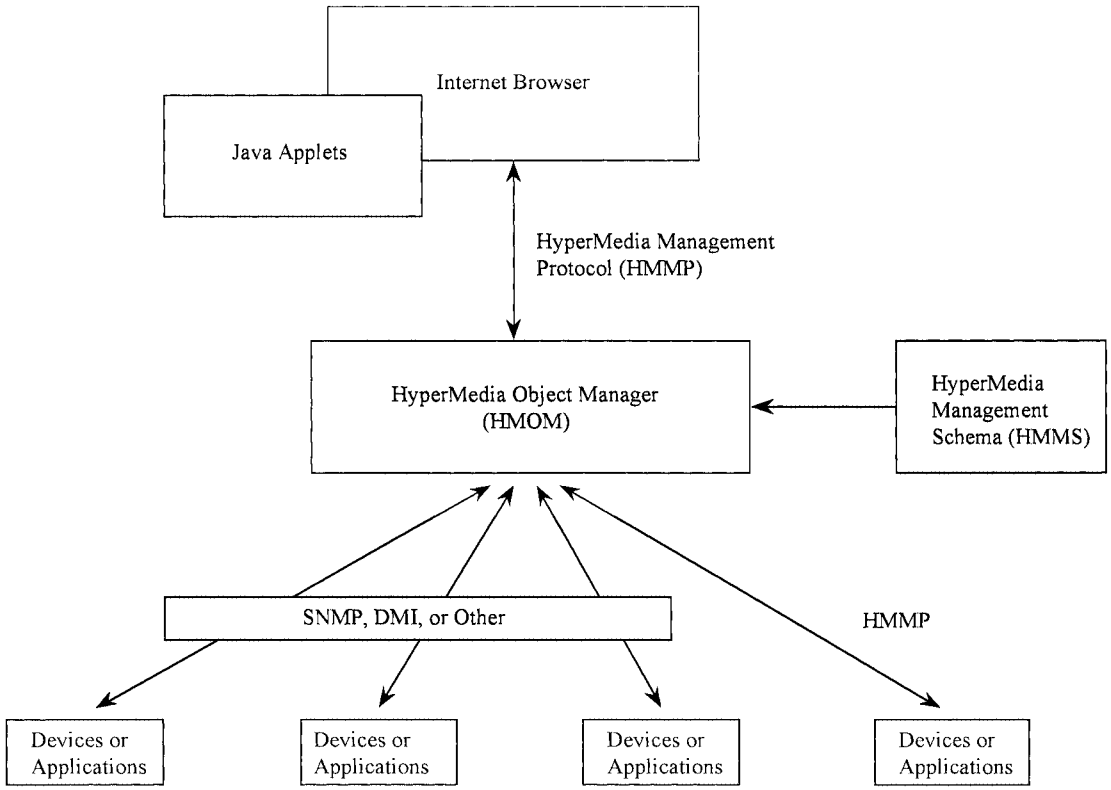


Figure 11-9
Web-based enterprise management architecture.

supports dynamic properties and associations. Work on HMMP is still being done, and there are working documents.

A server that implements most of the operations of the HMMP is known as an HMOM. A client makes a request to an HMOM (here, the HMOM is the server). If the HMOM cannot satisfy the client's request, then the HMOM switches the role to a client and forwards the client's request to another server to satisfy the client's request. Here, the HMOM acts as a proxy. The HMOM reference implementation in C++ is expected to be placed in the public domain.

Common Information Model (CIM) is the data model used along with HMMP. CIM is used to represent real-world objects on the lines of an object-oriented paradigm. The logical view of CIM is similar to the object-oriented database views. CIM is divided into metamodel and standard schema. The metamodel defines what constitutes a schema in terms

of classes, instances, and qualifiers and how these can be used to represent real-world objects. Here *qualifier* refers to a modifier, which can be applied to a class definition.

The standard schema refers to the well-published standard classes that represent well-known devices and other manageable objects. In the case of new devices, either the class definitions have to be inherited from the standard class hierarchy or an entirely new class hierarchy has to be created. All servers are expected to implement the standard schema. This requirement brings some standardization to the classes supported in servers. The components of a schema, such as object classes, object instances, and qualifiers, can be described in the textual format using Managed Object Format (MOF).

CIM supports references that are just a sort of pointer to other classes or instances within the schema. References are in the form of a string of object paths. A pair of references is known as an *association.* Instances of association classes are used to establish binding relationships between object instances or object classes.

Object instances are identified by keys, just as in relational databases. Classes and instances are grouped into namespaces. Namespaces can be nested in a hierarchical manner and can be manipulated by using a standard class _Namespace. HMMP protocol operation specifies in which namespace a given protocol operation is being performed. This is also known as *namespace binding.*

HMMP does not provide transport level functionality. For transport protocol, HMMP proposes to use HyperMedia Transport Protocol (HMTP). HMTP is basically a transaction-oriented transport protocol. A unique transaction identifier identifies the request-response transactions. This transaction identifier can be used to identify the responses with the requests. HMTP permits explicit and implicit acknowledgments. In the implicit acknowledgments, the response to a request is the acknowledgment of the request itself. The explicit acknowledgment is useful for lengthy operations. In addition to the HMTP for HMMP transport, mapping of other transport protocols such as TCP is also being considered. Use of different prevalent transport protocols is very essential if the HMMP is to cover a wide range of transport protocols in use.

In HMMP, a notification or event is referred to as an *indication.* The source of an indication is known as a producer and the receiver of the indication is known as the *consumer.* Registration to receive notifications is required to receive indications. An indication can be acknowledged or unacknowledged.

There are different levels of security. Level 1 security requires the authentication of the client. This is enough to protect a server from unauthorized access from average users with no mala fide intentions. Security level 2 is not yet defined. It is proposed to have this level use an RSA public key at first for authentication and RC4 symmetric cryptography for subsequent transfer of PDUs between client and server.

HMMS describes the management information data and is owned by Desktop Management Task Force (DMTF). DMTF is yet another computer industry consortium devoted to development, support, and maintenance of management standards for desktop computers and products. DMTF was started in 1992 by PC industry leaders. The prominent members of this consortium are Microsoft, Intel, IBM, Compaq, Dell, NEC, SunSoft, and others.

Desktop Management Interface (DMI) is the management standard for describing and accessing information about all types of PCs and PC components. DMI specification DMI 2.0 was completed in 1996. The latest document, DMI 2.0s, includes DMI 2-0 and security extensions. DMI further enables remote troubleshooting using DMI access to information about any system in the network. DMTF is also working on the common information model to model the management information data in a standard object-oriented manner. For more information on DMTF, DMI, and CIM, interested readers may visit the DMTF web site (http://www.dmtf.org).

11.8.1 WBEM and TMN

Our interest in WBEM is on how WBEM and TMN can be integrated. A large amount of the material discussed here is taken from a Vertal white paper on WBEM and TMN. Here we will only briefly discuss the WBEM and TMN integration.

In Figure 11-10, the HMMP client and HMOM are the WBEM components; we have already briefly looked into these. Managers and agents are the TMN components. Two new components are introduced in the architecture for protocol mapping and informational model mapping. The HMMP TMN gateway is an important component that acts as a link between WBEM and TMN environments. It performs mapping of HMMP operations to CMIP operations and vice versa.

Another important component required for the informational model mapping is the metadata compiler. TMN uses the GDMO/ASN.1 for defining MIBs. In the case of WBEM, CIM is the informational model.

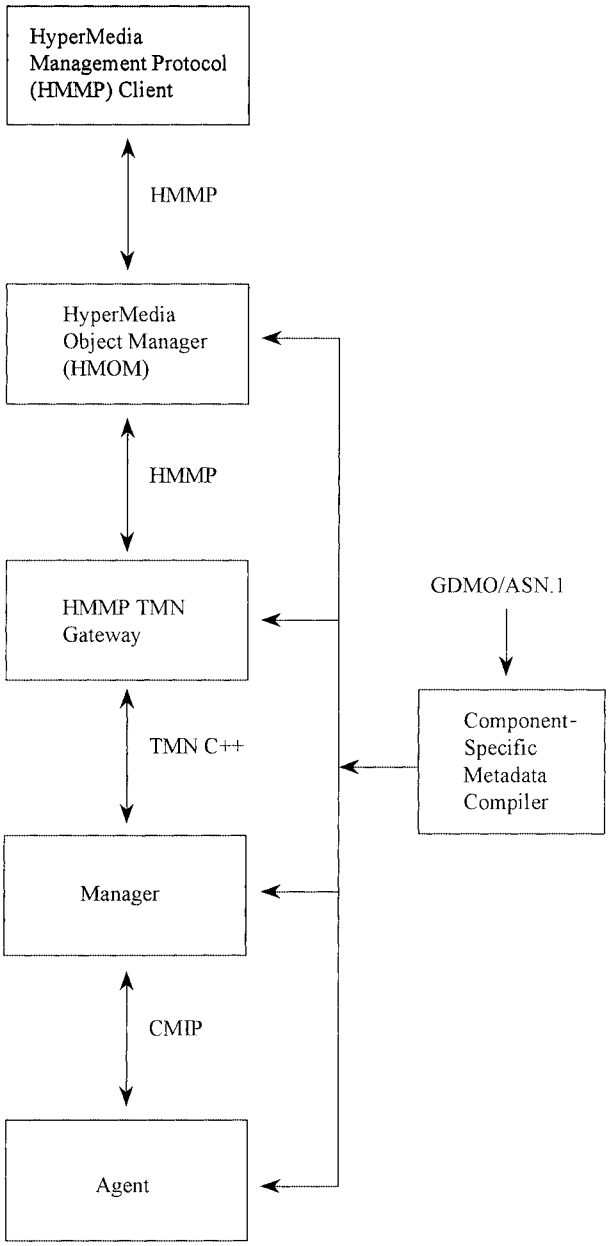


Figure 11-10 WBEM and TMN integration.

The metadata compiler produces C++ header files and code based on the TMN MIBs used, and makes them available to the manager, agents, and manager applications. However, for a WBEM environment, the metadata compiler must be able to generate the MOF from the MIB. This MOF is imported to the HMOM and the CMIP-TMN gateway.

Namespace and naming in CIM are slightly different than the RDN and DN. So mapping of names is one of the issues when WBEM is to be integrated into TMN.

11.9 Java and TMN

Java is useful in TMN applications for providing platform independence. Java applets, which are just TMN applications, can provide the fault, performance, service management, or similar applications.

Some of the salient points of Java are:

- Java is an interpreter language. Therefore execution of the code is slower than in those cases in which the run time codes of languages such as C++ are used.
- Java uses no pointers.
- Java has no union types.
- Java clients are platform independent.
- Java code executes where it resides.
- Java clients and CORBA objects can combine well to create dynamic content in Web pages.
- Java has a built-in native ORB called remote method invocation (RMI). RMI facilitates method invocation on remote objects. RMI provides similar functionality to ORB, but is limited to Java environment.
- Alerts and notifications between CORBA objects and Java clients can be easily implemented. This property is very useful for depicting details of alarms in Web browsers.

Java Abstract Window Toolkit (AWT) is available for developing application GUIs. Java AWT supports standard widgets such as scrollbar, checkboxes, and different layout options. Java also supports access to different databases. Java furnishes a set of classes for Internet connectivity, parsing of URLs, HTTP messages, and HTML.

Java is very popular in the programming community. From a TMN perspective, for providing dynamic capabilities to Web browsers and for taking advantage of distributed environment provided by CORBA, there is a need for Java to IDL mappings and vice versa. Because of the differences in the syntax and semantics of Java and IDL, some of the IDL features are mapped indirectly to Java as given in the following list:

- Unsigned IDL integers are mapped to Java signed counterparts.
- IDL structs are mapped to Java classes.

- IDL unions are mapped to Java classes.
- IDL sequences map to Java array.
- IDL exceptions are mapped to Java classes.
- As Java has no pointers and reference arguments, IDL out and inout are handled by Holder Objects.
- IDL permits aliasing by using typedef. However Java lacks the concept of aliasing. So while mapping IDL alias, the Java type is mapped to the underlying IDL type.

Java features have been extended by Sun Microsystems to provide network, systems, and service management solutions. This collection of extensible objects and methods is known as JavaManagement API (JMAPI).

11.10 Java Management API

JMAPI can be used on different operating systems, network protocols, and various computer architectures. JMAPI consists of the following features:

- *Admin view module (AVM):* An extension of Java Abstract Window Toolkit (AWT). AVM can be used to develop user interfaces for distributed management applications. The AVM base set consists of image button, multi-column lists, scrolling windows and panels, state button, toolbar, image canvas, convenience dialogs, and busy tool. In addition, applications are available to build help systems, tables, charts, and graphs.
- *Base object interface:* Supports construction of objects for distributed services and resources. These allow defining distributed attributes and methods and persistent attributes.
- *Managed container interface:* Permits users to perform management operations on a group of object instances of a managed object.
- *Managed notification interface:* Allows asynchronous event notifications between managed objects or management applications. This feature is useful for building event management services.
- *Managed data interface:* Useful for mapping managed object classes and instances to a relational database. These interfaces are available on different relational databases.
- *Managed protocol interface:* Classes provided in this feature can be used to build an infrastructure to perform distributed operations in

a secure manner. The security is provided by validating that only trusted Java code runs on a client. The remote invocation method authenticates all requests for Java classes and platform-specific native libraries in a server.

- *SNMP interface:* Can be used to extract information from SNMP agents. This feature has only limited application and can be used only where SNMP protocol is used.
- *Applet integration interface:* Permits integration of Java applets with JMAPI.

JMAPI is a useful toolkit for developing network management applications. It integrates well with Java as both have been developed by Sun Microsystems. For more information on the JMAPI, visit the Web page http://java.sun.com/products/JavaManagement/overview.html.

However, note that URLs of Web pages and contents of Web pages frequently change and that the latest information should be used when designing and developing TMN applications.

11.11 Summary

In this chapter we have looked at the recent trends in TMN. We have introduced the concepts of distributed processing and distributed network management. CORBA implementations are becoming popular in the TMN arena for providing distributed network management. Web-based TMN is also an increasingly popular trend. Many TMN applications, especially in the service management area, are being accessed via Web browsers. WBEM is another promising network management architecture in the desktop arena; therefore we have briefly looked into WBEM and also one of the ways WBEM and TMN can coexist. Finally, Java has generated a keen interest because of the platform independence and simplicity of the language. It is becoming popular as a language of choice in many network management and TMN areas. We have concluded the chapter by taking a look at JMAPI from the perspective of its use in TMN.

Integrated Network Management V, Integrated Management in a Virtual World, Proceedings of the Fifth IFIP/IEEE International Symposium on Integrated Network Management, has a number of very good articles on the CORBA, Java, and Web implementations in TMN. References to some of the important articles are furnished in the next section.

11.12 References

11.1. Soukouti, N. and U. Hollberg, *Joint Inter Domain Management: CORBA, CMIP and SNMP.* Integrated Network Management V, Integrated Management in a Virtual World, Proceedings of the Fifth IFIP/IEEE International Symposium on Integrated Network Management, San Diego, CA, May 12—16, 1997, Lazar, A., Saracco, R., and Stadler, R. (eds.), London: Chapman & Hall, pp. 153—164, 1997.

11.2. Chadha, R. and S. Wuu, *Incorporating Manageability into Distributed Software.* Integrated Network Management V, Integrated Management in a Virtual World, Proceedings of the Fifth IFIP/IEEE International Symposium on Integrated Network Management, San Diego, CA, May 12—16, 1997, Lazar, A., Saracco, R., and Stadler, R. (eds.), London: Chapman & Hall, pp. 489—502, 1997.

11.3. Whitner, R. B., *Designing Scalable Applications Using CORBA.* Integrated Network Management V, Integrated Management in a Virtual World, Proceedings of the Fifth IFIP/IEEE International Symposium on Integrated Network Management, San Diego, CA, May 12—16, 1997, Lazar, A., Saracco, R., and Stadler, R. (eds.), London: Chapman & Hall, pp. 503—514, 1997.

11.4. Feldkhun, L., M. Marini, and S. Borioni, *Integrated Customer-Focused Network Management: Architectural Perspectives.* Integrated Network Management V, Integrated Management in a Virtual World, Proceedings of the Fifth IFIP/IEEE International Symposium on Integrated Network Management, San Diego, CA, May 12—16, 1997, Lazar, A., Saracco, R., and Stadler, R. (eds.), London: Chapman & Hall, pp. 17—30, 1997.

11.5. ITU-T Recommendation X.703, Information Technology—Open Distributed Management Architecture, 1997.

11.6. ITU-T Recommendation X.901, Information Technology—Open Distributed Processing—Reference Model, Part 1: Overview and Guide to Use, 1997.

11.7. ITU-T Recommendation X.902, Information Technology—Open Distributed Processing—Reference Model: Foundations, 1995.

11.8. ITU-T Recommendation X.903, Information Technology—Open Distributed Processing—Reference Model: Architecture, 1995.

11.9. ITU-T Recommendation X.904, Information Technology—Open Distributed Processing—Reference Model: Architectural Semantics, 1997.

11.10. Revision 2.2, *The Common Object Request Broker: Architecture and Specification,* Framingham, MA: Object Management Group, 1998.

11.11. Updated Version, *CORBA Services: Common Object Services Specification,* Framingham, MA: Object Management Group, 1997.

11.12. Harssema, M., *Integrating TMN and CORBA,* Computer Science Masters Thesis, Enschede, The Netherlands: University of Twente, 1996.

11.13. Raud, R., *OSI and CORBA, Alternatives or Complementary.* Paper in Sonet Interoperability Forum (SIF), 1998.

11.14. Fielding, R., J. Gettys, J. C. Mogul, L. Masinter, P. Leach, H. Frystyk, T. Berners-Lee, N. Freed, and N. Borenstein, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies,* RFC 2045, 1996.

11.15. Freed N., and N. Borenstein, *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types,* RFC 2046, 1996.

11.16. Moore, K., *MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text,* RFC 2047, 1996.

11.17. Franks, J., P. Hallam-Baker, J. Hostetler, P. Leach, A. Luotonen, E. Sink, and L. Stewart, *An Extension to HTTP: Digest Access Authentication,* RFC 2069, 1997.

11.13 Further Reading

Web-Based Enterprise Management Initiative Web Page, http://wbem.freerange.com.

Accessing TMN Through Web-Based Enterprise Management, Vertel White Paper, 1997.

Web-Based Enterprise Management, Microsoft Web page, 1996.

Won-Ki Hong, J., J.-Y. Kong, T.-H. Yun, and S.-S. Kim, "Web-Based Intranet Services and Network Management," *IEEE Communications Magazine,* vol. 35, no. 10, pp. 100—110, 1997.

Mapping of IDL to Java. JavaSoft, 1996.

Weiss, M., A. Johnson, and J. Kiniry, *Distributed Computing, Java, CORBA and DCE,* 1996.

Raman, L., Information Modeling and Its Role in Network Management. In *Telecommunications Network Management,* Aidarous, S. and Plevyak, T. (eds.), New York: The Institute of Electrical and Electronics Engineers, Inc., pp. 1—62, 1998.

This page intentionally left blank.

CHAPTER 12

Software Management Frameworks, TMN Challenges, and Trends

12.1 Introduction

In this chapter, we discuss object-oriented software management frameworks. Many vendors are already providing different TMN solutions including software management frameworks based on object-oriented paradigms. Therefore, it is essential to take a look into the basic principles involved in these management frameworks.

Basically, these software management frameworks assist in developing a variety of management applications such as managers and agents. It is better to procure these management frameworks to develop TMN solutions instead of developing TMN solutions from scratch. A rule of thumb is: If you can procure a TMN solution that meets your needs at a competitive price, do so. As an extension to this rule, only if you cannot get the TMN solution you want on the market at a competitive price should you opt for developing network management solutions.

Intense competition in the telecommunications industry is playing a key role in the services provided and in improvements in technology. Every telecommunications service provider and vendor wants to expand its array of services and is trying to offer more and better services than its rivals. As a result, new technologies to cater to new customer requirements are being developed at a fast rate. These new technologies need network management. Also, in many cases, new services are built on existing network infrastructure. The network management solutions for these new services have to coexist with the legacy network management solutions.

In addition, TMN is evolving with many gray areas and a large number of unresolved issues. We need to take a close look at the most important of these issues in order to help designers and implementers to recognize the problems they may encounter while providing TMN solutions.

At the same time, it is absolutely essential to be aware of the trends in TMN. This will aid in providing solutions from a long-range perspective. These solutions, which take note of the future migration issues, will have fewer problems in migrating to the future TMN solutions. Such solutions will require minimal changes to accommodate new technologies. The unresolved issues and future trends in network management are also explained in Reference 12.1.

12.2 Management Frameworks

TMN information models and management protocols are based on object-oriented principles. Some of the major advantages of using object-oriented paradigms are:

- The code written for a particular object class can be reused to satisfy additional requirements by making minor changes.
- Extensibility means that new object classes can be easily derived using inheritance and polymorphism. This permits easy and elegant extensions to object class definitions.
- It is easy to model and map the real systems and resources to object class and objects. The definitions of object classes make sense because they relate to real systems.
- Encapsulation permits hiding of internal details including data. This enables data integrity, as only operations to be performed on an object are visible at the boundary of the object.

The trend in TMN is to use object-oriented management frameworks. These can be used to develop MIBs, managers, agents, and network management applications. The management frameworks provide very generic functionality. The generic managers, agents, or applications have to be modified to meet specific requirements.

Object-oriented methodologies are also used for the design and development of management frameworks. Some of the popular methodologies are based on Paradigm and Unified Modeling Language (UML).

The earlier practice was to use proprietary languages in the development of network management systems. This is slowly giving way to the use of generic programming languages such as C++ and Java. As it is easy to map object-oriented methodologies to C++, it is one of the most popular development languages.

Figure 12-1 gives an overview of the structure of a software management framework. However, each vendor incorporates slight variations in the actual implementations to distinguish its product offerings. Individual components are discussed in the following text.

12.2.1 Communication Infrastructure

The basic function of the communication infrastructure is to provide a communication stack on computer platforms to enable the transfer of

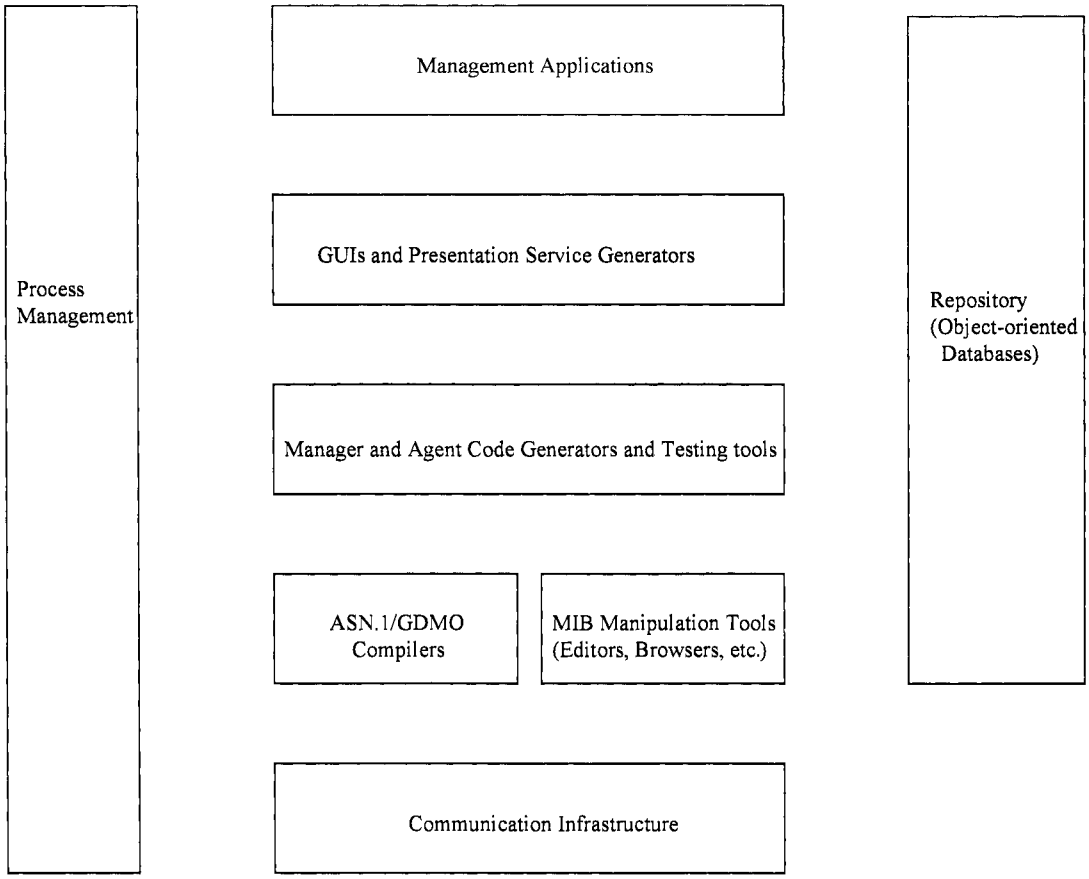


Figure 12-1
Software management framework components (© 1994 IEEE).

management information between managers and agents. This concept can also be extended to provide communication between other TMN layers such as NML, SML, BML, and network elements.

However, many computer platforms provide a variety of communication stacks such as ACSE/presentation layer stacks. There are software packages available for computer platforms to provide a Q3 interface along with the communication stacks. Some use TCP/IP or CORBA IIOP. Some of the computer platforms provide RFC 1006 to run CMIP over TCP/IP. These ready-to-use communication stacks are not enough to convey management information. In management frameworks, communication

adapter object classes have to be provided to integrate these ready-to-use communication stacks with the manager or agents.

12.2.2 MIB Manipulation Tools

Most of the management frameworks provide a variety of tools for the creation, editing, and updating of MIBs. There are tools to store the MIBs in a persistent storage. This persistent storage can be an object-oriented database or any of the popular relational databases. Some vendors support most of the popular databases.

GDMO/ASN.1 editors are a common feature used to develop the GDMO/ASN.1 definitions for the managed object classes. These editors have easy-to-use GUIs that assist in developing GDMO and ASN.1 definitions of object classes. Syntax checking is done while the object class definitions are being inputted through the GUI, and there are prompts for errors if wrong entries are made. After syntax checking, these managed object class definitions can be stored in a persistent storage with unique names. The set of object classes in persistent storage is also known as an *object class repository.*

GDMO/ASN.1 editors can also retrieve the object class definitions from persistent storage, update the definitions, and store them again in persistent storage. These GDMO/ASN.1 definitions become input to the GDMO/ASN.1 compilers. If the framework components are purchased from the same vendor, different components are properly integrated. Otherwise, if the framework components are from different vendors, there may not be tight integration of products. As a simple example, if the editors and compilers use different naming syntaxes for files in the repository, there can be problems.

GDMO browsers are another important feature. After error-free object class definitions are compiled, GDMO browsers display the object class inheritance hierarchy and containment hierarchy. These browsers have flexible capabilities to display the object class hierarchies with zoom in, zoom out, and partial displays. There are also capabilities for printing in different formats.

Note that in the preceding explanations, different vendors can use different functionalities in their editors and browsers. We have made the distinction between editors and browsers based on use. Browsers are used after the correct compilation of the GDMO and ASN.1 definitions to view and print the hierarchies, whereas editors are used to make changes to the GDMO and ASN.1 definitions.

12.2.3 GDMO and ASN.1 Compilers

Usually, TMN development frameworks have ASN.1 and GDMO compilers. These compilers accept GDMO/ASN.1 definitions, do the syntax checking, and produce header files and data structures and store them in repositories. These header files and data structures are used for developing managers and agents. Framework libraries also provide support for object filtering, scoping, persistence, object creation, and object deletion services.

The GDMO/ASN.1 definitions are normally stored in files in repositories: They are retrieved from the repositories and then compiled. The normal practice in GDMO and ASN.1 compilers is to prompt for an error, then invoke the GDMO or ASN.1 editors and point to the line/place where the error has occurred. The supporting ASN.1 definitions must be compiled first, using ASN.1 compilers, and then GDMO definitions are compiled. Once these GDMO/ASN.1 definitions are correctly compiled, browsers can be used to display the inheritance and containment hierarchies.

Most of the TMN frameworks are based on standard MIBs such as X.721 and M.3100. This dependency on the standard MIBs places restrictions on deriving new managed object classes, as the new managed object classes are inherited from the object classes defined in X.721 and M.3100.

12.2.4 Manager and Agent Code Generators

Once the GDMO and ASN.1 definitions are correctly compiled, code generators can be used to generate the manager and agent codes. These code generators produce C++ headers and implementation codes for managers and agents. The implementation codes are customized for manager and agent operations by adding further code for association, accessing and searching directories, and communication of management information.

Let us look into the issue of management association a bit further. As we saw in Chapter 7, an association has to be formed between a manager and an agent before any management information can be transmitted using CMIP. For this, the manager has to know the location of the agent. A manager needs the assistance of directory services to locate the agents. These directory services can be X.500 based. These directories have AE-titles or distinguished names used to form associations, so the manager has to have code to go to the directory and search for either the AE-title or the distinguished name of the agent. Here we have made the assump-

tion that the manager is initiating the association. However, there is no such restriction; agents can also initiate an association.

Usually, the implementation codes generated by code generators have markers indicating where the specific customized codes are to be added.

Figure 12-2 shows the flow for generating manager and agent codes. There are return paths to the beginning of the ASN.1 definitions and GDMO definitions from the manager code generator and agent code generator, the manager code, and the agent code. In many cases it is necessary to go back to the original ASN.1 and GDMO definitions and correct them, then go through the cycle again as shown in the figure.

The ASN.1 and GDMO compilers and manager and agent code generators can have enhancements so that they only recompile or generate the codes for the changes made and incorporate them appropriately in the compiled code or generated code.

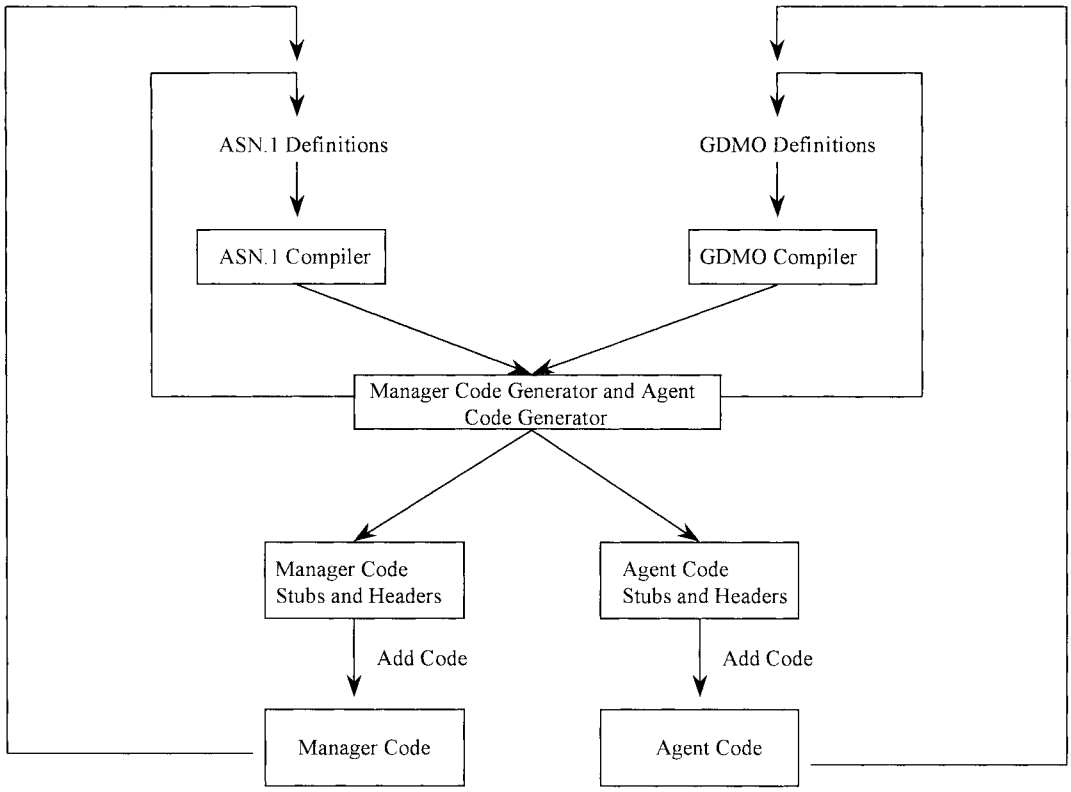


Figure 12-2
Generating manager and agent codes.

In addition to the code generators, many management frameworks provide tools for testing managers and agents in stand-alone environments. As an example, a tool for testing a manager provides for association and simulates the environment for an agent. The manager test tool also simulates different CMIP commands. For easy interface, GUIs are provided. These tools can be used to perform initial predeployment testing of managers and agents.

12.2.5 GUI and Presentation Service Generators

Usually, GUIs insulate the user from the complexities of implementations and present simple menu bars or icon objects to access the manager applications, ASN.1 and GDMO compilers, and code generators. The parameters are entered through dialog boxes. In GUIs, when erroneous data are entered, the user is prompted to correct the errors with appropriate and meaningful messages. Also, help is generally available to explain the steps involved in correctly entering the data and using operations by clicking buttons. Usually in GUIs different colors and different audio and video effects are included to depict different states.

There are many commercially available tools on the market to aid in rapid development of GUIs. The wisest practice is to develop these GUIs in the early stages and listen to customer feedback. In software development, GUI development is complex and very subjective. As everyone has a strong opinion on how a GUI should look, major problems can occur at a later date if customer buy-in does not occur early on.

12.2.6 Persistent Storage

MIBs are normally stored in persistent storage. The persistent storage can be object-oriented databases or relational databases. Object-oriented databases are best suited for storing MIB data structures, as the structures used in these databases are similar to the structures of object classes. When relational databases are used, conversions from the relational record structure to the object class structures are required while storing and retrieving MIB object classes. These conversions can affect the performance of management systems.

Logging of different types of data, including security audit trails, also necessitates storing data in persistent storage. These data can be stored in flat files or in databases as log records. The flat file approach is simple to implement and provides faster logs without size limitations, whereas data-

base logs can have limitations of size and require more time to process each record. The data stored in databases are easier to manipulate as some of the database operations are provided by database tools.

There are many vendors that provide object-oriented databases, and it is becoming a popular trend to use these databases in TMN for providing persistence.

12.2.7 Process Management

Communication between different components of a network management system is accomplished with the help of *processes*. Some of the examples of processes are manager processes, agent processes, client/server processes, and so on. Most of these are cooperative processes in the sense that they have to be well coordinated. Processes have to be maintained properly so that there is no failure in the computer platforms where they reside and execute. In addition, processes have major impact on performance and storage space utilization. Multithreading is an example of how processes can be used to improve performance. The designer of any management platform has to carefully examine the performance and storage space allocations of the end products generated by processes.

12.2.8 Management Applications

The development of managers and agents is a small part of the TMN development cycle. These days, there are many applications, such as fault, performance, and performing service-level activities such as telephone number administration and billing. New applications are frequently being developed to meet customer requirements.

The primary objective of network management application tools is to automate as much as possible the code generation phase for these applications. However, these management applications will have to be customized by adding—and sometimes changing—application code to meet specific requirements.

There are also management application platforms that provide libraries of object classes for implementing systems management functions such as event management, scheduling, software management, alarm management, and so on. As an example, let us take the case of alarm management. This application provides the ability to collect alarms, display the alarms in different order, display the alarm history, filter the alarms based on certain criteria, retire the alarms, and so on.

It is becoming a normal practice to buy different application packages to enhance management systems. Trouble ticket systems, which track problems from inception to closure, constitute one such package. Trouble tickets can be used to coordinate different activities such as managing field resources, providing customer-related reports, generating statistical reports, and so on. Trouble ticket applications can involve different levels of complexity and automation. As an example, when a problem is reported, an e-mail or paging system can notify the proper repair personnel to attend to the problem.

There are two possible scenarios in the development of management applications. In the first, the design has the tools on hand to generate management applications. The other case is that of packaged management software applications. These applications can be bought as separate pieces and integrated into network management applications; an example is the integration of trouble ticketing and fault management.

Application platforms invariably come with easy-to-use GUIs. These GUIs enable the use of the applications without requiring a high level of computer literacy and are designed to avoid the learning curve involved in becoming well-versed in the use of a tool.

12.2.9 Implementation Notes

Many factors must be considered in the selection of management frameworks to generate managers, agents, and management applications. Some of these are:

- The managers and agents generated are dependent upon the information model used in the management frameworks. As an example, if the management framework is based on the M.3100 information model, the generated managers and agents will work only with managed object classes derived from M.3100-based object classes.
- There should be support for multithreading. This is required for enhanced performance, especially when processing linked replies.
- A fast GUI development tool is required to do initial prototyping. This is essential because feedback provides direction on how to improve the GUIs to meet internal and customer requirements. Prototyping is also important to garner support for a project.
- A good GUI interface is required for debugging. Debugging is an important, time-consuming activity during the development process; easy GUI interfaces can facilitate quicker development.

- Database support is required for providing persistence. This database support has to be available in the platform on which EMs and NEs are deployed. As an example, if the generated code is dependent on *X* object-oriented databases, then *X* object-oriented database support should be available on the development computer platform and the production platform. By *production platform,* we mean the computer platform on which NEs and EMs are deployed. *X* object-oriented database support is also required if the EMs and NEs are to be ported to other platforms; otherwise the porting itself may involve a major rewrite and thus retesting the code.
- There is a need to consider computer platform incompatibilities. Management frameworks and management applications are normally developed in one computer platform and then ported to different computer platforms. Incompatibilities between computer platforms can pose major problems when porting to different platforms. Many computer platforms have apparent (and sometimes none-too-apparent) differences in their support for data models.
- Just as there are platform incompatibilities, there are software incompatibilities. As an example, Orbix implementations on Sun Solaris and Windows NT are slightly different.

Interested readers may also consult the chapter on management platforms in Reference 12.2.

12.3 Unresolved Issues and Challenges in TMN

Although from many standpoints TMN is in a fairly advanced stage, there are still many challenges to be faced. Let us examine some of these.

12.3.1 Integration of Legacy Systems and Standard TMN Solutions

Many telecommunications service providers are still clinging to the old proprietary systems. To complicate this problem further, some of the TMN solutions have been on legacy computer platforms and different forms of databases. These disparate TMN solutions have grown indepen-

dently over time. In addition, telecommunications service providers have made huge investments in proprietary solutions.

Competitive pressures compel these service providers to update their technology. Some service providers see this as an opportunity to change over to the latest technologies and standard solutions. However, due to the large investments of money, time, and personnel, transformation from proprietary legacy TMN solutions to standard solutions will be evolutionary.

Migrating data from old to new systems presents a major challenge in introducing new TMN solutions. As an example, some databases may be proprietary, but the latest trend in TMN is to use object-oriented databases. This variance in the formats of the databases can create a major migration nightmare.

12.3.2 Impact of Changes in Regulatory Environments

Many nations are instituting new regulations designed to increase competition. As an example, in the United States a customer can choose to use any local or interstate and international carrier. Historically, a telephone number has been assumed to be the property of the service provider; now, however, the customer is more and more being considered the owner of the telephone number.

As a first step, local telephone number portability is being implemented across the United States. This means a customer can use any local telephone service provider. This regulatory change has an impact on telephone number administration; in the years ahead, it will extend to the international scene. Such regulatory changes will have their own impact on TMN solutions.

12.3.3 Integration of TMN Solutions for Different Technologies

A large telecommunications network can be a combination of many different telecommunications and data communication networks with different technologies. As an example, one network may be using GSM in one portion, ATM in another portion, and SDH in still another portion. For seamless end-to-end data flow, management of these disparate networks must be integrated. In addition, network management solutions for the different technologies have to be integrated to meet customer

requirements for quality service at a reasonable price. To cater to increasing customer requirements, advances in technology will continue to provide more features and better services.

12.3.4 Automation

Automation is a very loosely used term. For some, it means replacing manual TMN work by some form of automatic process without any manual intervention. As an example, sometimes it is necessary to download the latest versions of software for a piece of equipment such as a switch. This can be done by manually loading the latest software using a tape or CD-ROM; however, the software can also be downloaded to the switch from a processor at a designated time.

Many telecommunications service providers and vendors use different levels of automation to provide customer services and to generate reports on telephone numbers, system availability, repairs, and billing. Many times these data are maintained in different systems operated by different people. For many services, there is a need for synchronization of data. Even in those cases where trouble ticketing is used for tracking the repairs, the degree of automation varies. Automation at the service management layer has the potential to provide significant savings to telecommunications service providers.

12.3.5 Information Model Differences

There are differences between different data models. For example, CORBA uses IDL to define the interface data model, compared to TMN, which uses GDMO and ASN.1. On top of this, there are proprietary data models. Every vendor claims that its solutions are the standard ones. However, these data models and language environments are incompatible with one another. These differences in data models create many problems when migrating from one data model to another, and when different TMN systems are to co-exist.

12.3.6 Protocol Differences

Other problems are presented when different management protocols have to interact. This is one of the major hurdles when different management

solutions are to interoperate with one another. As an example, SNMP and CMIP protocols are different. Many major telecommunications service providers have a mix of management protocols, such as SNMP, CMIP, and even proprietary management protocols, within the company.

Now imagine the problem of protocol differences if computer and telecommunications networks are to be integrated on a global level. The computer networks have their own variety of management protocols and applications developed just for network maintenance. These management protocol differences can be a major problem in providing an integrated network management solution.

12.3.7 Language Independence

There are subtle programming language differences. Compilers and linkers have some dependence on the computer platforms. Also, some computer platforms may not support all the features of a programming language. This close tie between programming languages and computer platforms makes TMN solutions both language and platform dependent.

Java is expected to be platform independent, which may solve some of the platform dependency problems of TMN.

12.3.8 Platform Independence

Despite all the talk of standardization of management and transport protocols, TMN solutions are in fact based on computer platforms. In the early days, UNIX was claimed to be platform independent; as time went by, however, we saw different flavors of incompatible UNIX systems. So platform independence in the sense that a TMN solution in one platform must be able to work seamlessly on another platform is still a long way off.

There are major differences especially in the communication stacks of different computer platforms. Many UNIX systems have ACSE/Presentation OSI-based stacks; however, these may not be available in non-UNIX platforms. So the communication stack itself can become a major hurdle in porting applications from UNIX to non-UNIX environments. Similarly, object-oriented databases are platform dependent in many cases, making provision of management solutions in a platform-independent manner problematic. As an example, if an object-oriented database vendor

supports only some platforms, it may be difficult to port management solutions from one platform to another.

12.3.9 Presentation Services/GUI

The user presentation services through which a network operator interfaces with the management applications should have a uniform look and feel. However, there are not many standards from a GUI point of view. Personal experience shows that GUIs are the most controversial aspect in software development. There is a need to focus on the standardization of presentation services and GUIs.

12.3.10 Standards Lag Behind Solutions

When new equipment is integrated into networks, network management becomes a major issue. Also, the size of networks makes manual intervention and repair difficult. At the same time, customer requirements for better and more reliable service at a cheaper price are continuously increasing. TMN will play a major role in satisfying these requirements.

However, TMN solutions have to be worked out in parallel with the development of equipment. Because of the complex, heterogeneous nature of the networks and the spread of networks over international borders, the solutions have to be open. Proprietary solutions, which were acceptable earlier, can no longer be tolerated in the fast-changing telecommunications industry. Here standards and standardization become critical requirements. The multiplicity of standards bodies and the overlapping work of these bodies are major impediments to speedy implementation of TMN solutions.

Also, in many cases, standards take a long time. They get bogged down in different levels of politics. In addition, the keen competition between different computer and telecommunications vendors also prevents uniform acceptance of standards. These factors impede the quick launch of products and the implementations of TMN solutions, especially when the technology is changing at a fast rate. Also, most telecommunications standards are based on ITU-T standards. However, due to the large size of ITU-T and the number of players involved in the standardization effort, sometimes ITU-T standards lag behind the TMN solutions required. To overcome the lacuna, ITU-T standardization efforts will have to keep pace with technological advances.

12.3.11 TMN Solutions Are Not Open

It is a common practice to claim that a given TMN solution is open. Despite a good amount of standardization in the TMN arena, seamless communication, interaction and management function between open systems is still not a reality. For various reasons, such as standards not being available for all the resources that have to be managed, proprietary solutions become part of the open system solutions to some degree. In some cases proprietary solutions are deliberately introduced for intellectual property and competitive reasons. Therefore it is necessary to be aware of the proprietary hooks in network management systems when evaluating TMN solutions.

12.3.12 Integrated Network Management for Computers, Telecommunications, and Televisions

As we have mentioned many times, the role of telecommunications is expanding; it is no longer limited just to POTS. Television is another common medium that has invaded a large number of homes worldwide, having penetrated more areas than telephony. Therefore cable operators see the opportunity to provide telephone-related services.

In addition to telecommunications and cable TV, there are many LAN-based computer networks that are expanding their spheres of influence. Service providers with data communication networks want to provide Internet telephone services. At the same time, many telecommunications service providers are already providing Internet services. Each sector is competing for services to provide to customers. As a result, new services are being offered to customers by a variety of service providers. These changes have a large impact in service provisioning, network architecture, and pricing (Reference 12.3). This will be a major challenge to some telecommunications service providers and vendors.

The demarcations or lines of separation between computers, cable TV, and telecommunications are becoming blurred. As a matter fact, they are slowly beginning to overlap with respect to services provided. Internet service is an example of this overlapping of the services provided by the computer, television and telecommunications industries. Internet service providers want to provide telephone service as well, and in some cases cable TV service providers want to provide Internet services. Each sector

is introducing new services; to cater to these new services, new types of equipment—Web TV, for instance—are introduced.

New equipment in this mixed scenario is being introduced at a fast rate, and this equipment needs network management. In some cases, the control of service providers can span a wide geographical and customer base. Under these circumstances, the role of network management covering a wide variety of equipment can be a very daunting task.

12.4 Future Trends

We need to look into the crystal ball and envision the future of TMN. This is essential for TMN practitioners: If we don't have much of an idea of the future, our TMN implementations will be incomplete when we have to accommodate future solutions. The forecast for TMN industry participants is very rosy in terms of revenue generation. At the same time, the future will be very challenging, with a need to keep pace with fast-changing technologies. From this angle, let us examine where the TMN industry is heading.

12.4.1 Focus on Standardization

Technological improvements in TMN are occurring at a fast rate. To keep pace with these developments, increasing effort is being directed toward TMN. Because of globalization and privatization, standardization is also becoming very important.

12.4.2 Distributed Network Management

In the distributed network management, one of the challenges is figuring out how legacy systems will co-exist with new systems built on the basis of new technologies such as CORBA. Another problem is determining how managers will interface with one another. The coordination of many distributed managers will be a daunting task. Yet another problem is deciding how backup will be activated when one of the managers is down. The issues of data synchronization and consistency will have to be taken into account in these cases.

CORBA will take an increasingly important role in distributed network management. Therefore TMN has to accommodate CORBA.

12.4.3 TMN Solutions on Windows-Based Platforms

Because of the high cost of UNIX-based platforms, Windows 95/98 and Windows NT are finding favor as preferred network management development and production platforms. Java is also becoming a popular programming language.

12.4.4 Increased Use of Object-Oriented Paradigms

Object-oriented methodology has gained acceptance in the design and development of TMN solutions. As a corollary, the use of object-oriented databases for providing persistence is also becoming the norm. These trends have been accelerated by the availability of object-oriented design and development tools on different popular computer platforms. There are also many vendors providing good object-oriented databases and database tools.

12.4.5 Integration of the Web and TMN

The use of Web technology for providing customer service is gaining popularity. A Web browser can also be used as an interface to enable customers to communicate about problems. A customer must be able to enter information about the problems he or she encounters, and this must be enough to ensure that the problems will be acted on. Billing is another area where Web browsers can be used. The trend for customers to receive and pay the bills and resolve billing problems using Web browsers will be growing at an enormous pace.

12.5 Summary

In this chapter we have discussed object-oriented software management frameworks and how they can be used to provide TMN solutions. TMN faces many challenges and opportunities ahead in the struggle to make an impact on telephone services. This is particularly true at a time when

"telephone" services include many types of services such as Internet service, wireless, and so on, and when distinctions between local and long-distance services are becoming fuzzy. We have discussed the unresolved issues and challenges confronting TMN solutions. The chapter ends with a discussion of future trends in TMN solutions.

12.6 References

12.1. Udupa, D. K., *Network Management Systems Essentials,* New York: McGraw-Hill, 1996.

12.2. Pauthner, G. and J. Power, Management Platforms. In *The Telecommunications Network Management, Technologies, and Implementations,* Aidarous, S. and Plevyak, T. (eds.), New York: The Institute of Electrical and Electronics Engineers, Inc., pp. 111—149, 1998.

12.3. Ejiri, M., The Paradigm Shift in Telecommunications Services and Networks. In *Integrated Network Management IV,* Sethi, A. S., Raynaud, Y., and Faure-Vincent, F. (eds.), Cornwall, UK: Chapman & Hall, pp. 688—699, 1995.

This page intentionally left blank.

APPENDIX A

How to Keep Up to Date and Procure the Latest TMN Standards

A.1 Symposiums and Conferences

Integrated management (IM) and network operations and management symposium (NOMS) conferences are held alternately; IM conferences convene in odd-numbered years, and NOMS conferences convene in even-numbered years. These conferences are organized by the International Federation for Information Processing (IFIP) Working Group (WG) 6.6 on Network Management for Communication Networks and the Institute of Electrical and Electronics Engineers Communications Society's (COMSOC) Committee on Network Operations and Management (CNOM). The conferences have the latest papers and discussions on network management. They also publish the proceedings of the conferences/symposiums. Attending these conferences or reading the symposium proceedings are good sources for keeping up to date on TMN.

A.2 The World Wide Web (WWW)

One of the easiest ways to access information on any topic—including TMN—is to use the World Wide Web. Large amounts of material are available on many specific topics on TMN. However, care has to be exercised regarding information gathered from the WWW. The material contained in the WWW is not always authentic. Some Web sites contain individual opinions—as contrasted to confirmed facts—so one has to be cautious about gathering information. Still, by and large, the information retrieved from the standards bodies, consortiums, and well-established vendors can be treated as authentic.

Many organizations associated with TMN activities have Web sites. Details of some of the important ones follow. Note that Web pages change frequently, so URLs may change.

The SimpleWeb

The SimpleWeb provides links and information on network management. The primary focus of this Web site is Internet management. The SimpleWeb is a good Web site, and one can find a wealth of information related to network management. The SimpleWeb is maintained by the Telematics Systems and Services management group (TSS) of the University of Twente (the Netherlands), in collaboration with Jürgen Schönwälder of TU Braunschweig (Germany).

The URL of the SimpleWeb is http://wwwsnmp.cs.utwente.nl/.

The URL of the standards page in The SimpleWeb is http://wwwsnmp.cs.utwente.nl/standard/.

NetMan

The University of Buffalo's NetMan is also a good repository of information on network management. The URL of NetMan is http://netman.cit.buffalo.edu/.

International Telecommunications Union—Telecommunications (ITU-T)

ITU-T standards are available only through subscription to the ITU-T. You need a user ID and password to access the ITU-T documents. Most of the vendors have access to the ITU-T standards—so check inside your company to learn whether access to the standards is available there. You can also individually purchase ITU-T documents. The URL for the ITU-T home page is http://www.itu.ch/.

International Standardization Organization (ISO)

ISO documents can be purchased from vendors. Visit the ISO home page for information on how to procure ISO documents. The ISO's home page URL is http://www.iso.ch/.

Internet RFCs

RFCs of the Internet Engineering Task Force are available on the WWW, and they can easily be downloaded. Be sure to check the standard track of an RFC before implementing it.

ATM Forum

Most of the network management forum technical specifications are available on the WWW. The URL of ATM Forum is http://www.atmforum.com/.

TeleManagement Forum

The name Network Management Forum (NMF) has been changed to TeleManagement Forum. TeleManagement Forum documents are available only for subscribers. The URL of TeleManagement Forum is http://www.tmforum.org/.

This page intentionally left blank.

APPENDIX B

Important TMN and Network Management Standards

Because the material on TMN and network management standards is voluminous, we are listing only some of the important network management documents by ITU-T, ISO, Internet network management—related RFCs, ETSI, and ANSI. For details on how to procure documents from other organizations, refer to Appendix C. Documents marked with asterisks are being revised.

B.1 ITU-T

Note that many ITU-T documents are paired with ISO.

G Series Documents

1. G.774, Synchronous Digital Hierarchy (SDH) Management Information Model for the Network Element View, 1992.
2. G.784, Synchronous Digital Hierarchy (SDH) Management, 1994.
3. G.803, Architectures of Transport Networks Based on the Synchronous Digital Hierarchy (SDH), 1993.
4. G.805, Generic Functional Architecture of Transport Networks, 1995.
5. G.831, Management Capabilities of Transport Networks based on the Synchronous Digital Hierarchy (SDH), 1996.

I Series Documents

6. I.321, B-ISDN Protocol Reference Model and its Application, 1991.
7. I.610, B-ISDN Operation and Maintenance Principles and Functions, 1995.
8. I.751, Asynchronous Transfer Mode Management of the Network Element View, 1996.

Q Series Documents

9. Q811, Lower Layer Protocol Profiles for the Q3 and X Interfaces, 1997.
10. Q812, Upper Layer Protocol Profiles for the Q3 and X Interfaces, 1997.
11. Q821, Stage 2 and Stage 3 Description for the Q3 Interface—Alarm Surveillance, 1993.
12. Q822, Stage 1, Stage 2, Stage 3 Description for the Q3 Interface—Performance Management, 1994.
13. Q823, Stage 2 and Stage 3 Functional Specifications for Traffic Management, 1996.
14. Q824.0, Stage 2 and 3 Description for Q3 Interface, Customer Administration—Common Information, 1995.

M Series Documents

15. M.3010, Principles for a Telecommunications Management Network, 1996.
16. M.3020, TMN Interface Specification Methodology, 1995.
17. M.3100, Generic Network Information Model, 1995.
18. M.3101, Managed Object Conformance Statements for the Generic Network Information Model, 1995.
19. M.3180, Catalogue of TMN Management Information, 1992.
20. M.3200, TMN Management Services and Telecommunications Managed Areas: Overview, 1997.
21. M.3300, TMN Management Capabilities Presented at the F Interface, 1992.*
22. M.3320, Management Requirements Framework for the TMN X Interface, 1997.
23. M.3400, TMN Management Functions, 1997.

X Series and ISO Documents

24. X.160, Architecture for Customer Network Management Service for Public Data Networks, 1996.

25. X.161, Definition of Customer Network Management Services for Public Data Networks, 1997.
26. X.162, Definition of Management Information for Customer Network Management Service for Public Data Networks to Be Used With CNMc Interface, 1997.
27. X.163, Definition of Management Information for Customer Network Management Services for Public Data Networks to Be Used With CNMe Interface, 1995.
28. X.200 (ISO/IEC 7498-1), Basic Reference Model: Basic Model, 1994.
29. X.207 (ISO/IEC 9545), Application Layer Structure, 1993.
30. X.208 (ISO/IEC 8824), Specification of Abstract Syntax Notation One (ASN.1), 1988.
31. X.209 (ISO/IEC 8825), Specification Basic Encoding Rules for Abstract Syntax Notation (ASN.1), 1988.
32. X.210 (ISO/IEC 10731), Basic Reference Model: Conventions for the Definition of OSI Services, 1993.
33. X.217 (ISO/IEC 8649), Service Definition for the Association Control Service Element, 1995.
34. X.217 Amendment 1 (ISO/IEC 8649 AM 1), Service Definition for the Association Control Service Element, Amendment 1: Support of Authentication Mechanisms for the Connectionless Mode, 1996.
35. X.218 (ISO/IEC 9066-1), Reliable Transfer: Model and Service Definition, 1993.
36. X.219 (ISO/IEC 9072-1), Remote Operations: Model, Notation, and Service Definition, 1988.
37. X.226 (ISO/IEC 8823-1), Connection-Oriented Presentation Protocol: Protocol Specification, 1994.
38. X.227 (ISO/IEC 8650-1), Connection-Oriented Protocol for the Association Control Service Element: Protocol Specification, 1995.
39. X.227 Amendment 1 (ISO/IEC 8650-1 AM 1), Connection-Oriented Protocol for the Association Control Service Element: Protocol Specification, Amendment 1: Incorporation of Extensibility Markers, 1996.
40. X.228 (ISO/IEC 9066-2), Reliable Transfer: Protocol Specification, Remote Operations: Protocol Specification, 1988.
41. X.229 (ISO/IEC 9072-2), Connection-Mode Protocol Specifications, 1988.

42. X.237 (ISO/IEC 10035-1), Connectionless Protocol for the Association Control Service Element: Protocol Specification, 1995.
43. X.237 Amendment 1 (ISO/IEC 10035-1 Amd. 1), Connectionless Protocol for the Association Control Service Element: Protocol Specification, Amendment 1: Incorporation of Extensibility Markers and Authentication Parameters, 1996.
44. X.500 (ISO/IEC 9594-1), The Directory: Overviews of Concepts, Models and Services, 1993.
45. X.650 (ISO/IEC 7498-3), Basic Reference Model, Naming and Addressing, 1996.
46. X.660 (ISO/IEC 9834-1), Procedures for the Operation of OSI Registration Authorities, General Procedures, 1992.
47. X.660, Amendment 1 (ISO/IEC 9834-1), Procedures for the Operation of OSI Registration Authorities, General Procedures, Amendment 1: Incorporation of Object Identifiers Components, 1996.
48. X.680 (ISO/IEC 8824-1), Abstract Syntax Notation One (ASN.1), Specification of Basic Notation, 1994.*
49. X.680 Amendment 1 (ISO/IEC 8824-1), Abstract Syntax Notation One (ASN.1), Specification of Basic Notation, Amendment 1: Rules of Extensibility, 1995.
50. X.681 (ISO/IEC 8824-2), Abstract Syntax Notation One (ASN.1), Information Object Specification, 1994.*
51. X.682 (ISO/IEC 8824-3), Abstract Syntax Notation One (ASN.1), Constraint Specification, 1994.
52. X.683 (ISO/IEC 8824-4), Abstract Syntax Notation One (ASN.1), Parameterization of ASN.1 Specifications, 1994.
53. X.690 (ISO/IEC 8825-1), ASN.1 Encoding Rules, Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER), and Distinguished Encoding Rules (DER), 1994.*
54. X.691 (ISO/IEC 8825-2), ASN.1 Encoding Rules, Specification of Packed Encoding Rules (PER), 1995.*
55. X.700, Management Framework for Open Systems Interconnection (OSI) for CCITT Applications, 1992.
56. X.701 (ISO/IEC 10040)—Systems Management Overview, 1992.
57. X.710 (ISO/IEC 9595), Common Management Information Service Definition for CCITT Applications. 1991.*
58. X.711 (ISO/IEC 9596), Common Management Information Protocol Specification for CCITT Applications, 1991.*

59. X.720 (ISO/IEC 10165-1), Management Information Model, 1992.
60. X.721 (ISO/IEC 10165-2), Definition of Management Information, 1992.
61. X.722 (ISO/IEC 10165-4), Guidelines for the Definition of Managed Objects, 1992.
62. X.723 (ISO/IEC 10165-5), Generic Management Information, 1993.
63. X.724 (ISO/IEC 10165-6), Requirements and Guidelines for Implementation Conformance Statement Proformas Associated with OSI Management, 1996.
64. X.725 (ISO/IEC 10165-7), General Relationship Model, 1995.
65. X.730 (ISO/IEC 10164-1), Object Management Function, 1992.
66. X.731 (ISO/IEC 10164-2), State Management Function, 1992.
67. X.732 (ISO/IEC 10164-3), Attributes for Representing Relationships, 1992.
68. X.733 (ISO/IEC 10164-4), Alarm Reporting Function, 1992.
69. X.734 (ISO/IEC 10164-5), Event Report Management Function, 1992.
70. X.735 (ISO/IEC 10164-6), Log Control Function, 1992.
71. X.736 (ISO/IEC 10164-7), Security Alarm Reporting Function, 1992.
72. X.737 (ISO/IEC 10164-14), Confidence and Diagnostic Test Categories, 1995.
73. X.738 (ISO/IEC 10164-13), Summarization Function, 1993.
74. X.739 (ISO/IEC 10164-11), Metric Objects and Attributes, 1993.
75. X.740 (ISO/IEC 10164-8), Security Audit Trail Function, 1992.
76. X.741 (ISO/IEC 10164-9), Objects and Attributes for Access Control, 1995.
77. X.742 (ISO/IEC 10164-10), Usage Metering Function for Accounting Purposes, 1995.
78. X.743. (ISO/IEC 10164-20), Time Management Function.
79. X.744 (ISO/IEC 10164-18), Software Management Function, 1996.
80. X.745 (ISO/IEC 10164-12), Test Management Function, 1993.
81. X.746 (ISO/IEC 10164-15), Scheduling Function, 1995.
82. X.750 (ISO/IEC 10164-16), Management Knowledge Management Function, 1996.
83. X.751 (ISO/IEC 10164-17), Change Over Function, 1995.
84. X.790, Trouble Management Function for ITU-T Applications, 1995.

85. X.791, Profile for Trouble Management Function for ITU-T Applications, 1996.
86. X.800 (ISO/IEC 7498-2), Security Architecture for Open Systems Interconnection for CCITT Applications, 1991.
87. ISO/IEC 13244, ITU-T Recommendation X.703, Information Technology—Open Distributed Management Architecture, 1997.
88. ISO/IEC 10746-1, ITU-T Recommendation X.901, Information Technology—Open Distributed Processing—Reference Model, Part 1: Overview and Guide to Use, 1997.
89. ISO/IEC 10746-2, ITU-T Recommendation X.902, Information Technology—Open Distributed Processing—Reference Model: Foundations, 1995.
90. ISO/IEC 10746-3, ITU-T Recommendation X.903, Information Technology—Open Distributed Processing—Reference Model: Architecture, 1995.
91. ISO/IEC 10746-4, ITU-T Recommendation X.904, Information Technology—Open Distributed Processing—Reference Model: Architectural Semantics, 1997.
92. ISO/IEC 7498-4, Basic Reference Model—Part 4: Management Framework, 1989.
93. X.749 ISO/IEC 10164-19, Management Domain and Management Policy Management Functions.
94. ISO/IEC 10646-1, Universal Multiple-Octet Coded Character Set (UCS): Architecture and Basic Multilingual Plane, 1993.

B.2 Internet RFCs

Here, we are listing only important TMN/Network Management RFCs.

1. RFC 1155, Structure and Identification of Management Information for TCP/IP-Based Internets, 1990.
2. RFC 1157, Simple Network Management Protocol, 1990.
3. RFC 1212, Concise MIB Definitions, 1991.
4. RFC 1213, Management Information Base for Network Management of TCP/IP-Based Internets: MIB-II, 1991.

5. RFC 1595, Definitions of Managed Objects for the SONET/SDH Interface Types, 1994.
6. RFC 1695, Definitions of Managed Objects for ATM Management Version 8.0 using SMIv2, 1994.
7. RFC 1901, Introduction to Community-Based SNMPv2, *SNMPv2 experimental,* 1996.
8. RFC 1902, Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2), 1996.
9. RFC 1903, Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2), 1996.
10. RFC 1904, Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2), 1996.
11. RFC 1905, Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2), 1996.
12. RFC 1906, Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2), 1996.
13. RFC 1907, Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2), 1996.
14. RFC 1908, Coexistence between Version 1 and Version 2 of the Internet-Standard Network Management Framework, 1996.
15. RFC 1909, An Administrative Infrastructure for SNMPv2, *SNMPv2u experimental,* 1996.
16. RFC 1910, User-based Security Model for SNMPv2, *SNMPv2u experimental,* 1996.
17. RFC 2089, Mapping SNMPv2 onto SNMPv1 within a Bilingual SNMP Agent, *SNMPv2 informational,* 1997.
18. RFC 2271, An Architecture for Describing SNMP Management Frameworks, 1998.
19. RFC 2272, Message Processing and Dispatching for the Simple Network Management Protocol (SNMP), 1998.
20. RFC 2273, SNMPv3 Applications, 1998.
21. RFC 2274, User-Based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3), 1998.
22. RFC 2275, View-Based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP), 1998.

B.3 ETSI Standards

1. EN 301 251 (GSM 12.11 Version 4.2.0 Draft), Fault Management of the Base Station System (BSS), 1998.
2. ETS 300 612-1 (GSM 12.00), Part 1: Objectives and Structure of Network Management, 1996.
3. ETS 300 612-2 (GSM 12.01), Part 2: Common Aspects of GSM/DCS 1800 Network Management, 1996.
4. ETS 300 612-3 (GSM 12.07), Part 3: Operations and Performance Management. (This is a draft document at the time of writing.)
5. ETS 300 613 (GSM 12.02), Subscriber, Mobile Equipment (ME) and Services Data Administration, 1996.
6. ETS 300 614 (GSM 12.03), Security Management, 1996.
7. ETS 300 615 (GSM 12.04), Performance Data Measurements, 1996.
8. ETS 300 616 (GSM 12.05 Version 4.3.1), Event and Call Data, 1998.
9. ETS 300 617 (GSM 12.06), GSM Network Configuration Management, 1996.
10. ETS 300 622 (GSM 12.20), Base Station System (BSS) Management Information, 1996.
11. ETS 300 623 (GSM 12.21), Network Management (NM) Procedures and Messages A-bis Interface, 1996.
12. ETS 300 624 (GSM 12.22), Interworking of GSM Network Management (NM) Procedures and Messages at the Base Station Control, 1996.
13. ETS 300 627 (GSM 12.08 Version 4.5.1), Subscriber and Equipment Trace, 1998.
14. ETR 128 (GSM 12.30), ETSI Object Identifier Tree, Common Domain, Mobile Domain, Operation and Maintenance (O&M), Managed Object Registration Definition, Edition 2, 1995.

B.4 ANSI Standards

1. T1-TR34 Technical Report: Network Capabilities, Architectures and Interfaces for Personal Communications, 1994.
2. T1.210-1993, Operations, Administration, Maintenance, and Provisioning (OAM&P)—Principles of Functions, Architectures, and Pro-

tocols for Telecommunications Management Network (TMN) Interfaces. 1993.

3. T1.227-1995, Operations, Administration, Maintenance, and Provisioning (OAM&P)—Extension to Generic Network Model for Interfaces between Operations Systems across Jurisdictional Boundaries to Support Fault Management—Trouble Administration, 1995.
4. T.228-1995, Operations, Administration, Maintenance, and Provisioning (OAM&P)—Services for Interfaces Between Operations Systems across Jurisdictional Boundaries to Support Fault Management—Trouble Administration, 1995.
5. T1.229a-1995, Operations, Administration, and Provisioning (OAM&P)—Performance Management Area Service for Interfaces, 1995.
6. T1.232-1996, Operations, Administration, and Provisioning (OAM&P)—G Interface Specification for Use with the Telecommunications Management (TMN), 1996.
7. T1.233-1993, Operations, Administration, and Provisioning (OAM&P)—Security Framework for Telecommunications Management Network (TMN) Interfaces, 1993.
8. T1.240-1996, Operations, Administration, and Provisioning (OAM&P)—Generic Network Information Model for Interfaces between Operations Systems and Network Elements, 1996.
9. T1.244-1995, Operations, Administration, and Provisioning (OAM&P)—Interface Standards for Personal Communication Services, 1995.
10. T1.245-1997, Directory Services for Telecommunications Management Network (TMN) and Synchronous Optical Network (SONET), 1997.
11. T1.247-1995, Operations, Administration, and Provisioning (OAM&P)—Performance Management Functional Area Service and Information Model, 1995.
12. T1.252-1996, Operations, Administration, and Provisioning (OAM&P)—Security for the Telecommunications Management Network (TMN) Directory, 1996.
13. T1.257-1997, Operations, Administration, and Provisioning (OAM&P)—Traffic Management Services and Information Model for Interfaces between Operations Systems and Network Elements, 1997.
14. T1.651-1996, Mobility Management Application Protocol (MMAP), 1996.
15. T1.651a-1996, Mobility Management Application Protocol (MMAP)—Extensions, 1996.

This page intentionally left blank.

APPENDIX C

Suggested Exercises

Chapter 1

1.1 Why is TMN needed?

1.2 Define what TMN is.

1.3 Explain the key differences between data communication network management and telecommunications network management.

1.4 Give the details of different standard bodies in the TMN arena. Include in this list the important consortiums that affect TMN.

1.5 Describe different TMN management layers.

1.6 Furnish details of the five systems management functional areas (SMFAs).

1.7 Explain the concepts of a manager and agent.

1.8 Research software tools available for implementing the SMFAs and make comparisons of the functionality provided.

Chapter 2

2.1 Explain the salient points of TMN functional architecture.

2.2 What is a TMN function block?

2.3 Define different TMN function blocks.

2.4 How are TMN functional components related to TMN function blocks? Provide the functional components in the OSF function block.

2.5 Provide the functional components in a WSF function block.

2.6 What is a reference point, and what is the relationship between reference points and interfaces?

2.7 Define TMN physical architecture and the different components of physical architecture.

2.8 List and describe different TMN interfaces.

2.9 Explain X interface, using examples.

2.10 Explain TMN information architecture.

2.11 Describe shared management knowledge, using an example.

2.12 What is OAM&P? Explain different OAM&P categories.

2.13 What is CNM?

2.14 Describe CNM functional and physical architectures.

2.15 What are CNMc and CNMe interfaces? Where are these interfaces used?

2.16 Map the SMFA and CNM capabilities.

Chapter 3

3.1 What is a TMN MS? How is it related to TMN management functions?

3.2 What is a TMN management function set group? Map SMFAs and different TMN management function set groups.

3.3 Define provisioning function set group.

3.4 Design a traffic management application. Use M.3400 and Q823. Provide use cases in the design.

3.5 Design an alarm surveillance application. Use M.3400 and Q821 for designing the application. Use any object-oriented design technique of your choice.

Chapter 4

4.1 What are service providers and service users?

4.2 State the differences between connection-oriented and connectionless communications.

4.3 What is a primitive?

4.4 What is a management information model?

4.5 What is a registration hierarchy?

4.6 Define containment hierarchy. What are the differences between registration and containment hierarchies?

4.7 What is meant by management information tree?

4.8 Explain the concept of scoping.

4.9 What is filtering? Where are scoping and filtering used?

4.10 Describe the concepts of polymorphism and allomorphism.

4.11 What is the idea behind synchronization? Can we do without this function?

4.12 Explain different generic state attributes.

4.13 Describe different status attributes.

4.14 What is an intelligent agent? Where is it used?

Chapter 5

5.1 Why is abstract syntax required?

5.2 Explain different structure types.

5.3 Why are tagged types used? Explain each of the tagged types.

5.4 Why are subtypes required? Describe different subtypes.

5.5 Why is BER required?

5.6 Explain the ILC fields.

5.7 Describe definite and indefinite length forms, using examples.

5.8 Discuss the issues involved in the design of an encoder/decoder.

5.9 What enhancements are made in X.680 over X.208?

5.10 Discuss the areas where the use of ASN.1 and BER may not be required. Discuss the areas where they are absolutely required.

5.11 List the requirements for an ASN.1 compiler.

5.12 From the list of requirements in the previous exercise, provide a high-level design for an ASN.1 compiler. State the assumptions you have made.

Chapter 6

6.1 Why do we need managed object class definitions?

6.2 What are the important issues involved in the definition of a managed object class?

6.3 Where are attribute groups used?

6.4 Why should the naming of a managed object class be unique?

6.5 What are the similarities between definitions of managed object classes used in ITU-T recommendations and managed object classes defined in object-oriented languages? What are the differences?

6.6 Write GDMO and ASN.1 definitions for a performance management—related object class. Use a scanner managed object class (defined in X.739) to a derive the managed object class. State the assumptions clearly.

6.7 For the performance management—related object class used in the previous exercise, provide the name binding. Here also furnish all the related GDMO and ASN.1 definitions.

6.8 Discuss the strategy to instantiate the performance management—related object class defined in exercises 6.6 and 6.7 such that efficient use of the performance management—related objects can be made.

6.9 Provide the list of requirements for a GDMO compiler. State the assumptions clearly.

6.10 From the list of requirements in the previous exercise, provide a high-level design for a GDMO compiler.

6.11 Make a comparative study of GDMO and ASN.1 compilers available from two important vendors. State the strengths and deficiencies of these compilers.

Chapter 7

7.1 Why are ACSE services required?

7.2 Explain different ACSE services.

7.3 List and explain each of the CMISE services.

7.4 State the error processing cases involved in each of the CMISE services.

7.5 State the advantages and disadvantages of using CMIP.

7.6 Explain why CMIP implementations have been slow to come by in the market.

7.7 Provide a high-level design for a manager and an agent using ACSE, ROSE, and CMISE services.

Chapter 8

8.1 State the important principles in Internet network management.

8.2 Explain the rationale for using UDP for SNMPv1.

8.3 Describe the SNMPv1 protocol messages, including their shortcomings.

8.4 Why are proxies required?

8.5 What is MIB-II? Explain the main object group classifications of MIB-II.

8.6 What are the differences between SNMPv1 and SNMPv2?

8.7 Describe SNMPv2 protocol messages.

8.8 State the extensions made to SMI in SNMPv2.

8.9 What are the differences between SNMPv2 and SNMPv3?

8.10 Which RFCs constitute SNMPv3? State the key features of SNMPv3.

8.11 Describe the salient points of SNMPv3 architecture, using RFC 2271.

8.12 Explain managers and agents as defined in SNMPv3 architecture.

8.13 Describe the primitives provided by dispatcher, message-processing subsystem, and security subsystem.

8.14 Explain the security features of SNMPv3, using RFC 2274 and RFC 2275.

8.15 Briefly describe the Internet standardization process.

8.16 Compare SNMP with CMIP protocols. State where each protocol is useful.

8.17 What are the advantages and limitations of SNMP protocols?

8.18 Download the SNMP implementation code from the WWW and study the code. List strengths and shortcomings in the code. Use the SNMP version of your choice.

8.19 Run the code downloaded in the previous exercise. Write the manager and agent code to run the downloaded SNMP code.

Chapter 9

9.1 What are the important issues involved in network management for mobile communications?

9.2 List some of the important requirements for PCS network management. Consult ANSI T1.244-1995 for answering this question.

9.3 List the protocols used in the protocol stack for PCS network management. Consult ANSI T1.244-1995.

9.4 Describe the ETSI-based OAM for PLMN.

9.5 Describe the PLMN information model and define MOCs.

9.6 List the cases where FTAM can be used in network management for mobile communications.

9.7 Describe different FTAM scenarios.

9.8 Explain the different protocol layers used in PLMN.

Chapter 10

10.1 Explain the different layers in the B-ISDN reference model.

10.2 What are the important requirements of SDH network management?

10.3 Explain the concepts of SMN and SMS.

10.4 List SDH-specific parameters for which alarm conditions are raised.

10.5 State SDH-specific management functions.

10.6 Describe the primary groups in Internet-based SONET/SDH MIB.

10.7 Explain SDH OAM. (Note that this explanation is to be based on I.610.)

10.8 List and explain different ATM Forum network management interfaces.

10.9 Describe ATM Forum CNM.

10.10 What is ILMI? Describe the primary object groups in ILMI.

10.11 Describe the primary managed object groups in Internet-based ATM MIB.

10.12 Explain the key issues involved in broadband network management.

Chapter 11

11.1 Explain the differences between distributed and centralized network management.

11.2 Why is distributed network management needed?

11.3 Explain different distributed network management architectures. Make a comparison highlighting the strengths and weaknesses of different architectures.

11.4 List and explain the requirements of distributed network management systems.

11.5 What are the different viewpoints in ODP?

11.6 List the different distribution transparencies.

11.7 List the ODP functions.

11.8 What is ODMA? Explain how different viewpoints defined in RM-ODP are handled in ODMA.

11.9 How can CORBA be used in TMN?

11.10 Compare different CORBA implementations, and pick one of the vendors for your needs.

11.11 Describe important CORBA services required for a TMN application.

11.12 List the TMN applications that can be used by Web browsers.

11.13 What is WBEM? Explain how it can be used in TMN.

11.14 How can you use Java in TMN?

11.15 What are the limitations of IDL to Java mapping?

Chapter 12

12.1 Explain the principles behind TMN management frameworks.

12.2 Compare two widely used management frameworks. Describe their strengths and weaknesses.

12.3 Provide the requirements for an object-oriented TMN management framework.

12.4 Provide a high-level design for the object-oriented based TMN management framework. Develop a high-level design from the list of requirements in the previous example.

12.5 Research different object-oriented database implementations available and compare the implementations.

12.6 What are the challenges facing TMN?

12.7 Highlight the important trends in TMN.

12.8 Explain the impact on TMN of integrating the cable, telecommunications, and computing industries.

This page intentionally left blank.

LIST OF ACRONYMS

AA	Application Association
AAL	ATM Adaptation Layer
AARE	A-ASSOCIATE-RESPONSE, ACSE PDU
AARQ	A-ASSOCIATE-REQUEST, ACSE PDU
ABRT	A-ABORT, ACSE PDU
AC	Application Context
ACSE	Association Control Service Element
ADC	Administration Center
AE	Application Entity
AIN	Advanced Intelligent Network
ANSI	American National Standards Institute
AP	Application Process
APDU	Application Protocol Data Unit
API	Application Programming Interface
APS	Automatic Protection Switching
ASE	Application Service Element
ASN.1	Abstract Syntax Notation One
ATM	Asynchronous Transfer Mode
AUC	Authentication Center
AUDT	A-UNIT-DATA APDU
AVM	Admin View Model
AWT	Abstract Window Toolkit
BER	Basic Encoding Rules
BER	Bit Error Rate
BICI	Broadband Inter Carrier Interface
B-ISDN	Broadband Integrated Services Digital Network
BISSI	Broadband Inter Switching System Interface
BML	Business Management Layer
bps	Bits per Second
BSC	Base Station Controller

BSS	Base Station System
BTS	Base Transceiver Station
CDMA	Code Division Multiple Access
CER	Canonical Encoding Rules
CF	Control Function
CGI	Common Gateway Interface
CIM	Common Information Model
CLTS	Connectionless-mode Transport Service
CMIP	Common Management Information Protocol
CMIPM	Common Management Information Protocol Machine
CMIS	Common Management Information Service
CMISE	Common Management Information Service Element
CNM	Customer Network Management
CNMc	Customer Network Management using CMIP
CNMe	Customer Network Management using EDI/MHS
CONM	Committee on Network Operations and Management
CONS	Connection-mode Network Service
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
CPDU	CMIP Protocol Data Units
CRC	Cyclic Redundancy Check
CSPDN	Circuit Switched Public Data Network
CV	Coding Violation
DAF	Directory Access Function
DCC	Data Communications Channel
DCF	Data Communication Function
DCN	Data Communication Network
DCOM	Distributed Component Object Model
DCS	Digital Cellular System
DER	Distinguished Encoding Rules
DIB	Directory Information Base
DII	Dynamic Invocation Interface
DISMAN	Distributed Management (Internet Network Management)

DMI	Desktop Management Interface
DMTF	Desktop Management Task Force
DN	Distinguished Name
DS3	Digital Signal 3
DSF	Directory System Function
DSL	Digital Subscriber Line
DSOM	Distributed System Object Model
ECC	Embedded Control Channel
EDI	Electronic Data Interchange
EFD	Event Forwarding Discriminator
EIA	Electronic Industry Association
EIR	Equipment Identity Register
EM	Element Management
EML	Element Management Layer
EOC	Embedded Operations Channel
ES	Errored Second
ETS	European Telecommunication Standard
ETSI	European Telecommunications Standards Institute
FSM	Finite State Machine
FTAM	File Transfer Access and Management
FTIF	File Transfer Initiator Function
FTRF	File Transfer Responder Function
GDMO	Guidelines for the Definition of Managed Objects
GIOP	General Inter-ORB Protocol
GNE	Gateway Network Element
GRM	General Relationship Model
GSM	Global System for Mobile Communications
GTP	Group Termination Point
GUI	Graphical User Interface
HEC	Header Error Control
HLR	Home Location Register
HMMP	HyperMedia Management Protocol
HMMS	HyperMedia Management Schema

HMOM	HyperMedia Object Manager
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IAB	Internet Activities Board
ICF	Information Conversion Function
IDL	Interface Definition Language
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IESG	Internet Engineering Steering Group
IETF	Internet Engineering Task Force
IFIP	International Federation for Information Processing
IIOP	Internet Inter-ORB Protocol
ILC	Identifier, Length, and Contents
ILCE	Identifier, Length, Contents, and End-of-Contents
ILMI	Integrated Local Management Interface
IM	Integrated Management
IME	Interface Management Entity
IN	Intelligent Network
IP	Internet Protocol
IPX	Internet Package Exchange
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ITU-T	International Telecommunications Union-Telecommunications.
JIDM	Joint InterDomain Management
JMAPI	Java Management API.
kbps	Kilo bits per second (10^3 bps)
LAN	Local Area Network
LAPD	Link Access Protocol on D Channel
LATA	Local Access and Transport Area
LCN	Local Communications Network
LLA	Logical Layered Architecture
LOF	Loss of Frame
LOS	Loss of Signal

MAF	Management Application Function
MAPDU	Management Application Protocol Data Unit
MCF	Message Communication Function
MD	Mediation Device
ME	Mobile Equipment
MF	Mediation Function
MF-MAF	Mediation Function-Management Application Function
MHS	Message Handling System
MIB	Management Information Base
MIME	Multipurpose Internet Mail Extensions
MIT	Management Information Tree
MMAP	Mobility Management Application Protocol
MML	Man Machine Language
MO	Managed Object
MOC	Managed Object Class
MOF	Managed Object Format
MS	Mobile Station
MS	Management Services
MSC	Mobile Services Switching Center
NE	Network Element
NEF	Network Element Function
NEF-MAF	Network Element Function-Management Application Function
NM	Network Management
NMF	Network Management Forum
NML	Network Management Layer
NMS	Network Management Station
NNE	Non-SDH Network Element
NOC	Network Operations Center
NOMS	Network Operations and Management Symposium
NPC	Network Parameter Control
NSAP	Network Service Access Point
OAM	Operation, Administration, and Maintenance
OAM	Operation and Maintenance (ITU-T)

OAM&P	Operation, Administration, Maintenance, and Provisioning
ODMA	Open Distributed Management Architecture
ODP	Open Distributed Processing
OID	Object Identifier
OMC	Operations and Maintenance Center
OMG	Object Management Group
OMT	Object Modeling Technique
OOB	Out of Band
ORB	Object Request Broker
OS	Operations Systems
OSF	Operations Systems Function
OSF-MAF	Operations Systems Function-Management Application Function
OSI	Open Systems Interconnection
OSS	Operation Support System
PBX	Private Branch Exchange
PCI	Protocol Control Information
PCS	Personal Communication System
PDN	Public Data Network
PDU	Protocol Data Unit
PDV	Presentation Data Value
PER	Packed Encoding Rules
PICS	Protocol Implementation Conformance Statement
PLCP	Physical Layer Convergence Protocol
PLMN	Public Land Mobile Network
PM	Performance Monitoring
POTS	Plain Old Telephone Service
PMC	Personal Mobility Controller
PMD	Personal Mobility Datastore
PPDU	Presentation Protocol Data Unit
PRM	Protocol Reference Model
PSAP	Presentation Service Access Point
PSC	PCS Switching Center
PSTN	Public Switched Telephone Network

PVC	Permanent Virtual Circuit
QA	Q Adaptor
QAF	Q Adaptor Function
QAF-MAF	Q Adaptor Function-Management Application Function
QOS	Quality of Service
RAS	Reliability, Availability, and Survivability
RASC	Radio Access System Controller
RDN	Relative Distinguished Name
RFC	Request For Comments
RLRE	A-RELEASE-RESPONSE, ACSE PDU
RLRQ	A-RELEASE-REQUEST, ACSE PDU
RM-ODP	Reference Model—Open Distributed Processing
RMI	Remote Method Invocation
RMON	Remote Network Monitoring
ROER	Remote Operation Error, ROSE APDU
ROIV	Remote Operation Invoke, ROSE APDU
RORJ	Remote Operation Reject, ROSE APDU
RORS	Remote Operation Result, ROSE APDU
ROS	Remote Operations
ROSE	Remote Operation Service Element
RPC	Radio Port Controller
SAP	Service Access Point
SDH	Synchronous Digital Hierarchy
SDU	Service Data Unit
SEFS	Severely Errored Framing Second
SES	Severely Errored Second
SF	Security Function
SIF	SONET Interoperability Forum
SM	Systems Management
SMFA	Systems Management Functional Area
SMI	Structure of Management Information
SMK	Shared Management Knowledge
SML	Service Management Layer

SNMP	Simple Network Management Protocol
SNMPv1	Simple Network Management Protocol Version 1
SNMPv2	Simple Network Management Protocol Version 2
SNMPv3	Simple Network Management Protocol Version 3
SMAE	Systems Management Application Entity
SMASE	Systems Management Application Service Entity
SMI	Structure of Management Information
SMN	SDH Management Network
SMS	SDH Management Sub-Network
SOM	System Object Model
SONET	Synchronous Optical Network
SPDU	Session Protocol Data Unit
SSAP	Session Service Access Point
SS No. 7	Signaling System No. 7
SVC	Switched Virtual Circuit
TC	Transmission Convergence
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TIA	Telecommunications Industry Association
TINA	Telecommunications Information Network Architecture
TL1	Transaction Language 1
TMC	Terminal Mobility Controller
TMD	Terminal Mobility Datastore
TMN	Telecommunications Management Network
TP	Termination Point
TPDU	Transport Protocol Data Unit
TR	Technical Report
UAS	Unavailable Second
UCS	Universal Multiple-octet Coded Character Set
UDP	User Datagram Protocol
UNI	User Network Interface
UPC	User Parameter Control
URL	Uniform Resource Locator

USM	User-based Security Model
UTC	Coordinated Universal Time
VACM	View-based Access Control Model
VCC	Virtual Channel Connection
VCI	Virtual Channel Identifier
VCL	Virtual Channel Link
VLR	Visitor Location Register
VPC	Virtual Path Connection
VPI	Virtual Path Identifier
VPL	Virtual Path Link
WAN	Wide Area Network
WBEM	Web-Based Enterprise Management
WSF	WorkStation Function
WSSF	WorkStation Support Function
WWW	World Wide Web

This page intentionally left blank.

LIST OF TRADEMARKS

AppleTalk	Apple Computer, Inc.
IBM	International Business Machines Corp.
IPX/SPX	Novell, Inc.
Java	Sun Microsystems, Inc.
Netware	Novell, Inc.
Windows 95/98	Microsoft Corporation
Windows NT	Microsoft Corporation

www.pcltools.com

www.pcltools.com

This page intentionally left blank.

INDEX

B

C

www.pcltools.com
www.pcltools.com

D

E

F

G

H

I

N

O

P

T

About the Author

Divakara K. Udupa has been working in the network management area for about 10 years. He has worked for IBM and ISR Global Telecom; currently, he is working in the TMN Solutions group in Siemens Telecom Networks. He has a wide range of experience as a designer and developer with networking protocols, network management protocols, and TMN.

Mr. Udupa is the author of the successful book *Network Management Systems Essentials,* published by McGraw-Hill. He owns one patent and has published many articles.

Mr. Udupa has an MS in computer science from Rensselaer Polytechnic Institute and ME in mechanical engineering from Calcutta University, and a BS in mechanical engineering from Banaras Hindu University. He is a member of the Association of Computing Machinery (ACM) and a senior member of the American Institute of Industrial Engineers (AIIE).