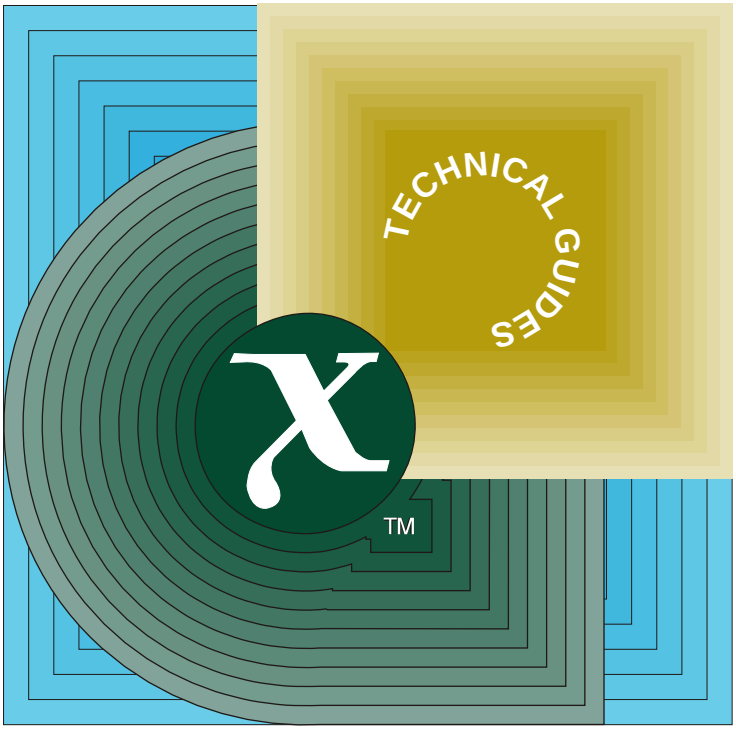
Guide

ISO and Internet Management: Coexistence and Interworking



THE Open GROUP

[This page intentionally left blank]

X/Open Guide

ISO/CCITT and Internet Management: Coexistence and Interworking Strategy

X/Open Company Ltd.

X/Open Guide

ISO/CCITT and Internet Management: Coexistence and Interworking Strategy

ISBN: 1 872630 67 7

X/Open Document Number: G211

Published by X/Open Company Ltd., U.K.

Any comments relating to the material contained in this document may be submitted to X/Open at:

X/Open Company Limited
Apex Plaza
Forbury Road
Reading
Berkshire, RG1 1AX
United Kingdom

or by Electronic Mail to:

XoSpecs@xopen.co.uk

Contents

List of Figures

List of Tables

Preface

X/Open

X/Open is an independent, worldwide, open systems organisation supported by most of the world's largest information systems suppliers, user organisations and software companies. Its mission is to bring to users greater value from computing, through the practical implementation of open systems.

X/Open's strategy for achieving this goal is to combine existing and emerging standards into a comprehensive, integrated, high-value and usable system environment, called the Common Applications Environment (CAE). This environment covers the standards, above the hardware level, that are needed to support open systems. It provides for portability and interoperability of applications, and allows users to move between systems with a minimum of retraining.

The components of the Common Applications Environment are defined in X/Open CAE Specifications. These contain, among other things, an evolving portfolio of practical application programming interfaces (APIs), which significantly enhance portability of application programs at the source code level, and definitions of, and references to, protocols and protocol profiles, which significantly enhance the interoperability of applications.

The X/Open CAE Specifications are supported by an extensive set of conformance tests and a distinct X/Open trademark - the XPG brand - that is licensed by X/Open and may be carried only on products that comply with the X/Open CAE Specifications.

The XPG brand, when associated with a vendor's product, communicates clearly and unambiguously to a procurer that the software bearing the brand correctly implements the corresponding X/Open CAE Specifications. Users specifying XPG-conformance in their procurements are therefore certain that the branded products they buy conform to the CAE Specifications.

X/Open is primarily concerned with the selection and adoption of standards. The policy is to use formal approved *de jure* standards, where they exist, and to adopt widely supported *de facto* standards in other cases.

Where formal standards do not exist, it is X/Open policy to work closely with standards development organisations to assist in the creation of formal standards covering the needed functions, and to make its own work freely available to such organisations. Additionally, X/Open has a commitment to align its definitions with formal approved standards.

X/Open Specifications

There are two types of X/Open specification:

- *CAE Specifications*

  CAE (Common Applications Environment) Specifications are the long-life specifications that form the basis for conformant and branded X/Open systems. They are intended to be used widely within the industry for product development and procurement purposes.

Developers who base their products on a current CAE Specification can be sure that either the current specification or an upwards-compatible version of it will be referenced by a future XPG brand (if not referenced already), and that a variety of compatible, XPG-branded systems capable of hosting their products will be available, either immediately or in the near future.

CAE Specifications are not published to coincide with the launch of a particular XPG brand, but are published as soon as they are developed. By providing access to its specifications in this way, X/Open makes it possible for products that conform to the CAE (and hence are eligible for a future XPG brand) to be developed as soon as practicable, enhancing the value of the XPG brand as a procurement aid to users.

- *Preliminary Specifications*

  These are specifications, usually addressing an emerging area of technology, and consequently not yet supported by a base of conformant product implementations, that are released in a controlled manner for the purpose of validation through practical implementation or prototyping. A Preliminary Specification is not a ''draft'' specification. Indeed, it is as stable as X/Open can make it, and on publication has gone through the same rigorous X/Open development and review procedures as a CAE Specification.

  Preliminary Specifications are analogous with the ''trial-use'' standards issued by formal standards organisations, and product development teams are intended to develop products on the basis of them. However, because of the nature of the technology that a Preliminary Specification is addressing, it is untried in practice and may therefore change before being published as a CAE Specification. In such a case the CAE Specification will be made as upwards-compatible as possible with the corresponding Preliminary Specification, but complete upwards-compatibility in all cases is not guaranteed.

In addition, X/Open periodically publishes:

- *Snapshots*

  Snapshots are ''draft'' documents, which provide a mechanism for X/Open to disseminate information on its current direction and thinking to an interested audience, in advance of formal publication, with a view to soliciting feedback and comment.

  A Snapshot represents the interim results of an X/Open technical activity. Although at the time of publication X/Open intends to progress the activity towards publication of an X/Open Preliminary or CAE Specification, X/Open is a consensus organisation, and makes no commitment regarding publication.

  Similarly, a Snapshot does not represent any commitment by any X/Open member to make any specific products available.

X/Open Guides

X/Open Guides provide information that X/Open believes is useful in the evaluation, procurement, development or management of open systems, particularly those that are X/Open-compliant.

X/Open Guides are not normative, and should not be referenced for purposes of specifying or claiming X/Open-conformance.

This Document

A major goal of the X/Open Systems Management programme is to enable the integration of Systems and Network Management. As part of the activities directed towards achieving this goal, X/Open has published a Management Protocols API (XMP) which allows uniform access to both the ISO/CCITT and the Internet management protocols - CMIP and SNMP respectively.

This document addresses some of the issues involved in making use of both the ISO/CCITT and Internet approaches in design and operation of management systems.

This document was developed in collaboration with the Network Management Forum and is also published by them as NMF Technical Report 107.

Trade Marks

X/Open and the 'X' device are trademarks of X/Open Company Limited in the U.K. and other countries.

Acknowledgements

This Guide was developed in collaboration with the Network Management Forum (NMF).

Referenced Documents

The following documents are referenced in this guide:

AO
: The Mystical Adaptor Object, Grainger, Pope, Santifaller, presented at the OSF MANSIG, September 8 1992.

APS
: ISO/IEC ISP 11183-1, Information Technology - International Standardized Profiles AOM1n OSI Management - Management Communications Protocols - Part 1: Specification of ACSE, Presentation and Session Protocols for the use by ROSE and CMISE, May 1992.

BMC
: ISO/IEC ISP 11183-3, Information Technology - International Standardized Profiles AOM1n OSI Management - Management Communications Protocols - Part 3: AOM11 - Basic Management Communications, May 1992.

CMIP
: CCITT Recommendation X.711, ISO/IEC 9596-1: 1991 (E), Information Technology - Open Systems Interconnection - Common Management Information Protocol Specification - Part 1: Specification, Edition 2.

CMIPI
: RFC 1189, CMIP For The Internet, Warrier, U., Besaw, L., LaBarre, L., Handspicker, B., October 1990.

CMISD
: ISO/IEC 9695: 1991, Information Technology - Open Systems Interconnection - Common Management Information Service Definition. CCITT Recommendation X.710 (1991), Common Management Information Service Definition for CCITT applications - General concepts.

Concise MIB
: RFC 1212, Concise MIB Definitions, Rose, M., McCloghrie, K., 1991 March.

CORBA
: The Common Object Request Broker: Architecture and Specification, published jointly by the Object Management Group (OMG) and the X/Open Company Limited, Document Number 91.12.1, Revision 1.1, 1992.

DM
: The Distributed Management Choice, Thomas, L., LAN Technology, 1992 April.

DMI
: CCITT Recommendation X.721 (1992), ISO/IEC 10165-2: 1992, Information Technology - Open Systems Interconnection - Structure of management information: Definition of management information.

EMC
: ISO/IEC ISP 11183-2, Information Technology - International Standardized Profiles AOM1n OSI Management - Management Communications Protocols - Part 2: AOM12 - Enhanced Management Communications, June 1992.

ERM

CCITT Recommendation X.734 (to be published), ISO/IEC 10164-5: 1992, Information Technology - Open Systems Interconnection - Systems Management: Event report management function.

GDMO

CCITT Recommendation X.722 (1992), ISO/IEC 10165-4: 1992, Information Technology - Open Systems Interconnection - Structure of Management Information: Guidelines for the Definition of managed objects.

GMI

CCITT Recommendation X.723, ISO/IEC DIS 10165-5: Information Technology - Open Systems Interconnection - Structure of Management Information - Part 5: Generic Managed Information, ISO/IEC JTC1/SC21 N6572, February 20,1992.

IIIMIBTR

ISO and Internet Management Coexistence: Translation of Internet MIBs to ISO GDMO MIBs, LaBarre, L, to be published December, 1992.

IIIMIB-IITR

ISO and Internet Management Coexistence: Translation of the Internet MIB-II (RFC 1213) to ISO GDMO, LaBarre, L, to be published December, 1992.

IIIPARTYTR

ISO and Internet Management Coexistence: Translation of SNMP Party MIB (RFC 1353) to ISO GDMO, LaBarre, L, to be published December, 1992.

IIMPROXY

ISO and Internet Management Coexistence: ISO/Internet Management Proxy, Chang, A., to be published December 1992.

IIOMIBTR

ISO and Internet Management Coexistence: Translation of ISO GDMO MIBs to Internet MIBs, Newnan, O., to be published December, 1992.

IPSNM

Network Management of TCP/IP Networks: Present and Future, Ben-Artzi, A., Chadna, A. & Warrier, U., IEEE Network Magazine, July 1990.

ISOTCP

RFC 1006, ISO transport services on top of the TCP: Version 3, Rose, M.T.; Cass, D.E., 1987 May.

ISTDP

RFC 1310, "Internet Standards Process", Chapin, L., March 1992.46. OSI Products, Release 7, June 1992.

LAN/MAN

IEEE 802.1B, LAN/MAN Management, January 27, 1992.

MEDIACC

US WEST Network Interface Specification - MEDIACC Trouble Administration (TA), Document Number 77302, Issue A, US WEST Communications, Inc., 1992 May.

MIB-II

RFC 1213, Management Information Base for Network Management of TCP/IP-based Internets: MIB-II, Rose, M., McCloghrie, K., 1991 March.

MIM

CCITT Recommendation X.720 (1992), ISO/IEC 10165-1: 1992, Information Technology - Open Systems Interconnection - Structure of management information: Management information model.

OAAC

CCITT Recommendation X.741 (to be published), ISO/IEC CD 10164-9, Information Technology - Open Systems Interconnection - Systems Management: Objects and Attributes for Access Control function.

O*P*1

Network Management Forum: Forum 006, Forum Library - Volume 4: OMNI*Point* 1 Definitions, Issue 1.0, August 1992.

O*P*1DG

OMNI*Point* 1 Guide for Developers, Issue 1.0, Network Management Forum, September 1992.

OSIPR

OSI Products, Release 7, June 1992, published by Technology Appraisals Ltd., 82 Hampton Road, Twickenham TW2 5QS, U.K. Also available via Corporation for Open Systems (COS) in U.S.A.

SAR

CCITT Recommendation X.736, ISO/IEC 10164-7: Information Technology - Open Systems Interconnection - Systems Management - Part 7: Security Alarm Reporting Function, ISO/IEC JTC1/SC21 N6367, October 15, 1991.

SAT

CCITT Recommendation X.740, ISO/IEC 10164-8: Information Technology - Open Systems Interconnection - Systems Management - Part 8: Security Audit Trail Function, ISO/IEC JTC1/SC21 N7039, June 2,1992.

SIMI

RFC 1155, Structure and Identification of Management Information for TCP/IP-based Internets, Rose, M., McCloghrie, K., 1990 May.

SM

Network Management Forum: Forum 016, Application Services: Security of Management, Issue 1.0, August 1992.

SMF

CCITT Recommendation X.731, ISO/IEC 10164-2: Information Technology - Open Systems Interconnection - Systems Management - Part 2: State Management Function, ISO/IEC JTC1/SC21 N6356, October 15,1991.

SMK

Network Management Forum: Forum 015, OMNI*Point* 1 Shared Management Knowledge, Issue 1.0, August 1992.

SMO

CCITT Recommendation X.701 (1992), ISO/IEC 10040: 1992, Information Technology - Open Systems Interconnection - Systems management overview.

SNMP
RFC 1157, Simple Network Management Protocol (SNMP), Case, J.D.; Fedor, M.; Schoffstall, M.L.; Davin, C., 1990 May.

SNMPI
Integration of OSI-Based and SNMP-Based Network Management Systems: an Example, Balazs, A., Beschoner, K., Muller, H., presented at the IFIP Workshop, October 1992.

SNMPMPC
Software Brings SNMP Management to PCs, Workstations and Servers, Mier, E., Communications Week, February 17 1992.

SNMPO
RFC 1161, SNMP Over OSI, Rose, M.,1990 June.

SNMPR
SNMP, From Counters to Clocks, Mier, E., Communications Week, January 27 1992.

SNMPS
SNMP Stations Pull It All Together, Mier, E., Communications Week, April 6, 1992.

SONM
Simply Open Network Management: An Approach for the Integration of SNMP into OSI Management Concepts, Abeck, S., Clemm, A., Holberg, U., submitted to the 3rd International Conference on Integrated Network Management, publication pending.

XMP
X/Open Preliminary Specification, Systems Management: Management Protocols API (XMP), X/Open Document Number P170, ISBN 1-872630-32-4, July 1992.

XOM
X/Open CAE Specification, OSI-Abstract-Data Manipulation API (XOM), X/Open Document Number C180, ISBN 1-8720630-17-0, November 1991.

Chapter 1

Introduction

1.1 Problem Statement

There are a variety of management protocols currently being used throughout the industry. Although most management protocols deployed today are still proprietary, there is an increasing demand for standard solutions which allow interoperable management in a multi-vendor environment. Two major sets of management standards exist today:

- **Internet Management:**
  Internet standards defined by the Internet Engineering Task Force (IETF) Internet Activities Board (IAB)

  and

- **ISO/CCITT Management:**
  Open Systems Interconnection - Systems Management International Standards and Recommendations defined jointly by ISO and CCITT.

Both ISO/CCITT and Internet Management standards specify protocols to transmit management information over a communications interface. The approach used to achieve this task evolved from different perspectives. ISO/CCITT Management was designed by a large, multi-national committee and was intended for any type of management environment. Internet Management was developed through testing and redesign by a small group within the Internet community and was designed to provide management of TCP/IP networks.

It seems inevitable that both Internet and ISO/CCITT management technologies will continue to be deployed throughout the industry, resulting in a multi-protocol distributed management environment. The challenge is therefore to provide for coexistence and interworking of these standards-based technologies so that users can deploy the management products and tools which best meet their needs, without being locked into a single solution.

1.2 Scope and Purpose

This document addresses some of the issues involved in designing and choosing management products in environments which may include both Internet and ISO/CCITT standards-based technologies.

- Background and tutorial information on each approach is provided for those readers who may not already be familiar with one or both management standards.
- The two management approaches are compared and contrasted from an application developer and end-user perspective, providing insight into strengths and weaknesses, as well as criteria which may assist in technology selection.
- Possible strategies for coexistence and interworking are identified, including discussion of both existing and future solutions.

This document is the first in a series of specifications intended to address the overall problem. Since standards-based management products are themselves relatively new, very little exists today in the way of agreed methodologies for coexistence and interworking. As such, this document represents an initial discussion of near-term issues and possible solutions. Future

work is expected in the following areas:

- The initial problem space has been restricted to ISO / CCITT and Internet Management, in order to focus on problems facing current product development / deployment. Future expansion is expected to cover issues related to multi-vendor management platform technologies such as the Object Management Group (OMG) Common Object Request Broker Architecture (CORBA), the Open Software Foundation (OSF) Distributed Management Environment (DME), and Unix International (UI) Atlas.
- Within the initial problem space, this document does not provide detailed specification of any existing or future methodologies for coexistence or interworking. Instead, this document references existing specifications and identifies areas which require further development. Detailed design specifications will be developed separately as needed to describe specific methodologies which are proposed but do not currently exist. Areas of future work are identified in Chapter 4.
- It is not the role of this document to provide transition plans. This document serves as a balanced technical specification focused on analysis, coexistence, and interworking. This document makes no statement on transition plans or policies. It is expected that individual organizations may publish their own transition plans which contain policy statements related to procurement or deployment of ISO / CCITT and Internet technologies.

This document is one of a set of interrelated documents that make up the OMNI*Point* 1 documentation. The overall structure of the OMNI*Point* 1 documentation is described in the OMNI*Point* 1 Developers' Guide (see reference **OP1**).

1.3 Concepts and Terminology

Proprietary management protocols have traditionally been developed for specific types of devices or product lines. Each proprietary protocol tended to have a wide variety of message types that were used to manage the devices. Accommodating new devices or new features required adding functionality to existing message types and, in most cases, adding new message types to the protocol. This meant that not only did adding new devices or new functionality require additional management applications, but also required modification to the existing management protocol. This greatly complicated quality assurance and version control, since enhancements to the protocol affected all products using the protocol, not just those that utilized the enhancements. This approach was expensive and time consuming. Further, this continual accretion of functionality did not have a positive effect on the quality of the products, management or otherwise.

The management standards defined by ISO / CCITT and the Internet Task Force are designed to allow extensibility without requiring all users of the protocol to understand everything. Both ISO / CCITT and Internet management standards are based on the same fundamental concept: an *object model*. In this approach, management operations are performed on *objects* that represent the elements of the managed system. Each request may contain a list of attributes (or parameters) on which the management operation is to be applied. The request is performed on the object, and a response may be returned to the requestor. Spontaneous occurrences or conditions in the managed resources are also represented as messages about *objects*.

The use of an object model rather than a functional model greatly simplifies the problem. By expressing all operations in terms of a very small number of operators on objects and their attributes, it is possible to make the implementation essentially table driven. Adding new devices or new features becomes a case of merely adding definitions of new objects or attributes. The protocol itself does not change. This also means that the protocol implementation does not

have to be changed every time enhancements are made. The protocol itself does not understand anything about the objects or the attributes. It merely carries the necessary information. As long as the managed resources know their object and attribute definitions and an application knows what to do with them, the application can cause the management protocol to carry the necessary information to perform the operations.

Although both Internet and ISO/CCITT management specifications are based on *object models*, terms often differ, and the same term can be used with a different meaning. To assist the reader and avoid possible confusion, the following table provides a summary comparison between the terms used in each approach.

Internet Management Term	**ISO/CCITT Management Term**
MIB (Management Information Base)	Library of Object Class definitions
MIB Groups	Object Classes
Tables	Object Classes
Tables Entries	Object Classes
Object Types	Attributes
Instantiated Group or MIB Variables	Object Instance

Table 1-1 Comparison of Terminology.

For brevity, this document sometimes uses the terms *object* and *MIB* in a generic sense, intending to encompass both approaches with a single term.

The term *standard* is used throughout this document in reference to both ISO/CCITT and Internet documents. However, it should be recognized that these documents undergo different procedures for development and approval.

- ISO/CCITT *standards* are developed and approved by international organizations which consist of formal representation from most national bodies. These *standards* include both ISO International Standards and CCITT Recommendations.
- Internet *standards* are *Request for Comments* (RFC) documents which have been formally approved by the IAB. These documents are developed by less formal working groups, meeting under the auspices of the IETF, consisting primarily of individual technical contributors. RFC 1310 defines the Internet Standards Process (see reference **ISTDP**).

Chapter 2

Background and Tutorial

This chapter provides background and tutorial information regarding ISO/CCITT and Internet Management standards. Readers who are familiar with both ISO/CCITT and Internet Management standards may skip this chapter.

2.1 Internet Management

This section provides background and tutorial information on the set of Internet standards which collectively specify "Internet Management".

2.1.1 Historical Perspective

Prior to the emergence of Internet Management standards, network management for Transmission Control Protocol/Internet Protocol (TCP/IP) was performed with a small set of tools such as those shown in Table 2-1. In many cases, a logon to remote machines is required in order to execute these commands.

ping	Verify the ability to communicate with another system on the IP network via a loop-back test. Response time is also reported.
netstat	Networking status information on packets leaving, arriving, and discarded at the link, IP, and TCP levels, and routing table information
ifconfig	Check the network address, network mask, and operational status
traceroute	Checks where IP is failing on the network

Table 2-1 Existing Tools for TCP/IP Management

In 1987, Simple Gateway Monitoring Protocol (SGMP) was introduced as an interim standard protocol for TCP/IP network management. One of the objectives of the interim protocol was to seek reactions from the user community. SGMP included not only the protocol for communication between network management entities, but also identified variables which were to be monitored. This protocol was experimented with by several groups; for example, the National Science Foundation Network (NSFnet) backbone used SGMP in 1988.

From 1988 to 1990, many vendors implemented the successor to SGMP: Simple Network Management Protocol (see reference **SNMP**). This widespread implementation resulted in the SNMP being named an Internet Standard for TCP/IP network management. Another Internet Standard, the Management Information Base I (MIB-I), subsequently revised and republished as Management Information Base II (see reference **MIB-II**), defines a base set of variables that represent a managed resource or device.

In 1989, ISO/CCITT management standards received attention within the Internet community. Some believed that migration to OSI would be easier in the future if OSI-style network management was used to manage TCP/IP networks. To facilitate this, a draft Internet Standard was developed which used the emerging ISO/CCITT management protocol (CMIP) over a specialized set of OSI upper layer protocols (LPP) and TCP/IP. This approach, known as CMOT, was not widely implemented and has since been abandoned.

Several trade publications (see references **SNMPR**, **SNMPMPC**, **SNMPS**) are available which list commercially-available SNMP products, including SNMP management systems and SNMP agents.

2.1.2 Internet Management Specifications

Network management in the TCP/IP environment is defined and governed by three Request For Comments (RFCs) adopted by the TCP/IP governing body, the Internet Activities Board (IAB).

RFC 1155	Structure and Identification of Management Information for TCP/IP-based Internets
RFC 1157	Simple Network Management Protocol
RFC 1213	Management Information Base for Network Management of TCP/IP-based Internets: MIB-II

Table 2-2 Internet Management Specifications

Each of these three RFCs are recommended as standards and areas applicable to particular products must be implemented to assure interoperability in a heterogeneous environment. Of these, the most widely-recognized RFC is the one which defines the Simple Network Management Protocol (SNMP). The term SNMP is sometimes used as a short-hand reference for the entire set of Internet Management standards.

2.1.3 Internet Management Model

The Internet Management model is based on a client/server paradigm, where servers own resources and clients request services from the resources. In the Internet model, servers are called agents and clients are called managers or management stations. The agent interfaces with the underlying system to obtain and manipulate the network management information. The agent exposes management information by communicating with managers using the SNMP protocol.

Agents are located at each network entity that will be managed or monitored. Typically, SNMP-based agents run on gateways and routers, as well as on important servers like file servers or mail gateways, to allow monitoring of network server connectivity. Dedicated agents may also monitor specific hardware or applications. The manager or managing station is application software that runs at the Network Operations Center. Today, there is no widely accepted mechanism for integrating multiple SNMP-based agents on a single system, or for communicating between management stations.

In the Internet Model, all management operations are represented as alterations or inspections of objects, sometimes called MIB variables. Variables are either retrieved (Get) or altered (Set). The monitoring of the state of the network is primarily accomplished by polling for information; however, a limited number of unsolicited messages (Traps) can guide the timing and/or frequency of the polling.

2.1.4 Internet Management Protocol

The management protocol, SNMP, is a request/reply protocol that reflects a simple fetch/store paradigm. More complex operations on an agent system are requested by the side effects of the Set operation. A GetNext command allows traversal of the MIB in lexical order, according to type name. Fields are provided in SNMP to permit elementary authentication and the correlation of replies with corresponding requests. When an agent receives an SNMP message, it uses the supplied community name (the name assigned to an arbitrary pairing between an SNMP-based agent and a set of management applications) to check if the requested operation is allowed.

The SNMP standard requires an agent to support all five SNMP messages shown in Table 2-3.

Message	Description
Get_Request	Retrieve MIB variable(s).
GetNext_Request	Interrogate MIB variable(s) for which you do not know the variable names and/or how many entries are in a table.
Get_Response	Response to a Get, GetNext, or Set request.
Set_Request	Request to change MIB variable(s) to the supplied value(s)
Trap	A hint indicating that something has happened: cold/warm start, link down/up, authentication failure, or EGP neighbour loss. An enterprise-specific trap allows for vendor-defined extensions.

Table 2-3 SNMP Messages

In general, SNMP Get, GetNext and Set messages are generated by a manager, and Get_Response and Trap messages are generated by an agent. Figure 2-1 shows the Internet Management Station and Agent roles.

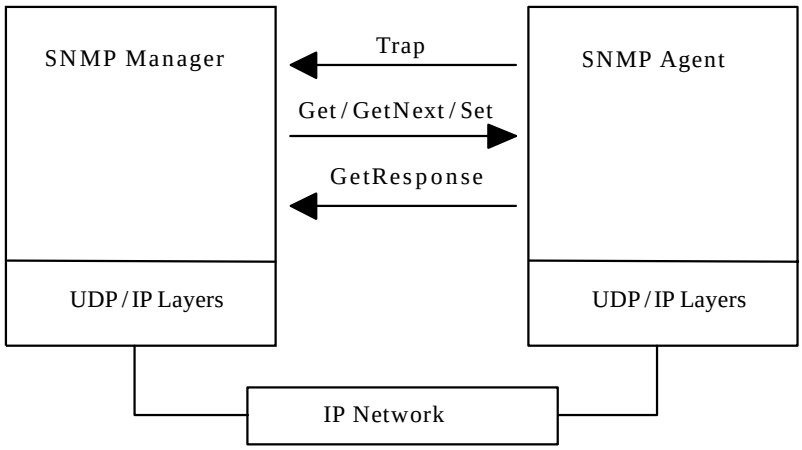


Figure 2-1 Internet Management Roles

An SNMP-based agent typically operates over the Internet connectionless transport protocol stack, known as UDP/IP. The agent interacts with the underlying system to obtain and manipulate the managed resources. An agent can send a trap to the manager if something important (such as a link failure) occurs; this is the exception rather the rule. More often, the manager sends some SNMP requests to the agent to obtain information (Get, GetNext) and/or change (Set) the value of some variables. To each of these requests, the agent responds with the Get_Response packet. Normally, an SNMP-based manager will be polling agents for information, so the agent generally takes a passive role. Only in a few cases will an agent take the initiative and send a trap (hint) to the manager to inform of the state change. If the manager receives the trap, it can take appropriate action to obtain more information - this approach is known as trap-directed polling. The SNMP standard discourages use of additional traps but

places no restrictions on their use.

2.1.5 Internet Management Information Model

The SNMP protocol does not specify which data, objects or variables are used for management, or how management information is represented. Instead, the Internet Structure of Management Information (SMI) standard (see reference **SIMI**) provides a methodology for defining the network management information as a Management Information Base (MIB). A MIB is built from a simple subset of Abstract Syntax Notation One (ASN.1) data types. Elements of management information within the MIB are defined as object types, with the following properties:

Object Descriptor	A textual, human readable name for the variable (like sysDescr)
Object Identifier	An ASN.1 identifier (like 1.3.6.1.2.1.1.1)
Syntax	One of the following ASN.1-defined types: number (integer), string, object identifier, IP address, counter (wraps on overflow), gauge (latches on overflow), time ticks (100ths of a second)

Table 2-4 Internet Object Type Definition

Complicated structures (tables) are created by aggregation into (records).

Another Internet standard, known as MIB-II (see reference **MIB-II**), defines the management information relevant for managing TCP/IP networks. Design objectives for MIB II included:

- maintain independence between management information and management protocol,
- define the basic set of information required to manage TCP/IP networks, and
- provide for the addition of enterprise-specific information.

MIB-II is an updated version of the original MIB-I. Changes between MIB-I and MIB-II involved deprecating one object group (the address translation group), adding a couple of new object groups, and modifying a few variables. Object groups currently included in MIB-II are summarized in the following table.

System	Description, Location, etc of Agent System
Interfaces	Description, Speed, etc of Interfaces
Address Translation (deprecated in MIB-II)	Maps Network and Physical Addresses
Internet Protocol (IP)	Datagrams in/out/forwarded, etc
Internet Control Message Protocol (ICMP)	ICMP input/output statistics
Transmission Control Protocol (TCP)	Max/current connections, input/output counts
User Datagram Protocol (UDP)	Datagrams in/out/invalid, etc
Exterior Gateway Protocol (EGP)	EGP messages received/errors, etc
Transmission	Media MIB plug-in (FDDI, Token-Bus, etc)
SNMP	SNMP Agent statistics/counts

Table 2-5 Internet MIB-II Object Groups

SNMP uses the ISO/CCITT registration tree defined in ISO/IEC 9834-1 (CCITT Recommendation X.660) for the purpose of naming MIB variables. Each object type is assigned an ASN.1 OBJECT IDENTIFIER from the Internet portion of the global registration tree. For example, the Internet MIB-II is assigned the following OBJECT IDENTIFIER:

iso(1) international-organization(3) dod(6) internet(1) mgmt(2) mib(1)

Object types within MIB-II are assigned OBJECT IDENTIFIERs subordinate to this arc, for example:

iso(1) international-organization(3) dod(6) internet(1) mgmt(2) mib(1) system(1) sysDescr(1)

Indirect addressing of data within the MIB is accomplished by using the OBJECT IDENTIFIER value assigned to the type name as an address. Since data addressing is on a single level, multiple instances of a data type cannot be directly addressed by this method. Addressing multiple instances of the same data type in a single MIB is accomplished by grouping the instances in the columns of a conceptual table and using row values of one of the columns as an additional, associative address (that is, appending an index to the end of an object identifier). Except where information is modeled as MIB tables, there is an assumption that a single instance exists within the agent. There is no mechanism in the protocol for global naming. All naming is local relative to the agent (system).

The separation of management information and the protocol used to monitor/manage that information allows new MIBs to be added without a need to change the protocol, and potentially allows MIBs to be managed by multiple protocols. Internet MIBs allow for enterprise specific extensions, so users can add new variables. The specific features and functions implemented by SNMP agents and managers are typically verified through pair-wise interoperability trials rather than formal conformance testing.

2.1.6 Internet Management Future Developments

SNMP management seems well-entrenched as the method to manage TCP/IP networks, but future extension to accommodate other types of networks seems questionable. Future developments are likely in the following areas:

- **New Standard MIBs:**
  Refer to Section 3.7 on page 35 for a sample list of Internet MIBs currently under development. Each new MIB will require changes and additions to the existing MIB specific applications in order to support these new functions.
- **OSI Internet Management:**
  The original CMOT approach has been replaced by *CMIP For The Internet*. This layered communication architecture utilizes a general-purpose mapping between OSI and Internet transport protocols (see reference **ISOTCP**) which appears to offer more promise of industry acceptance than the original CMOT. The GDMO translation of MIB-II has also been recently updated to align with the final ISO/CCITT specifications included in OMNI*Point* 1.
- **Secure SNMP:**
  A proposed Internet Standard has been developed which improves SNMP access control and authentication mechanisms through protocol encapsulation. This enhancement may increase the usage of SNMP for management control operations, but deployment may be delayed by the emergence of SMP (see below).

- **Simple Management Protocol (SMP):**
  This is a proposal developed by four developers for enhanced Internet management, including:
  - Modified Secure SNMP
  - Bulk Data Transfer
  - Primitive Manager to Manager Interaction (Inform-Request)
  - Better Managed Object definition
  - Better Error Handling
  - Configurable Exception Reporting
  - Less Memory
  - Generalized Addressing
  - Independent of Underlying Protocol

Coexistence between SNMP and SMP is defined using a proxy approach. The IETF/IAB has formed a new working group to develop SNMP version 2, starting with the SMP and Secure SNMP specifications as a basis. The intent is to transition and upgrade in a single step.

2.1.7 Internet Management Summary

Internet Management was designed as a solution to the urgent and tractable problem of managing bridges, routers and the like in TCP/IP networks; the protocol is deliberately simple and the initial MIB was tailored explicitly to accommodate only Internet addressable equipment. The number and complexity of functions realized by the management agent are minimized, focusing on monitoring and control in a debugging paradigm. This implies lower development costs and easier implementation of SNMP-based agents for developers of network management tools. However, the quest for simplicity, and the specific scope of the problem, has introduced limitations in addressing, requirements for event reporting, scalability to large networks, etc..

2.2 ISO/CCITT Management

This section provides background and tutorial information on the set of International Standards and CCITT Recommendations which collectively specify Open Systems Interconnection - Systems Management, referred to in this document as ISO / CCITT Management.

2.2.1 Historical Perspective

ISO / CCITT specification of management standards began in 1984, when management was included in the initial Open Systems Interconnection (OSI) Reference Model (RM). A Management Framework extension to the OSI RM was published in 1989. In this framework, systems management is accomplished through the exchange of object-oriented request / reply protocol between managing and managed systems. ISO / CCITT System Management Functional Areas (SMFAs) subdivide and summarize the requirements for fault, performance, security, accounting and configuration management. The ISO / CCITT Systems Management Overview reference *SMO*, published in 1991, further defines three different kinds of Systems Management standards:

Management Communication	Service and protocol for exchanging management operations and notifications
Management Information	Represents the resources that can be managed
Management Functions (SMFs)	Common tasks related to many resources or applications

Table 2-6 ISO / CCITT Management Overview

Version 1 of the ISO / CCITT Common Management Information Protocol (CMIP) was published in 1989, was used as the basis for the NM Forum Release 1 in June 1990, and was demonstrated at Interop '90 by a small number of portable stack vendors. By the end of 1990, defects discovered during initial implementation were corrected by the CMIP Version 2 and a four-year moratorium on CMIP extensions was put in place to ensure stability. Another dozen management standards reached final IS status by the end of 1991, defining functions and information for configuration and fault management, and fifteen vendors participated in the NM Forum R1 demonstration at Telecom '91. Deployment of ISO / CCITT management has been slow, but may be accelerated by the publication of OMNI*Point* 1 in 1992. By late 1992, almost three dozen management standards and profiles reached final IS or ISP status, including standard MIBs for the OSI Transport and Network Layers. A trade publication (see reference **OSIPR**) lists over a dozen commercially-available CMIP products.

2.2.2 ISO/CCITT Management Specifications

ISO / CCITT Systems Management is defined by a set of International Standards and Recommendations jointly developed by the two organizations. The following specifications have been published as full standards / recommendations and were included in OMNI*Point* 1:

CCITT	ISO/IEC	Content
X.700	7498	Management Framework
X.701	10040	Systems Management Overview
X.710	9595	Common Management Information Services (CMIS)
X.711	9596-1	Common Management Information Protocol (CMIP)
X.720	10165-1	Structure of Management Information (SMI)
X.721	10165-2	Definition of Management Information (DMI)
X.722	10165-4	Guidelines for the Definition of Managed Objects (GDMO)
X.730	10164-1	Object Management Function
X.731	10164-2	State Management Function
X.732	10164-3	Attributes for Representing Relationships
X.733	10164-4	Alarm Reporting Function
X.734	10164-5	Event Report Management Function
X.735	10164-6	Log Control Function
X.736	10164-7	Security Alarm Reporting Function
X.740	10164-8	Security Audit Trail Function

Table 2-7 ISO / CCITT Management Specifications

An International Standardized Profile (ISP) collects together a set of base standards, selects options within them, and specifies practical constraints. There are currently two profiles which collect together base standards for management protocol (CMIP) and the supporting upper layer protocols (ISO / CCITT ROSE, ACSE, Presentation, and Session). There are also five Management Function draft profiles which collect together base standards for systems management functions and the supporting management communications stacks defined by CMIP Profiles. All of these profiles are included in OMNI*Point* 1, along with several additional management function and information specifications which are based on the ISO / CCITT management model (e.g., NM Forum Path Tracing, Trouble Administration).

2.2.3 ISO/CCITT Management Model

The ISO / CCITT Management model also uses a client / server paradigm, where the client is known as the managing system and the server is known as the managed system. The managed system takes the agent role, receiving management operations targeted to managed objects and forwarding notifications emitted by managed objects. The managing system takes the manager role, invoking management operations and receiving notifications. The same system may take on different roles for each protocol exchange, thus manager-to-manager communication can be accomplished by one system briefly taking on the agent role for a given protocol exchange.

In the ISO / CCITT Model, management operations are performed on managed object instances and their properties, including *attributes*. As in the Internet Model, attributes are either retrieved or altered, and unsolicited messages (Notifications) can occur. However, despite their overall similarity, the two models differ in many ways, as described in the following sections.

2.2.4 ISO/CCITT Management Protocol

ISO/CCITT Common Management Information Protocol (CMIP) - see reference **CMIP** - defines a much larger set of messages than SNMP, and provides mechanisms for distribution of load between manager and agent, allowing optimization of the traffic load generated by management. In addition, CMIP distinguishes between operations on objects and operations on attributes of objects. A companion standard, the ISO/CCITT Common Management Information Service (CMIS) - see reference **CMISD** - describes the abstract services which are realized through the exchange of CMIP protocol. CMIP messages are summarized in Table 2-8 below. The syntax of the messages themselves and the management information they carry are defined in terms of Abstract Syntax Notation One (ASN.1).

Message	Description
Event Report	Report an event about a managed resource to another CMIS service user
Get	Retrieve management information from another CMIS service user
Set	Request modification of management information
Action	Request another CMIS service user to perform an action
Create	Request another CMIS service user to create a managed object
Delete	Request another CMIS service user to delete a managed object
CancelGet	Invoked by a CMIS service user to cancel a previously issued Get

Table 2-8 CMIS Services

Most CMIP messages contain the following parameters:

- the type of operation or notification,
- an *invoke* identifier for request/reply correlation,
- the identity of the target or source managed object(s),
- an optional timestamp, and
- object-specific or operation-specific information.

The identification of the target managed object(s) can be based on certain combinations of:

- its type (managed object class),
- its name (managed object instance),
- its position in the containment hierarchy (scope), and/or
- a predicate referring to its attributes (filter).

These features allow a managing system to affect a number of objects with a single operation and to select objects depending on their dynamic state (i.e., test-and-set). Object instances are organized within a containment hierarchy by their names (see Section 2.2.5 on page 16, ISO/CCITT Management Information Model).

In general, CMIP Get, Set, Create, Delete, Action, and CancelGet request messages are generated by the managing system in the Manager role, and response messages are generated by the managed system in the Agent role. The CMIP Event Report request message is typically generated by the managed system. Figure 2-2 shows the ISO/CCITT Manager and Agent roles.

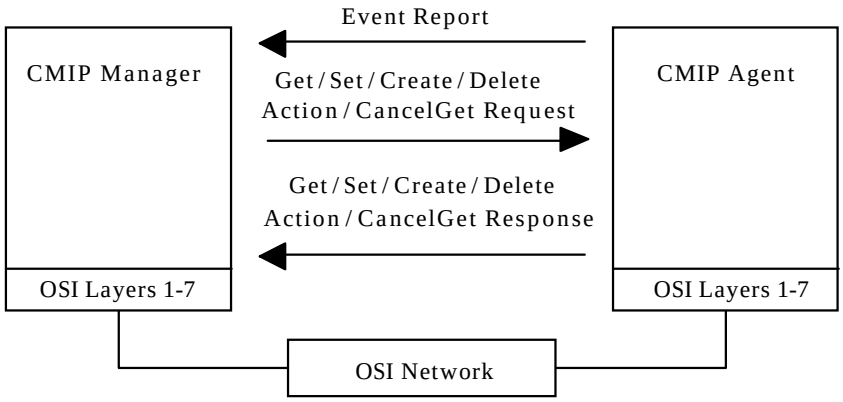


Figure 2-2 ISO / CCITT Management Roles

A CMIP Event Report is an unsolicited indication of some important event in the managed system. It contains an indication of the event and associated information that may be useful in understanding the event. CMIP supports both confirmed and unconfirmed Event Reports. The ISO / CCITT Model takes an event driven view of management, so there is wide use of the CMIP Event Report. However, other standards define control mechanisms which can be used to discard, log, or forward notifications as Event Reports, at the manager's discretion.

Scoping and Filtering allow CMIP operations to be applied to the whole allow CMIP operations to be applied to the whole MIB, such that all the attributes that an agent wishes to expose to a manager can be retrieved in a single management operation. The Filter parameter defines the attribute conditions that must be satisfied for the operation to take place. The Scope parameter allows the requester to specify where in the containment hierarchy to start performing the operation. Sets, Gets, Actions, and Deletes may optionally use Scope and / or Filter. When a request applies to multiple managed objects, these operations may generate multiple replies to a single request. For example, a Get of attribute values of several managed objects generates one reply for each managed object, each reply containing the attributes of a single managed object.

CMIP also provides a rudimentary Synchronization option that allows each operation to be done as best-effort or *atomic*. Best effort implies that the receiver will make a best effort to perform the operation on each managed object selected. Atomic implies that

> all managed objects selected for the operation are checked to ascertain if they are able to successfully perform the operation; if one or more is not able to successfully perform the operation, then none perform it, otherwise all perform it

- see reference **CMISD**.

ISO / CCITT specifications also include several Systems Management Functions which provide the additional management capabilities, built on top of the service provided by CMIS and CMIP.

- Management information which is common to many managed resources is often specified as a Systems Management Function. This allows the same management information to be manipulated in the same manner, independent of the underlying resource. For example, the ISO / CCITT State Management Function (see reference **SMF**) defines a generic state management model which includes common attributes for representing operational, administrative, and usage status, and a common notification for signaling changes in state.

- The ability to control management itself is also specified as Systems Management Functions. These functions usually define managed objects which represent the control capability. This allows the same management protocol and information model to be used to control the management service; no specialized control protocol is required. For example, the ISO/CCITT Event Report Management Function (see reference **ERM**) defines an Event Forwarding Discriminator managed object which allows managers to initiate, configure, suspend, resume, or terminate event reporting by the agent.

Systems Management Functions make use of the ISO/CCITT management information model for representing managed objects. They typically provides a standard representation for a task which is common to a number of management applications. For example, the event control service described above is relevant to any application which monitors events.

2.2.5 ISO/CCITT Management Information Model

The ISO/CCITT management information model an object-oriented approach which represents real-world systems and resources as managed objects. As described in the ISO/CCITT Management Information Model (see reference **MIM**), managed objects are characterized by:

- the operations they accept,
- the notifications they emit,
- the attributes (data) they make available, and
- the behavior they exhibit.

A key aspect of the object-oriented approach is that objects are characterized by means of an abstract interface specification. The actions, notifications, attributes, and behavior are those observed at the object boundary. ISO/CCITT Guidelines for the Definition of Managed Objects (GDMO) - see reference **GDMO** - defines the template notation used to express these properties. Objects that share identical specifications are grouped into classes. New class specifications may be derived from existing class specifications using strict multiple inheritance: new classes (subclasses) inherit all of the specifications of the original classes (superclasses), as well as additional specification. The object oriented design of the ISO/CCITT methodology provides for four key functions:

Encapsulation	Provide for access to data via methods and hides internal implementation from the user.
Object Class Structure	Provides the ability to extend and combine existing interface definitions to create new interfaces
Class Inheritance	Allows one class to be refined from another class providing for reuse of specifications.
Allomorphic Behavior	Provides a mechanism for migration and coexistence between multiple versions of class definitions.

Table 2-9 ISO/CCITT Information Modeling

One of the important characteristics of this model is Allomorphism. Allomorphism allows the same operation to be performed on objects of different classes; each object responds to operation in a way that is defined in a common *compatible class* definition. This technique allows implementation of common operations for use with many types of objects, so that new object

classes may be supported without requiring new application code.

In the ISO/CCITT model, objects may enter into relationships with one another. Binary relationships (e.g., supplier/consumer relationships) are typically represented as attributes or objects which refer to related objects. Containment relationships are typically reflected in the name of the managed object. A subordinate managed object is named by the combination of the name of its superior (containing) object, and information uniquely identifying this managed object within the scope of its superior object. These names are combined recursively into a single rooted hierarchy called the naming tree. The top level of the naming tree is referred to as root.

A template called a Name Binding defines the possible superior and subordinate objects which make up a naming tree, and the attributes which are used for naming. Each managed object instance named by a single-valued X.500-style Distinguished Name. A Distinguished Name is an ordered sequence of (attribute, value) pairs, where the order (the superior/subordinate structure) and attributes (of the pairs) are specified by Name Bindings. For example:

{ {systemTitle = { {countryName="US"},{organizationalUnitName="xyz"},{commonName="abc"}}}
,{discriminatorId = "efg"}}

identifies an Event Forwarding Discriminator called *efg* contained by a System called {US|xyz|abc}, which is itself contained by the global root. This naming architecture provides a flexible structure in which global uniqueness can be assured. In addition, relative and local form class and instance names are also permitted. Additional Name Bindings can be defined for existing managed object classes on an as-needed basis.

The ISO/CCITT management information naming tree is not related to the OBJECT IDENTIFIER (registration) tree. OBJECT IDENTIFIERS are assigned to ISO/CCITT managed object classes and their properties, but these identifiers represent types, not names. In this model, OBJECT IDENTIFIERS tend to be assigned from various arcs in the registration tree, depending upon the defining organization. For example, the OMNI*Point* 1 Library Volume 4 assigns OBJECT IDENTIFIERS starting from the arc:

iso member-body(2) canada(124) forum(360501)

Unlike Internet Management, which was initially designed for managing TCP/IP networks, ISO/CCITT Management was not designed specifically for managing OSI networks. As a result, there is no single ISO/CCITT MIB which is analogous to the Internet MIB-II. However, ISO/CCITT Definition of Management Information (DMI) - see reference **DMI** - includes definitions which apply to most ISO/CCITT-based management environments; a few are shown in Table 2-10.

System	Description, Status, etc of Agent System
Top	Attributes common to all object classes
Event Forwarding Discriminator	Controls forwarding of notifications as Events
Log	Controls logging of notifications as Records
Log Record	Attributes common to all records; the specific log records defined by SMFs are also defined

Table 2-10 ISO/CCITT DMI Object Classes

2.2.6 ISO/CCITT Management Future Developments

Future developments in ISO/CCITT Management will take two complementary paths: further development of the framework standards and recommendations within ISO and CCITT, and development of profiles and managed-object definitions by other groups. Work is already under way in the following areas:

- **Extended Systems Management Architecture:**
  Includes management domains, relationship models, and schema (management knowledge) management.
- **Additional System Management Functions**:
  Includes new functions related to management of performance (e.g., summarization, workload monitoring), security (e.g., access control), and accounting (e.g., accounting meter), as well as additional configuration/fault-related functions (e.g., software management, time management).
- **Managed Object Definition**:
  Underway throughout the industry, in a wide variety of technology and service domains (refer to Section 3.7 on page 35 for examples).
- **Management Profiles and Ensembles:**
  Will collect together new function specifications and managed object definitions, as they reach completion, into cohesive specifications which facilitate implementation, procurement, and interoperability.

2.2.7 ISO/CCITT Management Summary

ISO and CCITT have attempted to steer a difficult course by developing a framework that is specific enough to avoid protocol anarchy, but not so restrictive as to stifle management solutions. The object-oriented approach that it is based upon helps to separate generic and resource-specific issues, allowing for overall flexibility and reuse of common functions and information. This generic approach implies greater complexity and therefore increased costs (both development and operational). However, expansion to cover new applications, functions, and resources is greatly simplified, and flexible management tools can be used in many different ways, depending upon the environment to be managed.

Chapter 3

Comparison of Management Technologies

This Chapter compares and contrasts the characteristics of Internet and ISO/CCITT Management from an application developer and end-user perspective, providing insight into strengths and weaknesses, as well as criteria which may assist in technology selection. The following topics are covered:

- efficiency and performance consequences
- robustness of management solutions based on these standards
- flexibility and extensibility inherent in each model
- security-related issues relevant to each environment
- built-in functionality available to applications
- cost considerations
- likely technology and application domains for each approach.

The intent of this comparison is **not** to contribute to OSI versus TCP/IP protocol wars, but rather to provide guidance to those readers who need to make use of standards-based management solutions. However, it is inevitable that any subjective discussion of strength and weakness will be subject to debate and difference of opinion. When possible, this comparison is based on case study, implementation experience, and actual product usage. Often this is not possible due to the immaturity of products and corresponding deployment. The trade-offs are complex and cannot be resolved without considerable experimentation and analysis which has not yet occurred. It is expected that this Chapter will be updated over time to incorporate new information and further analysis. Readers are therefore cautioned to treat this Chapter as initial guidance, focusing not on which approach is by itself better, but instead on how to use both management approaches together in a cooperative and complementary fashion. The criteria provided by this Chapter should be used in conjunction with the coexistence and interworking strategies described in Chapter 4.

3.1 Efficiency

The overall efficiency and performance of a management system results from a combination of factors that must be balanced in relation to each other, including:

- consumption of network bandwidth by management traffic
- management software processor and memory usage
- product code size and complexity.

The following sections compare aspects of Internet and ISO/CCITT Management which affect these tradeoffs, such as:

- impact of the underlying protocol stack
- use of polling versus events
- operations affecting multiple objects.

3.1.1 Polling vs. Event-Driven

Both SNMP and CMIP include messages which can be used to signal events. The real difference is in how these are used. SNMP tends to be used in a polling mode, where the manager periodically polls agents for information of interest, and very few Traps are defined for the Internet MIB-II. CMIP tends be used in an event-driven mode, where agents automatically inform managers about unsolicited events of interest, and GDMO-based object classes tend to include many notifications which are likely to occur on a regular basis.

In a polling paradigm, the manager is responsible for requesting the appropriate information and filtering out any unwanted data. This approach can consume a great deal of network traffic and processing power to present useful information. For example, one experiment in a 500-node network polling every 10 seconds for a single variable generated 62kbps. While the polling overhead may not be significant in small networks, this overhead increases with network size, and polling cycles can become long and unresponsive in very large networks. However, polling is relatively straight-forward to implement, especially on the agent side, and is commonly used to reliably detect some types of failures (e.g., crashes, partitioned networks).

In an event-driven paradigm, agents are responsible for reporting any spontaneous events deemed important, as well as periodic reporting of performance information. Event traffic overhead can be considerably less than that which results from repeated polling in large networks. An event driven approach can be most effective in high uncertainty/stochastic environments, but a potential problem can arise when a resource fails without generating an event. Also, more agent-side complexity is required to support the event-driven model than the polling model.

ISO/CCITT system management functions allow defined events to be disabled, forwarded or logged, as configured by the manager. There are no corresponding standard mechanisms for Internet Management; ad hoc methods are evolving in implementations, but solutions are inconsistent. For example, RFC 1271, Remote Network Monitoring (RMON) MIB, addresses some of these aspects for Ethernet networks. Lack of standard management control mechanisms make it difficult to for an SNMP-based manager to control trap generation or to permit logging of traps on the agent side. However, agents which support these ISO/CCITT control functions are more complex because they require hooks into the local environment to detect events.

Ultimately, the actual traffic load for either paradigm is highly dependent upon the frequency of important events and fine tuning of management software to operate in a manner appropriate for the managed environment. A methodology known as trap directed polling represents a viable mix between these two paradigms. In this approach, a trap or event is used by the agent

to signal an error or interesting condition, followed by manager polling related to the subject or source of the trap or event. ISO/CCITT management can also be used in a polling mode and the decision on which paradigm to use depends on the management policy of the implementor and use (when the latter is given sufficient flexibility by an implementation).

3.1.2 Operations on Multiple Objects

While individual Gets and Sets provide good open-ended management tools, they can also be very inefficient in any real system where many individual pieces of data are needed to diagnose a problem or to change the configuration. In this case, generating a Get or Set message for each affected piece of data greatly increases not only the management traffic, but also the processing overhead. Since each message will be treated as a separate transaction, multiple messages tend to complicate the book-keeping on both sides. Therefore, one major optimisation is the ability to get or modify several pieces of data using a single message. The advantage is more efficiency in bandwidth and processing than any sort of implied synchronization or atomicity semantics.

Situations in which operations might be performed on multiple pieces of data include:

- requests on several attributes of the same object
- requests on closely related attributes of different but closely related objects
- requests on the same attribute(s) of different object instances.

Both SNMP and CMIP provide tools which allow operations to apply to more than one piece of data. However, the tools and the way in which they can be used are quite different.

SNMP allows operations on multiple variables in a single message; this approach decreases the number of messages actually sent, but the amount of data sent is not reduced significantly and messages can get quite lengthy. Since an SNMP message must fit in a single UDP message, there is a definite upper bound on the size of the list. In an SNMP environment, variables must be addressed within the context of agent, and different names are required to access each variable. Thus, performing an operation on a large number of related (or unrelated) variables requires transmission of considerable address information. In the case of a Set on related variables (such as a single element in multiple rows in a table), variables must also be duplicated in the message, even though they are being set to the same value. Using GetNext can minimize the processing requirements of the requester, but does not reduce the number of messages required or the amount of work required by the agent. This solution is optimized for simplicity of agent implementation, at the cost of greater complexity in the manager and less efficient use of network bandwidth.

Scope and Filter allow a single CMIP message to request an operation on a set of related attributes within the same or different objects. For example, an CMIP-based management application can request port information for all bridges in a network with a single scoped command. Filtering can be used to weed out unwanted objects or attributes to constrain the operation and prevent transmission of unnecessary information. This can greatly reduce the amount of traffic on the network, but is done at a significant cost in manager and agent size and complexity (the manager is impacted when the request requires sending messages to multiple agents). Implementation and use of CMIP Scope and Filter is optional; if an application depends upon Scope and Filter, it requires the AOM12 CMIP Profile (see reference **ERM**), otherwise either the AOM11 CMIP Profile (see reference **BMC**) or SNMP (see reference **SNMP**) can be used.

Scoping and filtering must be used with care to prevent undesirable results that may occur if the expression selects objects which were not intended. Although a waste of processing and bandwidth, if extra information is returned in response to a Get, one can simply throw away that which is of no interest. However, for Sets, Actions, or Deletes, unintentional results can be disastrous, and cannot be inverted algorithmically. It may be possible (if the original values

were known) to invert a Set operation by issuing a Set_Request which lists explicitly every (managedObjectId, attributeId) returned in the Set_Response. However, sending another Set_Request with the same scope and filter and the original value will not generally produce the desired result. Caution should be exercised when using scoped Set and Delete operations.

Scope and Filter are very powerful constructs, and may be better suited for communication between managers than between manager and agent. In the manager-agent interactions, it may be desirable to keep as much of the processing overhead as possible in the manager and minimize the processing in the agent (since the agent has other work to do besides management). However, there is a creative tension between this requirement and demands on network bandwidth. Complete implementation of scope and filter greatly increases the complexity of the agent; this is made somewhat better by the profiles which limit the complexity of the expressions possible in a Scope or Filter. On the other hand, presenting scoped and filtered requests to an agent will tend to minimize the network bandwidth consumed. For a manager, (or a manager functioning as an agent, as in manager-to-manager communication) the increased computational load may not be unreasonable. Ultimately, the ability to support Scoping and Filtering depends upon the size and complexity of the system on which the manager or agent process resides. Applications developed to manage low-end agents should not assume Scoping and Filtering.

When a CMIP Set, Get, Action, or Delete request retrieves large amounts of data from multiple managed systems via a single request, responses are sent back as multiple linked replies. The multiple reply facility allows the responder to generate a reply from each managed object as the request is being processed, rather than having to assemble all of the responses into a single reply which is sent as one message. The multiple reply is also used with operations that may take some time to complete, to indicate progress or to send intermediate results.

SNMP does not support multiple replies, but instead provides the GetNext operation as the means to traverse a table. Using SNMP, the manager sends a GetNext, and the Get_Response contains the value of the next lexicographical item in the SNMP naming (registration) tree. The manager then changes the Get_Response into a GetNext and sends it again. This process is repeated until the table has been traversed. Using CMIP, the manager would send a single Get operation with the appropriate scope and filter parameters (or the table may be defined as a single, complex attribute), and the entire table is returned either as a single Get Response or as multiple replies (each containing a table entry). In the worst case, CMIP scoped Get requests generate half as many messages as SNMP GetNext requests for the same table; in the best case, $^1/_n$ as many messages are sent (where "n" is the number of items in the table). The SNMP-2 proposal recognizes this problem, and defines a new GetBulk message to improve table retrieval.

3.1.3 Impact of Underlying Protocol Stack

CMIP is designed for use over a connection (usually, but not necessarily, ACSE), while SNMP is typically used over a connectionless datagram service (UDP). Connection-oriented communication requires explicit acknowledgment to ensure delivery and message order is preserved. In an efficient implementation, time-outs to detect lost messages in a connection-oriented transport protocol are kept near the order of the network round-trip time. In a connectionless environment, the time-out to detect a Get or Set lost by a failure of the agent cannot be less than the maximum time to fulfill a request and is less deterministic, taking into account that the agent may be busy with more urgent matters. There may be a significant difference in these two values, possibly one or more orders of magnitude. In most cases, the application delay incurred from loss of one connectionless message in the network will exceed the time required to establish a connection and deliver the first message over that connection. The connect-and-send time is of the order of at least two round trips; the agent response time-

out will seldom be that low. This makes the crossover point of the tradeoff very low and argues for the use of a connection oriented protocol. Some SNMP implementations optimize timeout and retransmission delays by re-using transport protocol algorithms designed for this purpose.

When CMIP is deployed over a full seven-layer OSI stack, including ACSE and ROSE, non-management protocol overhead is a significant proportion of the total size of messages. This is especially true for messages which involve only a few attributes (as is the case for many manager/agent interactions). The non-management protocol overhead incurred by SNMP over UDP is substantially lower.

Both SNMP and CMIP protocol messages are limited in length by the underlying protocols. The minimum-maximum protocol message length specified for SNMP over UDP is 484 octets; some implementations support up to 8k octets. The minimum-maximum protocol message length specified for CMIP profiles over ACSE/ROSE is 10k octets; some implementations support up to 64k octets. Application developers cannot generally assume product support for lengths greater than this minimum-maximum, except when developing for a specific target platform or network. CMIP is required in cases where protocol message lengths regularly exceed the limits imposed by SNMP/UDP.

It should be noted that both SNMP and CMIP are beginning to be deployed over a wide variety of underlying protocol stacks other than those mentioned above, including IEEE CMIP Over LLC (CMOL, see reference **LAN/MAN**), CMIP For The Internet (see reference **CMIPI**), and SNMP Over OSI (see reference **SNMPO**). This should be considered when developing management applications, so that the application minimizes assumptions made about the underlying stack, and that logic associated with the underlying stack is isolated.

3.2 Robustness

In this context, the term *robustness* refers to the strength and integrity of management services. There are many aspects which together determine robustness, including:

- reliable delivery
- synchronization and atomicity
- granularity of functions
- end-to-end application confirmation.

3.2.1 Reliable Delivery

There is considerable controversy over the use of connection-oriented and connectionless transport protocols to support management. CMIP was designed for use over a connection which provides reliable delivery, while SNMP uses UDP for request/response traffic, assuming that the response will serve as sufficient acknowledgment to ensure reliability. (Note this philosophy applies to Get and Set, but not to SNMP Traps, which have no response.)

Furthermore, the authors of SNMP believed that using UDP would allow management traffic to get through, even when network conditions were bad (that is, by avoiding connection establishment and retransmission). In contrast, the authors of CMIP believed that using a connection provides a better guarantee of delivery, since connection-oriented transport implementations are finely tuned to adjust to the vagaries of the underlying network.

If an application does not depend upon reliable delivery, then SNMP over UDP is undoubtedly more efficient (refer to Section 3.1.3 on page 22). However, if an application requires reliability, then CMIP over a connection provides built-in reliable delivery mechanisms. SNMP can also be used by applications which require Get or Set (but not Trap) reliability; in this case, the onus of ensuring reliable delivery falls to the application developer. Although relatively uncommon, SNMP can also be used over a TCP connection for reliable delivery.

3.2.2 Synchronization and Atomicity

When a given management operation can be performed using a single message, a rudimentary level of atomicity and synchronization is provided by both CMIP and SNMP agents, which first check to see if the operation can be performed before doing so. However, neither CMIP or SNMP provides true synchronization or two-phase commit semantics, and there is no synchronization provided at all for operations which require the exchange of multiple messages.

SNMP message complexity and size limitations make it difficult to perform complex management operations with a single message. The GetNext facility has a further disadvantage, especially when inspecting large tables such as OSPF routing tables or file system directories. Since there is no ability to lock the table before starting the process, the contents of the table may be changed before the entire table has been read. It is impossible, therefore, to tell if any inconsistencies in the table are because of the problem you are looking for or because the routing tables were updated during the process of Getting each row of the table. Neither protocol provides any table locking mechanism. The scoping, filtering, and synchronization parameters provided by CMIP allow for more complex management operations to be attempted with a single message. Using CMIP, the table access problem described above can be reduced by accessing the entire table at once with the appropriate use of scope and filter. There is no guarantee that changes will not occur during generation of multiple replies, but given the shorter timespan required for the operation, the probability of inconsistency is significantly lower.

If true synchronization and atomicity is required by the application, then CMIP must be used in conjunction with other OSI protocols tailored to this problem (e.g., OSI Session Layer Synchronization services, OSI Commitment and Recovery (CCR) services, OSI Transaction Processing (TP) services). However, no approved standards are currently available for this sort of combined usage, and it is unlikely that small agents will ever support these additional protocols (except perhaps for Session Synchronization services). There are no corresponding Internet RFCs for synchronization and atomicity protocols.

3.2.3 Granularity of Functions

As noted previously, CMIP provides in-built messages for Create, Delete, and Action operations. MIBs defined for use with SNMP typically use Set to provide the effect of these CMIP operations. Using the Set operation to invoke another operation as a side-effect is called *over loading*.

This approach makes it difficult to authorize access for normal Set operations, but not for Create or Delete Set operations. The increased granularity of CMIP operations makes it easier to establish selective access controls.

Provision of separate Create, Delete, and Action operations also offers some protection against unintentional side-effects that may result from over-loading the Set operation. It should be noted that both ISO/CCITT and Internet MIBs contain over-loaded Set operations; application developers are cautioned to recognize these cases and their side effects.

3.2.4 End-to-End Application Confirmation

Both CMIP and SNMP provide application (end-to-end) confirmation for Get and Set operations. However, CMIP also provides for non-confirmed Set operations and confirmed Event Reports. This allows greater flexibility for the application developer. Further, the inability to confirm SNMP Traps means that reliable delivery cannot be provided to event-driven applications.

Application developers should be aware that underlying protocol reliability does not substitute for end-to-end application confirmation. Even when CMIP is used over a reliable OSI stack, the manager application is still responsible for detecting a misbehaving agent which has an active, healthy ACSE association but is not responding to incoming requests for some other reason.

3.3 Flexibility and Extensibility

This section examines the flexibility and extensibility inherent in ISO/CCITT and Internet standards for management information and protocol.

3.3.1 Information Modeling Aspects

As detailed in Chapter 2, ISO/CCITT and Internet Management Information Models differ in representation, style, and capabilities. Table 3-1 summarizes these differences.

Features	ISO/CCITT	Internet
Attribute/Variable Syntax Type	Simple or Complex ASN.1 constructs	Simple ASN.1 Constructs
Attribute/Variable Syntax Uniqueness	Syntax is reusable among Attributes	Syntax must be unique among Variables
Attribute/Variable Optionality	Mandatory or Optional Attributes can intermingle within Managed Objects. Optional, can be static or dynamic.	Mandatory and Optional Variables cannot exist within the same Table. Optional supported statically only.
Create/Delete/Action Semantics	Defined as properties of the Managed Objects	No such concept exists (semantics can be achieved using Set of special-purpose MIB Variables)
Notification/Trap Semantics	Defined as properties of the Managed Objects	Independent of MIB Variables
Inheritance Relationship	Inheritance is used to successively refine and expand Managed Object definitions	No such concept exists (objects cannot be derived from other objects).
Containment Relationship	Containment used to organize Managed Objects by existence dependency	No such concept exists, though variables may be grouped into Tables
MIB Structure Style	Utilizes Object Oriented Techniques. Hierarchical in nature.	Utilizes a lexicographical schema order. Flat structure/Table in nature.
Modeling Technique	Greater independence from application usage (relatively generic)	Requires prejudgement of application usage (relatively specialized)

Table 3-1 Comparison of Information Models

ISO/CCITT adopts a model that is more aligned with the entity-relation and object-oriented data models. Internet standards adopt a model that is more aligned with the Relational data model. The following differences are especially relevant to flexibility and extensibility:

- Using ISO/CCITT inheritance, an existing definition may be expanded to encompass additional functionality provided by a new resource. A properly designed management application can manage the new resource as if it had the original class. Inheritance can be used to successively refine existing classes, adding new properties. However, it does not provide a mechanism for removing or deleting properties which are no longer deemed desirable. Internet management does not include inheritance, and requires that common properties be redefined as new object types in each MIB.

- ISO / CCITT allomorphism can be used to manage an object as though it were an instance of another compatible class (which may or may not be a superclass). This allows for versioning and vendor extension. Internet management does not include allomorphism, but allows for vendor extension by defining new object types, and vendors may support more than one object type for the same resource.
- ISO / CCITT object class definitions can be designed with conditional packages which allow for optionality and inclusion / exclusion based on an explicit condition. The SNMP MIB-II makes all object types within each supported object group mandatory, while GDMO MIBs tend to include optional properties within each supported object class. In both cases, vendors implement only those features which are relevant to their products and supported by the underlying resources. GDMO MIBs assign each conditional package an object identifier (OID) which can be used in protocol to control or monitor the packages present in a given instance of the class. No corresponding explicit mechanism is provided for Internet MIBs.
- Both CMIP and SNMP use the Abstract Syntax Notation One (ASN.1) to represent management information data types. SNMP supports a limited subset in order to reduce implementation requirements (see Table 3-2.) There are ways to represent the unsupported ASN.1 constructs in SNMP with the valid subset, but this adds significant complexity to applications of managers and agents.

ASN.1 Type	**CMIP**	**SNMP**
Boolean	Yes	No
Integer	Yes	Yes
Bit String	Yes	No
Octet String	Yes	Yes
Null	Yes	Yes
Object Identifier	Yes	Yes
Object Descriptor	Yes	No
External	Yes	No
Real	Yes	No
Enumerated	Yes	No
Sequence, Sequence of	Yes	No
Set, Set of	Yes	No
Choice	Yes	No
UTC, Generalized Time	Yes	No
ANY / ANY DEFINED BY	Yes	No
Numeric, Printable, Teletex, Videotex, IA5String, Graphic, Visible, General, and Character Strings	Yes	No
Tagged Types	Yes	Yes
SubTyping	Yes	Yes
Initial and Default	Yes	Only one Default value
Optional	Yes	Static support only

Table 3-2 Comparison of Data Types Supported By Protocols

3.3.2 Protocol Considerations

The following differences are especially relevant to flexibility and extensibility:

- SNMP does not provide for adding new management operations but rather for implicit operation initiated by performing a Set operation on an attribute value (e.g., the setting of a re-boot variable might imply the subsequent execution of the re-boot operation). CMIP Sets are used this way also, but CMIP Actions allow definition and execution of new management operations that are specific to a particular resource.
- Beyond the simple replacement operation, CMIP allows Set to also have the semantics of adding or removing values from an attribute, or Setting the attribute to a default value. The ability to add or delete values is especially useful for attributes which are of the data type *list*. In SNMP, there is no mechanism to get the same effect. The data type *list* is not a supported data element. Each list element is modeled as a separate attribute instance with its own managed object id. The richer semantics of Set found in CMIP will be most useful in exchanges between managers, but may also occur in some of the more complex managed resources.
- CMIP provides more flexible and extensible naming mechanisms than SNMP; Table 3-3 summarizes the differences between the two standards.

Feature	CMIP	SNMP
Naming Scheme	Global naming across all agents, Local naming also supported	Local naming within a particular agent
Naming Format	Different syntaxes including National Language Support	Restricted to ASN.1 object identifiers
Class/Type Naming	Object Identifier	Object Identifier
Instance/Variable Naming	X.500 style Distinguished Name	Object Identifier or none
Name Length	Relatively long names	Relatively short names

Table 3-3 Comparison of Naming

- The individual components that make up SNMP are tightly coupled and this has an effect on the extensibility of the protocol. For example, in order to add security to SNMP it is necessary to redefine the protocol. ISO/CCITT has a layered design which allows addition of new systems management functions without impacting the underlying management communications protocol and service. Similarly, this design allows for substitution of alternative management protocols which offer the same management services. However, layered design can lead to inefficient implementation -- it is easier to optimize a closely coupled design.
- ISO/CCITT standards include parameters which allow applications to negotiate the context and services to be used over a given connection. These mechanisms are designed to allow new functions to be requested or negotiated away without change in protocol, thus avoiding protocol transition/migration problems for end users. The negotiation mechanisms also allow dynamic discovery of system capabilities, reducing the need for manual configuration of system capabilities.
- In the ISO/CCITT management model, communication between managers can be accomplished by one system briefly taking on the agent role for a given protocol exchange. In the Internet model, there is no widely accepted mechanism for integrating multiple SNMP-

based agents on a single system, or for communicating between management stations.

3.4 Security

Application developers are often concerned about the security of the management services upon which they depend. The following discussion summarizes existing mechanisms which are provided by ISO/CCITT and Internet management standards to assure secure management services.

3.4.1 Management Protocol Security

Current SNMP security is handled via the *community name* mechanism. This mechanism provides a measure of protection that is at least consistent with the security found in most systems in the Internet environment. However, since community name is not very robust, control operations are generally performed via Telnet instead of SNMP. Some specific Internet MIBs provide additional access control via the community name. The community name is considered a portion of the attribute name, making it possible for an agent to support different sets of attributes for different community names. In general, users need access control on Read, Modify, Append, Create, and Delete level of the objects. For example users would like to provide some operators and applications append or modify access but not create or delete access. The current SNMP functionality makes that difficult. Secure SNMP, currently a proposed Internet Standard, addresses many of the security shortcomings of SNMP, providing both authentication and access control mechanisms.

CMIP relies on the OSI Association Control Service Element (ACSE) to provide connection oriented service. Since CMIP Manager-to-Agent communication occurs only over an association, it is relatively simple to implement a predefined list of agents which can start associations with a given manager, thus ensuring secure management operations. ISO/CCITT Management provides for authentication of the entities requesting the establishment of a management association based on simple credentials such as user name, passwords, and optional time-stamp and random number of fields. The Association Control Service Element (ACSE) standard defines protocol for peer-entity authentication at association establishment time. Authentication information is compared with a list of user names and passwords to authenticate the peer entity. Distribution and maintenance of passwords and user names is based on prior agreement between the communicating entities. The authentication information can be encrypted using an encryption mechanism.

The identity of the peer management entity can be used to authorize user permission to perform operations on objects and prevents unauthorized access to objects. CMIP also provides an access control parameter for each management operation. This parameter can be used to supplement the peer-entity information exchanged during association establishment; a standard method is defined by the OMNI*Point* 1 Security of Management specification (see reference **SM**). In addition, the ISO/CCITT Objects and Attributes for Access Control (see reference **OAAC**) draft standard specifies procedures to control access to management information (security of management) and procedures to manage access control type of information (management of security).

3.4.2 Underlying Security Services

Currently, the OSI Transport Layer Security Protocol (TLSP) specifies an extension of OSI transport layer protocol to support data confidentiality and data integrity using cryptographic techniques. OSI TLSP implementations are just starting. In the future, the OSI Network Layer Security Protocol (NLSP) is intended to provide an end-to-end secure connectionless and connection-oriented OSI network service including data origin authentication, access control, traffic flaw confidentiality, and data integrity protection.

Currently, SNMP uses the unreliable datagram (UDP) protocol to provide connectionless communications; no standard UDP security mechanisms have been defined or planned. This has led to concerns about the security of management information, and many Internet MIBs in the public domain do not allow the use of Set. Proprietary solutions are sometimes implemented to address the concerns about security when operating over UDP.

3.5 Application Functionality

Both ISO/CCITT and Internet management standards provide functionality which can be used as the basis for management application development. The following sections identify the built-in services provided by each standard for given areas of application functionality.

The different approaches taken by the two standards impact the way in which functionality is provided to applications. ISO/CCITT systems management functions provide generic services common to a number of applications. In the Internet environment, common functions are not defined in this manner. Instead, attributes necessary for these functions are defined on an as-needed basis in specific technology MIBs, without an overall consistent approach. The net result is that ISO/CCITT standards-based products offer more built-in tools to application developers, while Internet standards-based products keep agents simple by offering less to the application developer.

3.5.1 Configuration Management

Both Internet and ISO/CCITT style object definitions describe configuration information which can be manipulated by management protocols. Both standard protocols can be used to find and browse new managed device MIBs. CMIP does this via its multiple object selection capabilities, or through the use of OMNI*Point* 1 Shared Management Knowledge managed objects (see reference **SMK**). SNMP does this by using the Get and GetNext commands, or through the use of Internet routing and ping protocols. However, retrieving large MIBs with SNMP can be a long and tedious process for the management application, which must navigate the MIB in an orderly fashion, retrieving one object at a time (thus the SNMP-2 GetBulk proposal). CMIP allows a management application to perform multiple object selection with a wildcard and so retrieve the configuration more quickly.

ISO/CCITT defines CMIP messages for Create, Delete, and Action, plus several additional functions which identify common configuration-related attributes and events that can be included in object definitions (e.g., object creation/deletion events, generic state attributes). These ISO/CCITT functions allow generic configuration applications to be developed. Many Internet MIBs include this type of configuration information, but it is represented in a technology-specific manner (e.g., using Set or an Enterprise Trap).

There are many cases where a management application would like to activate/start/turn-on an object, or deactivate/stop/turn-off a protocol entity, test, etc. It is often the case that it is useful to create an object, but not activate it. Similarly, it is often useful to deactivate an object without deleting it. Both CMIP and SNMP can be used to provide this capability by setting attributes of an object. In addition, ISO/CCITT defines a generic state management function that specifies a common (technology-independent) representation for these attributes. Finally, the ISO/CCITT Generic Management Information MIB defines activate and deactivate actions which can be used for this purpose described above.

3.5.2 Performance and Accounting Management

Numerous public domain programs are available that measure network response time for TCP/IP networks; these programs can be used by Internet-based managers. Several technology-specific Internet MIBs include information relevant to performance and accounting. The representation of this information tends to be tailored to the technology being monitored. For example, a MIB often includes a variety of usage counters, and might include tables which store historical or cumulative information. External performance monitoring devices are also commonly used in the Internet management environment.

ISO/CCITT objects also include this sort of information relevant to performance and accounting management, but often generic counter, threshold, gauge, or tidemark attributes are used to provide a consistent representation across technologies. In addition, several specialized functions are included in ISO/CCITT which offer more complex capabilities, such as workload monitoring, summarization, and account metering. Management applications might use the managed objects defined by these ISO/CCITT functions to offer performance and/or accounting management. It is expected that future versions of standard managed objects (such as the OSI Network Layer MIB) will be extended to take account of these functions.

3.5.3 Problem Management

Most Problem Management applications involve monitoring of problem-related events. As mentioned previously, Internet management relies on manager polling to monitor IP-addressable components of a network. SNMP-based agent products must be extended if they are required to provide information on problem symptoms and probable causes and recommendations for problem resolution. Non-IP-addressable devices are often integrated in the Internet management environment through the use of external performance monitors.

ISO/CCITT management defines a generic alarm reporting function which specifies a common representation for problem-related events, including codes for problem symptoms and probable causes common to many technologies. These generic alarm reports are designed for inclusion in ISO/CCITT objects, and may be extended to include additional text or diagnostic information unique to a given product. This facilitates development of generic alarm monitoring applications; these can be either passive (which many early implementations are) or active, depending on the policy of the implementor (and possibly the user).

Neither ISO/CCITT or Internet standards define a method to track problems and build knowledge database from user experiences, although ANSI and the NM Forum have defined an ISO/CCITT-based Trouble Management function which provides a common format for trouble tickets exchanged between management systems. Management systems should allow operators to archive and retrieve historical information so problems can be tracked and symptoms can be correlated to the respective recovery procedures.

3.5.4 Security Management

Some management applications are responsible for managing the security of the managed network or system resources. ISO/CCITT and Internet management can be used to provide this functionality. ISO/CCITT defines a security alarm reporting and audit trail functions which can be used to monitor and record events of interest to the security management application. Similarly, security-related traps can be used with Internet management - community name violation is a standard trap; other enterprise traps are sometimes defined for specific products.

3.6 Cost Considerations

There are several cost factors to be considered when deciding to make use of standards-based management tools, including development, deployment, and operational costs. These are summarized in the following subsections.

3.6.1 Cost of Development

A major advantage of Internet management is the low cost of initial development. Because SNMP is relatively simple, agents are cheap to implement. However, to provide the same level of end-user functionality, complex implementation must occur somewhere - with SNMP, any complexity falls to the Management Station and to the application developer. Furthermore, each new SNMP MIB requires unique or altered application code to support. The first application may therefore be cheap, but each additional MIB is added cost. The comparatively small number of built-in SNMP functions also increases application development cost, especially for large applications which might otherwise use off-the-shelf or pre-existing ISO/CCITT functions.

ISO/CCITT management requires considerably more complex software to be developed on the agent side, significantly increasing initial development costs. The first management application is also relatively expensive, but the object-oriented approach allows objects to be added easily at a small incremental cost. ISO/CCITT functions also allow common tasks to be developed once, and then reused by a number of applications, thus reducing long-term development costs. However, for a typical systems builder/supplier, using off-the-shelf components (such as a portable protocol stack implementation or MIB compiler), the cost can be less.

3.6.2 Cost of Deployment

Currently, a large selection of Internet standards-based products are available, while comparatively few ISO/CCITT standards-based products can be purchased today. Similarly, several public domain SNMP implementations exist today, while only one public domain CMIP implementation is available. Product availability is directly related to the cost of deployment. According to the April 1992 issue of LAN Technology (see reference **DM**), SNMP source code is available for about $50k, while CMIP source code costs about $200k (sometimes discounted). The same article notes that SNMP implementations typically require about 50kb of memory, compared to 500kb for CMIP implementations. Thus, the article concludes:

> the cost of deployment is probably the single most important barrier to wider use of CMIP in managed devices

noting that

> several vendors are actively researching ways to reduce the cost of deploying CMIP - such as putting the entire CMIP and OSI stack on a chip

Readers are cautioned that the management market is extremely volatile, and the current cost of deployment is highly likely to change. SMP and SNMP version 2 will introduce a period of instability in the Internet management market, as users decide when and if to upgrade existing systems or build/buy new systems. The completion of OMNI*Point* 1 may spur release of CMIP-based products currently under development by several major vendors who have been waiting for market demand and stabilization of ISO/CCITT standards. New multi-vendor management platform technologies like OSF DME, OMG CORBA, and UI Atlas are also expected to have a significant impact on product availability and the management market in general. The recent IEEE standardization of CMIP over LLC (see reference **LAN/MAN**) is also likely to have an effect on the cost of deployment and availability of products in the LAN environment.

3.6.3 Cost of Operation

Performance costs and trade-offs were discussed previously in Section 3.1 on page 20. See also reference **IPSNM** for additional performance data. No public data is currently available regarding the total cost of operation for either Internet or ISO/CCITT-based management products. This section will be expanded in the future when additional information becomes available.

3.7 Technology and Application Domains

Since CMIP and SNMP cannot be used to manage without object definitions that represent the resources to be managed, the availability of object definitions gives some indication of the environments in which CMIP and SNMP are currently used. Currently, ISO/CCITT and Internet-based object definitions are available or underway for a large number of resource technologies, as shown in the Table 3-4. All of the MIBs listed are in the public domain, even when modeling proprietary resources.

ISO/CCITT GDMO-based MIBs		**Internet SMI-based MIBs**
10165-2 | X.721 - Definition of Management Information		MIB-II
M.3100 - Generic Network Model		Token Bus MIB
802.3H - Hub Management		Token Ring MIB
ANSI FDDI		DS1 MIB
Forum R1 Library		DS3 MIB
CNMA Library		Appletalk MIB
IDRP		OSPF MIB
ES-IS, IS-IS Routing		BGP v3 MIB
G.784 - SDH		Remote Network Monitoring MIB
T1.214		Ether-like MIB
T1.215		FDDI MI
IETF OIM MIB-II		Bridge MIB
ETSI Traffic Model		DECnet Phase IV MIB
ETSI Transmission Equipment		SMDS Interface Protocol MIB
10165-5 - Generic Management Information		RS-232-like MIB
10733 - OSI Transport Layer		Printer MIB
10737 - OSI Network Layer		Character Stream MIB
OSI Data Link, Physical Layer		X.25 MIB
OSF DCE Objects		Frame Relay MIB
OSF DME Objects		IS-IS MIB
OSF/1 Objects		Chassis MIB
CCITT SG XV SONET Objects		OSI IP MIB
P1003.7 System Administration Objects		Chassis MIB
Forum OMNI*Point* Libraries		PPP MIB
OIW OMNI*Point* Libraries		IDRP MIB
IEEE 802.x Objects		Ethernet Repeater MIB
Q.751 SS7 Objects		IP Forwarding MIB
Q.94x ISDN Objects		Token Ring Repeater MIB
T1X1.5 SONET Objects		

Table 3-4 Technology Domains

The MIBs shown in Table 3-4 are at various stages of definition and implementation, but some conclusions can be drawn from the technology coverage. Internet Management is used extensively in local area networks and their interconnection, to manage low-level protocol interfaces and devices. ISO/CCITT Management is more popular in the telecommunications industry, as evidenced by the large number of CCITT/ANSI/ETSI-defined managed objects.

The simplicity of Internet Management makes it desirable for small systems or networks which neither need nor can afford the cost of ISO/CCITT Management. The additional complexity of ISO/CCITT Management is suited for managing large, complex networks and systems which can take advantage of additional features and distribution of workload. Current practice does not reflect this, however. SNMP was designed, and is currently used, to manage the Internet - a very large and complex network indeed. Initial CMIP deployment has included not only large

telco environments, but also very small networks and LANs. Obviously, both sets of standards can be applied to any number of environments in a variety of ways. For example:

- Applications which operate in a polling mode, and do not require real-time notification of faults, are efficiently implemented using SNMP.
- Applications which perform complex performance management services can probably take advantage of ISO/CCITT functions like workload monitoring, scheduling, and test management.
- Applications which display network and system topology can be effectively implemented using either Internet or ISO/CCITT management, depending upon the protocol and MIB supported by agents in the distributed management environment.
- From a purely pragmatic viewpoint, SNMP-based agents are best managed by using Internet standards, while CMIP-based agents are best managed using ISO/CCITT standards. While this may seem obvious, it places the burden of coexistence and interworking on the application and end-user. Chapter 4 describes a number of strategies which are intended to reduce this burden.

3.8 Comparison Summary and Criteria

Which combination of standards is best suited to support a given application is a complex question with no simple answer. However, the comparison provided by this section is intended to help developers determine the right combination of tools to solve the problem at hand. Trade-offs are required; the following criteria must be balanced against one another. The weighting that should be applied to individual factors depends upon the situation - there is no single prioritized list which applies to every case.

Efficiency/Performance:
Select the event paradigm (polling versus event-driven) to be used by the application. If polling is to be used exclusively, current experience suggests that Internet Management may be more efficient. If the application is to be event-driven, then ISO/CCITT Management offers more functionality and control for this type of application. Weigh performance factors against intended usage.

Robustness:
Determine if reliable delivery is required by the application. If so, then connection- oriented management may be required; otherwise, connectionless management is probably sufficient. (Note that if reliability is mission-critical, out-of-band management may be required.) Synchronization and atomicity requirements should also be considered.

Flexibility/Extensibility:
Determine flexibility/extensibility requirements for the application. If the application is part of a management platform which must be capable of handling many new resources or functions, or is part of an agent system in a dynamic environment where new resources or new/changed management requirements are likely, then ISO/CCITT Management may pay off. Otherwise, Internet Management may be cheaper.

Security:
Identify the level of security required for management in the target environment, and compare this to the features provided by ISO/CCITT Management and Internet Management respectively.

Application Functionality:
Locate the area most closely associated with the application to be developed, and scan the features which are provided by ISO/CCITT and Internet Management. Determine whether any built-in features are available to simplify application development, and weigh these against the increased agent cost.

Cost Considerations:
Assess both the short and long term cost of development, deployment, and operation against the previous conclusions, to arrive at the right balance of cost versus function.

Technology/Application Domains:
Determine whether any standard MIBs exist for the technologies to be managed; in the short term, this alone may dictate a choice of ISO/CCITT or Internet Management, at least when interfacing to a given agent.

These criteria must be combined with other business-oriented criteria, such as:

- programmer experience
- products already in use
- installed base
- procurement requirements

- regulatory environment
- time to market

and additional criteria may apply for specific business cases.

In many cases, analysis of the pertinent factors will result in use of a combination of protocols and MIBs. For example, the April, 1992 issue of LAN Technology (see reference **DM**) describes a scenario where an application developer plans to build an application which monitors LAN-to-LAN performance over a wide area network. Based on the resources involved, the interconnections between them, and the functions which are required, the example chooses a combination of SNMP, CMIP, and RPC protocols, supporting a variety of Internet and ISO/CCITT style MIBs.

The following scenario is intended to illustrate from a highly simplified but practical perspective why ISO/CCITT and Internet coexistence and interworking are needed, and how this can be achieved. A fundamental premise is that integrated network management across the Internet and ISO/CCITT technology domains can deliver significant business benefit. It is also important to recognize that these benefits are only really accrued when management applications are developed to span these technology domains. Systems that support multiple protocols but do so by offering separate applications are limited in their ability to provide integrated end-to-end management.

A highly simplified example of a multi-protocol Internet is shown in Figure 3-1. Assume this network is used by an organization with four extended LAN sites, interconnected via public/carrier services such as T1 point to point managed bandwidth services (private circuits) and X.25 services.

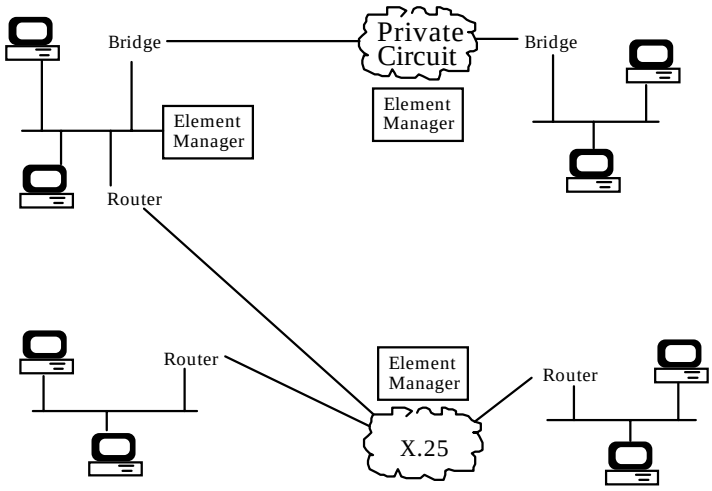


Figure 3-1 Scenario Without Integrated Management

Three types of element management systems might be at work in this example:

- carrier X.25 management systems which manage elements using CMIP
- carrier private circuit management systems which manage elements using CMIP
- LAN segment management systems which manage devices using SNMP.

A variety of possible integration scenarios could provide end-to-end monitoring and control of this sample network from a single point. The advantages of integrated management in this scenario include:

- cost savings due to rationalization/centralization of the network management force
- improved quality of service and faster resolution facilitated by an end-to-end view of the network
- cost saving due to better optimization of resources again facilitated by having an end-to-end view of the network.

One integrated management configuration is shown in Figure 3-2. In this scenario, capabilities at both the element management systems and the integrating management system are required to enable interworking.

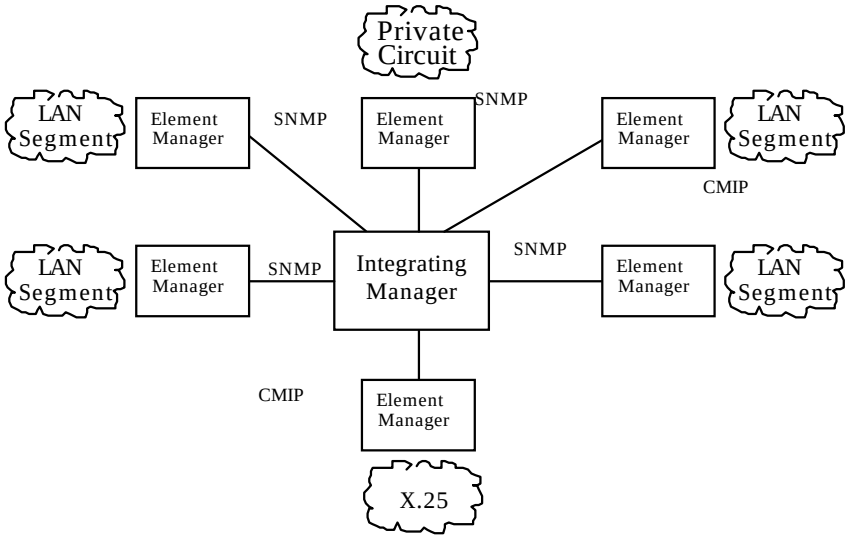


Figure 3-2 Scenario With Integrated Management

In this type of multi protocol management environment, there are many alternative methodologies and configurations possible to enable coexistence and interworking. This type of scenario is fairly typical, and illustrates the motivation for Chapter 4, which provides a summary overview of the practical options for achieving coexistence and interworking.

Chapter 4

Coexistence and Interworking Strategies

The problem of coexistence and interworking must be approached from various perspectives because network and systems management is not just an isolated function, protocol, communication layer, application, or collection of managed objects. Network and systems management involves all of these components, encompassing many aspects of internetworking, interoperability, and layered-communication management, as well as the management of new entities created to support various management functions. Therefore, there is no unique answer, solution or path to achieve coexistence or interworking.

This chapter describes various strategies for coexistence and interworking of Internet and ISO / CCITT Management. For each strategy, a number of existing or future methodologies are described which enable the strategy to be implemented. This chapter does not provide detailed specification of the methodologies for coexistence or interworking. Instead, it references existing specifications and identifies areas which require further development. Additional detailed design specifications will follow as needed to describe specific methodologies which are proposed but do not currently exist.

4.1 Coexistence Strategies

In this document, coexistence is defined as the ability for Internet and ISO/CCITT management to exist in the same distributed management environment, without interworking as described in Section 4.2 on page 47. That is, both sets of standards-based tools are available for application use, but they are used independently. For example, one set of resources might be managed with one management protocol, another set of resources might be managed with another management protocol. The application or end-user remains aware of this fact and must deal with the consequences.

Coexistence by itself is not very desirable. However, simple coexistence is likely to be the first strategy deployed, since it is much easier to accomplish than interworking. In order to implement the coexistence strategy, several methodologies are defined, including:

- mixed protocol stacks
- dual protocol stacks
- common APIs
- pass-through integration.

These methodologies are explored in the following subsections.

4.1.1 Mixed Protocol Stacks

There are several mixed protocol stack approaches currently defined throughout the industry, including:

- CMIP For The Internet (see reference **CMIPI**)
- CMIP Over LLC (see reference **LAN/MAN**)
- SNMP Over OSI (see reference **SNMPO**).

Each of these existing documents provides a layer-to-layer mapping which allows either Internet or ISO/CCITT management applications to be deployed over alternative transports. protocol stacks are shown in Figure 4-1, Figure 4-2 and Figure 4-3.

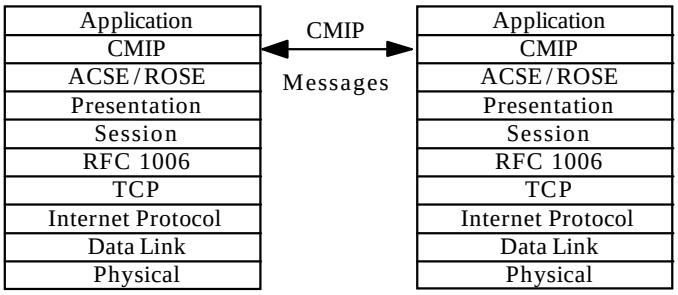


Figure 4-1 CMIP For The Internet

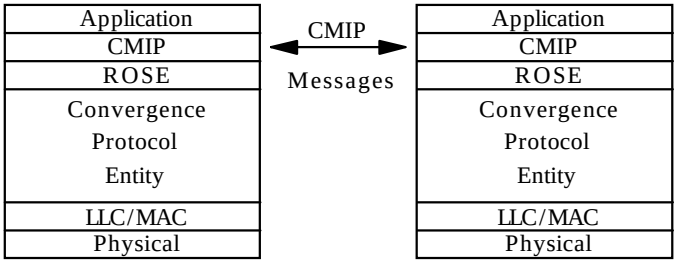


Figure 4-2 CMIP Over LLC

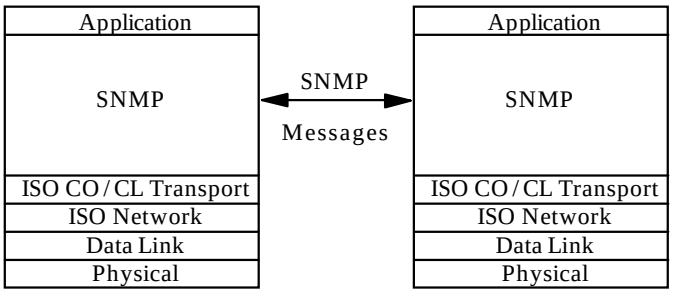


Figure 4-3 SNMP Over OSI

This methodology exists today, and allows management applications to be deployed over existing transports. However, this methodology does nothing to address coexistence of multiple management protocols. It also does not allow MIBs defined with one set of standards to be managed by applications which use another set of standards.

Note that Figure 4-1 on page 42 does not depict the last published CMIP For The Internet RFC (see reference **CMIPI**), but rather an updated proposal which uses RFC 1006 (see reference **ISOTCP**) to support CMIP and OSI upper layer protocols over TCP/IP. The Forum and X/Open intend to jointly publish a recommendation which recognizes this stack as the primary Internet IP-based approach for supporting CMIP. This proposal will refer to current CMIP standards and profiles for implementation agreements, enabling reuse of the same management protocol implementation over a variety of transports.

4.1.2 Dual Protocol Stacks

This approach assumes implementation of both the OSI and TCP/IP protocol stacks side-by-side in the same device to provide interconnection between two management processes. Both stacks share the same data-link and physical layers which will carry both Internet IP packets and OSI network layer packets. This approach is shown in Figure 4-4.

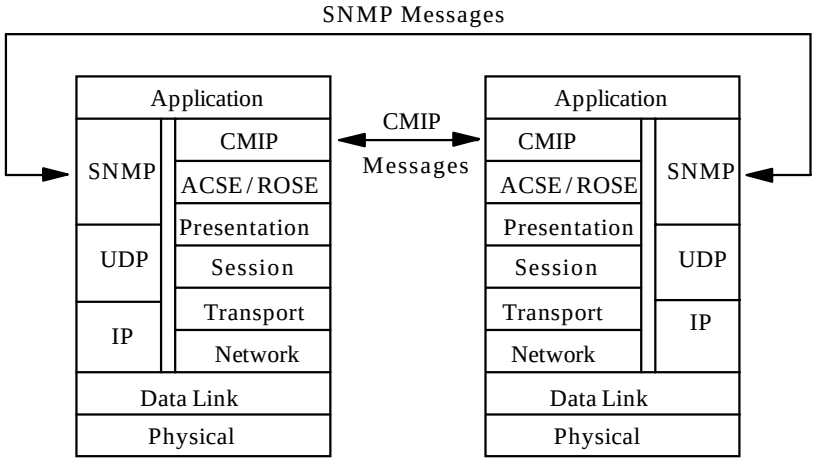


Figure 4-4 Dual Protocol Stacks

The clear disadvantage of this solution is the cost to implement it in simple devices and a possible performance degradation running two parallel stacks. However, the dual protocol stack is perhaps the most direct approach to coexistence, and therefore may be one of the first deployed in products. One strategy might be to have managers implement the dual stack approach, while agents implement only the management protocol which is native to their environment.

4.1.3 Common APIs

An extension to the dual stack methodology is the use of a common application programming interface (API) to assist application developers writing programs for this environment. Common APIs can provide a consistent set of verbs and data structures for application development, replacing distinct vendor-proprietary interfaces to each protocol stack. The X/Open Management Protocol API (XMP) (see reference **XMP**) exists currently and provides a common API to SNMP and CMIP services, shown in Figure 4-5.

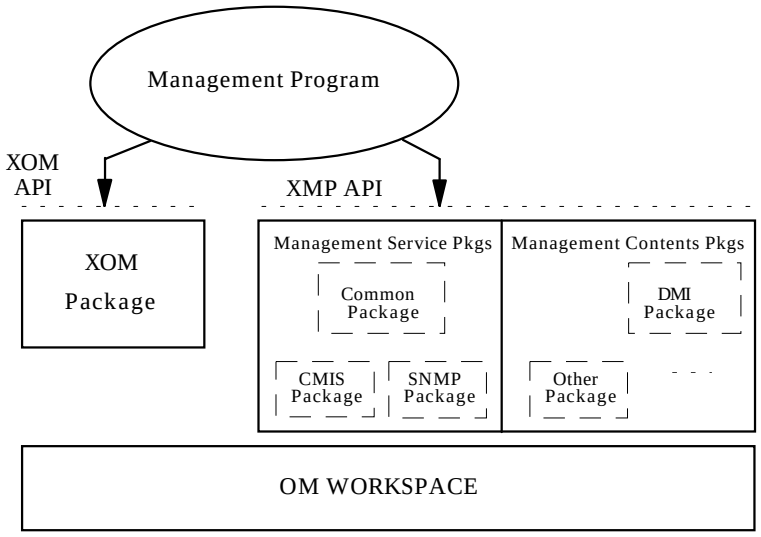


Figure 4-5 X/Open XMP API

XMP defines a set of C programming language functions that an application can use to invoke or respond to CMIS and/or SNMP service requests. The same function calls are used for CMIS and SNMP services that are common to both protocols (for example, Get-Request, Set-Request, and Event-Request). The parameters of each function call are based on another X/Open standard for representing data within API specifications: the X/Open OSI-Abstract-Data Manipulation (XOM) API (see reference **XOM**). XMP defines a common package containing parameters (called OM objects) which are common to both CMIS and SNMP. Other unique packages are defined for CMIS and SNMP-specific parameters, and management contents packages are defined for each MIB used with the XMP API. The XMP API approach allows an application to use the same set of function calls and some of the same data types, regardless of underlying protocol.

However, common APIs at this low level do not currently provide for protocol transparency or true service integration. Application developers must still be aware of the information model being used and the corresponding differences in services, naming, and error handling. This problem can be minimized by limiting use of those features which are unique to a given management standard. For example, an application which uses only simple Gets and Sets is less impacted by underlying protocols than an application which uses CMIP scoping and filtering or SNMP GetNext. This technique has the unfortunate effect of reducing application functionality to the lowest common denominator.

Increasingly, abstract APIs which provide greater application independence from the underlying protocols are a subject for future study. For example, the OMG CORBA (see reference **CORBA**) is being considered as the basis for distributed management environments like the OSF DME. CORBA specifies an object request broker which allows clients to send requests to servers modelled as objects. The interface for each object is specified using a notation called IDL (Interface Definition Language). In this way, CORBA IDL can be used to specify and generate APIs for each object. Since the object may provide a simplified interface to a complex set of underlying procedures, this approach might be used to define abstract APIs for specific services. Current abstract API design has yet to resolve trade-offs between protocol-independence and functionality, and many doubt that strict protocol transparency is achievable.

4.1.4 Pass-Through Integration

This approach involves limited integration of independent management subnetworks, where each subnetwork has its own management system. Integration is provided at the user-interface level by an application which provides a pass-through view of a subnetwork managed by another technology. Figure 4-6 shows an example of this methodology extracted from the paper ''Integration of OSI-Based and SNMP-Based Network Management Systems: An Example'' (see reference **SNMPI**).

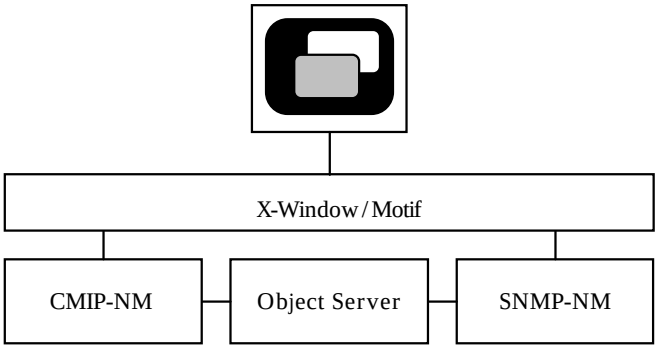


Figure 4-6 Pass-Through Integration

In this example, two autonomous management systems exist, one based on CMIP and the other based on SNMP. The SNMP-based management system (Transview-SNMP) makes events available to the CMIP-based management system (Transview-NMC) through an object-oriented interface. The interface is composed of a single GDMO-style managed object class which represents the entire network managed by SNMP. The primary purpose of this object class is to provide configuration and alarm event notification to the Transview-NMC. The events are made visible the end-user by Transview-NMC in a separate X-Window / Motif application window. That is, two windows are visible to the end-user, one window showing the CMIP-based managed network and another window showing the SNMP-based managed network.

Pass-through integration is a popular methodology for near-term integration of existing management systems which are based on different management protocols. Advantages of this approach include ease of implementation, ability to integrate both standard and proprietary protocols, and minimal management application impact. The primary disadvantage to this approach is that the end user remains aware of multiple management systems, and must adapt to the *look and feel* of each system. In addition, this methodology does not facilitate coordinated control of multiple subnetworks; for example, reconfiguration must be accomplished using the distinct tool set of each subnetwork management system.

4.2 Interworking

In this document, interworking is defined as the ability for Internet and ISO/CCITT management to exist in the same distributed management environment, with some level of integration between them. That is, both sets of standards-based tools are available for application use, without the application or end-user being required to deal with each individually. The objective of interworking is to provide a single, uniform end-to-end view of the managed network, irrespective of the underlying protocol(s).

In order to implement the interworking strategy, several methodologies are defined, including:

- protocol translation
- MIB translation
- service emulation.

These methodologies are not mutually exclusive, and are often used on conjunction with each other. The entity which performs emulation and/or translation is commonly referred to as a gateway or or proxy. For simplicity, this document will use the term proxy throughout, with no assumption made about the type(s) of emulation or translation performed by proxy. These interworking methodologies are also used in conjunction with coexistence techniques described previously in Section 4.1 on page 42 - for example, a proxy might use both dual protocol stacks and a common API.

4.2.1 Protocol Translation

Protocol translation involves simple conversion of SNMP protocol syntax into CMIP protocol syntax and vice versa, without any semantic interpretation. Figure 4-7 depicts a proxy translating between CMIP and SNMP protocols.

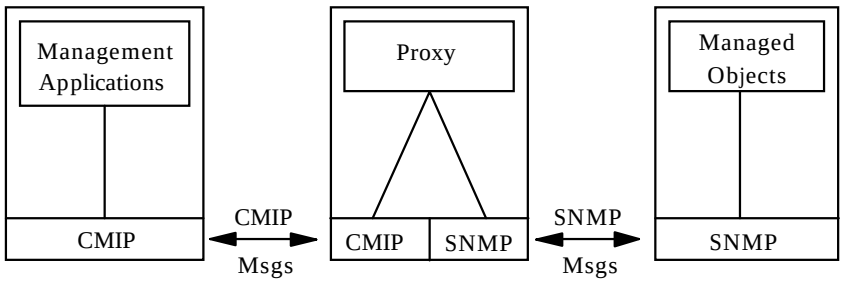


Figure 4-7 Protocol Translation

Direct protocol syntax translation is straight-forward for only a very small subset of the protocol (for example, simple Gets and Sets on individual variables/attributes). There are many cases in which messages cannot be directly translated (for example, CMIP Scoped Gets, SNMP GetNexts). In addition, even when protocol syntax translation is possible, the content of the message (the management information) and the semantics of the service request have not been addressed. Note that unless the naming mechanisms are the same, protocol translators also require some aspect of MIB translation.

For these reasons, protocol translation by itself is not a recommended interworking methodology. However, there are cases in which protocol translation plays an important part in an overall interworking solution. For example:

- Protocol translators are often used to convert between revisions of the same basic protocol. For example, a protocol translator proxy is proposed for SNMP and SNMP-2 interworking.
- Protocol translators are often used as a component of special-purpose tools which require only syntax mapping. For example, many alarm integrator products exist which forward foreign alarms in a standard format, sometimes passing along extra untranslated information which does not map directly but may be meaningful to the end-user.
- Protocol syntax translation may be considered one step in a complex, multi-step interworking effort also involving MIB translation and/or service emulation (refer to Section 4.2.2 and Section 4.2.3 on page 50). For example, the paper ''Simply Open Network Management'' (see reference **SONM**) describes an OSI-SNMP Gateway which includes a stateless mapping of CMIP M-Get to SNMP Get for the Internet MIB-II variable *tcpActiveOpens*. The draft ISO/Internet Management Proxy specification (see reference **IIMPROXY**) defines protocol translation between CMIP, SNMP, and Secure SNMP. In these examples, protocol translation is supplemented by both MIB translation and Service emulation to achieve the desired result.

4.2.2 MIB Translation

Translation between the ISO/CCITT and Internet MIBs is a key ingredient of interworking. This methodology involves mapping a MIB defined using ISO/CCITT MIM and GDMO into another MIB defined using Internet SMI and Concise MIB format, or vice versa. A translated MIB is intended to represent the same set of managed resources as the original MIB.

For example, the Internet MIB-II has been translated from Internet Concise MIB format into ISO GDMO format. A draft translation algorithm (see reference **IIIMIBTR**), translated OIM MIB-II (see reference **IIIMIB-IITR**), and translated SNMP Party MIB (see reference **IIIPARTYTR**) are now under development. Figure 4-8 shows a proxy providing both protocol and MIB translation for the Internet MIB-II.

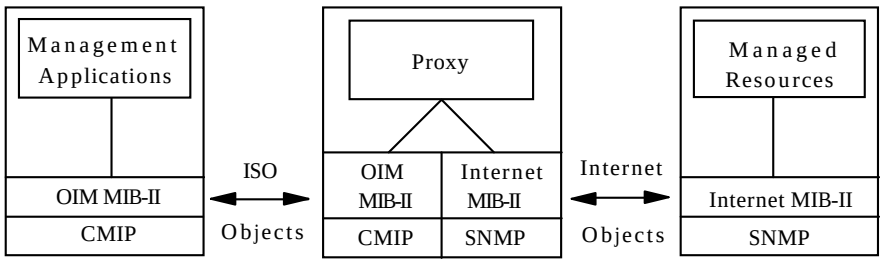


Figure 4-8 MIB Translation

The OIM MIB-II represents a one-to-one or direct MIB translation. A direct MIB translation attempts to represent nearly every aspect of the original MIB in the translated MIB, perhaps with some allowance made for differences in style. For example, object groups become object classes, object types become attributes, traps become notifications - these elements map one-to-one. However, since the Internet and ISO/CCITT naming models differ, new Id attributes must be added to each OIM MIB-II object class. Since the ISO/CCITT model is based on inheritance, and all GDMO object classes inherit from a class called Top, these Top attributes must also added to

each OIM MIB-II object class. These Id and Top attributes have no direct mapping back into the original Internet MIB-II, but must exist in the OIM MIB-II version for compatibility with the ISO/CCITT management model.

Another alternative is many-to-one or abstract MIB translation. An abstract MIB translation is a specialized mapping of objects which does not attempt to preserve every aspect of the original model in the translated model. Abstract MIB translation may be performed for a variety of reasons:

- The level of detail presented in the original model may be reduced in the translated model. This is typically used in a multi-tiered management system where the translated model is used by a higher level manager.
- The style of the translated management model may differ substantially from the original model. This is typically done to facilitate development of common management applications which assume a specific style or generic network model.

For example, the BT Concert product includes an X.700 LAN model which maps the Internet MIB-II into ISO/CCITT GDMO-based object classes which are subclasses of NMF Release 1 objects. This mapping imposes a new containment and relationship structure not present in the Internet MIB-II model, allowing the integrating manager to add value. The paper ''Simply Open Network Management'' (see reference **SONM**) also describes an abstract mapping for the Internet MIB-II, but maps into ISO/CCITT Generic Management Information (see reference **GMI**) standard object classes. OSF has also developed an abstract mapping for Internet MIB-II by translating selected object groups from RFC 1213 into the interface definition language (I4DL) that will be used with OSF DME (see reference **AO**).

Both methods of MIB translation offer advantages over simple protocol translation, because they preserve the semantics of the management information. On the other hand, neither method of MIB translation by itself addresses the translation of services required for true interworking (see Section 4.2.3). Comparing the two methods:

- Direct MIB translation by a proxy requires significant processing power and, in the best case, outputs as much information as it receives. Abstract MIB translation reduces the bandwidth consumption at the proxy. Here, the location of the proxy becomes key - the closer the proxy to the origin, the greater the bandwidth savings (and cost of deployment, unfortunately).
- Direct MIB translation is easier to define/automate than Abstract MIB translation, although it is extremely unlikely that even direct MIB translation can be fully automated. Semantic differences in information model require at least some human intervention, especially in the GDMO to Concise MIB direction; translation algorithms can only provide guidance in these cases.

It should be noted that MIB translation is valuable as an off-line tool as well. Many organizations have invested considerable time and money in information modelling, using either the ISO/CCITT model, the Internet model, or another pre-existing or proprietary model. This investment must be protected in cases where a different protocol will be used. For example, US West translated the ANSI Trouble Administration MIB from GDMO to Concise MIB format because cross-jurisdictional trouble ticket exchange is also required in the data services arena, this market segment is predominately managed by SNMP today, and no such Internet MIB existed (see reference **MEDIACC**). This GDMO MIB was translated into Concise MIB format using the guidance given in RFC 1212 (see reference **Concise MIB**) for *de-OSI-fying* MIBs, supplemented by additional rules required to fully address and preserve the content of the original ANSI MIB. A draft algorithm for GDMO to Concise MIB specification (see reference **IIOMIBTR**) is now under development.

In addition to those mentioned above, many MIB translation activities are underway throughout the industry, and specifications are likely to emerge for this methodology in the near future. MIB translation and common information model efforts are also likely to address other information models, such as the OMG CORBA model, the OSF DME model, and the SNMP-2 SMI model.

4.2.3 Service Emulation

This methodology involves mapping ISO/CCITT management services into Internet management services, or vice versa. This is accomplished by creating a MIB definition in one space for the services provided by a manager in the other space. The proxy acts as both the manager and the agent that represents the manager services. Figure 4-9 shows an example where an ISO/CCITT manager is being used to access objects that exist in an Internet object space through a proxy.

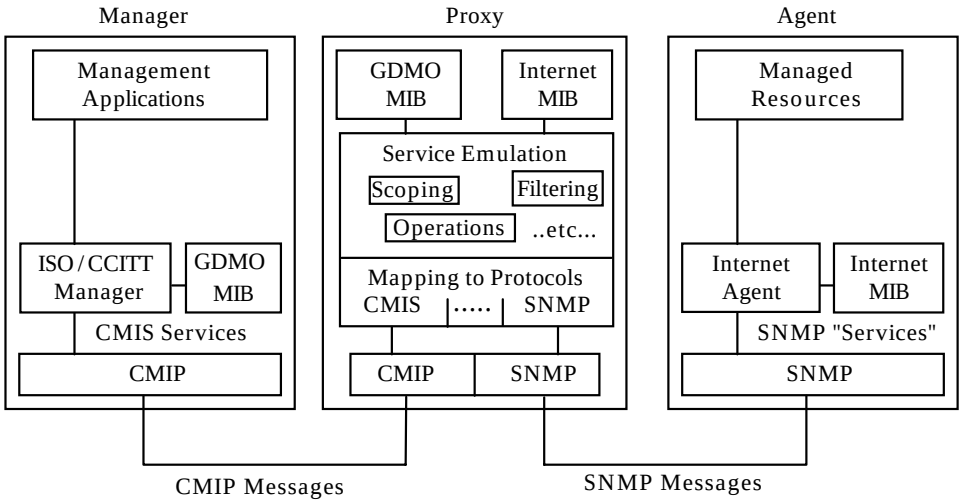


Figure 4-9 Service Emulation

The proxy represents the Internet agent in the ISO/CCITT space. For example, a table could be represented in the ISO/CCITT space as a single complex attribute. A CMIP M-Get request by the manager would result in an SNMP Get, followed by multiple SNMP GetNext requests. Note that the results could be cached by the proxy until the traversal is complete, or returned immediately as CMIP multiple replies. The overall effect of this is to present the ISO/CCITT manager with an object definition that is consistent with the style (syntax and semantics) of other ISO/CCITT objects.

There are a number of variations on this general approach:

- Direct modelling of a specific set of objects from one space into another. This is the general approach taken by the previous examples. The direct approach requires an object class specific proxy, although a given proxy can, of course, provide multiple object class mappings.
- Data-driven modelling of a specific set of objects from one space into another. In this case, mapping rules are configurable and can be set dynamically, but the proxy is still limited to translating specific object classes for which mapping rules have been defined. This implies that unexpected traffic (such as events from an unknown object class) cannot be forwarded through the proxy.
- Generic data-driven modelling for any object from one space into another. In this case, a general-purpose configurable rule base allows translation for any object class. This alternative is desirable from an application or end-user perspective, but is probably not feasible. There may be a small subset of services that can be mapped in this manner. The cost of developing such a proxy would be comparatively high.
- Abstract modelling of a specific set of objects from one space as a set of management services in the other. In this scenario, the actual number and structure of the objects in the one space is not reflected into the other. Rather, a more abstract set of services (operations of the manager side of the proxy) are projected (refer to Section 4.2.2 on page 48 for examples). In this case, some of the details of the structure may come across as attributes of the proxy object (for example, a list of the hosts known by the proxy). This approach is likely to yield the most efficient implementation of proxy communications since there is likely to be a fairly high ratio between the requests on one side and those on the other, yielding less overall traffic through the proxy.

None of these approaches is best for all situations, and any given situation may be handled in more than one way. For example, consider the problem of walking an Internet MIB from the ISO/CCITT space. In this case, direct modelling does not work very well, since the proxy must know in advance about the classes (MIBs) that may be present. Data-driven modelling is more workable, since the ISO/CCITT-based application can download into the proxy mapping(s) for the MIB(s) of interest. Abstract modelling is also appropriate, where the MIB-walking application itself is in the proxy and is modeled as a set of objects in the ISO/CCITT space. The process of downloading new MIBs to the proxy is straight-forward and the details of SNMP GetNext handling are isolated in the proxy. The downside, of course, is that objects cannot be easily browsed across the spaces.

Service emulation is, by nature, uni-directional. Thus, in the previous example, the proxy can process requests from a manager in the ISO/CCITT space, but cannot handle requests from managers in the Internet space. More than one proxy can be implemented in one process, or on one system, to provide bi-directional service emulation.

There are a number of issues which must be dealt with during service emulation:

Maintenance of State:
As described in the paper ''Simply Open Network Management'' (see reference **SONM**), proxies can be designed as either stateful or stateless. A stateless proxy is far less complex, but is constrained in terms of the services which it may provide. A stateful proxy retains an intermediate copy of the translated MIB which it uses to service requests. This approach allows provision of additional services, but requires more memory and introduces transient inconsistencies between the proxy MIB and the actual MIB. The paper recommends a combined approach where both methods are used, and decisions are made dynamically.

Name Mapping:
The proxy must understand the naming and addressing models used in both the source and target space, and provide for mapping between them. For example, ISO/CCITT Object Class and Attribute identifiers must be mapped to the corresponding Internet Object-Type identifiers, and ISO/CCITT Distinguished Names must be mapped to the corresponding (Internet Address, Object Identifier, Suffix) tuple. This mapping can be complex when services or objects do not map directly, and may require state retention as described above.

Service Mapping:
A key component of service emulation is mapping an incoming service request into the series of steps required to carry out and respond to the request, using another protocol and MIB. The following are examples of service mappings that may be performed by a proxy as part of service emulation:

- filtering of SNMP traps to simulate CMIP events
- scheduled polling and conversion of SNMP poll results into simulated CMIP events
- use of SNMP GetNext to simulate CMIP scoping
- caching of intermediate SNMP Get results to simulate CMIP filtering
- appropriate error handling and request rejection where services do not map.

The goal of service emulation is to provide a consistent service *look-and-feel* to the application and end-user. Although some work has been done in this area, much work still remains. A draft ISO/Internet Management Proxy specification (see reference **IIMPROXY**) is now under development, based on the MIB translation rules specified in the Internet to ISO GDMO MIB Translation (see reference **IIIMIBTR**).

Chapter 5

Conclusion

The combination of standards best suited to support a given application is a complex question with no simple answer. However, the comparison provided by Chapter 3 is intended to help developers determine the right combination of tools to solve the problem at hand. Trade-offs are required; criteria must be balanced against one another. The weighting of individual factors depends upon the situation - there is no single prioritized list which applies to every case.

Chapter 4 shows that coexistence is possible today, based on existing methodologies like mixed and dual protocol stacks, common APIs, and pass-through application integration. These methodologies should be viewed as near-term solutions to the immediate challenge of using multiple management solutions in a complementary fashion.

As stated in ‘‘Simply Open Network Management’’ (see reference **SONM**), in order to manage heterogeneous networks in an integrated fashion, it is desirable to have one global network model in which models of the specific networks are integrated. The purpose of interworking is to provide this consistent, integrated end-to-end view of the managed network to applications and end users.

Several interworking methodologies are currently being developed, and will provide long-term solutions which can be implemented in the form of a proxy. Direct protocol translation is already well understood, but is very limited. MIB translations exist today and methods are now being specified and refined. Service emulation appears to be best suited for achieving true integration.

This Guide is the first in a series of specifications intended to address these strategies and define these methodologies for general use.

Glossary

ACSE
Association Control Service Element

AOM
Application Profile OSI Management

AOM1x
Application Profile OSI Management Communications

AOM11
Basic Management Communications Profile

AOM12
Enhanced Management Communications Profile

AOM2xx
Application Profile OSI Management Functions

AOM211
General Management Capabilities Profile

AOM212
State Management and Alarm Reporting Profile

AOM213
Alarm Reporting Profile

AOM221
General Event Report Management Profile

AOM231
General Log Control Profile

API
Application Programming Interface

ASN.1
Abstract Syntax Notation 1

CCITT
International Telegraph and Telephone Consultative Committee

CCR
Commitment, Concurrency, and Recovery

CMIP
Common Management Information Protocol

CMIS
Common Management Information Service

CMOL
CMIP Over LLC

CMOT
CMIP Over TCP / IP

CORBA
Common Object Request Broker Architecture

DIS
Draft International Standard

DME
Distributed Management Environment

DMI
Definition of Management Information

EGP
Exterior Gateway Protocol

FDDI
Fiber Distributed Data Interface

GDMO
Guidelines for the Definition of Managed Objects

IAB
Internet Activities Board

ICMP
Internet Control Message Protocol

IDL
Interface Definition Language

I4DL
OSF DME Interface Definition Language

IETF
Internet Engineering Task Force

IP
Internet Protocol

ISO
International Organization for Standardization

ISP
International Standardized Profiles

LLC
Logical Link Control

LPP
Lightweight Presentation Protocol

MIB
Management Information Base

MIM
Management Information Model

NLSP
Network Layer Security Protocol

NM
Network Management

NMF
Network Management Forum

NSF
National Science Foundation

OID
Object Identifier

OIW
OSE Implementors Workshop

OMG
Object Management Group

OSF
Open Software Foundation

OSI
Open Systems Interconnection

RFC
Request For Comment

RM
Reference Model

ROSE
Remote Operations Service Element

RPC
Remote Procedure Call

SGMP
Simple Gateway Monitoring Protocol

SMF
Systems Management Function

SMFA
Systems Management Functional Area

SMI
Structure of Management Information

SMK
Shared Management Knowledge

SMO
Systems Management Overview

SNMP
Simple Network Management Protocol

SMP
Simple Management Protocol

TLSP
Transport Layer Security Protocol

TCP
Transmission Control Protocol

TP
Transaction Processing

UDP
User Datagram Protocol

UI
Unix International

XMP
X/Open Management Protocols API

XOM
X/Open OSI-Abstract-Data Manipulation API

Index