X/Open Snapshot

Systems Management:
Identification of Management Services (XIMS)

X/Open Company Ltd.

X/Open Snapshot

Systems Management: Identification of Management Services (XIMS)

ISBN: 1 872630 30 8
X/Open Document Number: S190

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 open 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 so protects investment in existing software while enabling additions and enhancements. It also allows users to move between systems with a minimum of retraining.

X/Open defines this CAE in a set of specifications which include an evolving portfolio of application programming interfaces (APIs) which significantly enhance portability of application programs at the source code level, along with definitions of and references to protocols and protocol profiles which significantly enhance the interoperability of applications and systems.

The X/Open CAE is implemented in real products and recognised by a distinctive trade mark — the X/Open brand — that is licensed by X/Open and may be used on products which have demonstrated their conformance.

X/Open Technical Publications

X/Open publishes a wide range of technical literature, the main part of which is focussed on specification development, but which also includes Guides, Snapshots, Technical Studies, Branding/Testing documents, industry surveys, and business titles.

There are two types of X/Open specification:

- *CAE Specifications*

  CAE (Common Applications Environment) specifications are the stable specifications that form the basis for X/Open-branded products. These specifications are intended to be used widely within the industry for product development and procurement purposes.

  Anyone developing products that implement an X/Open CAE specification can enjoy the benefits of a single, widely supported standard. In addition, they can demonstrate compliance with the majority of X/Open CAE specifications once these specifications are referenced in an X/Open component or profile definition and included in the X/Open branding programme.

  CAE specifications are published as soon as they are developed, not published to coincide with the launch of a particular X/Open brand. By making its specifications

available in this way, X/Open makes it possible for conformant products to be developed as soon as is practicable, so enhancing the value of the X/Open brand as a procurement aid to users.

- *Preliminary Specifications*

  These specifications, which often address an emerging area of technology and consequently are not yet supported by multiple sources of stable conformant implementations, are released in a controlled manner for the purpose of validation through implementation of products. A Preliminary specification is not a draft specification. In fact, 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 to the *trial-use* standards issued by formal standards organisations, and product development teams are encouraged to develop products on the basis of them. However, because of the nature of the technology that a Preliminary specification is addressing, it may be untried in multiple independent implementations, and may therefore change before being published as a CAE specification. There is always the intent to progress to a corresponding CAE specification, but the ability to do so depends on consensus among X/Open members. In all cases, any resulting CAE specification is made as upwards-compatible as possible. However, complete upwards-compatibility from the Preliminary to the CAE specification cannot be guaranteed.

In addition, X/Open publishes:

- *Guides*

  These 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 advisory, not normative, and should not be referenced for purposes of specifying or claiming X/Open conformance.

- *Technical Studies*

  X/Open Technical Studies present results of analyses performed by X/Open on subjects of interest in areas relevant to X/Open's Technical Programme. They are intended to communicate the findings to the outside world and, where appropriate, stimulate discussion and actions by other bodies and the industry in general.

- *Snapshots*

  These provide a mechanism for X/Open to disseminate information on its current direction and thinking, in advance of possible development of a Specification, Guide or Technical Study. The intention is to stimulate industry debate and prototyping, and solicit feedback. A Snapshot represents the interim results of an X/Open technical activity. Although at the time of its publication, there may be an intention to progress the activity towards publication of a Specification, Guide or Technical Study, X/Open is a consensus organisation, and makes no commitment regarding future development and further publication. Similarly, a Snapshot does not represent any commitment by X/Open members to develop any specific products.

Versions and Issues of Specifications

As with all *live* documents, CAE Specifications require revision, in this case as the subject technology develops and to align with emerging associated international standards. X/Open makes a distinction between revised specifications which are fully backward compatible and those which are not:

- a new *Version* indicates that this publication includes all the same (unchanged) definitive information from the previous publication of that title, but also includes extensions or additional information. As such, it *replaces* the previous publication.
- a new *Issue* does include changes to the definitive information contained in the previous publication of that title (and may also include extensions or additional information). As such, X/Open maintains *both* the previous and new issue as current publications.

Corrigenda

Most X/Open publications deal with technology at the leading edge of open systems development. Feedback from implementation experience gained from using these publications occasionally uncovers errors or inconsistencies. Significant errors or recommended solutions to reported problems are communicated by means of Corrigenda.

The reader of this document is advised to check periodically if any Corrigenda apply to this publication. This may be done either by email to the X/Open info-server or by checking the Corrigenda list in the latest X/Open Publications Price List.

To request Corrigenda information by email, send a message to info-server@xopen.co.uk with the following in the Subject line:

```
request corrigenda; topic index
```

This will return the index of publications for which Corrigenda exist.

This Document

This document is a Snapshot (see above). It is one of several documents within X/Open's Systems Management programme (XSM).

The XSM programme addresses distributed systems management. The primary requirement is to promote the development of management software that allows an administrator to manage a network of heterogeneous systems as a single logical system.

The XSM programme is concerned with the definition of those interfaces necessary for the portable implementation of distributed management systems. In many respects such systems are no different from other distributed applications, requiring a complete range of distributed services to support them. In addition to these general services, distributed management systems also require some specialised services, providing support for specific management functionality.

It is the purpose of this document to identify those services that are required to support such systems management activities.

This Snapshot is being published for information, and to stimulate comment and provide an opportunity for wider review throughout the industry.

Trademarks

UNIX® is a registered trademark of UNIX System Laboratories Inc. in the U.S.A. and other countries.

Palatino® is a registered trademark of Linotype AG and/or its subsidiaries.

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

Referenced Documents

The following documents are referenced in this guide:

ACSE
: ISO / IEC 8649: Association Control Service Element Service Definition

ACSEP
: ISO / IEC 8650: Association Control Service Element Protocol Specification

AM
: ISO / IEC 10164-4: Alarm Management Function

APS
: ISO / IEC DISP 11183-1: Specification of ACSE, Presentation and Session Protocols for use by CMISE and ROSE.

ASN.1
: ISO / IEC 8824, CCITT X.208: Specification of Abstract Syntax Notation One (ASN.1)

BMC
: ISO / IEC DISP 11183-3: AOM11 - Basic Management Communications

BRM
: ISO / IEC 7498: Open Systems Interconnection - Basic Reference Model

CDTC
: ISO / IEC JTC1 / SC21 N5518 Systems Management - Confidence and Diagnostic Test Classes

CMIP
: ISO / IEC 9596-1:1991 Version 2: Common Management Information Service Protocol

CMISD
: ISO / IEC 9595:1991 Version 2: Common Management Information Service Definition

CMISP
: "Common Management Information Services and Protocols for the Internet (CMOT and CMIP)", U.S. Warrior, L. Besaw, L. LaBarre, B.D. Handspicker, RFC 1189, October 1990

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.

DAF
: ISO / IEC 9594-8, CCITT X.509: The Directory - Authentication Framework

DCMS
ISO / IEC 9594-1, CCITT X.500: The Directory - Overview of Concepts, Models and Services (see also other parts of ISO 9594 / other CCITT recommendations X.5xx)

DES
Data Encryption Standard, Federal Information Processing Standards Publication 46, January 1977

DMI
ISO / IEC 10165-2: Definition of Management Information

DSPC
A Method for Obtaining Digital Signatures and Public Key Cryptosystems, R.L. Rivest, A.Shamir and L. Adleman, Communications of the ACM, February 1978 Vol. 21 nr. 2

ECAM
OSI / Network Management Forum, Application Services: Event, Configuration and Alarm Management, Forum 002, Issue 1.3, August 1991

EMC
ISO / IEC DISP 11183-2: AOM12 - Enhanced Management Communications

ERM
ISO / IEC 10164-5: Event Report Management Function

GDMO
ISO / IEC 10165-4: Guidelines for the Definition of Managed Objects

GDMO Algorithm
Algorithm for Translating ISO GDMO Templates to X / Open XOM Packages, L. Phifer and S. Warren, Bellcore, November 1991

Guide to Translating GDMO to XOM
Preliminary Specification, The X / Open Company Limited, in draft.

ITSEC
Information Technology Security Evaluation Criteria (ITSEC), Provisional Harmonised Criteria, version 1.2, Office for Official Publications of the European Communities, June 1991

KERB
Kerberos: An Authentication Service for Open Network Systems, J.G. Steiner, C. Neuman, J.I. Schiller, MIT Project Athena, March 1988

LC
ISO / IEC 10164-6: Log Control Function

MF
ISO / IEC 7498-4: OSI Basic Reference Model - Part 4: Management Framework

MIM
ISO / IEC 10165-1: Management Information Model

NA
ISO / IEC 7498-3: OSI Basic Reference Model - Part 3: Naming and Addressing

NTP
"Internet Time Synchronization: the Network Time Protocol", D.L. Mills, RFC 1129, October 1989

OAAC
ISO/IEC CD 10164-9: Objects and Attributes for Access Control

OM
ISO/IEC 10164-1: Object Management Function

Problem Statement
X/Open Systems Management: Problem Statement Snapshot, XO/SNAP/91/010, The X/Open Co. Ltd., 1991

SA
ISO/IEC 7498-2: OSI Basic Reference Model - Part 2: Security Architecture

SAR
ISO/IEC IS 10164-7: Security Alarm Reporting Function

SAT
ISO/IEC DIS 10164-8: Security Audit Trail Function

SCHM
OSI/Network Management Forum, Application Services: Scheduling Management Function. Forum 013, Issue 1.0 August 1991

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

STM
ISO/IEC 10164-2: State Management Function

SQL
Developers' Specification: Structured Query Language (SQL), the X/Open Company Ltd., 1990

TCS
DOD 5200.28-STD: Department of Defense Trusted Computer Systems Evaluation Criteria (the ''Orange Book'')

TESD
OSI/Network Management Forum, Forum Library - Volume 2: Testing Management Definitions. Forum 006, Issue 1.0, August 1991

TESM
OSI/Network Management Forum, Application Services: Testing Management Function. Forum 012, Issue 1.0, August 1991

TM
ISO/IEC DP 10164-12: Test Management Function

WM
ISO/IEC DP 10164-11: Workload Monitoring Function

X3

Accredited Standards Committee X3, Information Processing Systems, Data Base Systems Study Group Object Oriented Databases Task Group Final Report, ANSI, 1991

X.400

API to Electronic Mail (X.400), the X/Open Company Ltd. and the X.400 API Association, November 1991

XDS

API to Directory Services (XDS), the X/Open Company Ltd. and the X.400 API Association, November 1991

XMP

X/Open Systems Management: Management Protocols API, The X/Open Co. Ltd., in draft

XMPP

X/Open Systems Management: Management Protocol Profiles, XO/PRELIM/91/080, The X/Open Co. Ltd., in preparation

XOM

X/Open OSI-Abstract-Data Manipulation API (XOM), the X/Open Company Ltd. and the X.400 API Association, November 1991

XRM

X/Open Systems Management: Reference Model, XO/SNAP/91/040, The X/Open Co. Ltd., 1991

Chapter 1

Introduction

The X/Open Systems Management Reference Model (**XRM**) describes a model of systems management based on the use of managed objects to represent the real resources present in the system. This document identifies the services required in order to implement such a distributed management system.

It is intended to satisfy several high-level system requirements:

Portability	The ability to make software on managed and managing systems portable in source code form between different vendors' systems by extending the X/Open Common Applications Environment (CAE). An important additional aspect of portability is the "portability" of human administrators, that is, the ability of an administrator to move between systems and benefit from a consistent user interface.
Interoperability	The ability of management systems, and components of such systems from different vendors, to interwork, thus allowing a network of heterogeneous systems to be managed as a single system.
Location Transparency	The ability to administer resources without the need to be explicitly aware of their location.
Extensibility	The ability to extend the scope and capabilities of the management system and to implement different management policies as required. This includes the ability to make use of new communications protocols.
Robustness	The ability of the management system to provide integrity and the necessary levels of security and reliability.

The following requirements relate to the form of the interfaces that will be provided to access the management functionality:

Ease of Use	The services and APIs should be simple to use, consistent with the complexity of the underlying functionality.
Consistency	Wherever appropriate, stylistic inconsistency should be avoided in specification of interfaces.

The X/Open Systems Management Programme is defined in terms of a suite of documents that, taken together, will describe all the components needed to achieve the goals listed above.

The first of these documents is the X/Open Systems Management Problem Statement (reference **Problem Statement**). The Problem Statement provides an overview of the problem and a review of current activities.

The X/Open Systems Management Reference Model builds on the Problem Statement, providing a framework in which the various components of the solution can be identified. The individual components will be defined in subsequent documents.

The Reference Model is based on the use of object-oriented specification techniques. This in no way requires an implementation to use object-oriented technology. Object-oriented techniques have been adopted in this area by several other bodies, including vendors, standards bodies, and other industry consortia.

The Reference Model will be followed by several documents that can be grouped together according to three main headings:

Managed Objects — This group of documents will provide guidelines for the definition of Managed Objects, as well as the definition of basic Managed Objects for systems and network management. These definitions will be based on, and make reference to, the work of other bodies such as IEEE P1003.7 and the OSI Network Management Forum.

Management Services — The Management Services specifications will contain the definition of the intrinsic services that need to be provided by the management system. They will also define the APIs to those services.

Management Protocols — This group of documents will define the interoperability requirements in terms of profiles of management protocols such as CMIP. These profiles may be defined by reference to existing work within X/Open or elsewhere.

1.1 Scope and Purpose

The X/Open Systems Management programme (XSM) is concerned with the definition of those interfaces necessary for the portable implementation of distributed management systems. In many respects such systems are no different from any other distributed applications, requiring a complete range of distributed services to support them. In addition to these general services, distributed management systems also require some specialised services, providing support for specific management functionality.

The problem space that XSM addresses is that of distributed systems management. The primary requirement is to promote the development of management software that allows an administrator to manage a network of heterogeneous systems as a single logical system.

It is the purpose of this document to identify those services that are required to support such systems management activities.

Chapter 2

Management Services Overview

This Chapter provides a ‘‘quick reference’’ overview of the management services that are described in more detail in Chapter 3.

2.1 Management Services

The underlying capabilities that allow resources to participate in management are termed *management services*. They provide a generic set of functions that give the support needed to implement management functionality in an autonomous or distributed manner.

A management service is dedicated to assisting multiple applications (or other management services) in their management tasks by providing functionality common to many of them. A management service resides as an underlying component of one or more management solutions, and as such, is typically not exposed to end users. It often acts on a large number or large variety of managed objects or data.

Examples of management services are:

- an event forwarding service
- a data storage facility
- a data format translator
- a directory service
- an event logging and lookup facility
- a task automation service
- security services

A wide variety of services are required in order to develop distributed management systems. Some of these services are general in nature, while others are specific to management. This document identifies services of both kinds but distinguishes between them.

For each service that is identified, the requirement for that service and its relation to other management services are described. The options for meeting that requirement are then discussed. Where the requirement can be met using services or APIs defined in other publications from either X/Open or non-X/Open sources, those services and APIs are briefly described and the sources of their definitions are identified. Examples are the X/Open **OSI-Abstract-Data Manipulation (XOM)** service (reference **XOM**) and the OMG **Object Request Broker** (reference **CORBA**).

This document distinguishes between

- Generic Services
  These are services which provide functionality required by many applications and which are not specific to the management domain. Some of these services may have

management-specific components, for instance management-specific extensions to a generic naming service. Generic services should be provided in an integrated manner to both management and non-management service users.

- Management-specific Services
  These are services which are normally only made use of, directly or indirectly, by ‘‘management applications’’.

The following table lists the services discussed, and, for each service, gives:

- a brief description of its functionality
- an indication as to whether it is management-specific or generic.

Service	Functionality	XSM Specific
Alarm Management	filtering, reporting and logging of alarms (special case of events)	Yes
Basic Systems Management Service	manipulation of and interaction with managed objects	Yes
Data Abstraction Service	abstract data manipulation	Partly
Data Store Services	storing schema information (metadata), persistent object information, data dictionary, historical information, instance attribute information, etc.	Partly
Dispatcher / Object Messenger	sending messages to objects	No
Event Management	filtering, reporting and logging	Yes
Filtering Service	event and operation target selection by attribute	Yes
Instance Enrolment Service	entry of objects into the MIB	Yes
Management Communications Service	conveying management information between systems	Yes
Name Resolution (Directory) Service	determination of location and other information about an object from its name	Partly
Scheduling Management	task automation	Possibly
Scoping Service	operation target selection by position in the name space	Yes
Security Service	authentication, authorisation, privacy (encryption), integrity, non-repudiation, audit, etc.	No
State Management	operability and availability	Yes
Testing Management Service	diagnostics, confidence tests etc.	Yes
Timing Service	distributed timing	No

Note that the scheduling service is identified as possibly management specific since one of the options for realising it is management specific while another - the ‘‘cron’’ facility - is not.

Table 2-1 List of Management Services

2.2 Grouping Management Services

This document identifies those services that are required in order to construct management systems. It is not necessarily the case that every management service identified in this document will be provided separately with its own API. Grouping of individual services into useful, integrated, aggregate service sets may be appropriate. At the present time, it is not clear how such a grouping should be performed, and several possible grouping mechanisms can be imagined. Within the following Chapter, the services are simply presented in alphabetic order.

One possible grouping mechanism divides services into two groups:

1. Aggregate Services.
   These are services that are realised by the definition of particular object classes, attributes and behaviour.

2. Core Services.
   These are services that provide underlying mechanisms.

The definition of particular classes of managed objects associated with particular types of managed resource allows aggregate services to be defined that can be used in the performance of many administration tasks. Such object classes would be defined in the Managed Objects group of documents produced by the X/Open Systems Management programme. Aggregate services can also be defined by describing attributes, operations, notifications and behaviour which are common to managed objects representing some defined set of (or all) managed resources. The State Management Service described in Section 3.13 on page 47 is an example. In some cases, an aggregate service can be further refined to provide another, more specific, aggregate service. For example, the Alarm Management service is effectively a special case of the more general Event Management service.

The core services include services for interacting with and manipulating managed objects. They also include services that are not related to managed objects, such as Distributed Timing.

2.3 Relation to OSI Management Functional Areas

In the **OSI Reference Model Management Framework** (reference **MF**), five OSI Management Functional Areas are identified. These are:

- fault management
- accounting management
- configuration management
- performance management
- security management.

These Functional Areas serve as a way of categorising administration tasks but do not form a convenient starting point for the definition of management services because there is no specific and distinct set of services associated with each area. Rather, there is a set of generic services, each of which applies to most or all areas.

For example, the Testing Management service could be used in fault management (for diagnostic testing), configuration management (for confidence testing) and performance management (for performance testing).

In the case of Security Management, some clarification is required. The Security Management functional area is concerned with the management of security services. This can be distinguished from the provision of security services, which are required potentially to support all applications and services in a system. Those required to support management applications and services are discussed in Section 3.12 on page 41. Security Management is also discussed briefly in that section.

Chapter 3

Description of Services

3.1 Alarm Management

3.1.1 The Requirement

Alarms are the means by which system administrators are notified of abnormal conditions. They are events generated as notifications by managed objects. Early detection of faults, before significant effects have been felt by the user, is highly desirable. The Alarm Management service is concerned with the collection of alarms and with forwarding them to applications that will handle them (typically, by bringing them to the attention of administrators). It enables abnormal functioning of a system to be reported in a timely manner. It is essential for fault management.

The information conveyed in an alarm must include the source of the alarm and its severity, such as critical, major or minor. Users must be able to track the status of an alarm until the abnormal condition is restored to normal. Analysis of statistical alarm data can be useful in predicting problems for corrective action before any significant impact on service occurs.

3.1.2 Relation to Other Services

The Alarm Management service uses the Event Management service for reporting and logging of alarms. This in turn uses the filtering service to determine which events will be reported or logged.

3.1.3 Options for the Service

ISO has defined an **OSI Alarm Reporting** function (reference **AM**) which uses the CMIS event reporting service. It is briefly described below.

3.1.3.1 Types of Alarm

The following types of Alarm are identified:

Communications Alarm:
: an alarm associated with the process of sending information from one point to another.

Quality of Service Alarm:
: an alarm associated with the degradation in the quality of operation of a service.

Processing Error Alarm:
: an alarm associated with a software or processing fault.

Equipment Alarm:
: an alarm associated with an equipment fault.

Environment Alarm:
an alarm associated with a condition relating to an enclosure in which equipment resides.

3.1.3.2 Information Associated with Alarms

Each alarm will have associated with it the following information. The specific information values generated for each alarm may be specified in the definition of the object class of the managed object that generates it or may be assigned by the managed system in a system-dependent way, for example as determined by

- the system designer
- the system administrator
- the user.

Probable Cause

This parameter gives further details on the underlying problem; for example, for a communications alarm this field may be used to indicate, say, a framing or a call set-up error.

Specific Problems

This parameter qualifies the probable cause and consists of a set of ASN.1 object identifiers (which may be registered) or integers (which may, for example, index further information in a system's documentation).

Perceived Severity

The severity level of an alarm indicates the extent to which a fault that gave rise to the alarm impacts the normal operation of the object reporting a fault. The following 6 levels of severity are identified:

Cleared:
indicates the clearing of one or more previously reported alarms.

Indeterminate:
indicates that the system cannot determine the severity of the fault.

Critical:
indicates that service is affected and immediate corrective action is required.

Major:
indicates that service is affected and urgent corrective action is required.

Minor:
indicates that a fault has occurred that does not affect service and that corrective action should be taken to prevent a more serious fault.

Warning:
indicates the detection of a potential or impending fault.

Backed Up Status

An alarm can indicate whether the service represented by a managed object that has failed is now being provided by an alternative managed object (for example, a redundant spare).

Backup Object

This parameter is only present when the ‘‘Backed Up Status’’ parameter is true. It specifies the identity of the managed object instance providing the backup service.

Trend Indication

This is an optional parameter that can be used when one or more alarms have not yet been cleared and continue to be outstanding to indicate the severity trend of those alarms. It can take the following values:

More Severe:
: the severity in the current alarm is higher (more severe) than that in previously reported alarms.

No Change:
: the severity in the current alarm is the same as the highest of any previously reported alarms

Less Severe:
: there is at least one outstanding alarm that has higher severity (more severe) than the current alarm.

Threshold Information

When an alarm is the result of a threshold being crossed, this parameter indicates which threshold attribute triggered the alarm and the values associated with the threshold being crossed.

Notification Identifier

This parameter provides an identifier for the alarm which may be carried in the ‘‘Correlated Notifications’’ parameter of future alarms. It may be used to refer to the alarm later for, say, alarm clearing.

Correlated Notifications

This parameter identifies a set of previous alarms to which the current alarm is related.

State Change Definition

When the alarm is associated with a change of state of a managed object, this parameter may be used to describe that change.

Monitored Attributes

Managed object designers may specify a set of attributes, changes to which will be reported using this parameter.

Proposed Repair Actions

The object class designer may specify possible repair actions. They are identified in this parameter by ASN.1 object identifiers or by integers.

Additional Text

This parameter allows a free form text description of the problem being reported.

Additional Information

This parameter allows additional information relating to a fault to be reported in coded form.

3.2 Data Abstraction

3.2.1 The Requirement

Data Abstraction Services provide representations of management information by data structures and provide facilities for the creation and manipulation of those data structures. They are required to enable information to be passed between programs in a way that is independent of:

- hardware architectures
- programming languages and language translators
- communications protocols

in order to promote portability of and interworking between programs of various types concerned with Systems Management, including

- managers
- agents
- protocol handlers.

The Data Abstraction services used in connection with Systems Management must allow all types of Systems Management information to be represented. This includes managed objects, their attributes and the events that they generate.

The requirement for interworking includes a requirement for interworking between applications in separate, communicating Open Systems and a requirement for interworking between applications within a single Open System.

While enabling interworking between Management Applications, the Data Abstraction service should not constrain the implementation of an application's internal data structures. Nor should it inhibit the ability of a supplier to add value to systems and applications in ways that do not affect their ''open-ness'' to other suppliers' products.

The Data Abstraction service should include the following features:

- It should provide a standard, vendor-independent mapping between any managed object definition and the supporting X/Open API(s). Such mapping services should ideally be capable of automation, starting from an X/Open-recognised object definition format such as ISO GDMO.
- Wherever possible, it should perform protocol encoding/decoding within the API without requiring application intervention, or provide protocol encoding/decoding services which an application can invoke. This includes content-specific information (that is, specific attribute, notification action and parameter syntaxes).
- It should allow consistency across the programming interface (that is, it should be possible to build content-independent and content-specific management service parameters using the same set of Data Abstraction services).
- It should support both static and dynamic definition of data abstractions (that is, it should be possible to add new managed object definitions, and to update existing ones, both at compile time and at run time).

3.2.2 Relation to Other Services

The Data Abstraction Service does not use and is not directly used by other Systems Management services. It is potentially usable, however, in the definition of the API of any service.

3.2.3 Options for the Service

A data abstraction service has been defined and is specified by X/Open (jointly with the X.400 API Association) for use in conjunction with the X/Open Message Handling Service (reference **X.400**) and the X/Open Directory Service (reference **XDS**). It is described in the X/Open OSI-Abstract-Data Manipulation (XOM) specification (see **Referenced Documents**). It is the same service as that of the draft IEEE standard 1224 which is used in the draft IEEE message handling and directory service API standards (1224.1 and 1003.17). This service is potentially suitable for use in Systems Management and is used in the X/Open Management Protocols API specification (reference **XMP**).

XOM is independent of implementation considerations. It is defined in a programming language independent manner. (Also, a C language binding is defined for it. The definition of other language bindings is possible but no work on other bindings is currently planned within either X/Open or POSIX.)

XOM provides a way of representing information that is in some senses (though not in the fullest sense) ''object-oriented''. Information is represented as *objects* which have *attributes*. Each object is a member of a particular *object class* which determines the set of attributes that it may have. The object classes are arranged in a hierarchy and an object that is an instance of one class may have any of the attributes associated with that class or with any of the superiors of that class in the hierarchy (it need not, however, have values for all such attributes).

The XOM concept of object thus incorporates a form of class inheritance but it does not incorporate the concept of *method* (found in ''object-oriented'' programming languages such as Smalltalk) in which a set of information processing functions (methods) are associated with each object class.

Instead, XOM has a concept of *workspace*: an area of storage in which objects can be created and manipulated by a set of information processing functions. That set of functions is associated with the workspace rather than with the object classes. This facilitates operations (such as copy) that involve more than one object.

The workspace concept allows each software vendor to implement objects in a proprietary way and still present a standard interface to other software. A class of objects may be implemented differently by two different vendors. The two implementations will interwork and the software of one vendor can manipulate the objects implemented by the other. An applications program can interwork simultaneously with both implementations and pass information from one to the other.

XOM provides for representation of any information that is describable using ASN.1. The ASN.1 notation is used by CMIP to describe management information so that XOM provides the same generality of information representation as CMIP and CMIS. In particular, it can be used to describe managed objects, attributes and events and is in fact used in this way in the **XMP** API.

XOM provides static package definitions which are used by XMP to represent both protocol parameters and managed objects. XMP permits these OM packages to be added dynamically to the workspace at run time.

XOM has many qualities that would make it an appropriate choice for the X/Open Systems Management Data Abstraction service. Its chief drawbacks are its conceptual complexity, and the fact that it does not wholly shield the user (programmer) from the complexity of the data. While these drawbacks are outweighed by the advantages, they may indicate a requirement for work on higher level tools to assist the interface definer and the programmer, for example by automatically generating XOM representations of managed objects and attributes from their specifications.

One such tool has been produced by Bellcore (reference **GDMO Algorithm**, shortly to be published by X/Open as **Guide to Translating GDMO to XOM**). This translates ISO GDMO templates (reference **GDMO**) and ASN.1 syntax definitions (reference **ASN.1**) into XOM package definitions. It could prove of material assistance to specification writers defining XOM packages for systems management - and other - services and applications. It could thus be a starting point for meeting the high level tools requirements of the Data Abstraction service.

3.3 Data Store

3.3.1 The Requirement

An important aspect of systems management is the maintenance of considerable quantities of non-volatile data. The ‘‘life-span’’ of entities that require management varies from the very short (for example, processes), through the transient (for example, print queue entries), to the long-lived (for example, users).

The data storage requirements for these entities are obviously very varied. They can perhaps be distinguished by whether the data is required to persist across a system reboot.

Data on short-lived entities may be simply maintained in memory for the life of the entity. A log entry or accounting record may be created to record its brief existence. Such an entry would presumably be recorded on non-volatile storage for later analysis. For semi-permanent entities such as users, the data must be stored in permanent storage.

Systems Management applications must be capable of adapting to new devices and subsystems introduced into a system. To adapt, the applications will need data that defines how to monitor and control new entities. The Data Store service must be able to cater for the new types of information that are associated with these entities.

While the particular implementation of the storage is very much an implementation-dependent issue, the interface by which it is accessed is an important factor in providing portability. In order to provide a coherent programming environment, common access methods should be used to store different forms of data, with the developer having the capability to specify the requirements that are placed on particular items of data. Within a management system based on the concept of managed objects, the interface to the data storage function should be structured in order to reflect the use of managed objects, providing simple means to store and retrieve either complete or partial data about an object.

The data store service must cater for different versions of information, representing the changing state of the managed system through time. These include:

- current state - to reduce management traffic in a networked system
- previous state - for fault management comparison
- initial state - for startup and configuration of the system.

3.3.2 Relation to Other Services

The Data Store service may be used by any other management service and by applications.

3.3.3 Options for the Service

There are a number of data store services that could be used. They include the following.

File Handling

The simplest form of Data Store Service that can be envisaged comprises the file handling services of the X/Open CAE (*open*( ), *close*( ), *read*( ), *write*( ), etc).

SQL

The database management service provided by SQL is capable of handling information that is structured in accordance with the Relational model of data. This service (reference **SQL**) will be available to, and will probably be used by, managers, agents and applications in XPG compliant systems.

Object-Oriented Data Base

A more ''object-oriented'' approach to the Data Store service is provided through the concept of an object-oriented database front-end that serves as a logical central collection point for management information. A manager, agent or application invokes methods on objects in the data store. It is the responsibility of the method within the data store to determine the means for manipulating management information. In many cases a management protocol flow may be necessary in which case the invoked method may in turn invoke a method of a particular ''management protocol'' object.

The ANSI Object-Oriented Database Task Group (OODBTG) is a task group of the Database System Study Group, an advisory body to the Accredited Standards Committee X3 of the American National Standards Institute. The group published its final report in September 1991 (reference **X3**). It consists, in particular, of:

- recommendations for standards in Object Information Management
- Object Data Management reference model
- Object Data Management glossary
- Object Data Management bibliography.

The ODM reference model is an abstract object model applicable to any and all application areas. It represents object concepts in a concise, precise and explicit manner.

3.4 Event Management

3.4.1 The Requirement

Events occur throughout a system in response to stimuli such as values being input, signals being received, invalid or unauthorised operations being performed or scheduled operations being undertaken. They are detected by managed objects and typically result in notifications being issued. Some of those notifications will be processed by the management subsystem itself. Many, however, are required, at least potentially, to be brought to the attention of an administrator.

In a large and complex system, the rate at which events are generated can be very high, making it impossible for a human system administrator to deal with - or even to become aware of - all of them as they occur. It is thus necessary to select the information that is brought to the attention of human system administrators. It is also necessary to keep a historical record of events for subsequent reference, for example to assist with fault diagnosis or to investigate a breach of security.

The event management service provides the facilities of selective reporting and historical record keeping (logging) that enable administrators to deal with events occurring in a system without being swamped by too much information. As well as being for use by human system administrators, they also form a useful event pre-processing service for systems management applications.

The services required include the reporting, logging and subsequent analysis of events.

3.4.1.1 Event Reporting

The Event Reporting service determines, by examining the content of event notifications, where, if anywhere, they should be forwarded. To receive reports, recipients must express an interest in them. Forwarding is decided by looking at filter criteria supplied by those entities that have registered an interest. If the Notification matches the filter criteria then a report is issued.

3.4.1.2 Event Logging

The Event Logging service keeps a historical record of events in some form of permanent or semi-permanent store. A filtering mechanism is applied to event reports before committing them to store. Event reports need to be stored with associated time-stamps. The store may be considered to be of finite capacity and the Event Logging service should enable the user to determine the course of action to be taken when available storage capacity is used up.

3.4.1.3 Event Analysis

Event Analysis looks at the event notifications in order to allow for statistics gathering and possibly to schedule follow-up actions to be performed. Statistics may be used for many purposes such as indicating problems due to rising error rates. Follow-up actions may be required in situations that are critical in security or operational terms; for example, an alarm could be generated if more than a certain number of invalid log-in attempts were made over a given period.

3.4.1.4 Control of Event Management

Each Event Management service should be able to be

- initiated
- stopped
- suspended
- resumed
- configured
- queried.

3.4.2 Relation to Other Services

In addition to being used directly by applications, the Event Management service is used by the Alarm Management service.

The Event Management service uses the basic Systems Management Service to receive notifications of events. It uses the filtering service to select the events to forward to each recipient. It uses the state management service to enable an administrator to monitor its operation and to bring it into use and out of use. It uses the scheduling service to turn reporting or logging off and on periodically. It may also use the Data Store service to store event records and the timing service to enable it to apply time stamps.

3.4.3 Options for the Service

ISO has defined an OSI event reporting control function (reference **ERM**) and an OSI log control function (reference **LC**).

In the ISO approach, the Event Management services are realised by defining discriminator objects to control the forwarding of notifications as events and defining log objects to control the logging of events. These objects do not directly correspond to managed resources but can be operated on to provide Event Management functions by the OSI Object Management service (reference **OM**) which in turn uses CMIS, and by the OSI State Management service (reference **STM**).

The notifications generated by each managed object are passed to discriminator objects within the system that contains the managed object. The discriminators determine which notifications are to be forwarded as events and the destinations to which they are to be forwarded. This is done on the basis of

- the class or specific instance of managed object that generated the notification
- the parameters of the notification (for example - in the case of alarm events - severity, backed up status, probable cause, etc.)
- the time at which the notification was received (the discriminator uses the scheduling service to turn reporting on and off).

The notifications generated by each managed object are passed to log objects within the system that contains the managed object. Protocol engines are represented by managed objects so that receipt or generation of PDUs can result in notifications where

appropriate. Event reports received by the system are passed to its log objects also. The log objects include discriminators that determine which events are to be logged, using the criteria described above. Log objects also include defined behaviour that they exhibit when they become full and may also include behaviour (generation of alarms etc.) that they exhibit when they become almost full.

3.5 Filtering

3.5.1 The Requirement

A basic requirement in systems management is to apply a management function to a number of managed objects. For example, a system administrator might wish to gather utilisation statistics from all systems in a particular department of his organisation or to update routing tables throughout a network or subnetwork.

This requirement is met by the Filtering and Scoping services. The Scoping service allows managed objects to be selected by their position in the global name space. The Filtering service then allows selection of managed objects whose attributes meet specified conditions.

A further requirement is to filter the events that are input to particular applications or brought to the attention of system administrators. The Filtering service meets this requirement by enabling events to be filtered in the same way as managed objects are filtered.

3.5.2 Relation to Other Services

The Filtering service is used by management applications only through other systems management services.

The basic Systems Management Service allows filtering constraints to be placed on requests to delete objects, examine and modify attributes, and invoke operations. In order to meet these constraints, the filtering service is used (often together with the Scoping service) by the basic Systems Management Service and by the Communications services which are also used by the basic Systems Management Service.

The precise relationship between the Filtering service, the Dispatcher and the Communications service is currently unclear. The OSI communications service, CMIS, explicitly supports filtering but other communications services, such as that provided through SNMP, do not. Whether (and how) filtering can be applied when CMIS is not used requires further study. Also, given that the Dispatcher provides communications protocol transparency, there is a question of what form of filtering it should provide and of how it should provide it.

Other services that use the basic Systems Management Service, such as State Management and Scheduling, may also allow the user to specify filtering constraints when they are invoked.

The basic Systems Management Service does not apply Filtering to notifications. The Event Management service, however, applies Filtering in selecting events to be logged and reported. The Alarm Management service, which uses Event Management, also allows for alarms to be filtered.

3.5.3 Options for the Service

The CMIS Specification (reference **CMISD**) implicitly includes a Filtering service. This is described below.

Each of the CMIS M-SET, M-GET M-DELETE and M-ACTION services supports filtering. When both scoping and filtering are requested, scoping is applied first. This means that a single message can be transmitted to each management agent involved and that agent will then apply the message to a number of managed objects. In such a case, CMIS also allows the managing system to specify how the requested operation will be synchronised across the selected managed objects.

A filter is a set of one or more assertions about the presence or the values of attributes of a managed object or event. If the filter contains more than one assertion, the assertions may be combined using the logical operators AND, OR and NOT. Because nested logical operations are permitted the filter expression allows for arbitrarily complex filtering. Assertions may be applied to multivalued attributes, in which case the assertion is TRUE if it matches one or more of the values for that attributes (it need not match all of the values). If an attribute value assertion is present in the filter but that attribute is not present in the scoped managed object, then the result of the assertion is FALSE.

This concept of a filter is similar to that used in the CCITT X.500 Series Recommendations for the Directory Service (reference **DCMS**) and in the X/Open API to Directory Services (reference **XDS**).

Attribute Value Assertions in filters may use the following tests:

- check for the presence of an attribute

  or

- take a value and compare it against an attribute value and
  - check if the value is equal to an attribute value
  - check if the value is greater than or equal to the attribute value
  - check if the value is less than or equal to the attribute value
  - check if the value is a substring of the attribute value
  - check if the value is a subset of the attribute value
  - check if the value is a superset of the attribute value
  - check if the union of the value and the attribute value does not yield a null set.

These tests can only applied to attributes that specifically allow them. The definition of an attribute includes its syntax which defines the type of assertions that can be applied to it.

The synchronisation parameter is used to determine how an operation will be applied to a number of managed objects at once. Synchronisation can be:

- *Atomic:*
  the requested operation is applied only if it can be successfully applied to all selected managed objects.
- *Best Effort:*
  the requested operation is applied to all the selected managed objects but some operations may fail.

3.6 Instance Enrolment

3.6.1 The Requirement

Managed objects can be created and deleted in three ways:

- through the normal operation of the resources they represent
- through the use of the Systems Management Services
- through local information processes that are outside the scope of X/Open specifications.

The creation of a managed object consists of placing its name/identifier and set of attributes appropriate to its class in the Management Information Base (MIB). In some cases, this will mean the creation or modification of data that is used only for management purposes. In other cases (as, for example, when a process is created), the overhead of maintaining separate management records may be considered to be too great; in these cases only operational records (for example, process table entries) are created and the MIB entries exist in concept only.

When a managed object is created, it may be necessary to update records used by the Name Resolution service so that the object can be accessed through the Systems Management services (again, this will not be desirable for all objects). It may also be necessary to update relations that include the object and, in particular, the containment relation.

Similarly, the deletion of a managed object consists (conceptually) of removing its identifier and associated set of attributes from the Management Information Base and, in some cases, implies updating Name Resolution records to remove those pertaining to the object and possibly updating the containment relation and other relations.

The creation and deletion of a managed object (and also changes in the object's definition that affect naming) may thus affect parts of the MIB other than just the part that consists of the object itself. In many cases (for example, when managed objects are created through the normal operation of the resources that they represent), these changes are instigated by an entity playing an agent role, rather than a manager role. The Instance Enrolment service is required in order to to handle any necessary notification and updating of such parts.

The implications of object creation and deletion in OSI terms are as follows:

- The creation of an object includes the completion of a previously defined Name Binding Template, and registering the information in the Management Information Base (MIB). The Name Binding template specifies an object class of which the managed object is an instance, managed objects which may be directly superior to it in the global naming tree, and attributes that can be used for naming (ie attributes that can be used to construct a Relative Distinguished Name - RDN). When the addressing information in a completed name binding template, together with the address of the agent that serves the object, are placed in a directory, it can subsequently be accessed by the Name Resolution service or serve as a ''junction object''.

- The deletion of an object involves the removal of Name Binding Template information previously entered into the MIB.

3.6.2 Relation to Other Services

The Instance Enrolment service is used by the basic Systems Management service to generate creation, deletion and change notification events. It may update directories used by the Name Resolution service.

3.6.3 Options for the Service

In its Object Management function definition (reference **OM**), ISO has defined object change notification services which are tools that could be used by some other entity or service to update the MIB. This approach is summarised below.

The Object Management service defined in **OM** enables an entity responsible for a managed object in one system to report its creation, deletion or modification to systems management entities in other systems. It includes the following particular services, each of which is mapped onto the CMIS M-EVENT REPORT service using appropriate parameters:

- Object Creation
- Object Deletion
- Attribute Change (which is capable of reporting modification, addition or removal of attribute values).

All the above services specify, as parameters in the reporting message:

- a message identifier
- the mode of the operation (confirmed or non-confirmed)
- the identity of the object class of the object
- the identity of the instance of the managed object
- the event type
- the event time
- event information that is specific to each type of event.

3.7 Managed Object Interaction

In the X/Open model of Systems Management (reference **XRM**), managed resources are represented by managed objects. Management of managed resources takes place by performing operations on, and receiving information from, the managed objects that represent them.

The characteristics of a managed object can be expressed in terms of its attributes, operations, notifications and behaviour. The behaviour of a managed object defines how the attributes, operations and notifications work together and how they affect the managed resource that the managed object represents.

The X/Open systems management services must:

- enable managed objects to be created, modified and deleted
- enable the values of their attributes to be examined and changed
- enable their operations to be invoked
- handle notifications.

Two services that meet these requirements have been identified. They are alternatives, in the sense that, in any particular instance of interaction with a managed object, a user service or application will invoke either one or the other. However, it is possible for both to be available in a single system. The X/Open view of distributed systems management comprehends them both.

The two services are the Basic Systems Management Service and the Dispatcher.

The Basic Systems Management Service provides functionality that is similar to that provided by the OSI Object Management service defined in reference **OM**. That is, it includes functionality similar to that of CMIS and adds functionality relating to the creation and deletion of managed objects. The essential differences between the Basic Systems Management Service identified by X/Open and the OSI Object Management service are that the Basic Systems Management Service:

- applies to the management of entire distributed systems rather than just to the communications aspects of such systems
- allows for communications that use protocols other than OSI protocols (notably, IPS protocols) as well as for communications that use OSI protocols.
- applies to management interactions that take place within a single system as well as to those that take place between interconnected systems.

The Dispatcher (also referred to as the Object Messenger) is the key component of an object oriented architecture. When a request is made for an object to perform an operation, it is the Dispatcher that determines where the object is located, arranges for a message containing the request to be conveyed to it, queues that message for attention by the object and, finally, returns the results of the operation to the requester. Where this means sending information between communicating systems, the dispatcher selects the communications protocols to be used.

The relationships between the Basic Systems Management Service, the Dispatcher and the Management Communications services are illustrated in Figure 3-1.

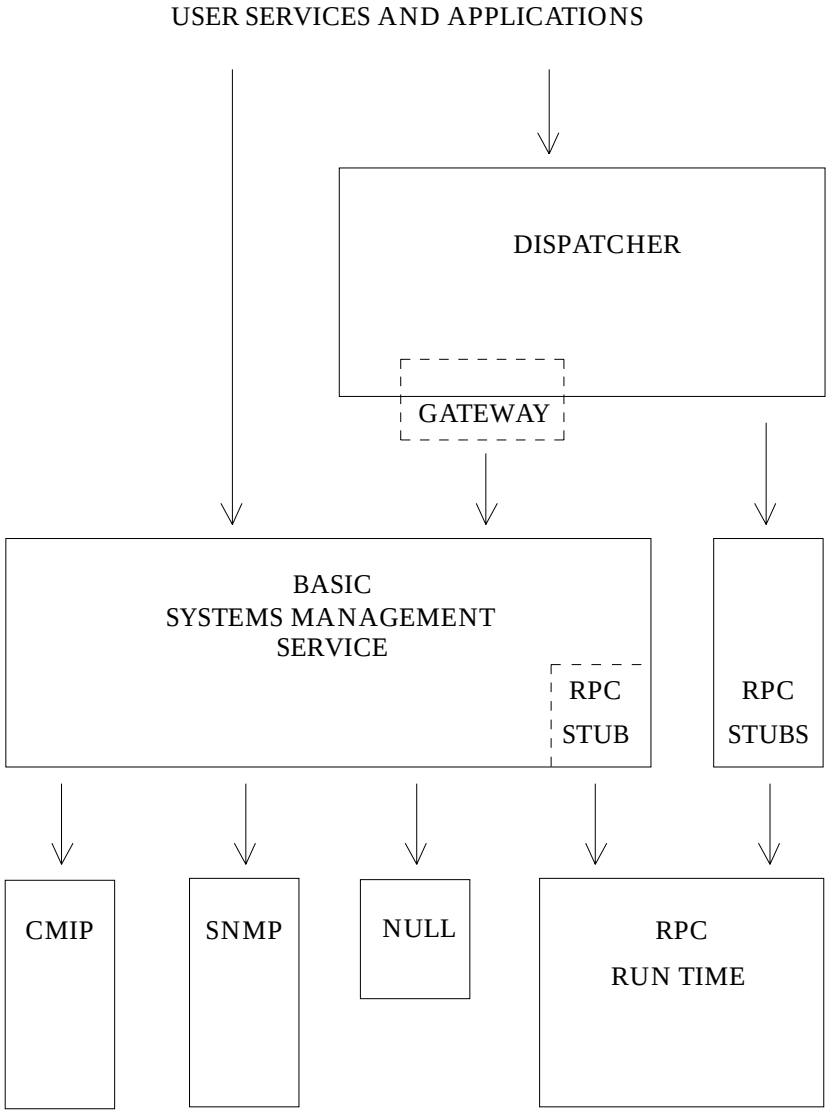

Figure 3-1 Managed Object Interaction Service Relationships

Other management services and applications may use either the Basic Systems Management Service or the Dispatcher in order to interact with managed objects.

If the Basic Systems Management Service is used, it will provide communications with an agent through which the managed object can be accessed. If that agent resides in the local system then communications are purely internal to that system and no protocols are required. In other cases, the communications may use CMIP or SNMP as appropriate. Alternatively, the Basic Systems Management Service may use RPC to invoke a procedure in the remote system to perform the managed object interaction.

If the Dispatcher is used, it will send a message to the target managed object, either by using RPC or by using a gateway to invoke the Basic Systems Management Service.

3.7.1 Basic Systems Management Service

3.7.1.1 The Requirement

The basic Systems Management Service must satisfy the following requirements.

Object Creation and Deletion

The service must enable an application to create and delete objects.

Creation or deletion of an object implies its addition to or removal from the MIB and typically affects the MIB in other ways also. The service must take responsibility for all actions consequent on the creation or deletion of objects.

The service must provide for the reading of attribute values by applications. For those attributes that can be modified by applications, it must provide for the addition, deletion and modification of attribute values.

Operation Invocation

The service must enable an application to invoke an operation provided by an object and to receive any results that the operation generates.

Notifications

The service must enable objects to issue notifications and enable applications to receive those notifications.

Kinds of Object

The above facilities must be provided for all kinds of objects defined for Systems Management. This includes managed objects that correspond directly to managed resources. In addition, it includes other objects (such as the discriminators used by Event Management described in Section 3.4 on page 18 that affect the way in which management services operate or that play some other Systems Management role.

Communications Protocols

When communication between systems is required, the service must be able to use the appropriate protocol stack. This could be an OSI protocol stack supporting CMIP, an IPS protocol stack supporting SNMP or some other protocol stack, such as one that supports RPC.

Security

The service must operate in a secure way, having regard to the fact that the whole of a system often becomes vulnerable once the security of its management services is breached.

Selection of Target Objects

The service must be able to apply a single request from an application to a number of objects and to select those objects in accordance with criteria which the application supplies.

Location Transparency

The service should not require an application to be aware of the physical location of the managed resources with which it interacts.

3.7.1.2 Relation to Other Services

The State Management, Testing Management, Event Management, Alarm Management and Scheduling Management services use the basic Systems Management Service, either directly or indirectly.

The basic Systems Management Service uses the following other management services (either directly or indirectly):

Name Resolution:
to provide location transparency

Communications:
when managers and agents are located in different systems

Scoping:
to select target objects

Filtering:
to select target objects

Instance Enrolment:
when objects are created and deleted

Security Services:
to ensure that everything it does is done in a secure way.

The relationship between the Basic Systems Management Service, the Dispatcher and the Communications Service is discussed in Section 3.7 on page 26.

3.7.1.3 Options for the Service

The X/Open Management Protocol (XMP) API Specification (reference **XMP**) defines an API that can be used over either CMIP or SNMP. It thus realises a significant part of the Basic Systems Management Service and forms a starting point for the definition of that service.

3.7.2 The Dispatcher

3.7.2.1 The Requirement

The Dispatcher must act as a high level Object Request Broker, allowing other Systems Management services to send messages to and receive messages from managed objects in a manner that is both location independent and transparent of internal structure.

The messages sent by the dispatcher will invoke the functionality described in Section 3.7.1.1 on page 28. However, the Dispatcher is not required to be aware of their significance, it simply provides a generic object-oriented messaging mechanism.

The role of the Dispatcher in handling notifications generated by objects requires further study.

The Dispatcher must handle both the synchronous and the asynchronous mode of operation and must manage linked replies for a single request.

In order to provide location independence, it must be able to resolve a managed object name to a specific piece of code at a specific address in the system or network.

It is responsible for determining the management protocol necessary for delivering each message and formatting the message accordingly. Currently, X/Open has identified the need to support the SNMP and CMIP management protocols in addition to RPC (reference **XMPP**). Hence the Dispatcher must at least support RPC and provide a gateway to the Basic Systems Management Service for use when SNMP or CMIP is required. The Dispatcher must also support the case where the object is in the same system as the requester and no communication is required and it must be flexible enough so that, in the future, other protocols can be added.

The Dispatcher must manage the manager-agent connection and share that connection among applications if and when appropriate. In the OSI environment it will use the Association Control Service Element (ACSE).

The need for security of Systems Management services in general has specific implications for the Dispatcher. For example, it may have to determine the responder's authentication scheme and authenticate the responder to the requester and/or authenticate the requester to the responder. It may have to check the authorisation of the requester to access the particular method of the specific object instance, or pass sufficient information to the responder to enable the responder to perform the check.

The dispatcher may be defined to perform services in addition to the above. For example, it could, given a request, select the most appropriate object to perform the operation.

3.7.2.2 Relation to Other Services

The Dispatcher will be used by applications for interactions with objects. It may also be used in the same way as the Basic Systems Management Service by aggregate services such as State Management and Event Management. While this is feasible in principle, the aggregate services are largely defined in terms of OSI management and would therefore most naturally be used with the Basic Systems Management Service and CMIP. The possibility of their using the Dispatcher requires further study.

The Dispatcher will use the Name Resolution service to resolve names. To provide protocol independence, it must be able to discover the type of management communications protocol to be used for communications with a managed object. One approach would be for the Name Resolution Service to return the type of protocol to be used as part of the location information it normally returns.

In order to meet the needs for secure operation, the Dispatcher will use the Security service.

The relationship between the Dispatcher, the Basic Systems Management Service and the Communications Service is discussed in section Section 3.7 on page 26.

3.7.2.3 Options for the Service

The Object Management Group (OMG) has defined an ‘‘Object Request Broker’’ (reference **ORB**) that provides the mechanism by which managed objects transparently make requests and receive responses.

The ORB definition can be used as a source document for identifying the services required by a Dispatcher/Object Messenger, but it may not be possible to adopt it unmodified and in its entirity. The ORB definition goes further than identifying architectural issues and includes C++ and C interface specifications. X/Open would wish to base its interface specification on other X/Open Management Services which the ORB does not use. Furthermore, deficiencies have been identified when compared with current work within ISO. For example it does not specify the capability to allow an object instance, completely independent of any client or other object, to issue a notification. Also, it does not support the OSI concept of filters.

Further work is required to refine the Dispatcher service definition, in the context of the X/Open Systems Management Reference Model (reference **XRM**) as a basis for the selection of a particular detailed specification for the service.

3.8 Management Communications

3.8.1 The Requirement

Management of a distributed system clearly requires a management communications service to enable management information, requests to perform operations and event notifications to be conveyed between the distributed parts of the system.

Several different services have been or are being developed for this purpose. In particular, there are the network management communication services CMIS (reference **CMISD**) which is an OSI International Standard, and SNMP (reference **SNMP**) which is an IPS standard that is popular on account of its practical functionality and comparative simplicity. There are also communications services based on RPC that are used for management purposes, often within products that conform to a ‘‘client-server’’ architecture.

Within a single distributed system, different items of equipment may use different management communications services. Hence, the Communications Service defined by X/Open for distributed systems management must be a general service that comprises a number of different particular communications services, including any or all of those mentioned above.

The Communications Service must enable the following to be communicated between managers and agents within a distributed system:

- requests to create, delete or modify the definitions of managed objects
- requests to examine or modify the attributes of managed objects
- requests for managed objects to perform operations
- notifications issued by managed objects.

The communications service should allow a single instance of communication to apply to a single object or (by applying scoping and filtering) to a set of objects.

In addition to manipulating and interacting with managed objects, management services and applications sometimes need to transfer large amounts of data from one system to another (for example, to download a program). Protocols that are oriented towards managed object manipulation are generally unsuitable for this purpose in that their overheads are much too high. There is therefore a need for a bulk data transfer service for use by management services and applications.

Although the Communications Service must incorporate all the above facilities, it should be stressed that they will not in general be relevant for all managed resources in a system or part of a system and that they will not necessarily be provided by all managers and agents. In particular, each agent is likely to provide only a subset of the above facilities, probably depending on the particular communications service(s) (such as CMIS or SNMP) that it supports.

3.8.2 Relation to Other Services

The Communications Service is used by the basic Systems Management Service and the Dispatcher. Through these two services, it is used indirectly by other services - such as Event Management, Alarm Management, Scheduling and Test Management - and by management applications.

The relation of the Communications Service to the Basic Systems Management Service and the Dispatcher is discussed in section Section 3.7 on page 26.

The relation of non-OSI Communications Services to the Filtering Service is unclear to some extent. This is discussed further in Section 3.5 on page 21.

The Security services that provide security of management use communications services. However, these are general communications services, such as might be used by any application or service, rather than Management Communications Services that are specifically designed to meet the needs of management applications and services.

The Timing service also uses communications services. These are likely to be provided through special-purpose communications protocols. The needs in this area can not be assessed until the precise form of the Timing Service has been determined.

3.8.3 Options for the Service

The X/Open Systems Management Protocol Profiles Specification (XMPP) (reference **XMPP**) specifies a number of management protocols (and supporting communications protocols) that are appropriate for use in distributed Systems Management. They currently consist of the following:

CMIP The OSI Common Management Information Protocol (reference **CMIP**). This should be implemented in conjunction with OSI ACSE and over OSI Presentation and Session layers in accordance with either of International Standardised Profiles AOM-11 and AOM-12, which are defined in **APS**, **EMC**, and **BMC** (see Referenced Documents).

SNMP The Simple Network Management Protocol (reference **SNMP**), in conjunction with other IPS Protocols.

The X/Open Management Protocol (XMP) API (reference **XMP**) provides access to the communication services of CMIS and SNMP has been defined as a means of promoting portability of systems management software.

XMPP recognises that other management protocols than those listed above may also be appropriate in certain circumstances. In particular, this includes those providing RPC.

RPC is a general mechanism rather than a specific management service. However, it is certainly capable of supporting management services. This could be done either by:

- defining specific remote procedures for management services

  or

- defining the remote procedures required to provide a general object management service (such as that defined in the ORB Specification, reference **CORBA**), which would become a systems management service by being applied to objects that

represent managed resources (that is, to managed objects).

In either case, there would probably be no need for an explicit definition of management protocols.

3.9 Name Resolution

3.9.1 The Requirement

3.9.1.1 Functional Requirements

In the X/Open model of Systems Management, each instance of management information exchange takes place between an entity acting in manager role and an entity acting in agent role. In general, the information exchange relates to a managed object or to a set of managed objects. These may be directly controlled by an agent entity that resides in the same system as the manager entity so that no communication between systems is necessary. In general, however, the manager entity will communicate with an agent entity in another system. This may control the managed objects directly or may in turn communicate (acting now in manager role) with another entity that acts as agent in a third system. In a similar way, this entity may in turn communicate with another entity in a fourth system, and so on.

In order to support these modes of operation, the Name Resolution service must function in two distinct ways.

First, given the name of an entity (agent, manager or managed object), it supplies that entity's location. (In some contexts, such as that of the X/Open Management Protocol API, reference **XMP**, the term ''title'' is used to refer to the name of an agent or manager). The Name Resolution service may also supply additional information, such as the managed object class of a managed object or the protocol to be used for communications with a remote entity.

Secondly, given the name of a managed object, the Name Resolution service supplies the name and location of an entity through which it can be accessed, either directly or indirectly. In the latter case, that entity may again use the Name Resolution service to determine the name and location of a further entity. In general, access to a managed object may involve several invocations of the name resolution service, each of which gives the name of a further entity in the path between the original application or service and the managed object.

3.9.1.2 Name and Address Formats

A name (of a managed object, an agent or a manager) may take one of several forms. Those currently envisaged are:

- a ''Distinguished Name'' (which is a node of a directed acyclic graph called the naming tree)
- an SNMP Object Name (this is an administratively assigned object identifier)
- a string of characters.

A distinguished name may be globally unique or unique only within a particular pre-defined naming context. For OSI systems management, the context for the local form is the system managed object. This represents a managed system which is uniquely and unambiguously identified by its system title, as described in ISO/IEC 7498-3 (reference **NA**). The use of global and local name forms is discussed in the Management Information Model (reference **MIM**).

A name consisting of a string of characters could be structured as a Distinguished Name (for example, "workstation12.headquarters.widgetco"). However, there may also be a requirement for system administrators to be able to use nicknames as aliases to refer to entities. The conversion of these nicknames to full Distinguished Names for input to a directory service would be performed as part of the Name Resolution service.

A location consists of a network address. This can be either an OSI presentation address or an Internet address.

There will be numerous ways to name objects in the rapidly evolving distributed object world. The X/Open Systems Management Name Resolution Service must be flexible enough to support multiple naming schemes and adapt to the new naming schemes that are sure to arise as the distributed object world evolves and stabilises. In addition, to achieve interoperability, implementors' agreements will have to be forged and/or detailed mapping schemes developed.

3.9.2 Relation to Other Services

The Name Resolution service is available for use by management applications in general. However, since the Systems Management functions can provide location transparency, an application does not need to use the Name Resolution service in order to use them since it can supply them with names rather than with locations. The main users of the Name Resolution service are thus the Systems Management functions themselves and, in particular, the Dispatch/Object Messenger Service.

3.9.3 Options for the Service

3.9.3.1 Use of Directories

The Name Resolution service can be realised by one or more Directory Services, possibly supplemented by some algorithmic rules.

It would be possible to create a directory in which there was an entry for each managed object, each agent and each manager. A Directory Service that operated on such a directory would realise the complete Name Resolution Service.

This scheme has the disadvantage of requiring a large number of directory entries. This disadvantage can be overcome if the directory has entries for only some managed objects and also has entries for ''Junction'' objects. These are objects that allow the addresses of subordinate objects to be resolved without requiring a directory entry for each subordinate object. This can be achieved by naming managed objects using a hierarchical scheme based on the containment tree and choosing as Junction objects the objects near the root of that tree (typically, the managed objects representing systems).

It should be noted that, when managed object instances are placed in a directory, relating managed object instances to agents instead of directly to Network Addresses adds a level of independence for the movement of agents without affecting managed object instance directory entries.

3.9.3.2 The X/Open Directory Service

The X/Open Directory Service (reference **XDS**) could be used to operate on the directory (or directories) required for Systems Management. This service provides the means of storing information in a directory, of modifying that information and of retrieving it. The directory is assumed to have a hierarchical structure and each entry is identified by a Distinguished Name, formed by adding a Relative Distinguished Name to the Distinguished Name of its immediate superior in the hierarchy. The directory may be a distributed directory containing several Directory System Agents as envisaged by the CCITT X.500 series recommendations (reference **DCMS**).

The XDS is defined using the concepts of the X/Open OSI-Abstract-Data Manipulation service (reference **XOM**). XOM object classes are defined corresponding to the information stored in an X.500 directory. It is likely that additional object classes would have to be defined to cater for the directory requirements of Systems Management.

3.9.3.3 Global and Local Location

It may be that the full distributed directory approach imposes performance penalties and that these will be unacceptable in cases where directory entries are modified or accessed frequently. For this reason, it may be necessary to define two different directory services:

- a Global Location Service
- a Local Location Service.

The Local Location Service would give access to information local to a single system or possibly to a single local area network. Such a service could be implemented using a directory (or part of a directory) contained within a single system, efficiently and with a low response time. The information accessed could therefore include information that was modified or accessed frequently.

The Global Location Service would provide access to information that was modified or accessed relatively infrequently. Such information could include information about a number of systems connected by wide or local area networks and could be stored in a distributed directory.

3.10 Scheduling Management

3.10.1 The Requirement

Administration tasks often require that an activity be scheduled for performance at a particular time, when a particular condition arises or regularly at specified intervals. The Scheduling Management service meets this need.

For certain operations of managed objects, it is desirable to be able to control:

- the time interval over which the operation operates,
- the periodicity of the operation, and
- the conditions that trigger the operation.

The Scheduling Management service must meet these requirements.

3.10.2 Relation to Other Services

The Scheduling Management service uses the basic Systems Management Service to cause managed objects to perform operations and adds value by enabling the user application or service to specify complex conditions under which the operation is to be performed. It may also use the timing service and the data store service. It may be used by other, administration task oriented, services, including Event Management, Testing Management, Performance Management and Security Management.

3.10.3 Options for the Service

3.10.3.1 The cron Facility

The UNIX operating system and its derivatives provide a scheduling service based on the concept of a daemon (called ''*cron*'') that executes commands at regular intervals as specified by system users. This facility forms part of the X/Open CAE as specified in the XPG. It allows a user to specify that a command be scheduled at particular times on particular days in each week or month or in particular months. Any command that could be given via the user interface may be specified.

3.10.3.2 The OSI Approach

The OSI NMF has produced a Scheduling Management specification (reference **SCHM**) that provides the functionality outlined in Section 3.10.1 and has defined associated Scheduling Management object classes.

This work has drawn upon existing standards wherever possible and has in turn been input to ISO.

There are scheduling capabilities defined within ISO System Management standards to control the period of event forwarding (reference **ERM**), and to control the period of logging (reference **LC**). Scheduling capabilities are also used in the draft ISO Workload Monitoring Function Specification (reference **WM**) to control the period of monitoring and in the ISO Summarisation Function to control the period of summarisation and to trigger summarisation reporting. The syntax of these schedulers is documented in the

ISO Definition of Management Information (reference **DMI**).

The OSI NMF model defines the Scheduled Managed Object (SMO) as the managed object whose function(s) are to be scheduled. The scheduling functionality may be defined within the SMO, or outside the SMO in a separate object, called a Scheduler Object (SO). One SO may control functions in many SMOs. Furthermore, an SMO may have several functions requiring scheduling and each may have its own SO, however, a given function within an SMO can only be controlled by one SO.

The service:

- allows a repetition cycle for a schedule to be defined. The OSI NMF has identified the need to define schedule functions with either:
  - — a repetition period selected from a predefined list (namely daily, weekly and monthly cycles)

    or
  - — arbitrary repetition periods.
- for fixed period scheduling:
  - — allows the interval of operation of a scheduled function to be defined. That is, for each day within the periodicity of a scheduled function, an administrator should be able to define the number of intervals and the start and stop times of each interval.
  - — allows a user to control the triggering of a scheduled function. That is, for each day within the periodicity of a scheduled function, an administrator should be able to define triggering conditions and the maximum number of times a function may be triggered.
- for arbitrary period scheduling:
  - — allows the repetition period to be defined.
  - — allows the duration of the scheduled operation to be defined.
  - — allows a user to control the triggering of a scheduled function. An administrator should be able to defined triggering conditions and the maximum number of times a function may be triggered during each repetition period of the schedule.

3.11 Scoping

3.11.1 The Requirement

The Scoping service, in conjunction with the Filtering service, allows selection of particular sets of managed objects. The Scoping service allows managed objects to be selected by their position in the global name space. The Filtering service then allows selection of managed objects whose attributes meet specified conditions.

3.11.2 Relation to Other Services

The Scoping service is used by management applications only through other systems management services. It determines, in conjunction with the Filtering service, the managed objects (and hence the managed resources) that will be affected when another systems management service is invoked.

The basic Systems Management Service allows scoping constraints to be placed on requests to delete objects, examine and modify attributes, and invoke operations. In order to meet these constraints, the Scoping service is used by the basic Systems Management Service and by the Communications services which are also used by the basic Systems Management Service.

Other services that use the basic Systems Management Service, such as State Management and Scheduling, may also allow the user to specify scoping constraints when they are invoked.

3.11.3 Options for the Service

CMIS (reference **CMISD**) implicitly includes a Scoping service. This is described below.

Each of the CMIS M-SET, M-GET M-DELETE and M-ACTION services supports scoping and also filtering. When both types are requested, scoping is applied first. (This is discussed in Section 3.5.3 on page 22).

The scoping parameter allows a managing system to select a part of the global name space by specifying a base node and subordinate nodes to specified depth to be selected. The CMIS scoping parameters enable the following selections to be made:

- the base node alone
- the base node and all its subordinates nodes
- The base node and all its subordinates down to and including the *n*th level
- The nth level subordinates of the base node only.

The synchronisation parameter may be used to determine how an operation will be applied to a number of managed objects at once, as described in Section 3.5.3 on page 22.

3.12 Security

3.12.1 The Requirement

In the area of systems management the Security Services includes both the management of security and the security of management. In general, security can be seen as comprising:

1. management of the security functions
2. the provision of security to systems management
3. the provision of security to any user or system program that requires it.

The XSM security service deals with the first two of these. It presents both a management interface (the management of security) and a functional interface (the security of management). The services identified under the latter heading (security of management) are likely to be similar to the security services required for purposes other than systems management.

3.12.1.1 Management of Security

No comprehensive requirement in this area has yet been formulated, although the requirements for two specific services (Alarm Reporting and Security Audit Trail) are discussed in ISO/IEC 10164-7 & -8, which specify the Security Alarm Reporting (**SAR**) and Security Audit Trail (**SAT**) functions. It is clear, however, that a comprehensive requirement exists and that the management of security must incorporate flexible facilities for specification of varied security policies.

Alarm Reporting

The security management user needs to be informed of the occurrence of a number of events that are relevant to security policy. These may include misoperations in security services and mechanisms and detected attacks on or breaches of system security. The security alarm reporting service enables such events to be brought to the attention of a security management user or to be processed by an application.

Security Audit Trail

The security audit trail service is required to record security-related events in a security audit trail log. Such events may, for example, include connections, disconnections, security mechanism utilisation, management operations and usage accounting.

3.12.1.2 The Security of Management

In a distributed systems environment, access to systems services and resources cannot be enforced at a single point as is done in the kernel of a non-networked system. The distributed systems management function must have adequate security services available to it to enable it to operate in a secure manner.

The security services required for managing a distributed system can be classified as a number of distinct services.

Authentication:
: confirmation of the identity of the managing and agent systems involved in a management interaction.

Authorisation:
: confirmation that the managing system has permission to access a managed object.

Auditing:
: the recording of management activity, to show the managed object operations performed, who performed them and when they were performed.

Data confidentiality:
: the protection of the information represented by a managed object from unauthorised disclosure.

Data integrity:
: the capability to detect active attacks, that is, the capability to detect the modification of received data or specified fields of received data.

Data origin authentication:
: provides assurance that the data source is the one claimed.

Non-repudiation:
: the provision of proof that will protect against either the sender of management information falsely denying sending the data or recipient falsely denying receiving the data.

Denial of Service Prevention
: preventing a user or group of users from denying the systems' services to administration tasks (for example by creating so many processes that management processes have insufficient CPU time).

Authentication

The Communication Services must support heterogeneous operation in which different vendors may supply different authentication protocols (for example, the Communication Services may transparently negotiate the authentication protocol used.) In addition, authentication requirements for management functions, services, and applications should be specified in a protocol-independent fashion, further supporting interoperability and portability. The Communication Services must be capable of supporting a variety of authentication protocols (for example, Kerberos from MIT's Project Athena) and be capable of extending the set of protocols supported as new protocols are developed.

The Communication Services must be capable of restricting the authentication protocols used in a remote access according to site-specific policies set by an administrator. As different managed objects have different security requirements, authentication requirements must be settable on a per-object basis. It is essential that the identity of management services and applications users be confirmed before invoking operations on managed objects. It is also important that the managed object confirm its identity to the invoking process.

Authorisation

The security services should support flexibility in the access controls assigned to use of management functions and applications. It must be possible to assign access privileges to any user within the management domain, and access requirements to any managed object instance within the management domain.

Different administrative domains may wish to operate different security policies. Furthermore, a given administrative domain may wish to impose different security requirements on different managed objects. These differences are evident in which managed objects and operations can be accessed by which user or users. It should be possible to grant authorisation to users to perform specific operations on specific objects, thereby allowing those users to be administrators of the objects. Authorisation may be used to meet many aims, for example to impose security controls, licencing restrictions or device permissions.

Audit

Security Auditing provides for the collection and review of security related events for the purpose of monitoring operation on managed objects. Auditing can take a simple form of maintaining a journal of all system management activities or a more sophisticated form of recording only security-relevant events. The former allows an authorised systems administrator to track the activities of managed object users, identify access violations and those responsible for them and thus determine security exposure. The latter allows security to be imposed in a more controlled manner, minimising the cost of auditing and allowing prompt response to access violation attempts (including the generation of access violation alarms).

Audit trails are used to promote user accountability and provide a history of changes to the system.

Security auditing for the purpose of ensuring security of management is a special case of the general requirement for security auditing which is discussed in Section 3.12.1.1 on page 41.

Data Confidentiality

The function performed by systems management can directly affect the service provided by a system. In a distributed systems environment, data will be sent to and received from managed objects on a common communications medium. Therefore data to and from managed objects may have to be protected from anyone who has access to the communications medium. This will typically involve the use of encryption.

Data Integrity

In a distributed systems environment, systems management may become a target for ''hackers''. Therefore, steps may need to be taken to detect active attacks and use the Logging and Alarm services to provide records of and prompt notification of such attacks.

Data Origin Authentication

The provision of assurance that the data source is the one claimed will be important if an organisation's security policy is to allow authorised systems managers access to a managed object only from specified source systems. This will be of particular use in situations where higher levels of security are required.

Non-Repudiation

In the context of systems management, the degree of non-repudiation provided by authentication and authorisation checks, coupled with an audit trail, will often be sufficient. However, more precise techniques that provide non-repudiation for specific transactions are available (typically, based on public key encryption technology).

Denial of Service Prevention

There have in the past been some spectacularly successful instances of denial of service by ''hackers'', such as the ''Internet Worm'' of 1988, which made a large number of important systems temporarily unusable by rapidly propagating copies of itself on many of the systems attached to the Internet. Management applications are as vulnerable to such threats as any other kind of application is and the results of denial of service to management applications are potentially as serious - not least from the point of view of repairing any damage caused by an attack.

3.12.1.3 Security Standards

The OSI model includes security functions and protocols to support these are being developed. Work on security within ISO is currently at a fairly early stage. In collaboration with CCITT, ISO is defining a Security Exchange Application Service Element; this work is currently at Working Draft stage. ISO has also recently modified ACSE (reference **ACSE**) to include an authentication function and is considering the provision of confidentiality and integrity services at the Presentation layer.

Other standards bodies - most notably POSIX - are also working on the definition of secure systems. The management of security is lagging behind the provision of security. However, groups such as POSIX are now raising the issues.

De-facto standards are arising from the requirements of government organisations. Among the most influential are those defined by the US Department of Defense and administered by the National Computer Security Centre (NCSC) of the National Security Agency (NSA). The DOD and NSA are widely recognised as leading authorities on computer security both in the USA and in other countries. The NCSC published in 1985 the Trusted Computer System Evaluation Criteria, commonly known as the Orange Book (reference **TCS**). This defines the basic requirements and evaluation classes for assessing the security level of commercially produced computer systems.

More recently, a set of criteria obtained by harmonising the criteria used by the governments of France, Germany, the Netherlands and the United Kingdom, and known as ITSEC, has been published by the Commission of the European Communities (reference **ITSEC**).

3.12.1.4 Related Work

The Trusted Session Working Group (TSWG) of the Trusted Systems Interoperability Group (TSIG) - see **Glossary** - is working on profiles to promote interoperability between trusted systems from different vendors.

3.12.2 Relation to Other Services

Potentially, the security of management services will be used by all other systems management services. They may in turn use other management services, in particular the event and alarm management services.

3.12.3 Options for the Service

3.12.3.1 Alarm Reporting

ISO has defined a security alarm reporting function for OSI (reference **SAR**). This is a variant of the general OSI Alarm Reporting function described in Section 3.1.3 on page 9. It replaces many of the parameters of the Alarm Reporting function with similar ones that are specific to security. So, for example, there is a Security Alarm Type parameter that defines the type of the security alarm as one of the following:

- integrity violation
- operational violation
- physical violation
- security service or mechanism violation
- time domain violation.

As with alarms in general, security alarms are issued as notifications by managed objects using the CMIS event report service. They are forwarded by the OSI Event Reporting function (described in Section 3.4.3 on page 19, which may apply filtering and may use scheduling to turn reporting off and on.

3.12.3.2 Security Audit Trail

The OSI Security Audit Trail function is defined in (reference **SAT**). It uses the OSI Log Control function (described in Section 3.4.3 on page 19, and defines a number of security events that are to be logged. (These events will be generated by managed objects using the CMIS event report service). They comprise:

- service reports (requests for service, acceptance or rejection by services of such requests, service failures and recoveries from failure, and other events appertaining to the provision, denial or recovery of a service)

- usage reports (statistical information).

3.12.3.3 Access Control

ISO is defining objects and attributes for access control (reference **OAAC**). This covers the access control requirements associated with the establishment of management associations using ACSE, the requesting of management operations and the generation of management notifications. It defines three classes of managed object:

- access control policy, which contains access rules
- targets, which identify managed objects that may be the targets of association requests, operation requests or notifications. They are identified by name, containment, filtering, the operations that may be performed on them and the initiators that are authorised to request those operations.
- initiators, which identify authorised initiators by identity (individual, anonymous or group name), by security label or by capability.

3.12.3.4 Other Security Services

The Kerberos service defined for Project Athena (reference **KERB**) has considerable support as an Authentication service for use in Open Systems. Kerberos uses a private key encryption algorithm (such as DES). It has been successfully implemented and used, at least on an experimental basis.

Although standards work on security in general is not mature, there is one area in which mature standards do exist. This is encryption, which is generally used to provide data confidentiality and data integrity and which to a certain extent also provides data origin authentication.

The two most widely used standard encryption algorithms are the DES algorithm (reference **DES**) and the RSA public key algorithm which is described in (reference **DSPC**) and in Annex C of CCITT Recommendation X.509 (reference **DAF**). The RSA public key algorithm can also be used to provide data origin authentication and non-repudiation (digital signature).

Although these algorithms are standardised, organisations may not wish to use them or may in some countries be prevented from doing so by government restrictions. Security services must therefore be sufficiently flexible to allow a choice in the encryption algorithms used.

3.13 State Management

3.13.1 The Requirement

For the performance of many administration tasks, there is a need to monitor the operability and usage of the system resources represented by managed objects. Monitoring capabilities should include the ability to receive notifications when operability or usage states change as well as the ability to inquire at any time what the operability and usage states are. In addition to monitoring, it is also sometimes necessary to control the availability of managed resources (for example, to take a piece of equipment off-line for re-configuration or repair). These facilities are provided by the State Management service.

3.13.2 Relation to Other Services

The State Management service may be used by services such as Event Management, Alarm Management, Scheduling Management and Testing Management that use control objects in order to monitor the operability and usage of those objects and in order to allow those objects to be administratively barred from use. This last facility provides a means of suspending the operation of the services that use the objects.

The State Management service uses the basic Systems Management service to examine and modify state and status attributes.

3.13.3 Options for the Service

An appropriate service is defined in ISO/IEC 10164-2 (reference **STM**). Although defined for OSI management, the service can be made applicable to Open Systems management in general. It is summarised below.

While each type of system resource has a different set of characteristics, reflected in the attributes of the corresponding class of managed objects, there are common characteristics of operability, usage and availability which are reflected in three state attributes which any managed object may have. These are the Operational State, the Usage State and the Administrative State. In addition, there are a number of common conditions, which apply to objects' operability, usage or availability states and which are reflected in status attributes.

The operational state describes the operability of the managed resource represented by a managed object. It reflects factors beyond the control of administrators or management applications. Its possible values are:

Disabled:
: The managed resource is totally inoperable (for example, because it is defective or not installed or because some other resource on which it depends is not available) and is unable to provide service.

Enabled:
: The managed resource is partially or fully operable and available for use.

The usage state reflects the extent to which the managed resource is actually being used. Its possible values are:

Idle:
The managed resource is not currently in use.

Active:
The managed resource is in use and has sufficient spare capacity to provide for additional users.

Busy:
The managed resource is in use but does not have sufficient spare capacity to provide for additional users.

The administrative state reflects factors within the control of administrators or management applications. Its possible values are:

Locked:
The managed resource is administratively prohibited from use.

Shutting down:
Use of the managed resource is administratively permitted to existing instances of use only.

Unlocked:
Normal use of the managed resource is administratively permitted.

The status attributes qualify the three main state attributes described above. Each status attribute except the ''unknown'' attribute is set-valued and represents a set of conditions that may apply to the resource. The presence of a particular value in a status attribute may imply a particular value for the resource's operational, usage or administrative state. The following status attributes are defined.

Alarm Status:
The title of this attribute is self explanatory. Its possible values are: under repair, critical, major, minor, alarm outstanding.

Procedural Status
This attribute is supported by managed objects that represent some procedure (such as a test process) which progresses through a sequence of phases. Its possible values are: initialisation required, not initialised, initialising, reporting, terminating.

Availability Status:
This attribute reflects the availability of the resource in terms of the following values: in test, failed, power off, off line, off duty (ie. made inactive by an internal control process), dependency (the resource can not operate because another resource on which it depends is unavailable), degraded, not installed, log full.

Control Status:
This attribute reflects what the resource is currently being controlled to do in terms of the following values: subject to test, part of services locked, reserved for test, suspended.

Standby Status:
The title of this attribute is self explanatory. Its possible values are: hot standby, cold standby, providing service.

Unknown Status:

This is a Boolean attribute that indicates whether the state of the resource represented by the managed object is known. When the value of this attribute is true, the values of the state attributes may not reflect the actual state of the resource.

3.14 Testing Management

3.14.1 The Requirement

Testing is the critical examination or trial of the capabilities or status of managed resources. It is an integral part of any Systems Management philosophy and is used in many administration tasks at every phase of the service or equipment lifecycle.

For example, prior to activating a service or equipment, testing may be used to confirm that it can operate according to specifications. When faults are detected, tests may be used as part of Fault Management to verify that a failure has occurred, as well as to perform analysis to isolate the failing component. Upon completion of a repair action, testing may be used to verify that the repair has actually corrected the fault condition. In Performance Management, tests may be used to monitor the health of equipment as well as to ensure service quality is maintained.

The Testing Management service provides the capability of requesting that tests be performed on managed resources to determine whether those resources can perform their functions or to assist in the diagnosis and isolation of faults. It handles test events, keeping them separate from operational events, and records them in a test log. It should cater for both single step and multi-step tests and for tests that are repetitive in nature.

Note that the definition of the Testing Management service does not include definitions of the tests that are performed. These form part of the definition of the behaviour of the managed objects that represent the resources to be tested. Some managed object definitions may not include tests; the resources that they represent will then not be testable using systems management services.

Testing Management should not be confused with conformance testing, which is done to establish the level to which an implementation conforms to X/Open APIs and international standards.

3.14.2 Relation to Other Services

Testing Management uses the basic Systems Management Service to enable applications to interact with managed objects. It may use Scheduling Management to schedule test actions, Event Management to control logging and reporting of events and Security Management to ensure that tests do not compromise systems' security.

3.14.3 Options for the Service

The OSI NMF has produced Testing Management specifications (reference **TESM**) and associated object class definitions (reference **TESD**), drawing on ISO work on Testing Management, as described in references **TM** and **CDTC**. However, the ISO work has not yet reached IS status. This work is summarised below.

3.14.3.1 Overall Approach

As defined by the OSI NMF, the Testing Management Service enables one system (the Test Manager) to request another system (the Test Performer) to invoke, control, monitor, and obtain results of tests on managed resources. It includes definitions of messages to be exchanged by the Test Manager and the Test Performer. It also includes definitions of objects that do not directly correspond to managed resources but which affect the operation of the Testing Management service. These include Test Definition Objects (objects that define the testing capability of a managed object), Test Control Objects (objects that define a specific instance of a test) and Initial Value Managed Objects (objects that represent test parameters).

Tests may be disruptive in nature, hence access control mechanisms must be established to ensure that only authorised users can invoke a test and that they are invoked only at appropriate times. In instances where the performance of a test affects the resource(s) being tested, Testing Management requires that the impact of the test on resource being tested should be formally specified; for example, if the OSI NMF OBJECT Template is used, this would be done in the BEHAVIOR clause.

3.14.3.2 Mode of Operation

A test is typically composed of a test cycle that will initialise the testing environment (preamble phase), perform the active test, and restore the environment to the pre-test state (postamble phase).

In general, tests can be executed on demand or scheduled for subsequent execution. Testing may be Asynchronous or Synchronous. A Synchronous test is a test that returns its results in the response to the test initiation request, while an Asynchronous test is a test in which the results are returned as a separate message. Current OSI NMF work on Testing Management has concentrated exclusively on an Asynchronous mode of operation. This mode of operation will meet current X/Open requirements.

3.14.3.3 Test Inputs

To perform a test, a Test Performer requires a number of inputs. These include the following:

Selection of Test:
The Test Manager must be able to specify the test to be performed.

Subject of test:
the Test Manager must be able to specify the resource to be tested.

Test Start Trigger:
the Test Manager must be able to specify when the test will be performed.

Priming Values and Limits:
the Test Manager must be able to specify parameter required to perform the test.

Execution Priority:
the Test Manager must be able to assign a priority to the execution of a test.

3.14.3.4 Starting Tests

Tests may be started through each of the following methods:

1. when the Test Performer receives a test request message from a Test Manager
2. at a preset time
3. periodically (for example, daily at 1:00 AM)
4. wherever a preset condition is met (for example, when a threshold is exceeded)
5. when the Test Performer receives a specified event report, or a specified event within the system occurs.

When a test is performed, a number of parameters may be used. These may be included in test request messages transmitted by the Test Manager, or stored at the Test Performer as priming information. In the latter case, the test initiation message will contain a pointer that can be used by the Test Performer to retrieve the stored priming information.

3.14.3.5 Test Control Capability

Tests may be controlled either explicitly or implicitly. A Test Manager may issue explicit commands to:

- terminate a test and return any final results to a specified destination
- suspend a test
- resume a suspended test
- modify the parameters of a test in progress
- abort a test and discard any pending results.

An implicit control is initiated upon the occurrence of a predefined condition or event specified by the Test Manager in the test request message and/or test parameters modification message. The Test Performer may initiate any of the following actions upon detection of the corresponding condition:

- a test may be terminated when a specified condition is met (eg normal completion of test, preset time, threshold crossed)
- a test may be suspended when a specified condition is met
- a test may be resumed upon detection of a condition
- a test may be aborted when an abnormal condition is detected.

Some tests may be comprised of 2 or more steps that are externally visible, in which case controls may apply to individual test steps.

3.14.3.6 Test Output

A Test Performer may produce three types of output:

Results:

There are two types of result: final and intermediate. The Test Manager may specify the content of the test results (for example, summary versus detailed). Results may be sent to the Test Manager, sent to a destination address specified by the Test Manager or the Test Performer may be polled for the results. Also, the Test Manager must be able to suspend and later resume the generation of results.

Status:

The Test Performer may use the test status to indicate such things as the correct functioning of a test, milestones reached in the test procedure, and estimated time to completion. Status notification may be requested, triggered by an event, or scheduled. Status notifications may be sent to the Test Manager, or sent to a destination address specified by the Test Manager, or the Test Performer may be polled for the results. The Test Manager may be able to select the content type.

Errors:

This message allows a Test Performer to report errors related to a test that has been requested. Error messages may be sent to the Test Manager or to a destination address specified by the Test Manager. Error messages may be sent in response to a test request or at a later time. The reporting or errors may be suspended (and error messages discarded) and later resumed.

3.15 Timing

3.15.1 The Requirement

The timing service is required in order to synchronise the local system's clock with Universal Time Coordinated, as well as with the clocks of other computers connected to the system.

Many aspects of management require the time to be accurately known and recorded. For instance, the value of the information contained in system logs is dependent on the reliability of the timestamp information associated with it. This is especially so in a distributed management system where events may be reported from different physical systems throughout the network. In order to make sense of the sequence events, there must be absolute confidence in the accuracy of the recorded times. Similarly, many distributed authentication technologies rely on accurate timestamping in order to prevent security breaches such as the replaying of message sequences.

For the purposes of Testing Management (reference **TESM**) and hence also of Scheduling Management (reference **SCHM**), the OSI NMF has identified a requirement for resolution of time up to the granularity of one nanosecond. Note that this is simply the resolution used in recording times and does not imply that times need be measured to that degree of accuracy. In practical systems today, an accuracy of milliseconds or tens of milliseconds can be considered to be ''state of the art''.

The requirements of management systems for a distributed timing service are no different to those of many other distributed applications. In addition to using the service, the management system will also be required to manage it, but this is beyond the scope of this particular document.

3.15.2 Relation to Other Services

The timing service may be used by any other Systems Management service and by applications.

3.15.3 Options for the Service

There are two distributed timing services that should be considered as candidates: the Internet Network Time Protocol (reference **NTP**), and the Distributed Time Service (DTS) of DCE.

Both services provide the application with an approximation of coordinated universal time. They use very different algorithms and communications protocols in order to do so. From the point of view of the application, however, the main difference is that, while NTP simply provides an approximate value of UTC, DTS provides an interval within which the true value is guaranteed to lie.

It is desirable for X/Open to recommend a standard distributed timing service for use not only in distributed systems management but also by other applications and systems programs. The choice can not be made on the basis of the systems management requirement alone but must take other requirements into account. Neither NTP nor DTS has to date had significant commercial use (NTP is the more widely used but is not commercially oriented; DTS is commercially oriented but has not been used

extensively). X/Open therefore considers it premature to make a definitive choice at the present time.

Glossary

API
Application Programming Interface

ASN.1
Abstract Syntax Notation 1

ACSE
Association Control Service Element

ACSEP
Association Control Service Element Protocol

ANSI
American National Standards Institute

BRM
Basic Reference Model

CAE
Common Applications Environment

CCITT
The International Telephone and Telegraph Consultative Committee

CM-API
The Consolidated Management API Specification

CMIP
The OSI Common Management Information Protocol

CMIS
The OSI Common Management Information Service

CMISE
Common Management Information Service Element

CMISP
Common Management Information Services and Protocols

CMOT
CMIS Over TCP / IP

DCE
The Distributed Computing Environment (of OSF)

DES
The Data Encryption Standard

DISP
Draft International Standardised Profile

DoD
The (United States of America's) Department of Defense

DTS
The Distributed Time Service (of OSF's DCE)

ECAM
Event, Configuration and Alarm Management

GDMO
The ISO Guidelines for the Definition of Managed Objects

IAB
The Internet Advisory Board

IEC
The International Electrotechnical Commission

IEEE
The Institute of Electrical and Electronics Engineers

IETF
The Internet Engineering Task Force

IP
The Internet Protocol

IPS
The Internet Protocol Suite

ISO
The International Organisation for Standardisation

ISP
International Standardised Profile

JTC1
Joint Technical Committee 1 (of ISO and IEC)

MIB
Management Information Base

MIT
Massachusetts Institute of Technology

NCSC
The (United States of America's) National Computer Security Centre

NMF
(See OSI NMF)

NSA
The (United States of America's) National Security Agency

NTP
The (Internet) Network Time Protocol

ODM
Object Data Management

OMG
The Object Management Group

OODB
Object Oriented Data Base

OODBTG
The ANSI Object-Oriented Database Task Group

OODM
Object Oriented Data Model

ORB
Object Request Broker

OSF
The Open Software Foundation

OSI
Open Systems Interconnection (based on the ISO reference model)

OSI NMF
The OSI Network Management Forum

PDU
Protocol Data Unit (usually, in the context of OSI)

POSIX
The Portable Operating System Interface defined by the IEEE Computer Society

RDN
Relative Distinguished Name

RFC
Request for Comments

ROSE
Remote Operations Service Element

RPC
Remote Procedure Call

RSA
The encryption algorithm described by Rivest, Shamir and Adleman

SA
Security Architecture

SMO
Scheduled Managed Object

SNMP
The Simple Network Management Protocol (of IPS)

SO
Scheduler Object

SQL
Structured Query Language

TCP
The Transmission Control Protocol (of IPS)

TCSEC
Trusted Computer System Evaluation Criteria (of the NCSC)

TSIG
Trusted Systems Interoperability Group
This Group was formed following an initiative of Secureware, a US-based company specialising in "plug-in" security extensions for UNIX. Their work has been adopted by OSF and will probably be part of the next OSF/1 release, and by DIA (US Government Defense Information Agency) as a building block for their CMW (Compartmentalised Mode Workstation) work which aims to provide a secure windows enviroment. Part of this work has resulted in the definition of the MAXSIX security extensions to TCP/IP.

TSWG
Trusted Session Working Group - a sub-group of TSIG.

UTC
Co-ordinated Universal Time

X.400
The X/Open API to Electronic Mail

XDS
The X/Open Directory Service

XMHS
Message Handling Service - see **X.400**

XMP
The X/Open Systems Management: Management Protocol API Specification

XMPP
The X/Open Systems Management: Management Protocol Profiles Specification

XOM
The X/Open OSI-Abstract-Data Manipulation Service

XPG
The X/Open Portability Guide

XSM
The X/Open Systems Management Programme

Index