Working Implementation Agreements for Open Systems Interconnection Protocols: Part 18 - Network Management

Output from the June 1994 Open Systems Environment Implementors' Workshop (OIW)

SIG Chair: **Paul Brusil, The Mitre Corporation**
Editing Changes: **Kellie Beall, NIST**

Foreword

This part of the Working Implementation Agreements was prepared by the Network Management Special Interest Group (NMSIG) of the Open Systems Environment Implementors' Workshop (OIW). See Part 1 – Workshop Policies and Procedures in the "Draft Working Implementation Agreements Document" for the workshop charter.

Text in this part has been approved by the Plenary of the above–mentioned Workshop. This part replaces the previously existing chapter on this subject.

To highlight textual changes since the last Workshop output, additions to the text in this part are marked with shading; deleted text is left in but marked with strikeouts.

Table of Contents

Annex E (informative)

List of Figures

18 Network Management

0 Introduction

(Refer to the Stable Implementation Agreements Document.)

1 Scope

(Refer to the Stable Implementation Agreements Document.)

2 Normative References

The following documents are referenced in the statements of the agreements relating to OSI sytems management.

[AMF] ISO/IEC CD 10164–10, Information Technology – Open Systems Interconnection – Systems Management – Part 10: Accounting Meter Function, ISO/IEC JTC1/SC21 N4958, 4 July 1990. (Document name has been changed to "Usage Metering Function". See [UMF].)

[AMWD] Information Processing Systems – Open Systems Interconnection – Accounting Management Working Document (Fourth Version), ISO/IEC JTC1/SC21, May 30, 1990.

[AOM12] DISP 11183–2, Information Technology – International Standardized Profiles AOMnn OSI Management – Management Communications Protocols – Part 2: AOM12 – Enhanced Management Communications, September 1991.

[ARF] ISO/IEC IS 10164–4, Information Technology – Open Systems Interconnection – Systems Management – Part 4: Alarm Reporting Function, ISO/IEC JTC1/SC21 N6359, August 19, 1991.

[ARR] ISO/IEC IS 10164–3, Information Technology – Open Systems Interconnection – Systems Management – Part 3: Attributes for Representing Relationships, ISO/IEC JTC1/SC21 N5186, September 1991.

[ATSS] ISO/IEC DIS 9646–2, Information Technology – Open Systems Interconnection – Conformance Testing Methodology and Framework – Part 2: Abstract Test Suite Specification, ISO/IEC JTC1/SC21 N5867, 10 April 1991.

[CDTC] ISO/IEC CD 10164–cdt, Information Processing Systems – Open Systems Interconnection – Systems Management – Part cdt: Confidence and Diagnostic Test Classes, ISO/IEC JTC1/SC21 N1394, December 1991.

[CMO] Information Processing Systems – Open Systems Interconnection – Working Draft of the Configuration Management Overview, ISO/IEC JTC1/SC21 N3311, 16 January 1989.

[DMI] ISO/IEC IS 10165–2, Information Technology – Open Systems Interconnection – Structure of Management Information – Part 2: Definition of Management Information, ISO/IEC JTC1/SC21 N6363, August 1991.

[ENSCON] Forum 025, The "Ensemble" Concepts and Format, Issue 1.0, Network Management Forum, July 1992.

[ERMF] ISO/IEC IS 10164–5, Information Technology – Open Systems Interconnection – Systems Management – Part 5: Event Report Management Function, ISO/IEC JTC1/SC21 N6360, August 1991.

[FMWD] Information Processing Systems – Open Systems Interconnection – Systems Management – Fault Management Working Document, ISO/IEC JTC1/SC21 N4077, December 1989.

[GDMO] ISO/IEC IS 10165–4, Information Technology – Open Systems Interconnection – Structure of Management Information – Part 4: Guidelines for the Definition of Managed Objects, ISO/IEC JTC1/SC21 N6309, July 30, 1991.

[GULS–1] ISO/IEC CD 11586–1, Information Technology – Open Systems Interconnection – Generic Upper Layers Security – Part 1: Overview, Models and Notation, ISO/IEC JTC1/SC21 N8182, August 9, 1993.

[GULS–2] ISO/IEC CD 11586–2, Information Technology – Open Systems Interconnection – Generic Upper Layers Security – Part 2: Security Exchange Service Element (SESE) Service Definition, ISO/IEC JTC1/SC21 N8183, August 9, 1993.

[GULS–3] ISO/IEC CD 11586–3, Information Technology – Open Systems Interconnection – Generic Upper Layers Security – Part 3: Security Exchange Service Element (SESE) Protocol Specification, ISO/IEC JTC1/SC21 N8184, August 9, 1993.

[GULS–4] ISO/IEC CD 11586–4, Information Technology – Open Systems Interconnection – Generic Upper Layers Security – Part 4: Protecting Transfer Syntax Specification, ISO/IEC JTC1/SC21 N8185, August 9, 1993.

[IIMCIMIBTRANS] ISO/CCITT and Internet Management Coexistence (IIMC): Translation of Internet MIBs to ISO/CCITT GDMO MIBs, Forum 026, Issue 1.0, November 1993.

[IIMCMIB–II] ISO/CCITT and Internet Management Coexistence (IIMC): Translation of Internet MIB–II (RFC1213) to ISO/CCITT GDMO MIB, Forum 029, Issue 1.0, November 1993.

[IIMCOMIBTRANS] ISO/CCITT and Internet Management Coexistence (IIMC): Translation of ISO/CCITT GDMO MIBs to Internet MIBs, Forum 030, Issue 1.0, November 1993.

[IIMCPROXY] ISO/CCITT and Internet Management Coexistence (IIMC): ISO/CCITT to Internet Management Proxy, Forum 028, Issue 1.0, November 1993.

[IIMCSEC] ISO/CCITT and Internet Management Coexistence (IIMC): ISO/CCITT to Internet Management Security, Forum 027, Issue 1.0, November 1993.

[LCF] ISO/IEC IS 10164–6, Information Technology – Open Systems Interconnection – Systems Management – Part 6: Log Control Function, ISO/IEC JTC1/SC21 N6361, June 1991.

[MICS] ISO/IEC CD 10165–6, Information Technology – Open Systems Interconnection – Structure of Management Information – Part 6: Requirements and Guidelines for Implementation Conformance Statement Proformas Associated with Management Information, ISO/IEC JTC1/SC21, 10 April 1992.

[MIM] ISO/IEC IS 10165–1, Information Technology – Open Systems Interconnection – Management Information Services – Structure of Management Information – Part 1: Management Information Model, ISO/IEC JTC1/SC21 N6351, June 1991.

[MOA] ISO/IEC IS 10164–11, Information Technology – Open Systems Interconnection – Systems Management – Part 11: Metric Objects and Attributes, ISO/IEC JTC1/SC21 N7533, February 1993. (Previously entitled "Workload Monitoring Function". See [WMF].)

[MTAMIBTRANS] RFC 1566 MADMAN MTA MIB GDMO translation, xxx

[OAAC] ISO/IEC CD 10164–9, Information Technology – Open Systems Interconnection – Systems Management – Part 9: Objects and Attributes for Access Control, ISO/IEC JTC1/SC21, February 1992.

[OMF] ISO/IEC IS 10164–1, Information Technology – Open Systems Interconnection – Systems Management – Part 1: Object Management Function, ISO/IEC JTC1/SC21 N5184, September 1991.

[OP1LIB] Forum 006, Forum Library – Volume 4: OMNIPoint 1 Definitions, Issue 1.0, Network Management Forum, August 1992.

[PMWD] Information Processing Systems – Open Systems Interconnection – Performance Management Working Document (Seventh Draft), ISO/IEC JTC1/SC21 N6306, June 24, 1991.

[SARF] ISO/IEC IS 10164–7, Information Technology – Open Systems Interconnection – Systems Management – Part 7: Security Alarm Reporting Function, July 1991.

[SATF] ISO/IEC DIS 10164–8, Information Technology – Open Systems Interconnection – Systems Management – Part 8: Security Audit Trail Function, ISO/IEC JTC1/SC21 N7039, June 1992.

[SF] ISO/IEC CD 10164–13.2, Information Technology – Open Systems Interconnection – Systems Management – Part 13: Summarization Function, ISO/IEC JTC1/SC21 N6485, November 12, 1991.

[SMWD] Information Processing Systems – Open Systems Interconnection – Systems Management – OSI Security Management Working Document – 7th Draft, ISO/IEC JTC1/SC21 N4091, 15 November 1989.

[STMF] ISO/IEC IS 10164–2, Information Technology – Open Systems Interconnection – Systems Management – Part 2: State Management Function, ISO/IEC JTC1/SC21 N5185, September 1991.

[TMF] ISO/IEC DIS 10164–12, Information Processing Systems – Open Systems Interconnection – Systems Management – Part 12: Test Management Function, ISO/IEC JTC1/SC21 N6558, November 1991.

[UMF] ISO/IEC 2ndDIS 10164–10, Information Technology – Open Systems Interconnection – Systems Management – Part 10: Usage Metering Function, ISO/IEC JTC1/SC21 N????, October 1993. (Previously entitled "Accounting Meter Function". See [AMF].)

[WMF] ISO/IEC DIS 10164–11, Information Technology – Open Systems Interconnection – Systems Management – Part 11: Workload Monitoring Function, ISO/IEC JTC1/SC21 N6677, February 3, 1992. (Document name has been changed to "Metric Objects and Attributes". See [MOA].)

3 Status

The following clauses were moved into the Stable Agreements in June 1990:

0 INTRODUCTION

2 NORMATIVE REFERENCES (i.e., only those relevant to the Stable Agreements)

6 MANAGEMENT COMMUNICATIONS

6.2 General Agreements on Users of CMIS

6.3 Specific Agreements on Users of CMIS

6.4 Specific Agreements on CMIP

The following clauses were moved to the Stable Agreements in December 1990:

1 SCOPE

1.1 Phased Approach

1.1.1 Alignment With Evolving Standards

1.1.2 Definition of Phase 1

1.1.3 Future Phases

2 NORMATIVE REFERENCES (i.e., only those relevant to the newly added Stable Agreements)

5 MANAGEMENT FUNCTIONS AND SERVICES

5.1 General Agreements

5.2 Object Management Function Agreements

5.3 State Management Function Agreements

5.4 Attributes For Representing Relationships Agreements

5.5 Alarm Reporting Function Agreements

5.6 Event Report Management Function Agreements

6 MANAGEMENT COMMUNICATIONS

6.1 Association Policies

7 MANAGEMENT INFORMATION

7.1 The Information Model

7.2 Principles of Naming

7.3 Guidelines for the Definition of Management Information

The following clause was added to the Stable Agreements in March 1991:

6 MANAGEMENT COMMUNICATIONS

6.5 Services Required by CMIP (added as subclause 13.7 of part 5, Upper Layer Agreements)

The following clauses were added to the Stable Agreements in September 1991:

6.1.3 Security Aspects of Associations

6.2.4 CMIS Subsets

6.4.5 Parameters

6.4.6 Access Control Parameter

8 CONFORMANCE

8.1 Introduction

8.2 General Requirements of Conformance

8.3 Specific Conformance Categories

8.3.1 Management Communication Categories

8.3.3 Management Information Conformance Category

8.3.3.1 MOCS Proforma

8.3.4 Management Application Contexts

The following clauses were added to the Stable Agreements in December 1991:

5.7 Log Control Function Agreements

5.8 Security Alarm Reporting Function Agreements

8.3.2 Management Functions and Services Conformance Categories

8.3.2.1 General Management Capabilities Conformance Category

8.3.2.2 Alarm Reporting and State Management Capabilities Conformance Category

8.3.2.3 Alarm Reporting Capabilities Conformance Category

8.3.2.4 General Event Report Management Conformance Category

8.3.2.5 General Log Control Conformance Category

The following clauses were added to the Stable Agreements in June 1992:

5.9 Security Audit Trail Function Agreements

6.4.7 Action Error Info

6.5 Services Required by CMIP

6.5.1 P–DATA Encoding

6.6 CMIP PICS

ANNEX A Management Information Library

ANNEX A.4 Harmonized Library

ANNEX A.5 OIW NMSIG IVMO Definitions

ANNEX B NMSIG Object Identifiers

ANNEX B.1 Introduction

ANNEX B.2 Harmonized MIL Object Identifiers

ANNEX B.3 Phase 1 MIL Object Identifiers

The following clause was added to the Stable Agreements in September 1992:

ANNEX C MOCS Proforma

Text was added to the following clause of the Stable Agreements in December 1992:

5.7.1 General Agreements

The following clauses are planned to be added to the Stable Agreements in September 1993:

8.4 Demonstration of Conformance

8.4.1 Management Communication

8.4.2 Management Functions and Services

8.4.3 Management Information

The following clauses were added to the Stable Agreements in September 1993:

8.4 Demonstration of Conformance

8.4.1 Management Communication

8.4.2 Management Functions and Services

8.4.3 Management Information

ANNEX D.2 Systems Management for OSI Transport and Network Layers Ensemble

The following clauses were added to the Stable Agreements in December 1993:

6.1.3 Security Aspects of Associations

10 Management Coexistence and Interworking

10.1 Internet MIB Translation

10.2 ISO/CCITT to Internet Management Proxy

ANNEX E Translated Management Information Libraries

ANNEX E.1 Introduction

ANNEX E.2 MIBs Translated By Organizations Other Than OIW

4 Errata

(Refer to the Stable Implementation Agreements Document.)

5 Management Functions and Services

ISO has partitioned network management into five Specific Management Functional Areas (SMFAs) as a convenience for developing requirements particular to configuration management (CM), fault management (FM), performance management (PM), security management (SM), and accounting management (AM). These requirements are specified in five separate SMFA standards ([CMO], [FMWD], [SMWD], [AMWD], and [PMWD]). Since the SMFAs have overlapping requirements, management functions and management information applicable to one SMFA are often applicable to other SMFAs. Therefore, the SMFAs point to separate standards that contain the management functions needed to satisfy particular requirements.

This set of management functions is referred to as the System Management Functions (SMFs). They provide a generic platform of common network management capabilities available to any management application. For example, the event report management function [ERMF] may be used to report events to satisfy FM, PM, AM, and SM requirements. The log control function [LCF] may be used to satisfy both FM and SM requirements.

The following schematic (figure 1) depicts the functional hierarchy of SMFs and SMFAs. There are currently seven SMF International Standards: Object Management [OMF], State Management [STMF], Attributes For Representing Relationships [ARR], Alarm Reporting [ARF], Event Report Management [ERMF], Log Control [LCF], and Security Alarm Reporting [SARF]. These SMFs provide much of the network management capabilities needed by CM and FM. When additional requirements are identified in other SMFAs, additional SMFs may be developed. Security Audit Trail [SATF] is a Draft International Standard. Committee drafts are currently in progress for the following additional SMFs: Objects and Attributes For Access Control [OAAC], Usage Metering [UMF], and Metric Objects and Attributes [MOA]. Working drafts are currently in progress for the following additional SMFs: Confidence and Diagnostic Testing (consisting of two documents, one specifying a Test Management Function [TMF], and the other defining related management support objects classes and attributes [CDTC]), and Summarization [SF].

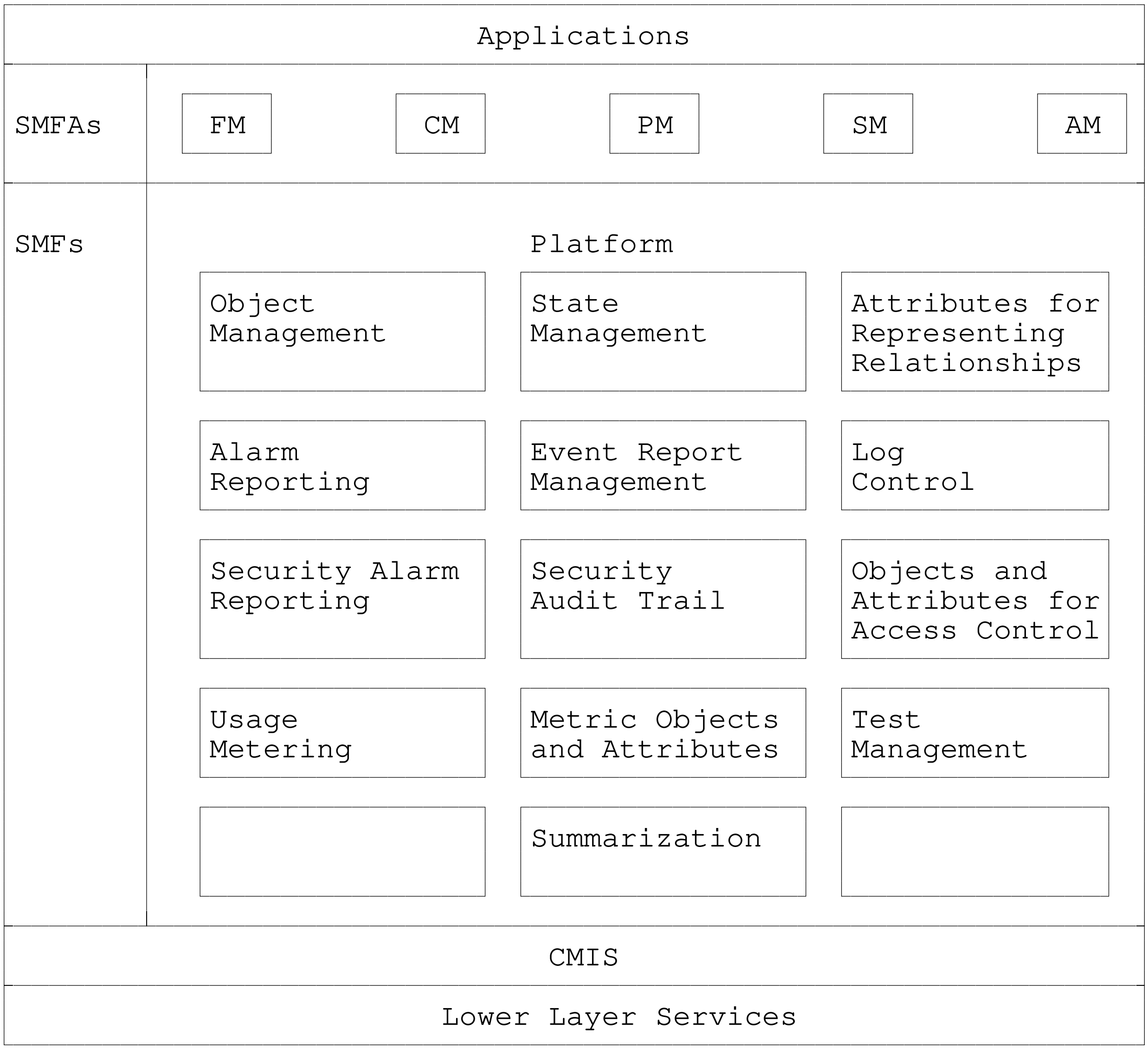

Figure 1 - Functional hierarchy of SMFs and SMFAs

5.1 General Agreements

(Refer to the Stable Implementation Agreements Document.)

5.2 Object Management Function Agreements

(Refer to the Stable Implementation Agreements Document.)

5.3 State Management Function Agreements

(Refer to the Stable Implementation Agreements Document.)

5.4 Attributes For Representing Relationships Agreements

(Refer to the Stable Implementation Agreements Document.)

5.5 Alarm Reporting Function Agreements

(Refer to the Stable Implementation Agreements Document.)

5.6 Event Report Management Function Agreements

(Refer to the Stable Implementation Agreements Document.)

5.7 Log Control Function Agreements

(Refer to the Stable Implementation Agreements Document.)

5.8 Security Alarm Reporting Function Agreements

(Refer to the Stable Implementation Agreements Document and online profile document referenced in editor's not below.)

Note: **[**The agreements in this clause are contained in the Security Alarm Reporting profile. The text for this profile is available on-line by anonymous ftp from the OIW document store. The document can be retrieved as follows: ftp to nemo.ncsl.nist.gov [129.6.58.136]; login as "anonymous" with password "guest"; cd to pub/oiw/agreements; retrieve the file "readme.sar" and read that file for instructions as to which files to retrieve.**]**

5.9 Security Audit Trail Function Agreements

(Refer to the Stable Implementation Agreements Document.)

5.10 Objects and Attributes for Access Control Agreements

5.10.1 Introduction

This subclause provides agreements pertinent to Objects and Attributes for Access Control defined by [OAAC].

Objects and Attributes for Access Control:

* defines a conceptual model for the administration of managed object access control; and

* provides the Access Control Descriptor, Target Access Control Information, and Authorized Initiators management support object classes to facilitate object access control.

There is a need to prevent unauthorized access to management resources at various levels:

* management notifications must not be sent to unauthorized recipients,

* unauthorized initiators must not have access to management operations, and

* management information must be protected from unintended disclosure.

This function defines mechanisms for controlling access to management associations and operations.

Objects and Attributes for Access Control makes use of the following management support objects:

accessControlDescriptor,
targetACI, and
authorisedInitiators.

Objects and Attributes for Access Control makes use of the following attributes, in addition to those attributes defined for the object class top:

accessControlDomainNames,
accessControlPolicyName,
ACDName,
ACDRules,
ACIOperations,
ACIRules,
AIName,
defaultRules,
globalRules,

initiatorACI,
initiatorList,
MIOperations,
MIRules,
objectList, and
targetACIName.

Objects and Attributes for Access Control makes use of the following notification types:

objectCreation,
objectDeletion,
attributeChange, and
securityServiceOrMechanismViolation.

5.11 Usage Metering Function Agreements

Editor's Note: **[**The material in this clause is out-of-date. The clause will be updated when the OIW NMSIG has the resources available to renew activity regarding its contents.**]**

5.11.1 Introduction

This subclause provides agreements pertinent to the Accounting Meter Function defined by [AMF].

The Accounting Meter Function:

* defines a conceptual model for collecting, recording, and reporting accounting information;
* provides a set of management information pertinent to account metering;
* provides the Accounting Record, Accounting Meter Control, and Accounting Meter Data management support object classes;
* provides a number of notifications regarding account metering; and
* provides a set of services to effect account metering.

In general, any accounting activity begins by monitoring resources to identify who is using them and to what extent they are being used. An accounting meter records the use of a resource in the form of accounting records or logs. Accounting meters record information such as:

* the identity of the user and the resource,
* the quality and type of service requested and provided,
* the usage start time and current time,
* the current state of usage (running or suspended), and
* the unit of measurement and number of units consumed.

The Accounting Meter Function defines the following management support objects:

accountingMeterControlObject,
accountingMeterDataObject, and
accountingRecordObject.

The Accounting Meter Function defines the following attributes:

controlObjectReference,
dataObjectReference,
dataObjectState,
meterInfo,
notificationCause,
notificationTime,
recordingTrigger,
reportingTrigger,
requesterId,
responderId,
resourceName,
serviceProvided,
serviceRequested,
subscriberId,
unitsOfUsage,
usageMeterTime, and
usageStartTime.

The Accounting Meter Function defines the following notification types:

accountingStarted,
accountingSuspended,
accountingResumed,
accountingRecord, and
accountingInfoLost.

The Accounting Meter Function defines the following actions:

startMetering,
suspendMetering, and
resumeMetering.

5.12 Metric Objects and Attributes Agreements

Note: **[**The OIW NMSIG is participating in the development of ISPs for Metric Objects and Attributes (ISO/IEC 10164-11). ISPs for Metric Objects and Attributes are numbered in the AOM252x series.

The latest drafts of this activity are available from nemo.ncsl.nist.gov via anonymous FTP. Documents can be retrieved as follows:

FTP to nemo.ncsl.nist.gov [129.6.58.136];
login as "anonymous" with password "guest";
cd pub/oiw/agreements;
retrieve the file "perfmgmt.readme";
read that file for instructions as to which further files to retrieve

Since the ISP activity in this area is relatively immature, these drafts are subject to change, especially with regard to base standard ICS proforma style.**]**

Editor's Note: **[**The material in this clause is out-of-date. The clause will be updated when the OIW NMSIG has the resources available to renew activity regarding its contents.**]**

5.12.1 Introduction

This subclause provides agreements pertinent to the Workload Monitoring Function defined by [WMF].

The Workload Monitoring Function:

* defines three conceptual models for the monitoring of system resources;

* provides the Gauge Monitor Metric and Mean Monitor Metric management support objects to facilitate workload monitoring;

* provides a number of notifications regarding workload monitoring; and

* provides a set of services to effect workload monitoring.

Three conceptual models are defined within the Workload Monitoring Function.

* Utilization Model: Provides monitoring of instantaneous use of an OSI resource.

* Rejection Rate Model: Provides monitoring of service request rejection.

* Resource Request Rate Model: Provides monitoring of requests for usage of OSI resources.

Together, these three models provide an estimate of the workload for a managed resources.

The Workload Monitoring Function defines the following management support objects:

gaugeMonitor, and
meanMonitor.

The Workload Monitoring Function defines the following attributes:

administrativeState,
counterT,
counterTMinusDT,
derivedGauge,
derivedGaugeThold,
estimateOfMean,
estimateOfMeanThold,
gaugeMonitorId,
granularityPeriod,
meanMonitorId,
observedAttributeId,
observedObjectClass,
observedObjectInstance,
schedularName, and
timeConstant.

The Workload Monitoring Function references the following notification types:

attributeChange,
stateChange,
qualityOfServiceAlarm,
objectCreation, and
objectDeletion.

5.13 Summarization Function Agreements

Note: **[**The OIW NMSIG is participating in the development of ISPs for the Summarization Function (ISO/IEC 10164-13). ISPs for the Summarization Function are numbered in the AOM253x series.

The latest drafts of this activity are available from nemo.ncsl.nist.gov via anonymous FTP. Documents can be retrieved as follows:

FTP to nemo.ncsl.nist.gov [129.6.58.136];
login as "anonymous" with password "guest";
cd pub/oiw/agreements;
retrieve the file "perfmgmt.readme";
read that file for instructions as to which further files to retrieve

Since the ISP activity in this area is relatively immature, these drafts are subject to change, especially with regard to base standard ICS proforma style.**]**

Editor's Note: **[**The material in this clause is out-of-date. The clause will be updated when the OIW NMSIG has the resources available to renew activity regarding its contents.**]**

5.13.1 Introduction

This subclause provides agreements pertinent to the Summarization Function defined by [SF].

The Summarization Function:

* defines a conceptual model for the summarization, reporting by notification, and logging of measurements pertaining to managed objects;

* provides the Measurement Summarization, Measurement Request, Observed Object Request, Running Summary Metric, Measures Threshold Control, and Measurement Object Summary Record management support object classes;

* provides a Measurement Summary notification to report summary information; and

* provides a set of services to effect measurement summarization.

The Summarization Function defines the following management support objects:

measurementSummarizationObject,
measurementRequest,
observedObjectRequest,
runningSummaryMetric,
measuresThresholdControl, and
measurementObjSummRecord.

At this time, the Summarization Function does not contain a complete list of services, attributes, or notifications.

5.14 Test Management Function Agreements

Editor's Note: **[**The material in this clause is out-of-date. The clause will be updated when the OIW NMSIG has the resources available to renew activity regarding its contents.**]**

5.14.1 Introduction

This subclause provides agreements pertinent to the Test Management Function defined by [TMF].

The Test Management Function:

* defines a conceptual model for the initiation, control and execution of tests and reporting of test results;

* provides the Test Results Record management support object;

* provides a Test Result notification for information reporting;

* provides a set of services to effect test management.

The Test Management Function defines the following management support objects:

testResultsRecord.

The Test Management Function defines the following attributes:

testSessionId,
testState,
testOutcome,
mOTS,
associatedObjects, and
timeoutPeriod.

The Test Management Function defines the following notification types:

testResultNotification.

The Test Management Function defines the following actions:

testRequestAsyncAction,
testRequestSyncAction,
testSuspendResumeAction, and
testTerminateAction.

5.15 Confidence and Diagnostic Test Classes Agreements

Editor's Note: [The material in this clause is out-of-date. The clause will be updated when the OIW NMSIG has the resources available to renew activity regarding its contents.**]**

5.15.1 Introduction

This subclause provides agreements pertinent to the Confidence and Test Classes defined by [TMF].

Confidence and Diagnostic Test Classes:

* identifies certain characteristics which are common to all classes of tests;

* identifies general test categories;

Confidence and Diagnostic Test Classes defines the following management support objects:

internalResourceResultsRecord,
connectivityResultsRecord,
dataIntegrityResultsRecord,
loopbackResultsRecord, and
protocolIntegrityResultsRecord.

Confidence and Diagnostic Test Classes defines the following attributes:

effectiveTime,
establishmentTime,
testDuration, and
loopCounter.

6 Management Communications

(Refer to the Stable Implementation Agreements Document.)

6.1 Association Policies

(Refer to the Stable Implementation Agreements Document.)

6.1.1 Application Context Negotiation

(Refer to the Stable Implementation Agreements Document.)

6.1.2 Functional Unit Negotiation

(Refer to the Stable Implementation Agreements Document.)

6.1.3 Security Aspects of Associations

(Refer to the Stable Implementation Agreements Document.)

7 Management Information

(Refer to the Stable Implementation Agreements Document.)

8 Conformance

8.1 Introduction

(Refer to the Stable Implementation Agreements Document for additional introductory text.)

Clause 8 also includes a discussion of conformance requirements for demonstration of conformance. These requirements are imposed on implementors to assure that implementations can be tested in an agreed consistent manner.

8.2 General Requirements of Conformance

(Refer to the Stable Implementation Agreements Document.)

8.3 Specific Conformance Categories

(Refer to the Stable Implementation Agreements Document.)

8.3.1 Management Communication Categories

(Refer to the Stable Implementation Agreements Document.)

8.3.2 Management Functions and Services Conformance Categories

(Refer to the Stable Implementation Agreements Document.)

8.3.2.1 General Management Capabilities Conformance Category

(Refer to the Stable Implementation Agreements Document.)

8.3.2.2 Alarm Reporting and State Management Capabilities Conformance Category

(Refer to the Stable Implementation Agreements Document.)

8.3.2.3 Alarm Reporting Capabilities Conformance Category

(Refer to the Stable Implementation Agreements Document.)

8.3.2.4 General Event Report Management Conformance Category

(Refer to the Stable Implementation Agreements Document.)

8.3.2.5 General Log Control Conformance Category

(Refer to the Stable Implementation Agreements Document.)

8.3.3 Management Information Conformance Category

(Refer to the Stable Implementation Agreements Document.)

8.3.3.1 MOCS Proforma

(Refer to the Stable Implementation Agreements Document.)

8.3.4 Management Application Contexts

(Refer to the Stable Implementation Agreements Document.)

8.4 Demonstration of Conformance

(Refer to the Stable Implementation Agreements Document.)

8.4.1 Management Communication

(Refer to the Stable Implementation Agreements Document.)

Editor's Note: **[**The NMSIG should align with CTS-3 and EWOS Conformance Testing Project Team Results. The NMSIG will examine CTS-3 CMIP project for a test object. (The OSI/NM Forum uses an upper tester test object for CMIP conformance testing.)**]**

8.4.2 Management Information

(Refer to the Stable Implementation Agreements Document.)

Editor's Note: [The availability of test cases for managed objects is TBD.**]**

8.4.3 Management Functions and Services

(Refer to the Stable Implementation Agreements Document.)

Editor's Note: [There may be requirements for test objects. The NMSIG should examine the results of the CTS-3 and EWOS Conformance Testing Project Team efforts.**]**

9 Management Ensembles

This clause, which is based on the NM Forum Ensemble Concepts and Format specification [ENSCON], contains agreements regarding the basic concepts and modelling techniques related to management ensembles. These agreements apply to developers of contributions to Annex D, Management Ensemble Annex.

It is not within the scope of this clause to make agreements about or to define specific management ensembles. Such definitions and/or agreements can be obtained via the Management Ensemble Library.

9.1 Management Ensemble Concepts

When modelling management ensembles, these agreements require the use of [ENSCON] with the following additional constraints.

Editor's Note: [Constraints will be added as subclauses, as they are identified. If no constraints are identified, the phrase "with the following additional constraints" will be deleted.**]**

9.2 Management Ensemble Format

When defining management ensembles, these agreements require the use of the format defined by [ENSCON] Annex C, with the following additional constraints.

9.2.1 Use of Boiler Plate Text

The common "boiler plate" text defined in Annex C of [ENSCON] shall be considered optional for inclusion in specific ensembles. Use of the boiler plate text is recommended, but only that text which is relevant to the ensemble need be included. The boiler plate text may be revised as appropriate for the specific ensemble.

10 Management Coexistence and Interworking

(Refer to the Stable Implementation Agreements Document.)

10.1 Internet MIB Translation

(Refer to the Stable Implementation Agreements Document.)

10.2 ISO/CCITT to Internet Management Proxy

(Refer to the Stable Implementation Agreements Document.)

10.3 ISO/CCITT MIB Translation

When translating management information from ISO/CCITT GDMO format to Internet MIB macro format, these agreements allow the use of [IIMCOMIBTRANS] with the following additional constraints.

Editor's Note: **[**Constraints to be added as subclauses, as they are identified. If no constraints are identified, the phrase "with the following additional constraints" will be deleted.**]**

Editor's Note: **[**Should we constrain MIB translation algorithms or approaches?**]**

Annex A (informative)

Management Information Library (MIL)

A.1 Scope of Activities

The OIW NMSIG may:

- a) Develop product level specifications and international Profiles for implementations, relating to common services/protocols for exchanging management information between OSI nodes;

- b) Develop product level specifications and associated international Profiles for implementations relating to systems management functions;

- c) Define, encourage and promote the development of requirements for new Managed Objects (MOs), MO Profiles and MO Ensembles (bundles of Profiles). As required, collect and/or disseminate this information to appropriate bodies in which it is expected that formal definition and registration of such management information can occur;

- d) Support and/or lead the development of definitions for new MOs, MO implementation agreements, MO Profiles and MO Ensembles;

- e) Support the cataloguing of new MOs, MO Profiles and MO Ensembles.

As necessary, the SIG will:

a) Establish liaisons with various standards bodies;

b) Provide feedback for additional/enhanced services and protocols for OSI management.

-

Examples of Specific Activities
1. Requirements Definition

- (a) Work with other OIW SIGs (potentially via TLC) and with EWOS & AOW NM groups to develop concepts/guidelines for developing internationally harmonized MO Profiles and MO Ensembles.

Example: TAX 3
MO Profile Guidelines

- (b) Actively solicit contributions that delineate new requirements for new MOs, MO Profiles, MO Ensembles, e.g., via letters to NMSIG membership, NMForum UAC, Open Systems User Alliance (Houston 30/Dallas 800), OIW membership, press releases, CBD announcements, ...

Example: X.400 MTA contribution (NMSIG–92/178, –92/179)
FAA Enterprise OA&M contribution (NMSIG–92/113)

- (c) Promote need to develop requirements for new MOs, Profiles, Ensembles, e.g., via OIW banquet presentations.

2. MO, Profile, Ensemble Definition Activities

- (a) On an as-interested basis (e.g., in response to requirements identified via example 1), the NMSIG may:

- (i) Develop MO, Profile, and/or Ensemble definitions, *when* no relevant standards or consortia activities exist;

Example: FAA Enterprise Management Information

- (ii) Collaborate with other OIW SIGs, or consortia, to provide MO definition contributions to standards, or consortia, to accelerate progress, when standards, or consortia, activities are immature or stagnated;

- [Consider registering contributions when, in the judgment of the NMSIG, standards activities are lagging *extremely* behind (e.g., > 3 years) *urgent* requirements. This would allow associated products to have useful market life cycles.]

- Example: X.400 MTA MOs

- (iii) Critique relevant MO, Profile, and Ensemble work ongoing in other groups;

- Example: OMNIpoint 1 Document Reviews

- (iv) Lead/support MO implementation agreements, Profiles, Ensemble development, *when* supporting standards, or consortia, activities are sufficiently mature.

- Example: M.TA51

- (b) On an as-interested basis (e.g., in response to requirements identified via example 1), the NMSIG may develop translation algorithms for automatically converting extant MO definitions from one community's object model (e.g., SNMP SMI) into OSI compatible, GDMO MOs.

3. Catalogue

- (a) Request EWOS & AOW to announce availability of catalogue.

- (b) Solicit further inputs to be fed to OPn cataloguer.

Editor's Note: **[**The following information in Annex A is residual information following the movement of clauses A.4 and A.5 to the Stable Agreements. This remaining text (i.e., clauses A.1.2, A.2, and A.3) needs to be reviewed for possible updates or deletion.**]**

A.1.1 Background

The Management Information Library provides definitions of management information - managed object classes, name bindings, attributes, actions and notifications. Provision of these definitions is made by a) references to standards' documents that contain these definitions, or b) inclusion of the actual definitions in this document; in which case they are registered in the NMSIG arc of the ISO ASN.1 Object Identifier Tree.

The reasons why the NMSIG has opted to define management information are as follows:

(i) There is an urgent need for network management within the community. Managed objects are critical ingredients of network management; but standards' defined managed objects that represent network/system resources are not available yet. However, there does exist an ISO standard that specifies guidelines for defining managed objects : [GDMO]. Different organizations, including private companies, etc, can use [GDMO] to define their own managed objects. However, two network management implementations can interoperate only if there is a common subset of managed objects supported on both sides. The NMSIG has used the [GDMO] standard to define "public domain" managed objects that meet the needs of the community and foster interoperability.

(ii) Standards' groups are not addressing all the network/system resources that need to be managed; i.e. there is no standards' activity for defining managed objects that represent such resources. The NMSIG has attempted to fill these holes by defining managed objects for these resources, and thus fulfil the needs of the community.

As mentioned earlier, managed objects in the MIL have been provided to foster interoperability. They are not normative as far as the NMSIG IAs are concerned. Implementors do not have to support any of the MIL managed objects; they may choose to define their own managed objects using the agreements on [GDMO] specified in Section 18.7. However, supporting managed objects from the MIL will increase the potential for interoperability with other network management implementations.

The NMSIG defined managed objects in the MIL are intended to be implementable but they also serve as a basis from which other implementations may define refinements or alternatives. These definitions do not override or duplicate those provided by standards' groups or other OIW SIGs.

More specifically, the transport and network layer managed objects that have been defined in the MIL are "generally applicable" objects, in that they do not represent any particular transport or network layer protocols, but contain characteristics common across different transport or network layer protocols. These managed objects provide a high level view of the transport and network layers, and are especially useful in managing heterogeneous networks that support various different types of transport and network layer protocols. These managed objects do not override the OSI Transport and Network Layer managed objects that are being defined in ISO. The ISO specified OSI Transport and Network Layer managed objects are "specific" managed objects that represent strictly the OSI Transport and Network protocol layers.

A.2 Rules and Procedures

Editor's Note: **[**The text contained in this clause is relatively old and requires update to accurately reflect the rules and procedures used to define the current MIL.**]**

The following rules and procedures apply to managed object class definitions that are to be included in the MIL :

(i) All managed object class definitions provided by the MIL must comply with ISO [GDMO] object templates.

(ii) A managed object class definition provided by the MIL must represent an abstraction of an identifiable logical or physical resource that can be managed via OSI management.

(iii) All managed object classes in the MIL will have registered ASN.1 object identifiers assigned either by a standards' body if it is defining the managed object class, or, if the managed object class definition is being progressed within the NMSIG, by the NMSIG in its branch of the ISO Registration Tree.

(iv) A managed object class will be selected as a candidate for inclusion into the MIL if there are at least two NMSIG members from different companies who express a requirement (strong interest) for the managed object class. If this is not a standards' defined managed object class, then there must be at least one NMSIG member who is committed to developing the definition of the managed object class.

(v) A managed object class selected for the MIL will be given a priority based on the number of members who express interest in it.

(vi) All managed object class definitions that are proposed for inclusion into the MIL will undergo a review process within the NMSIG. NMSIG member defined managed object classes will additionally undergo a balloting process. If problems are found with a standards' defined managed object class, the appropriate standards' body will be approached. If problems are found with a member defined managed object class, it will be returned with comments.

(vii) Based on its priority, there will be a call for contributions on the definition of a managed object class at an NMSIG meeting. Contributions could be in the form of a) identification of a standards' body that is currently working on the definition, or b) an NMSIG member definition of the managed object class.

(viii) An element of management information, once registered, i.e., given an ASN.1 Object Identifier, will never be deleted from the Registration Tree (ASN.1 Object Identifier tree). It may, however, fall into disuse due to lack of requirements for it.

A.3 General Guidelines

Editor's Note: **[**The text contained in this clause is relatively old and requires update to accurately reflect the general guidelines used to define the current MIL.**]**

It is recommended that the following guidelines be used in general for all managed object definitions, unless there is a specific exception condition:

a) For the objectCreation Notification, send all the attributes of the created managed object instance in the Attribute List parameter.

b) For the objectDeletion Notification, send all the attributes of the deleted managed object instance in the Attribute List parameter.

c) For the attributeValueChange Notification, send the Attribute Identifier List parameter.

d) Use the attributeValueChange Notification to signal counter attribute wrap, and include the maximum counter value in the Old Attribute Value parameter.

e) Include the Alarm Status attribute in all object class definitions which also contain one or more Alarm Notifications.

f) Include the State ATTRIBUTE GROUP in all object class definitions which also include one or more state attributes defined by [STMF].

g) Include the Relationship ATTRIBUTE GROUP in all object class definitions which also include one or more relationship attributes defined by [ARR].

h) Usage State, when used, is contained in a conditional (not mandatory) package.

A.4 Harmonized Library

(Refer to the Stable Implementation Agreements Document.)

A.5 OIW NMSIG IVMO Definitions

(Refer to the Stable Implementation Agreements Document.)

A.6 OIW NMSIG Shared Management Knowledge (SMK) Definitions

Editor's Note: **[**Requirements for a discovery object have been met by the discovery object defined and registered in the OP1 Library Volume 4 [OP1LIB] of the NM Forum and, therefore, the discovery definition and object ID in the NMSIG agreements have been deleted.**]**

Editor's Note: **[**To conserve resources, we have not reproduced the old text here that has been deleted from Annex A.6. For those wishing to review the deleted text, the old text can be found in the June 1991 Working Implementors' Agreements.**]**

Annex B (informative)

NMSIG Object Identifiers

(Refer to the Stable Implementation Agreements Document.)

B.1 Introduction

(Refer to the Stable Implementation Agreements Document.)

B.2 Harmonized MIL Object Identifiers

(Refer to the Stable Implementation Agreements Document.)

B.2.1 Object Class Object Identifiers

(Refer to the Stable Implementation Agreements Document.)

B.2.2 Package Object Identifiers

(Refer to the Stable Implementation Agreements Document.)

B.2.3 Name Bindings Object Identifiers

(Refer to the Stable Implementation Agreements Document.)

B.2.4 Attribute Object Identifiers

(Refer to the Stable Implementation Agreements Document.)

B.2.5 Action Object Identifiers

(Refer to the Stable Implementation Agreements Document.)

B.2.6 Parameter Object Identifiers

(Refer to the Stable Implementation Agreements Document.)

B.2.7 Response Code Object Identifiers

(Refer to the Stable Implementation Agreements Document.)

B.2.8 Module Object Identifiers

(Refer to the Stable Implementation Agreements Document.)

B.3 Phase 1 MIL Object Identifiers

(Refer to the Stable Implementation Agreements Document.)

B.3.1 Object Class Object Identifiers

(Refer to the Stable Implementation Agreements Document.)

B.3.2 Name Bindings Object Identifiers

(Refer to the Stable Implementation Agreements Document.)

B.3.3 Attribute Object Identifiers

(Refer to the Stable Implementation Agreements Document.)

B.3.4 Module Object Identifiers

(Refer to the Stable Implementation Agreements Document.)

Annex C (informative)

MOCS Proforma

(Refer to Stable Implementation Agreements Document.)

Annex D (normative)

Management Ensemble Annex

D.1 Introduction

This Annex contains specific management ensembles defined and published by the OIW NMSIG. Management ensembles contained in this Annex shall be defined using the concepts and formats specified in clause 9 of these agreements.

D.2 Systems Management for OSI Transport and Network Layers Ensemble

(Refer to the Stable Implementation Agreements Document.)

D.3 Allomorphism Sensitive Event Forwarding Discriminator (EFD) Ensemble

Editor's Note: **[**Because the Allomorphism Sensitive Event Forwarding Discriminator (EFD) Ensemble is intended to be a self-contained, standalone document, the clauses and subclauses of the Allomorphism Sensitive Event Forwarding Discriminator (EFD) Ensemble (as shown here in Annex D.3) are numbered as they would be in a separate, standalone document, and not as they would be according to their position in Annex D.3.**]**

Revision History

Issue 1.0, Draft 1 – December 1992

This is the first draft of this Ensemble, generated as output from the December 1992 OIW NMSIG meeting. The proposed schedule for this document is as follows:

1) Draft presented to OIW NMSIG. Initial comments generated. Ensemble added to the working IAs. December 1992 OIW NMSIG.

2) OIW NMSIG to prepare comments on the Ensemble. Comments to be placed on the OIW NMSIG exploder. December 1992 – March 1993.

3) EWOS EG–NM, AOW NMSIG, OSF, X/OPEN, OMG, NMF to generate comments. December 1992 – March 1993.

4) OIW NMSIG to review all comments, and resolve comments. March 1993.

5) Attempt to harmonize ensemble at RWNMCC.

6) Resolve comments. Move to stable IAs.

1 Introduction

Ensembles provide a top down view of a particular solution to a management problem. In order to focus on the solution to this management problem, specific restrictions are placed upon particular referenced definitions. The concepts and format of ensembles are described in Forum 025 – The "Ensemble" Concepts and Formats – Issue 1.0.

Each ensemble contains general text in each section that is common to all ensembles. By convention this common text is portrayed in bold italic characters.

This ensemble, wherever possible, references documents which define the components of the ensemble.

The management problem is identified as a set of requirements and constraints. In defining the solution to this management problem, the resources to be managed, the functions to be applied, and the scenarios describing the interactions are all identified. The ensemble references base standards and international standardized profiles (isps). It also references libraries containing definitions expressed by gdmo (guidelines for the definition of managed objects) templates.

The purpose of this document is to collect management information definitions and profiles, and show how they can be applied to manage the resources identified in this ensemble.

This document is organized as follows:

Section 1, "Introduction"	Provides a high level overview describing the ensemble and the structure of the document.
Section 2, "Management Context"	Identifies the managed resources and management capabilities of the ensemble.
Section 3, "Information Model"	Specifies all management information components of this ensemble.
Section 4, "Ensemble Conformance Requirements"	Provides or references statements of conformance for this ensemble. The managed object conformance statements (MOCS) proformas specific to the ensemble are provided in Annex B.

1.1 Unique Identity

The unique identity is a registered object identifier used to identify this ensemble.

An object identifier has not been assigned yet to this ensemble.

1.2 General Description of the Ensemble

This ensemble describes the functional capabilities of the allomorphismSensitiveEFD managed object class. The allomorphismSensitiveEFD is a subclass of the standardized eventForwardingDiscriminator managed object class defined in ISO 10165–2. This ensemble describes how:

- o the decision to forward an event report can be made based upon the valid allomorphic classes of a notification,
- o allomorphic event reports are generated at an agent,
- o a manager configures an allomorphismSensitiveEFD to generate allomorphic event reports, and
- o allomorphism is employed to manage an allomorphismSensitiveEFD.

1.3 Scope and Purpose

Ensembles represent specific solutions to particular problems. Thus, an ensemble is the complete description of the problem and the solution to that problem.

This section describes the requirements of the problem. It includes the definition of the information model that represents the solution to a problem. These definitions comprise references to one or more management information libraries which contain definitions of managed object classes expressed in gdmo templates, packages, attributes, name bindings, etc. Also,included in the ensemble definition are statements of conformance and suitable proformas.

The requirements driving the design of the ensemble are as follows:

1. Develop a discriminator managed object class that allows for filtering on the list of allomorphs emitted with a notification by an extended managed object that acts allomorphically.
2. Develop a means of determining the valid value to be placed into the "managed object class" field of an allomorphic event report. Should the value be the actual class or an allomorphic class?
3. To describe allomorphic operations, manager and agent responsibilities, to manage an allomorphismSensitiveEFD.

This ensemble references 10165–2, DMI which contains GDMO for the eventForwardingDiscriminator class from which allomorphismSensitiveEFD is derived.

This ensemble references protocol data units required by ISP 11183–2, "CMISE/ROSE for AOM12 – Enhanced Management Communications" as a basis for conformance requirements.

1.4 Relationships With Other Ensembles

This section identifies the relationships of this ensemble to other ensembles.

This ensemble can be used with other ensembles that require the forwarding of unsolicited management information. For example, this ensemble can be used in conjunction with the OSI Interworking Ensemble.

2 Management Context

The "management context" describes why the ensemble is required. The description of the "management context" includes the definition of the resources to be managed, the management functions to be performed, the scope of the problem to be solved, and the management view or level of abstraction from which the problem is to be approached.

2.1 General Introduction

2.1.1 Allomorphic Behaviour of Managed Objects

Allomorphism is the ability of a managed object that is an instance of a given class to be managed as an instance of one or more other managed object classes. For example, if a manager product only understands a printer managed object class, and an agent supports a subclass of printer called superDuperPrinter, allomorphism allows the manager to manage instances of the superDuperPrinter managed objects as instances of the printer managed object class.

While allomorphic behaviour represents some implementation cost to both the manager and agent products, its benefits outweigh the costs. The chief benefit is that of decoupling the delivery of enhancements in an agent product with specific support enhancements in a manager product, providing a seamless migration strategy. In other words, when the agent product is upgraded to allow printers to be modelled as superDuperPrinter managed objects, it is not a requirement to simultaneously upgrade the manager to understand superDuperPrinter at the same time. The manager can manage superDuperPrinter managed objects as if they were members of the printer managed object class until its code can be updated to manage instances of superDuperPrinter class. By supporting allomorphic behaviour, the agent product will be able to receive a default level of management from a manager product which only supports the allomorphic class, thus making possible an easy migration path for installing updated agent and manager products.

2.1.2 Allomorphism Sensitive EFD

The allomorphismSensitiveEFD managed object class will provide capabilities above and beyond those of the standardized eventForwardingDiscriminator managed object class defined in ISO 10165–2.

2.1.2.1 Enhanced filtering capability

The allomorphismSensitiveEFD managed object class provides enhanced filtering capabilities.

When both the manager and agent support allomorphism, there will frequently be cases where a manager wishes to receive unsolicited information about a particular type of resource. For example, a manager might wish to receive all notifications emitted by managed objects representing printers. The allomorphismSensitiveEFD provides a mechanism for allowing a manager to receive notifications for a printer resource, regardless of whether the printer is represented at an agent by a printer managed object or a superDuperPrinter managed object.

2.1.2.2 Allomorphic Notification Support

The allomorphismSensitiveEFD managed object class provides a deterministic mechanism for an agent to provide allomorphic event reports to a manager.

Allomorphic event reports differ from non–allomorphic event reports only in the value of the managedObjectClass parameter of the event report. For example, an allomorphic event report corresponding to a notification emitted by a superDuperPrinter managed object would have the managedObjectClass parameter of the event report equal to printer, since this is the class that the manager understands. The other parameters of the event report are not altered as a result of allomorphism. If the notification is extendable, the manager may receive additional parameters in eventInfo associated with the notification as it is defined for superDuperPrinter, that are not defined for printer. The manager must be capable of receiving the event report in its totality and utilize the parameters as it sees fit.

An example of an extendable notification is the standardized communicationsAlarm. The communicationsAlarm has an extendable parameter defined called additionalInformation. The syntax of additionalInformation is SET OF managementExtension. The additionalInformation parameter contains more subparameters in a communications Alarm emitted from a superDuperPrinter than it would if emitted from a printer. The definition of communicationsAlarm is extended using the NOTIFICATION template, and PARAMETER template.

Please see the second edition of CMIPrun for a tutorial on the use of SET of ManagementExtension.

A manager that only understands the printer class will receive a communicationsAlarm notification that has additional subparameters in the additionalInformation parameter that applies to the superDuperPrinter class, and not to the printer class. The manager must be able to understand these additional subparameters (or display them to an operator who can understand them) as it sees fit.

An example of additional subparameters that a manager must pay attention to and process are the additional communicationsAlarm subparameters that are a part of the additionalInformation parameter, defined with the significance subparameter=true. The significance subparameter is a boolean value which is set to true if the receiving system (manager) must be able to parse the contents of the additional subparameter for the event report to be fully understood.

2.1.2.3 Compatibility with Managers that only support EFDs

Instances of the allomorphismSensitiveEFD managed object class can act allomorphically themselves. This allows a down–level manager that only understands the eventForwardingDiscriminator class to manage instances of allomorphismSensitiveEFD as if they were instances of eventForwardingDiscriminator.

2.2 Management View and Level of Abstraction

This section indicates the management view of the ensemble which includes information on the level of abstraction. For example, in an hierarchically organized system this section would indicate if the ensemble deals with the management of equipment, the management of the networks, or the management of services. It may also indicate management perspectives and roles.

This ensemble deals with the discrimination and forwarding of unsolicited information from managed objects acting allomorphically, and from managed objects not acting allomorphically. This ensemble is general purpose, and can be used in any management environment where systems playing the manager and agent role have the capabilities to support managed objects acting allomorphically.

This ensemble addresses the provider viewpoint, describing the responsibilities of a system playing the agent role that provides the event report discrimination function. This ensemble also details the user viewpoint, describing the responsibilities of a system playing the manager role that uses the discrimination function.

2.3 Resources

This section defines all the resources or components of resources that are to be the subject of the ensemble. The definition of the resources contains all the resources and only those resources that are relevant to the ensemble. The resources are defined by textual descriptions or by reference to other documents containing descriptions of the resources. When other documents are referenced statements are provided to indicate any restrictions and constraints on those source definitions.

This ensemble models the discrimination functionality realized by an agent system.

2.4 Functions

This section defines the management functions that can be performed on the resources described in section 2.3, "Resources." These functions may be primitive functions for osi systems management (e.G., Event management), higher level functions for general network management (e.G., Alarm surveillance), or other functions unique to the problem of the ensemble addresses.

These definitions consist of a brief textual description of each function. In some cases these descriptions will include a set of references to other documents. For example:

ISO system management functions

Telecommunications management network (tmn) ccitt rec. M.3020

Other standards

When other documents are referenced, statements are required to indicate the restrictions and constraints to the function definitions to the ensemble.

This ensemble utilizes the functions that are defined for the event forwarding discriminator managed object class as defined in ISO/IEC 10164–5. In addition, this ensemble defines a new function, the Allomorphism Sensitive EFD Function, comprised of:

- o allowing a manager to set a discriminator construct to apply a filter to the set of valid allomorphic classes for a notification.
- o enabling an agent to fill in the managedObjectClass parameter of a notification with an allomorphic class, if appropriate.
- o enabling a manager to manage an instance of allomorphismSensitiveEFD as an instance of eventForwardingDiscriminator using allomorphism.

2.5 Other Requirements

This section contains any other management context requirements than functions, resources or level of abstraction. These may be business requirements or performance requirements, for example.

This ensemble also fills in several gaps in the current definition of the eventForwardingDiscriminator:

- o defines precisely the object identifiers that correspond to potential event report attributes mapped from attributes of top.
- o Clarifies that local time instead of GMT time is to be used for attributes of the daily and weekly scheduling packages for instances of allomorphismSensitiveEFD that implement these packages.

3 Management Information Model

The information model focuses on the real world under study. It contains information about both the elements of the model and their interrelationships. The elements of management information are defined using gdmo templates and their interrelationships are graphically illustrated.

3.1 General Introduction

The allomorphismSensitiveEFD managed object class provides capabilities above and beyond those of the standardized eventForwardingDiscriminator managed object class defined in ISO 10165–2.

3.1.1 Enhanced Event Filtering Capability

The allomorphismSensitiveEFD managed object class provides enhanced event filtering capabilities.

When both the manager and agent support allomorphism, there will frequently be cases where a manager wishes to receive unsolicited information about a particular type of resource. For example, a manager might wish to receive all notifications emitted by managed objects representing printers. The allomorphismSensitiveEFD provides a mechanism for allowing a manager to receive notifications corresponding to a printer resource regardless of whether the printer is represented at an agent by a printer managed object, or a superDuperPrinter managed object.

When a superDuperPrinter managed object acting allomorphically as a printer emits a notification, it makes available two things at the managed object boundary:

1. the notification as defined for the superDuperPrinter class, and
2. an unordered list of valid allomorphs for the notification.

The list of valid allomorphs may differ from the value of the allomorphs attribute of the superDuperPrinter managed object. For example, the allomorphs attribute value may include printer, superPrinter, and function. The notification being emitted is printerReport which is inherited from printer, superPrinter, and not from function. Therefore, when the superDuperPrinter managed object emits the printerReport notification, it makes available at the managed object boundary:

1. the printerReport notification as defined for the superDuperPrinter class. This notification will include managedObjectClass parameter equal to superDuperPrinter. The notification will also include any additional parameters added as a result of subclassing from printer, and superPrinter.

2. the "list of valid allomorphs for the notification" with printer and superPrinter as the only set elements.

The notification information must then be transformed into a potential event report as described in ISO/IEC 10164–5, Event Report Management Function by the conceptual event pre–processing function. A potential event report is considered a "discriminator input object" that has attributes that reflect the notification parameters, and additional information that the allomorphismSensitiveEFD can discriminate on. The allomorphismSensitiveEFD can discriminate on the following attributes of a potential event report:

- managedObjectClass – corresponds to the value of the objectClass attribute of the superDuperPrinter emitting the notification. The value would be superDuperPrinter.

- managedObjectInstance – the distinguished name of the instance of superDuperPrinter emitting the notification

- eventType – the value would be printerReport

- validAllomorphs – corresponds to the list of valid allomorphs that accompanied the notification. The value would be {printer, superPrinter}, where {} denotes a SET.

- Event type–specific attributes – these are attributes that correspond to parameters of the notification. These notification parameters must have syntax associated with them. This is accomplished when defining the notification using the GDMO NOTIFICATION template constructs of WITH INFORMATION SYNTAX and AND ATTRIBUTE IDS.

Once the potential event report is formed, then the conceptual event pre–processing function routes it to all allomorphismSensitiveEFD managed objects, and any eventForwardingDiscriminator managed objects (if the system supports them).

Each allomorphismSensitiveEFD managed object applies the discriminator construct specified by the discriminatorConstruct attribute to the attributes of the potential event report to determine whether it meets the criteria for forwarding to the manager.

An enhancement offered by allomorphismSensitiveEFD over the eventForwardingDiscriminator is the ability to discriminate on values of the validAllomorphs. To continue the example, the manager wishes to receive printer reports from managed objects that are either printers, or act as printers allomorphically. The manager specifies the following value for the discriminatorConstruct attribute of an allomorphism SensitiveEFD:

```
((managedObjectClass Equal printer)
   or
(set membership ({printer}, validAllomorphs)))
   and
((eventType Equal printerReport))
```

where set membership refers to the matching rules for set valued attributes:

- equality

- o present
- o subset of
- o superset of
- o non–null set intersection

The (managedObjectClass Equal printer) comparison fails since the potential event report managedObjectClass attribute value is equal to superDuperPrinter. The (set membership (printer, validAllomorphs)) comparison passes, since printer is listed as an element of the validAllomorphs set–valued attribute of the potential event report. The (eventType Equal printerReport) comparison also passes. As a whole, the discriminator construct is satisfied, allowing the allomorphismSensitiveEFD to pass the notification.

```
((managedObjectClass Equal printer)
    or
(set membership ({printer}, validAllomorphs)))
    and
((eventType Equal printerReport))

resolves to    ((false)or(true))and(true)
resolves to      (true) and (true)
resolves to           true
```

3.1.2 Allomorphic Event Report Capability

The allomorphismSensitiveEFD managed object class provides a deterministic mechanism for an agent to provide allomorphic event reports to a manager. This is accomplished with semantics associated with a new attribute of allomorphism SensitiveEFD called switchMOCTo.

The switchMOCTo attribute is set by the manager to denote the managed object classes that it understands and desires to have present in the allomorphic event report. For example, the manager sets switchMOCTo to {printer} to indicate that it is interested in receiving notifications with the managedObjectClass parameter set to printer, as opposed to superPrinter or superDuperPrinter, for notifications emitted from instances of superPrinter or superDuperPrinter that can be managed as a printer allomorphically.

Allomorphic event reports differ from non–allomorphic event reports only in the value of the managedObjectClass parameter of the event report. In the example, an printerReport emitted by a superDuperPrinter managed object would have the managedObjectClass parameter of the event report switched to printer by the allomorphismSensitiveEFD, since this is the class that the manager understands. The other parameters of the event report are not altered as a result of allomorphism. Therefore, the manager may receive additional parameters in the eventInfo parameter associated with the notification as it is defined for superDuperPrinter, that are not defined for printer. The manager must be capable of receiving the event report and handling extraneous parameters of interest.

If the processing of the discriminatorConstruct determines that an event report is to be generated, then allomorphismSensitiveEFD takes the following processing steps in determining if an allomorphic event report or a non–allomorphic event report should be emitted:

1. determine if the value of the managedObjectClass attribute of the potential event report is a set element of the switchMOCTo attribute of the allomorphism SensitiveEFD.

   o If TRUE, then a non–allomorphic event report is issued. The managedObjectClass parameter of the event report will contain the value of the actual class of the managed object, not an allomorphic class.

   o If FALSE, then proceed to the next step

   In the example, the value of switchMOCTo is {printer}. The value of the managedObjectClass attribute of the potential event report is superDuperPrinter. Since switchMOCTo does not contain superDuperPrinter, then it is still possible that an allomorphic event report might be issued.

2. compare the value of the switchMOCTo attribute of allomorphismSensitiveEFD to the value of the validAllomorphs attribute of the potential event report.

   (switchMOCTo) NON–NULL INTERSECTION (validAllomorphs)

   o If TRUE, then an allomorphic event report will be issued. Proceed onto the next step.

   o If FALSE, then a non–allomorphic event report will be issued. The managedObjectClass parameter of the event report will contain the value of the actual class of the managed object, not an allomorphic class.

   Continuing the example, the manager previously set the value of switchMOCTo to {printer} to indicate that if the notification passes the discriminatorConstruct, then it wants to receive event reports from those managed objects of printer class, or allomorphic event reports from managed objects that can be allomorphically managed as instances of the printer class. The NON–NULL INTERSECTION test is applied to determine if a non–allomorphic event report, or alternatively, an allomorphic event report is issued:

   (switchMOCTo) NON–NULL INTERSECTION (validAllomorphs)

   same as

   {printer} NON–NULL INTERSECTION {printer, superPrinter}

   yields

   TRUE

   In the example, an allomorphic event report will be issued.

3. The candidate values for insertion into the managedObject Class field of the allomorphic event report are the result of a logical operation:

   (switchMOCTo) LOGICAL INTERSECTION (validAllomorphs)

If multiple values result from the operation, then it is a local implementation option to choose one of the values.

Editor's Note: [The following comments were generated at the December OIW NMSIG. The comments have not been harmonized yet within the OIW NMSIG. These comments will appear in the text of the working agreements as an editors note. Other consortia/workshops are asked to comment on the OIW NMSIG comments as well.

1. Examine the applicability of the switchMOCTo attribute to other support objects such as:

   - access control objects
   - scheduling objects
   - management knowledge management

2. Redo the syntax and/or semantics of the switchMOCTo attribute so that it represents a prioritized list of classes instead of a set of classes. This would allow a manager to give its "preferred order" of classes to which the managedObjectClass parameter value would be switched to for an allomorphic event report.**]**

Completing the example, the result of the LOGICAL INTERSECTION is printer. Therefore, the allomorphismSensitiveEFD will switch the value of the managedObjectClass parameter of the allomorphic event report from superDuperPrinter to printer.

3.1.3 Other Requirements

3.1.3.1 Package Requirements

This ensemble requires that the following packages must be dynamically present in an instance of allomorphismSensitiveEFD :

- o top package
- o packages package
- o allomorphic package
- o discriminator package
- o efd package
- o allomorphism sensitive EFD package

3.1.3.2 Name Binding Requirements

The following name binding requirements apply:

- o at least one name binding must be supported
- o any managed object class can be listed as the SUPERIOR managed object class. However, an instance of this class must be the managed object that "represents the system". In addition, an instance of this class must be compatible with the system managed object class.

3.1.3.3 Potential Event Report Attribute Requirements

The ensemble requires that an instance of allomorphismSensitiveEFD must be able to discriminate on at least the following attributes of a potential event report derived from notifications. This is a minimum set:

Table 3-1. Minimum PER Attributes required by the Profile

attribute	Object Identifier
managedObjectClass	{smi2AttributeID 60}
eventType	{smi2AttributeID 14}
managedObjectInstance	{smi2AttributeID 61}
perceivedSeverity	{smi2AttributeID 17}
securityAlarmSeverity	{smi2AttributeID 23}

The ensemble allows for a supplier to specify additional attributes derived from notifications. This ensemble defines the validAllomorphs as one such attribute. Other attributes derived from notifications must be specified as part of the GDMO NOTIFICATION template constructs of WITH INFORMATION SYNTAX and AND ATTRIBUTE IDs.

Table 3-2. Additional PER attributes required by this Ensemble

attribute	Object Identifier
validAllomorphs	{XXXXXXXXXXXXXXX}

3.1.3.4 Discriminator Construct Requirements

The manager sets the filter to be applied to the attributes of a potential event report by setting the discriminatorConstruct attribute value. The filter takes the same form as the filters that are supplied in CMIP operations, the CMISFilter syntax. The following filter items must be supported:

- o equality
- o substrings
- o greaterOrEqual
- o lessOrEqual

- o present
- o subsetOf
- o supersetOf
- o nonNullIntersection

The following CMIS filter parameters must be supported:

- o item – refers to one of the above listed filter items
- o and
- o or
- o not

The following example is used to clarify the difference between a filter item and a filter parameter in a filter expression present as a value of the discriminatorConstruct attribute:

(filter item) (managedObjectClass Equal EFD)
(filter parameter) OR
(filter item) (setOperation) ({ALLOEFD}, allomorphs))

The number of filter items in this example is two and the level of nesting in this example is one.

An instance of allomorphismSensitiveEFD must be capable of supporting at least:

- o sixteen filter items in a discriminatorConstruct attribute value
- o four filter items joined by the AND filter parameter
- o four filter items joined by the OR filter parameter

An instance of allomorphismSensitiveEFD must be able to support at least two levels of nesting when the filter parameter at the first level of nesting is an AND or an OR.

The filter parameter of NOT may be used at any level of nesting without any restrictions.

3.1.3.5 Support of Allomorphism

Instances of allomorphismSensitiveEFD must support being managed allomorphically as an instance of eventForwardingDiscriminator. As a result:

- o the allomorphs attribute of an instance of allomorphismSensitiveEFD must at least contain a value for eventForwardingDiscriminator.

- o the validAllomorphs PER attribute must at least contain a value for eventForwardingDiscriminator for notifications emitted by an instance of allomorphismSensitive EFD.

3.1.3.6 Daily Scheduling and Weekly Scheduling Packages

Unless specified otherwise in a managed object behaviour definition, the values of the following components of weekMask and IntervalsOfDay are interpreted as local time:

- o Interval–start,
- o Interval–end, and
- o days of week

3.2 Relationships

This section defines the relationships between the components of the model. These may be expressed in entity relationship (er) diagrams or other similar graphical representations.

Three types of diagrams are used:

- o one for the relationships inherent in the underlying resources,
- o one for the relationships among the classes representing these resources,
- o and one for the naming schema.

3.2.1 Relationships Among The Resources

3.2.2 Relationships Among Classes Representing The Resources

3.2.3 Naming Schema

3.3 Scenarios

This section defines the ensemble scenarios. Each of these definitions consists of a brief textual description and message flow diagrams. The scenarios are used to show the managed object in the information model can be used to accomplish the functions listed in section 2.4, "Functions".

Note: **[**Instances of the allomorphismSensitiveEFD managed object class can act allomorphically themselves as instances of the eventForwardingDiscriminator class. This allows a manager that only understands the eventForwardingDiscriminator class to manage instances of allomorphismSensitiveEFD as if they were instances of eventForwardingDiscriminator.**]**

The following scenarios summarize the exchanges between a manager and agent. The exchanges consider an agent that has implemented allomorphismSensitiveEFD. The agent only has instances of allomorphismSensitiveEFD instantiated, and not any instances of eventForwardingDiscriminator. The case of a manager that only understands eventForwardingDiscriminator and manages instances of allomorphismSensitiveEFD as if they were instances of eventForwardingDiscriminator is examined. In addition, the case of the manager that understands allomorphismSensitiveEFD is also explored.

The following abbreviations will be used:

ABBREVIATION	DESCRIPTION
EFD	Denotes the eventForwardingDiscriminator object class defined in ISO 10165–2.
ASEFD	Denotes allomorphismSensitiveEFD object class. Managed objects of this class are compatible with the eventForwardingDiscriminator managed object class.
ACTUAL	Refers to the "actual class", as documented in clause 7.4.4 of GDMO.

The protocol mechanisms are documented by management operation.

3.3.1 Event Forwarding Scenarios Overview

The first scenario provides an overview of event forwarding in an allomorphismSensitiveEFD environment where both the manager and agent understand the allomorphismSensitiveEFD, but only the agent implements instances of allomorphismSensitiveEFD:

1. The Managing Application MgrApplT creates an eventForwardingDiscriminator (EFD T1) at the managing system (or some other local mechanism to route events) to receive event reports (ERs) forwarded from the agent system.

2. Managing Application MgrApplT creates an allomorphismSensitiveEFD (ASEFD T2) at the agent system to receive ERs. The managers sets the values of discriminatorConstruct and switch MOCTo on the create operation.

3. Notifications with validAllomorphs attribute are generated by the managed objects in the agent system. These notifications become the potentialEventReports and are inputted to ASEFD.

4. The allomorphismSensitiveEFD T2 tests the attributes of the potential event report relative to the value of the discriminatorConstruct attribute. If the discriminatorConstruct resolves to true, then the allomorphismSensitiveEFD T2 will forward an event report.

   The allomorphismSensitiveEFD T2 tests to see if the value of the managedObjectClass attribute of the potential event report is a set element of the switchMOCTo attribute.

   - If TRUE, then a non–allomorphic event report will be issued. The managedObjectClass parameter of the event report will contain the value of the actual class of the managed object, not an allomorphic class.

o If FALSE, then the value of the switchMOCTo attribute is compared to the value of the validAllomorphs attribute of the potential event report.

(switchMOCTo) NON–NULL INTERSECTION (validAllomorphs)

– If TRUE, then an allomorphic event report will be issued.

The candidate values for insertion into the managedObjectClass field of the allomorphic event report are the result of a logical operation. The result of the operation is a set of one or more elements, where each element corresponds to a candidate allomorphic class for insertion:

(switchMOCTo) LOGICAL INTERSECTION (validAllomorphs)

If multiple elements result from the operation, then it is a local implementation option to choose one of the elements.

– If FALSE, then a non–allomorphic event report will be issued. The managedObjectClass parameter of the event report will contain the value of the actual class of the managed object, not an allomorphic class.

For example, assuming that

– object A belongs to the object class mocA, object B belongs to mocB, and so on.

– mocA is a superclass of mocB, mocB is a superclass of mocC, and so on.

The EFD T1 at the managing system performs the filtering based on its discriminatorConstruct which has a test for managedObjectClass = mocA, and forwards the event reports that passed to the manager application MgrApplT. The manager system can have some other local mechanism for handling event reports in a similar fashion.

If the switchMOCTo attribute value of { mocA } is specified for an allomorphismSensitiveEFD instance T2 at the agent, then the notifications from objects E and D will be forwarded to MgrAppl T as allomorphic event reports. Notifications from object A are forwarded to MgrAppl T as non–allomorphic event reports.

3.3.2 Create operation - Case 1

A manager that only understands the eventForwardingDiscriminator class and not allomorphismSensitiveEFD will issue an M–CREATE operation with the parameter,

managedObjectClass = eventForwardingDiscriminator

If the agent supports allomorphismSensitiveEFD, then the agent creates an extended managed object and sets attributes as follows:

objectClass = allomorphismSensitiveEFD

allomorphs = { eventForwardingDiscriminator }

Where the brackets { } denote a set. The agent issues an CREATE response that includes the parameter:

managedObjectClass = allomorphismSensitiveEFD

Since the manager requested the creation of a managed object of class eventForwardingDiscriminator, but was told by the agent that the class is allomorphismSensitiveEFD, the manager knows that the managed object is acting allomorphically, and can be managed as an instance of eventForwardingDiscriminator. If the manager wishes further verification, it can perform a GET operation to retrieve the value of the allomorphs attribute which will have a value of { eventForwardingDiscriminator }.

3.3.3 Create operation - Case 2

A manager that understands allomorphismSensitiveEFD will issue an M–CREATE operation, with the parameter:

managedObjectClass = allomorphismSensitiveEFD

The agent will create an instance of allomorphismSensitiveEFD, and sets attributes as follows:

objectClass = allomorphismSensitiveEFD

allomorphs = { eventForwardingDiscriminator }

The agent issues an M–CREATE response with the parameter:

managedObjectClass = allomorphismSensitiveEFD

3.3.4 Delete operation

For a manager to delete an instance of an extended managed object of allomorphismSensitiveEFD it need to know only the distinguished name. The manager will issue an M–DELETE operation, with the parameter:

baseManagedObjectClass = eventForwardingDiscriminator or

baseManagedObjectClass = allomorphismSensitiveEFD or

baseManagedObjectClass = ACTUAL or

baseManagedObjectClass = any class listed in the allomorphs attribute for which the operation is valid.

The agent will then delete the managed object.

For scoped operations, each allomorphismSensitiveEFD managed object that falls within the specified scope that meets the filter criteria, and has an active name binding that permits deletes will be deleted.

3.3.5 GET with no attributes (Scope="base object" only) - Case 1

If the manager only understands eventForwardingDiscriminator, then it wants to retrieve only those attributes of the extended managed object that apply to eventForwardingDiscriminator, and not to allomorphismSensitiveEFD. The manager requests an M–GET operation, with the parameters:

baseManagedObjectClass = eventForwardingDiscriminator and

scope = base object (or is absent and defaults to base object).

The extended managed object acts allomorphically, and returns in the M–GET response the attribute identifiers and either values/error indications of eventForwardingDiscriminator, and not those of allomorphismSensitiveEFD.

3.3.6 GET with no attributes (Scope = "base object" only) -Case 2

If a manager understands allomorphismSensitiveEFD, then it wants to retrieve all of the attributes of the managed object. The manager requests an M–GET operation, with the parameter:

baseManagedObjectClass = allomorphismSensitiveEFD or

baseManagedObjectClass = ACTUAL.

The managed object acts as a member of its actual class, and returns in the M–GET response the attribute identifiers and either values/error indications of allomorphismSensitiveEFD.

3.3.7 GET with no attributes (Scoped operation) - Case 1

If a manager only understands eventForwardingDiscriminator, and it wants to retrieve all attributes from all managed objects that it considers members of the eventForwardingDiscriminator class in a scoped operation, then it issues an M–GET operation, with the parameters:

baseManagedObjectClass = System (for example) and

scope = first level only, or whole subtree, or individual levels, or base to nth level.

The manager must specify as a value for the M–GET Filter parameter the following:

((managedObjectClass Equal eventForwardingDiscriminator)
OR
(non–null set intersection ({eventForwardingDiscriminator}, allomorphs)))

Note: **[**Please note that the allomorphs refers to the attribute inherited from top. This is a different attribute than validAllomorphs.**]**

Note: **[**Agents that conform to this ensemble will not create instances of eventForwardingDiscriminator, only instances of allomorphismSensitiveEFD.**]**

Therefore, when instances of allomorphismSensitiveEFD within the scope of the request apply the filter, the filter will resolve to true as follows:

((managedObjectClass Equal eventForwardingDiscriminator)
OR
(non–null set intersection ({eventForwardingDiscriminator}, allomorphs)))

Resolves to: (false) or (true) ––> true

The allomorphismSensitiveEFD managed objects will not act allomorphically as eventForwardingDiscriminator managed objects, but as members of their actual class, allomorphismSensitiveEFD. The manager will know that all of the objects that are responding are either members of or are compatible to the eventForwardingDiscriminator class by the virtue of how the CMIP filter was constructed on the request. Managed objects of allomorphismSensitiveEFD will return attribute identifiers and either values/error conditions of allomorphismSensitiveEFD. The manager will receive the managedObjectClass parameter equal to allomorphismSensitiveEFD in the linked replies from the agent, and must not discard the linked replies because of the presence of this parameter value. In addition, the manager must gracefully handle the unexpected information or attributes. For example, the switchToMOC attribute value.

3.3.8 GET with no attributes (Scoped operation) - Case 2

If a manager understands allomorphismSensitiveEFD, and it wants to retrieve all attributes from all managed objects that it considers members of allomorphismSensitiveEFD in a scoped operation, then it issues an M–GET operation, with the parameters:

baseManagedObjectClass = System (for example) and

scope = first level only, or whole subtree, or individual levels, or base to nth level.

To retrieve all attributes from all managed objects of allomorphismSensitiveEFD, then the manager must specify as a value for the M–GET Filter parameter the following:

(managedObjectClass Equal allomorphismSensitiveEFD)

The managed objects that meet this filter will act as members of their actual class, allomorphismSensitiveEFD. The manager will know that all of the objects that are responding are members of allomorphismSensitiveEFD. Managed objects of allomorphismSensitiveEFD will return attribute identifiers and either values/error conditions of allomorphismSensitiveEFD.

3.3.9 Replace Attribute Value operation

For this operation, the extended managed object only acts as a member of its actual class, allomorphismSensitiveEFD. Therefore, the manager issues an M–SET operation, with the parameter:

baseManagedObjectClass = eventForwardingDiscriminator or

baseManagedObjectClass = allomorphismSensitiveEFD or

baseManagedObjectClass = ACTUAL or

baseManagedObjectClass	=	any managed object class listed in the allomorphs attribute for which the operation is valid.

The extended managed object performs the operation as allomorphismSensitiveEFD.

For scoped operations, each allomorphismSensitiveEFD managed object that falls within the specified scope that meets the filter criteria will perform the operation as allomorphismSensitiveEFD.

3.3.10 Replace-with-default value operation

For this operation, the extended managed object only acts as a member of its actual class, allomorphismSensitiveEFD. Therefore, the manager issues an M–SET operation, with the parameter:

baseManagedObjectClass	=	eventForwardingDiscriminator or
baseManagedObjectClass	=	allomorphismSensitiveEFD or
baseManagedObjectClass	=	ACTUAL or
baseManagedObjectClass	=	any managed object class listed in the allomorphs attribute for which the operation is valid.

The extended managed object replaces the attribute values with the default values of allomorphismSensitiveEFD.

For scoped operations, each allomorphismSensitiveEFD managed object that falls within the specified scope that meets the filter criteria will perform the operation as allomorphismSensitiveEFD.

3.3.11 Add member operation

For this operation, the extended managed object only acts as a member of its actual class, allomorphismSensitiveEFD. Therefore, the manager issues an M–SET operation, with the parameter:

baseManagedObjectClass	=	eventForwardingDiscriminator or
baseManagedObjectClass	=	allomorphismSensitiveEFD or
baseManagedObjectClass	=	ACTUAL or
baseManagedObjectClass	=	any managed object class listed in the allomorphs attribute for which the operation is valid.

The extended managed object performs the operation as allomorphismSensitiveEFD.

For scoped operations, each allomorphismSensitiveEFD managed object that falls within the specified scope that meets the filter criteria will perform the operation as allomorphismSensitiveEFD.

3.3.12 Remove member operation

For this operation, the extended managed object only acts as a member of its actual class, allomorphismSensitiveEFD. Therefore, the manager issues an M–SET operation, with the parameter:

baseManagedObjectClass	=	eventForwardingDiscriminator or
baseManagedObjectClass	=	allomorphismSensitiveEFD or
baseManagedObjectClass	=	ACTUAL or
baseManagedObjectClass	=	any managed object class listed in the allomorphs attribute for which the operation is valid.

The extended managed object performs the operation as allomorphismSensitiveEFD.

For scoped operations, each allomorphismSensitiveEFD managed object that falls within the specified scope that meets the filter criteria will perform the operation as allomorphismSensitiveEFD.

3.3.13 Notifications

Instances of allomorphismSensitiveEFD emit notifications as they are defined for allomorphismSensitiveEFD. AllomorphismSensitiveEFD does not introduce additional notifications over the eventForwardingDiscriminator. Therefore, every notification that an instance of allomorphismSensitiveEFD emits will be accompanied at the managed object boundary with {eventForwardingDiscriminator } as the list of valid allomorphs for the notification.

3.4 Management Information References (and Definitions)

This section references all the definitions of management information relevant to the ensemble. The definitions may be provided as references to other documents which contain gdmo specifications. This section may contain references to definitions that are relevant to the ensemble. Thus, this section also contains statements about any additional restrictions or constraints to those definitions.

This ensemble departs from standard ensemble format, and defines the GDMO specification of the allomorphismSensitiveEFD here.

3.4.1 Managed Object Classes

3.4.1.1 allomorphismSensitiveEFD

```
allomorphismSensitiveEFD MANAGED OBJECT CLASS
   DERIVED FROM
      "CCITT REC. X.721 (1992)|ISO/IEC 10165–2:1992"
                   :eventForwardingDiscriminator;
   CHARACTERIZED BY
      allomorphismSensitiveEFDpkg;
REGISTERED AS {xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx}
```

3.4.2 Packages

3.4.2.1 allomorphismSensitiveEFDpkg

```
allomorphismSensitiveEFDpkg PACKAGE
   BEHAVIOUR
      allomorphismSensitiveEFDBhv;
   ATTRIBUTES
      switchMOCTo
         REPLACE–WITH–DEFAULT
         DEFAULT VALUE ASEFDmodule.emptySet
         GET
         ADD–REMOVE;
   REGISTERED AS {xxxxxxxxxxxxxxxxxxxxxxxxxxxxx }
```

3.4.3 Attributes

3.4.3.1 switchMOCTo

```
switchMOCTo  ATTRIBUTE
   WITH ATTRIBUTE SYNTAX
      ASEFDmodule.SetOfManagedObjectClasses;
   MATCHES FOR
      EQUALITY,
      SET–COMPARISON,
      SET–INTERSECTION;
   BEHAVIOUR
      switchMOCToBhv;
REGISTERED AS {xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx}
```

3.4.3.2 validAllomorphs

```
validAllomorphs  ATTRIBUTE
   WITH ATTRIBUTE SYNTAX
      ASEFDmodule.SetOfManagedObjectClasses;
   MATCHES FOR
      EQUALITY,
      SET–COMPARISON,
      SET–INTERSECTION;
   BEHAVIOUR
      validAllomorphsBhv;
REGISTERED AS {xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx}
```

3.4.4 Behaviours

3.4.4.1 allomorphismSensitiveEFDBhv

allomorphismSensitiveEFDBhv BEHAVIOUR
DEFINED AS

"

An instance with this behaviour provides a deterministic mechanism for an agent to provide allomorphic event reports to a manager. Allomorphic event reports differ from non–allomorphic event reports only in the value of the managedObjectClass parameter of the event report. An allomorphic event report will contain a valid allomorphic class in the managedObjectClass parameter. A non–allomorphic event report will contain the actual class of the managed object in the managedObjectClass parameter. The information content of the event report will be exactly that defined in the managed object class definition for the managed object that emitted the notification, i.e. it is not modified as a consequence of allomorphism.

An instance with this behaviour realizes allomorphic event reports by being able to operate on the validAllomorphs attribute of a potential event report. The validAllomorphs attribute value is mapped from the set of valid allomorphic classes for which the notification is defined. The set of valid allomorphic classes for which the notification is defined is made available by a managed object acting allomorphically, in conjunction with the notification at the managed object boundary. An instance with this behaviour decides whether an allomorphic event report, or alternatively, a non–allomorphic event report is issued.

An instance with this behaviour takes the following processing steps in determining if an allomorphic event report should be emitted if the processing of the discriminator Construct attribute resolves to true:

1. determine if the value of the managedObjectClass attribute of the potential event report is a set element of the switchMOCTo attribute.

   o If TRUE, then a non–allomorphic event report will be issued. The managedObjectClass parameter of the event report will contain the value of the actual class of the managed object, not an allomorphic class.

   o If FALSE, then proceed to the next step

2. compare the value of the switchMOCTo attribute to the value of the validAllomorphs attribute of the potential event report.

   (switchMOCTo) NON–NULL INTERSECTION (validAllomorphs)

   o If TRUE, then an allomorphic event report will be issued. Proceed onto the next step.

   If FALSE, then a non–allomorphic event report will be issued. The managedObjectClass parameter of the event report will contain the value of the actual class of the managed object, not an allomorphic class.

3. The candidate values for insertion into the managedObjectClass field of the allomorphic event report are the result of a logical operation. The

result of the operation is a set of one or more elements, where each element corresponds to a candidate allomorphic class for insertion:

(switchMOCTo) LOGICAL INTERSECTION (validAllomorphs)

If multiple elements result from the operation, then it is a local implementation option to choose one of the elements. An instance of this behaviour supports discriminating on a number of attributes mapped from notification parameters:

Table 3-3. Minimum PER Attributes required by the Profile

attribute	Object Identifier
managedObjectClass	{smi2AttributeID 60}
eventType	{smi2AttributeID 14}
managedObjectInstance	{smi2AttributeID 61}
perceivedSeverity	{smi2AttributeID 17}
securityAlarmSeverity	{smiAttributeID 23}
validAllomorphs	{XXXXXXXXXXXXXXXXXX}

Other attributes derived from notifications must be specified as part of the GDMO NOTIFICATION template constructs of WITH INFORMATION SYNTAX and AND ATTRIBUTE IDS.

Unless otherwise specified, the allomorphs attribute cannot be set from a value specified by an explicit CREATE operation.
";

3.4.4.2 switchMOCToBhv

switchMOCToBhv BEHAVIOUR
DEFINED AS

" The value of an attribute with this behaviour indicates managed object classes that are eligible to be placed into the managedObjectClass parameter of an event report. ";

3.4.4.3 validAllomorphsBhv

validAllomorphsBhv BEHAVIOUR
DEFINED AS

" The value of an attribute with this behaviour is mapped from the set of valid allomorphic classes for which the notification is defined. The set of valid allomorphic classes for which the notification is defined is made available by a managed object acting allomorphically, in conjunction with a notification at the managed object boundary. ";

3.4.5 ASN.1 Syntax Definitions

```
--
-- Allomorphism Sensitive Event Forwarding Discriminator
-- Ensemble
--
-- ASN.1 Module Definitions
--

ASEFDmodule {XXXXXXXXXXXXXXXXX}


          DEFINITIONS ::= BEGIN

                    -- EXPORTS everything

                    SetOfManagedObjectClasses ::= SET OF OBJECT IDENTIFIER

                    -- This ASN.1 is designed to negate the use of the
                    -- localForm of ObjectClass.

                    emptySet SetOfManagedObjectClasses ::= {}

          END
```

4 Ensemble Conformance Requirements

4.1 General Conformance Requirements

The general conformance requirements for omnipoint 1 are specified in forum 020 – OMNIPoint 1 conformance requirements – Issue 1.0. All the conformance requirements identified in this part of the document are based on that document and Forum 025 – The "Ensemble" Concepts and Format – Issue 1.0.

In general, an implementation supporting this ensemble must prove conformance to:

- o all of the object classes representing the resources of the ensemble
- o all the functionality representing the management of the ensemble resources

The conformance requirements of an ensemble, either reference a set of existing ISPs (AOM2x OSI management–management functions), or define specific ensemble conformance requirements which are based on existing ISPs.

The conformance requirements are presented in a tabular fashion forming the implementation conformance statement (ICS) proformas.

An ensemble may also include other implementation conformance statement (ICS) proformas for components of the ensemble other than system management functions. These ICS proformas will also be specified in a tabular format.

The supplier of an implementation that claims conformance to this ensemble must complete these tables, indicating which options and capabilities have been implemented.

It is the proformas that identify which role (manager/agent) the implementation supporting this ensemble adopts.

The capabilities of the underlying object classes, ISP functions and management communication protocols that are not explicitly required for this ensemble are left "beyond the scope" of conformance to this ensemble.

4.2 Specific Conformance Requirements

This section presents the specific conformance requirements for this ensemble. The relationship of ensemble conformance to OSI management functions ISP conformance is discussed, and ensemble function support requirements are presented.

The detailed managed object conformance statements are provided in Annex B.

4.2.1 Common Conditions List Conventions

The table below lists the common conditions that are defined in other profiles and used within this ensemble:

NOTATION	DESCRIPTION
c1	Support of at least one of these options is required. This condition is specified in DISP 12059–0.
c2	Support of the feature in at least one management role is required. This condition is specified in DISP 12059–0.

4.2.2 Specific Conditions List Conventions

The table below lists the specific conditions that are uniquely defined for this ensemble:

NOTATION	DESCRIPTION
c70	Present if the ROIV–m–CREATE (sending) contained a value in the managedobjectclass parameter that differs from the actual class of the object that was created.
c71	If M–GET is supported, then M–CANCEL–GET is optional,else out of scope.

c72 If a name binding that supports create operations is supported, then M–CREATE is mandatory, else out of scope.

c73 If a name binding that supports delete operations is supported, then M–DELETE is mandatory, else out of scope.

c74 Present if the ROIV–m–GET (sending) contained EFD or a compatible class listed in the allomorphs attribute as the value for the baseManagedObjectClass parameter

4.2.3 OSI Management Functions Profiles Conformance

The table below, lists all the current ISPs and identifies which profiles are required to be supported when the implementation adopts a manager or agent role.

The following notation convention has been used:

NOTATION	DESCRIPTION
m	defines a mandatory requirement
i	stands for out–of–scope

Table 4-1. Ensemble functional ISP conformance requirements

ISP Supported	Manager role	Agent Role
AOM211 - General Management Capabilities	i	i
AOM212 - Alarm Reporting and State Management Capabilities	i	i
AOM213 - Alarm Reporting Capabilities	i	i
AOM221 - General Event Report Management	i	i
AOM231 - General Log Control Management	i	i

4.2.4 Ensemble Functions Conformance

The table below lists all of the ensemble functions, and identifies which are mandatory, optional or conditional in the manager or agent roles.

The following notation convention has been used:

NOTATION	DESCRIPTION
m	defines a mandatory requirement
o	defines an optional requirement
c	defines a conditional requirement

Table 4-2 Ensemble Function Requirements

Ensemble Specific Functions	Manager Role	Agent Role
allomorphism Sensitive EFD function	m	m

4.2.5 Management Conformance Summary

Table 4-3. System Conformance Statement/Management Conformance Summary

Index	Ident.	Ident. of Std.	MO Class Label / MOCS Proforma	Base	Profile	Additional Info
4.3.1	CMIP	ISO/IEC 9596-1	ISO/IEC 9596-2	-	m	
4.3.2	ROSE	ISO/IEC 9072-2	ISO/IEC 9596-2	-	m	
4.3.3	ACSE	ISO/IEC 8650	ISO/IEC 8650-2	-	m	
4.3.4	Pres.	ISO/IEC 8823	ISO/IEC 8823-2	-	m	
4.3.5	Sess.	ISO/IEC 8827	ISO/IEC 8827-2	-	m	

4.2.6 Management Capability Support/SMFUs Support

Table 4-4. Management Capability Support/SMFU Support Summary

Index	Functional Unit	Base Name	MAPDU Standard	CMIPDU Support	Profile Indexed by CMIS
4.4.1	-	-	-	-	-

4.2.7 MOCS Proforma For Ensemble Managed Object Classes

Table 4-5. MOCS Proforma for Ensemble MO classes

Index	Class Name	Base Standard		Profile	
		Manager role	Agent role	Manager role	Agent role
4.5.1	allomorphism SensitiveEFD	-	-	c2	c2

c2 - support of the feature in at least one management role is required

4.2.8 Association Initiator/Responder

Table 4-6. Association Initiator/Responder

Capability	Base Standard		Profile	
	Initiator	Responder	Initiator	Responder
What type of association does the implementation support?	c1	c1	c1	c1

4.2.9 CMIS Services (CMIP pdu) Requirements

Table 4-7. Manager CMIS Services (CMIP PDU) Requirements

Index	CMIS Service	pDISP 12059-0 Draft 5.0 Table Reference		Conditions mandated relevant to ISP 11183-2
		Manager Role	Profile	
4.7.1	M-GET	Table 13	c1	none
4.7.2	M-SET	Table 15	c1	none
4.7.3	M-CREATE	Table 7	c1	none
4.7.4	M-EVENT-RPT	Table 11	c1	none
4.7.5	M-CANCEL-GET	Table 5	c71	none
4.7.6	M-DELETE	Table 9	c1	none

c71 – If M–GET is supported, then M–CANCEL–GET is optional, else out of scope.

Support for modified ISP 11183–2 tables as defined in 4.2.9.1 is required for the supported CMIS services.

Table 4-8. Agent CMIS Services (CMIP PDU) Requirements

Index	CMIS Service	pDISP 12059-0 Draft 5.0 Table Reference		Conditions mandated relevant to ISP 11183-2
		Agent Role	Profile	
4.8.1	M-GET	Table 14	m	none
4.8.2	M-SET	Table 16	m	none
4.8.3	M-CREATE	Table 8	c72	none
4.8.4	M-EVENT-RPT	Table 12	m	none
4.8.5	M-CANCEL-GET	Table 6	c71	none
4.8.6	M-DELETE	Table 10	c73	none

c71 – If M–GET is supported, then M–CANCEL–GET is optional, else out of scope.

c72 – If a name binding that supports CREATE operations is supported, then M–CREATE is mandatory, else out of scope.

c73 – If a name binding that supports DELETE operations is supported, then M–DELETE is mandatory, else out of scope.

Support for modified ISP 11183–2 tables as defined in 4.2.9.1 is required for the supported CMIS services.

4.2.9.1 Modifications To ISP 11183-2 Tables

This ensemble specifies the use of the protocol elements of CMIP. The requirements are stated by reference to tables in the general CMIP Profile ISP 11183–2. The following tables modify the tables in ISP 11183–2 for the purposes of this ensemble.

Abbreviation	Description
EFD	denotes the eventForwardingDiscriminator class.
ASEFD	denotes the allomorphismSensitiveEFD class. Managed objects of this class are compatible with the eventForwardingDiscriminator managed object class.
ACTUAL	refers to the "actual class", as documented in clause 7.4.4 of GDMO.

4.2.9.1.1 ROIV-m-Create (sending)

Table 4-9. Modifications to ISP 11183-2, Table 14

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
14.4.1	managedObject Class	m	mm	mm	(3)

(3) – The parameter is either ASEFD or a class which is compatible with an instantiation of ASEFD. EFD is a compatible class to an instance of ASEFD.

4.2.9.1.2 ROIV-m-Create (Receiving)

Table 4-10. Modifications to ISP 11183-2, Table 15

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
15.4.1	managedObject Class	m	mm	mm	(3)

(3) – The following values must be statically supported:
- EFD
- ASEFD

Note: **[**Other values of compatible classes that are supported by the receiving implementation may also be specified.**]**

4.2.9.1.3 ROIV-m-Delete (sending)

Table 4-11. Modifications to ISP 11183-2, Table 16

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
16.4.1	baseManaged ObjectClass	m	mm	mm	(2)

(2) – The parameter must take one of the following values when scope = baseObject only:
- EFD
- ASEFD
- ACTUAL or any compatible class listed in the allomorphs attribute

4.2.9.1.4 ROIV-m-Delete (receiving)

Table 4-12. Modifications to ISP 11183-2, Table 17

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
17.4.1	baseManaged ObjectClass	m	mm	mm	(2)

(2) – The following values must be statically supported when scope = baseObject only:

– EFD
– ASEFD
– ACTUAL

Note: **[**Other values of compatible classes that are listed in the allomorphs attribute may also be specified.**]**

4.2.9.1.5 ROIV-m-Get (sending)

Table 4-13. Modifications to ISP 11183-2, Table 22

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
22.4.1	baseManaged ObjectClass	m	mm	mm	

Note: **[**For an allomorphic operation with scope = baseObject only, the value can be any compatible class listed in the allomorphs attribute. The RORS-m-Get (sending) will contain only the attribute identifiers and values for the requested class.**]**

4.2.9.1.6 ROIV-m-Get (receiving)

Table 4-14. Modifications to ISP 11183-2, Table 23

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
23.4.1	baseManaged ObjectClass	m	mm	mm	

Note: **[**For an allomorphic operation with scope = baseObject only, the value can be any compatible class listed in the allomorphs attribute. The RORS-m-Get (sending) will contain only the attribute identifiers and values for the requested class.**]**

4.2.9.1.7 ROIV-m-LinkedReply-Delete (sending)

Table 4-15. Modifications to ISP 11183-2, Table 26

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
26.4.1.1	managedObject Class	m	mm	mm	(2)
26.4.2.1	managedObject Class	m	mm(1)	mm(1)	(2)
23.4.3.1	managedObject Class	m	mm(1)	mm(1)	(2)

(2) – The value of this parameter is the value of the objectClass attribute.

4.2.9.1.8 ROIV-m-LinkedReply-Get (receiving)

Table 4-16. Modifications to ISP 11183-2, Table 28

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
28.4.1.1	managedObject Class	m	mm(1)	mm(1)	(2)
28.4.2.1	managedObject Class	m	mm(1)	mm(1)	(2)
28.4.1	managedObject Class	m	mm(1)	mm(1)	(2)

(2) – The value of this parameter is the value of the objectClass attribute.

4.2.9.1.9 ROIV-m-LinkedReply-Set (sending)

Table 4-17. Modifications to ISP 11183-2, Table 30

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
30.4.1.1	managedObject Class	m	mm(1)	mm(1)	(4)
30.4.2.1	managedObject Class	m	mm(1)	mm(1)	(4)
30.4.3.1	managedObject Class	m	mm	mm	(4)

(4) – The value of this parameter is the value of the objectClass attribute.

4.2.9.1.10 ROIV-m-Set (sending)

Table 4-18. Modifications to ISP 11183-2, Table 32

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
32.4.1	baseManaged ObjectClass	m	mm	mm	(3)

(3) – The following values must be statically supported when scope = baseObject only:
- EFD
- ASEFD
- ACTUAL or any compatible class listed in the allomorphs attribute for which the operation is valid.

4.2.9.1.11 ROIV-m-Set (receiving)

Table 4-19. Modifications to ISP 11183-2, Table 33

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
33.4.1	baseManaged ObjectClass	m	mm	mm	(3)

(3) – The following values must be statically supported when scope = baseObject only:
- EFD
- ASEFD
- ACTUAL or any compatible class listed in the allomorphs attribute for which the operation is valid.

4.2.9.1.12 ROIV-m-Set-Confirmed (sending)

Table 4-20. Modifications to ISP 11183-2, Table 34

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
34.4.1	baseManaged ObjectClass	m	mm	mm	(3)

(3) – The following values must be statically supported when scope = baseObject only:
- EFD
- ASEFD
- ACTUAL or any compatible class listed in the allomorphs attribute for which the operation is valid.

4.2.9.1.13 ROIV-m-Set-Confirmed (receiving)

Table Table 4-21. Modifications to ISP 11183-2, Table 35

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
35.4.1	baseManaged ObjectClass	m	mm	mm	(3)

(3) – The following values must be statically supported when scope = baseObject only:
- EFD
- ASEFD
- ACTUAL or any compatible class listed in the allomorphs attribute for which the operation is valid.

4.2.9.1.14 RORS-m-Create (sending)

Table 4-22. Modifications to ISP 11183-2, Table 40

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
40.3	CreateResult	m	mo	mc70	
40.3.1	managedObject Class	m	oo	mc70	(2)

(2) – The parameter value must take the value of the objectClass attribute

C70 – present if the ROIV–m–CREATE (sending) contained a value in the managedObjectClass parameter that differs from the actual class of the object that was created.

4.2.9.1.15 RORS-m-Delete (sending)

Table 4-23. Modifications to ISP 11183-2, Table 42

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
42.3.1	managedObject Class	o	oo(2)	oo(2)	(2)

(2) – The parameter value must take the value of the objectClass attribute

4.2.9.1.16 RORS-m-Get (sending)

Table 4-24. Modifications to ISP 11183-2, Table 46

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
46.3	GetResult	m	mo	mc74	
46.3.1	managedObject Class	o	oo(2)	mc74(2)	(5)
46.3.4	attributeList	m	mm(3)	mm(3)	(6)

c74 – present if the ROIV–m–Get (sending) contained EFD or a compatible class listed in the allomorphs attribute as the value for the baseManagedObjectClass parameter.

(5) – The value of this parameter is the value of the objectClass attribute

(6) – the attributeList only contains the set of attributeId and attributeValue pairs defined for requested compatible class. The requested compatible class is specified in the ROIV–m–Get (sending) baseManagedObjectClass parameter, and must be listed in the allomorphs attribute.

4.2.9.1.17 RORS-m-Set-Confirmed (sending)

Table 4-25. Modifications to ISP 11183-2, Table 48

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
48.3.1	managedObject Class	o	oo(2)	oo(2)	(3)

(3) – The parameter value must take the value of the objectClass attribute

4.2.9.1.18 ROER-classInstanceConflict (sending)

Table 4-26. Modifications to ISP 11183-2, Table 52

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
52.3.1	baseManaged ObjectClass	m	mm	mm	(1)

(1) – The value of this parameter is the same as was present on the invoking operation.

4.2.9.1.19 ROER-getListError (sending)

Table 4-27. Modifications to ISP 11183-2, Table 58

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
58.3.1	managedObject Class	o	oo(1)	mc74(1)	(2)
58.3.4.1.2	attributeId	m	mm	mm	(3)
58.3.4.2.1	attributeId	m	mm	mm	(3)

(2) – The value of this parameter is the value of the objectClass attribute

(3) – only attributeId values defined for the requested compatible class are present if:

- scope = baseObject only
- the requested compatible class that is specified in the ROIV–m–Get (sending) baseManagedObjectClass parameter is listed in the allomorphs attribute
- the value of the errorStatus parameter is 2 (accessDenied)
- no attributes were specified in the attributeIdList on the ROIV–m–Get (sending)

c74 – The managedObjectClass parameter shall be present if the ROIV–m–GET (sending) contained EFD or a compatible class listed in the allomorphs attribute as the value for the baseManagedObjectClass parameter.

4.2.9.1.20 ROER-noSuchObjectClass (sending)

Table 4-28. Modifications to ISP 11183-2, Table 84

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
84.3	ObjectClass	m	mm	mm	(1)

(1) – The parameter value is the same as was present on the invoking operation

4.2.9.1.21 ROER-processingFailure (sending)

Table 4-29. Modifications to ISP 11183-2, Table 92

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
92.3.1	managedObject Class	m	mm	mm	(1)

(1) – The value of this parameter is the value of the objectClass attribute

4.2.9.1.22 ROER-setListError (sending)

Table 4-30. Modifications to ISP 11183-2, Table 94

ISP 11183-2 Index	Parameter name	Base std.	ISP 11183-2	Ensemble	Type, value(s) & range(s)
94.3.1	managedObject Class	o	oo(3)	oo(3)	(4)

(4) – The value of this parameter is the value of the objectClass attribute

D.4 Service Request Management Ensemble

Editor's Note: **[**Because the Service Request Management Ensemble is intended to be a self-contained, standalone document, the clauses and subclauses of the Service Request Management Ensemble (as shown here in Annex D.4) are numbered as they would be in a separate, standalone document, and not as they would be according to their position in Annex D.4.**]**

SERVICE REQUEST MANAGEMENT ENSEMBLE - DRAFT 3

Table of Contents

List of Figures

List of Tables

REVISION HISTORY

Issue 1, Draft 1, December 1992

Issue 1, Draft 2, February 1993 - the major changes in this draft were the incorporation of review comments, expanding and revising the text from Draft 1, an attempt to broaden the scope of the ensemble to support more than just network services, and the addition of draft text to Sections 2.1 and 2.2.

Issue 1, Draft 3, March 1993 - the changes in this draft were the incorporation of review comments obtained and discussed in the March 1993 OIW meeting.

1. INTRODUCTION

Ensembles provide a top down view of a particular solution to a management problem. In order to focus on the solution to this management problem, specific restrictions are placed upon particular referenced definitions.

The concepts and format of Ensembles are described in the "NM Forum Ensemble Concepts and Format" [n1] specification document.

This Ensemble, wherever possible, references documents which define the components of the Ensemble.

The management problem is identified as a set of requirements and constraints. In defining the solution to this management problem, the resources to be managed, the functions to be applied, and the scenarios describing the interactions are all identified. The Ensemble references base standards and International Standardized Profiles (ISPs). It also references libraries containing definitions expressed by GDMO (Guidelines for the Definition of Managed Objects [n2]) templates.

The purpose of this document is to collect management information definitions and profiles, and show how they can be applied to manage the resources identified in this Ensemble.

This document is organized as follows:

Section 1, "General Information", provides a high level overview describing the Ensemble and the structure of the document.

Section 2, "Management Context", identifies the managed resources and management capabilities of the Ensemble.

Section 3, "Information Model", specifies all management information components of this Ensemble.

Section 4, "Ensemble Conformance Requirements", provides or references statements of conformance for this Ensemble. The Managed Object Conformance Proformas that are specific to this Ensemble are provided in Annex B.

1.1 UNIQUE IDENTITY

The unique identity is a registered object identifier used to identify this Ensemble.

Editor's Note: [identity to be provided**]**

1.2 GENERAL DESCRIPTION

This Ensemble specifies the managed objects and the application functions that define a service request interface between a provider and a customer. Such capabilities allow a customer to submit a service request to a provider, exchange information regarding the requrest, modify the request, obtain periodic information on the status of a request, and be notified by the provider that a request has been satisfied.

This ensemble specifies a standardized means for a customer to request, change, and track services provisioned by a service provider. For example, a customer contracts with a provider to supply services upon request, i.e., to provision or allocate the resources necessary to provide the elements of the services. This ensemble defines a standard customer/provider interface that specifies how a customer requests elements of the contracted (i.e., pre–authorized) service and is informed of its status. This ensemble addresses the customer's view of the customer/provider interface for processing service requests.

Many of the terms used in this Ensemble (e.g., service request, service, goods, user, etc.) have different meanings to different readers. Therefore, to set the context for the scope, purpose, requirements to be satisfied, and functions needed for this Ensemble, a number of terms are defined below and are defined from a user perspective.

For the purposes of this ensemble the following definitions apply:

– Service Request – a request for the provisioning of one or more services, connections, and goods to one or more users.

– Service – a specific functionality available to one or more users. Examples of the types of services that could be requested include electronic mail, voice mail, user privileges (e.g., long distance access, file access, and security privileges), video and teleconferencing, and application usage (e.g., SNA). (Note: this list should not be construed to be all inclusive of the services that could be requested. In fact, it is expected that the list of possible services will be continually changing and may span several other areas of information technology and possibly maintenance services.) In this Ensemble, the term service is not intended to represent OSI Layer Service Access Points.

– Connection – refers to a user's access (attachment) to a network. Examples of the types of connections that could be requested include dedicated leased lines, voice connections, packet switched services (e.g., X.25, frame relay, or ATM), LAN connections, and multidrop connections. (Note: this list should not be construed to be all inclusive of the connections that could be requested. In fact, it is expected that the list of possible connections will be continually changing and may span several other areas of information technology.)

– Goods – refers to physical items. These physical items may be necessary to provide services and connections. Examples of the types of goods that could be requested include equipment/hardware (e.g., muxes, switches, modems, bridges, routers, cables, computers and peripheral supplies, phone sets, encryption devices, and network interface cards), software, and people. (Note: these lists should not be construed to be all inclusive of the goods that could be requested. In fact, it is expected that the list of possible goods will be continually changing and may span several other areas of information technology.)

– Customer – a corporation, organization, or individual with needs to be satisfied by some services, connections, and goods. A customer is the procurement agent for some group of users.

– Requester – a requester is a person or process authorized to submit a specific service request on behalf of a user.

– User – a person or process that uses services, connections, and goods.

– User device – a resource to which a specific service is delivered. Not all services require an end user device.

– Provider – an organization responsible for supplying some service, connection, or goods that are visible to management. Services, connections, and goods provided may be tariffed or non–tariffed, public or private, and may be provided to one or more customers. The same organization can be both a customer and a provider.

Editor's Note: **[**From comments from BT: In Section 1.2 (or somewhere else Scope ?? Context ??), a couple of diagrams would be useful, perhaps showing the 'requester-provider' relationship.**]**

1.3 SCOPE AND PURPOSE

Ensembles represent specific solutions to particular problems. Thus, an Ensemble is a complete description of the problem and the solution to that problem.

This section describes the requirements of the problem. It includes the definition of the information model that represents the solution to a problem. These definitions comprise references to one or more management information libraries that contain definitions of managed object classes expressed in GDMO templates, packages, attributes, name bindings, etc. Also included in the Ensemble definition are statements of conformance and suitable proformas.

The purpose of this Ensemble is to define a general purpose management service that will allow:

– A requester to submit a service request to a provider for the purpose of adding, modifying, or deleting a preauthorized service, connection, or goods

– A requester to submit a service request to a provider for the purpose of modifying or canceling an outstanding service request

– A requester to receive feedback on the status of a service request and pertinent implementation information

This Ensemble does not address:

– A customer's internal mechanism for tracking service requests

– The accounting, pricing, billing, or other contractual issues related to service, connection, and goods provisioning

1.4 RELATIONSHIPS WITH OTHER ENSEMBLES

This section identifies the relationships of this Ensemble to other Ensembles.

At this time, this Ensemble is not related to any other Ensembles.

2. MANAGEMENT CONTEXT

The "Management Context" describes why the Ensemble is required. The description of the "Management Context" includes the definition of the resources to be managed, the management functions to be performed, the scope of the problem to be solved, and the management view or level of abstraction from which the problem is to be approached. The influence of the Management Context on the Ensemble is shown in Figure 1.

```
                                             MANAGEMENT TOOLS
                                         {Standards: GDMO, Objects,
                                        System Management Functions,
                                               Profiles, ...}
                                                      |
                                                      V
    MANAGEMENT CONTEXT                  -----------------------------
                                       |           ENSEMBLE           |
                                       |                              |
VIEWPOINT                              | - Requirements               |
  ------------------------->           |                              |
{User, Provider, Element,              | - Scenarios                  |
     Network, ...}                     |                              |
                                       | - Resources                  |
                                       |                              |
RESOURCES                              | - Information Models         |
  ------------------------->           |                              |
{Equipment, Software,                  | - Entity Relationship        |
     Applications, ...}                |   Diagrams                   |
                                       |                              |
                                       | - Object Specifications      |
FUNCTIONS                              |                              |
  ------------------------->           | - Managed Object             |
{Fault, Configuration,                 |   Conformance Statements     |
      Performance, ...}                |                              |
                                       | - Ensemble Conformance       |
                                        -----------------------------
```

Figure ??. Management Context Overview

2.1 GENERAL INTRODUCTION

A general description for the steps involved in processing a service request is given below. Not all of the steps listed below will necessarily be required or taken for each request. In addition, steps 2 though 6 can occur in any order.

1. INITIATE A SERVICE REQUEST – A requester submits a request for a service, connection, or good.

2. EXCHANGE INFORMATION ABOUT A SERVICE REQUEST – Information exchange can happen zero or more times throughout the life of a service request and can be initiated by either the requester or the provider. Examples of information exchange are:

   – A provider may request clarification or additional information about a service request; in turn, the requester provides the desired information

   – A provider provides pricing, scheduling, or other implementation information concerning the service request

3. MODIFY (ADD TO, CHANGE, DELETE FROM, AND DELETE) AN OUTSTANDING SERVICE REQUEST – A requester initiates a modification to an outstanding service request

4. PROVIDER PROVISIONS SERVICE, CONNECTION OR GOODS – The provider designs and costs the requested service, connection, or good; orders required goods; schedules the provisioning activities; and provisions the service, connection, or goods. (Note: These functions are outside the scope of this Ensemble.)

5. GET STATUS INFORMATION – A customer requests status information from the provider

6. STATUS NOTIFICATIONS – A provider sends the customer status notifications when the status of a service requests changes

7. PROVISIONING COMPLETED – The provider completes all the necessary steps to provision the requested service, connection, or goods

Editor's Note: **[**Add a diagram depicting the steps described above. Also add text describing why the ensemble is required.**]**

2.2 MANAGEMENT VIEW AND LEVEL OF ABSTRACTION

This section indicates the management view of the Ensemble, which includes information on the level of abstraction. For example, in a hierarchically organized system, this section would indicate if the Ensemble deals with the management of equipment, the management of networks, or the management of services. It may also indicate the management perspectives and roles.

Editor's Note: **[**Add text describing whether the ensemble is from the user or provider point of view and the expected level of detail.**]**

The management view that this ensemble addresses is based on the interface between two (or more) cooperating management systems operating in some sort of requester–provider relationship, where the provider is to operate on a set of services, connections, and goods on behalf of the requester. The requester is able to monitor and control the progress of that order; and, where appropriate, to cancel or modify the order.

This requester–provider relationship is appropriate to an interface between any management system architecture or any interface between user and provider domains (as in the Reconfigurable Circuit Service Ensembles), and is not limited to the provisioning of network services. This model is not restricted to the layer, purpose of the interaction, or the services, connections, or goods affected.

Editor's Note: **[**State what the model is targeted toward.**]**

2.3 RESOURCES

This section defines all the resources or components of resource that are to be the subject of the Ensemble. The definition of the resources contains all of the resources and only those resources that are relevant to the Ensemble. The resources are defined by textual descriptions or by reference to other documents containing descriptions of the resources. When other documents are referenced, statements are provided to indicate any restrictions and constraints on those source definitions.

Editor's Note: **[**The resources to be managed are service requests. Possible structures for managed objects representing service requests include:

- A base service request managed object class with more detailed subclasses for different types of service requests or for requests for different types of services
- One (or more) base service request managed object class(es) with relationship/referential "pointers" to other classes providing more detailed description of the type of service request or the type of service requested
- Some combination of the approaches described above

Regardless of the approach, it is not the intent of this Ensemble to define every possible type of service that a customer might wish to request. However, it is the authors' intention to include the detailed definition of at least one service in this Ensemble to serve as an example of how other services may be defined.**]**

Editor's Note: **[**Comment from BT: The SRM mechanism should be capable of supporting any sort of request (order) for any sort of service, connection, or good. It is therefore important that the resources section does not specify service-specific resources. For this type of mechanism the resources involved should be the order itself, not the subject of the order. As listed in the BT contribution this could include:

- a resource defining the orders that the provider is capable of performing
- a resource defining the progress of an order
- a resource representing the changes to be made
- resources representing the real resources to be affected

These would provide a basic mechanism to be used in the ensemble which would support a wide range of possible resources, changes, etc.. The exact nature of these resources would need to be further defined, but see the BT contribution for more details.**]**

2.4 FUNCTIONS

This section defines the management functions that can be performed on the resources described in Section 2.3. These functions may be primitive functions defined for OSI systems management (e.g., event management), higher level functions for general network management (e.g., alarm surveillance), or other functions unique to the problem the Ensemble addresses.

These definitions consist of a brief textual description of each function. In some cases, these descriptions will include a set of references to other documents, for example:

ISO System Management Functions

Telecommunications Management Network (TMN) CCITT
M.3020 [4]

Other standards

When other documents are referenced, statements are required to indicate the restrictions and constraints to the function definitions in the Ensemble.

Editor's Note: **[**The figure below is included to provide an overview of the functions to be addressed by this Ensemble. Descriptions of these functions will be provided in a later draft.**]**

```
=============================================================

REQUESTER                                        PROVIDER


INITIATE A SERVICE REQUEST:

-----   Requester submits request for service      ---->
<----   Optionally, provider acknowledges request   -----


EXCHANGE INFORMATION ABOUT A SERVICE REQUEST:

<----   Provider requests clarification/           -----
        additional info
-----   Requester provides clarification/          ---->
        additional info
<----   Optionally, provider acknowledges          -----
        additional info


<----   Provider provides pricing, scheduling,     -----
        installation and other info
-----   Optionally, requester acknowledges/        ---->
        confirms information


MODIFY (ADD TO, CHANGE, DELETE FROM, AND DELETE) AN OUTSTANDING
SERVICE REQUEST:

-----   Requester submits request to modify an     ---->
        outstanding service request
<----   Optionally, provider acknowledges request   -----


GET STATUS INFORMATION:

-----   Requester requests status information      ---->
<----   Provider sends status response             -----


STATUS NOTIFICATIONS:

<----   Provider sends status (change)             -----
        notifications
-----   Optionally, requester acknowledges/        ---->
        confirms information
```

Figure ??. Overview of the Service Request Management Ensemble Functions

Editor's Note: **[**Comment from BT: The list of functions should include:

Both Asynchronous (Controlled) and Synchronous (Uncontrolled) functions:

- Create order
- Order rejected by performer
- Modify order
- Suspend/Resume order
- Report on order progress
- Monitor order progress
- Delete order
- Report on failure
- Report on completion (partial success and complete success)**]**

2.5 OTHER REQUIREMENTS

This section contains requirements not covered in functions, resources, or level of abstraction. For example, these may be business or implementation requirements.

Editor's Note: **[**Requirements related to security need to be addressed.**]**

3. MANAGEMENT INFORMATION MODEL

For the purposes of defining an Ensemble, an Information Model can be thought of as focusing on the real world under study. An information model contains information about both the elements of the model and the relationships between them. For a management information model the elements of management information are defined using GDMO and the relationships are graphically illustrated.

Editor's Note: **[**Comment from BT: This model could be very similar to the testing management type mechanism which allows a range of tests to be performed on a range of resources. This sort of mechanism should be applicable to the order handling type work. The classes will of course be different but it may save effort if the same principles were applied.**]**

Editor's Note: **[**This proposed approach requires further investigation. Testing model will be kept in mind, but there questions as to whether it is the best or most appropriate model for SRM.**]**

3.1 GENERAL INTRODUCTION

3.2 RELATIONSHIPS

This section defines the relationships among the components of the model. These may be expressed in Entity–Relationship (ER) diagrams or other similar graphic representations.

Three types of diagrams may be used:

- One for the relationships intrinsic to the underlying resources. In this representation of the model, the entities (resources represented by managed object classes) making up the Ensemble are identified along with the relationships between the entities.

- One for the relationships among the classes representing the resources.

- One for the naming schema. The naming model to be used by this ensemble is described, which is a subset of all possible naming relationships. This is expressed graphically and by listing references to those name bindings selected for use with the ensemble.

The management information described in this section is defined to have the following inter–relationships.

3.3 SCENARIOS

This section defines the scenarios associated with this Ensemble. The scenarios are used to show how the managed objects in the information model can be used to accomplish the function listed in section 2.4. The scenarios may be defined in the standards or defined specifically for the ensemble.

Each of the scenario definitions consist of a brief textual description and message flow diagrams. In some cases, these description will include a set of references to other documents. When other documents are referenced, statements are required to indicate the restrictions and constraints in this Ensemble to the function definitions in the referenced document.

In the scenarios that follow, CMIP flows between (and corresponding CMIS primitives within) manager and agent systems are indicated by arrows with a three character abbreviation for request (Req), indicate (Ind), response (Rsp), and confirm (Cnf) primitives shown at the head and tail of the arrow. For example:

```
o-- Req --------------- Ind -->
        CMIS request
<-- Cnf --------------- Rsp --o
        CMIS response
```

Editor's Note: [Comment from BT: Scenarios required for each function.**]**

3.4 MANAGEMENT INFORMATION REFERENCES

This section references all the definitions of management information relevant to the Ensemble. The definitions will be provided entirely by references to other documents which contain GDMO specifications.

This section contains only references to definitions that are relevant to the Ensemble. Thus, this section also contains statements about any additional restrictions or constraints to those definitions.

4. ENSEMBLE CONFORMANCE REQUIREMENTS

Editor's Note: [Comment from BT: Should at least refer to AOM211, and 221 - likely that 231 should be included depending on exact functions adopted.**]**

4.1 GENERAL CONFORMANCE REQUIREMENTS

4.2 SPECIFIC CONFORMANCE REQUIREMENTS

4.2.1 OSI Management Functions Profiles Conformance

4.2.2 Ensemble Functions Conformance

4.2.3 Management Conformance Summary

4.2.4 Management Capability Support/SMFUs Support

4.2.5 MOCS Proforma for Ensemble Managed Object Classes

4.2.6 Association Initiator/Responder

4.2.7 CMIS Services (CMIP PDU) Requirements

Editor's Note: [Unresolved Comments, Discussion Points, Issues, and Action Items:

1) Comment from BT:
Location. Title page
Comment. Title should be changed to reflect that the mechanism specified is more generally applicable. The title could be changed to :

- Order Handling Management Ensemble
- Generic Order Handling Management Ensemble
- Order Request Management Ensemble
- Order Request Handling Ensemble

Rationale. This mechanism could be used for any interface where two (or more) systems were involved in some sort of user-provider relationship. See following comments.

2) Provider frequently has to deal with one or more end users, particularly in later stages of the provisioning activities. What if any impact does that have on this ensemble?

3) Need to apply model & scenarios to "customer-provider-vendor" arrangement.

4) Can/should this ensemble be broadened to include all types of services, connections and goods and not just those that are network and telecommunications related? If so, some of the definitions in Section 1.2 may need to be modified to reflect this broadened scope.

5) What is the relationship between this ensemble and phone calls/email service requests??

6) What (if any) language considerations are needed? (Is foreign language support needed?)

7) Is the "send request" and "status always open until instance deleted" the simplest scenario or is "send request, status open" and "notify of completion the simplest"?

8) Is the Management Context Diagram in the Section 2.0 Ensemble template intended to be used verbatim or "customized" for the particular Ensemble being documented? What are the management context functions? (Is there a "standard" list?)

9) Need to look at if and how to handle a single request that is broken up by the provider into the ordering and/or provisioning of multiple services, connections, and goods.

10. Look into the use of EDI, TMN, and the Trouble Ticketing concept

11. Add a discussion about the relationship between this ensemble and EDI, when each might be used, etc.

12. Identify which model (e.g., ISO, CCITT) is being used.**]**

Annex E (informative)

Translated Management Information Libraries

E.1 Introduction

(Refer to the Stable Implementation Agreements Document.)

E.2 MIBs Translated By Organizations Other Than OIW

RFC 1566 MADMAN MTA MIB GDMO translation, as specified by [MTAMIBTRANS]

(Refer to the Stable Implementation Agreements Document.)

E.3 OIW NMSIG Translated MIBs

Editor's Note: **[**MIBs which may be translated by the OIW NMSIG have yet to be determined.**]**

Editor's Note: **[**The OIW NMSIG expressed a strong interest in initially translating the RMON MIB (The Internet Remote Monitoring Management Information Base, RFC 1271), the MADMAN Network Services Monitoring MIB (NMSIG-93/301), the MADMAN Directory Monitoring MIB (NMSIG-93/302), and the MADMAN Mail Monitoring MIB (NMSIG-93/303). An electronic call has been distributed to identify other candidate MIBs to be considered for translation.**]**

E.3.1 Translated MIB #1