Frameworx How-to Guide

Multi Technology Network Management (MTNM)

Implementation Statement Templates and Guidelines

MTNM 4.5

TMF814A

November 2014

Latest Update: Frameworx Release 14.5	**Member Evaluation**
Version 3.3.0	**IPR Mode: RAND**

Notice

Direct inquiries to the TM Forum office:

240 Headquarters Plaza,
East Tower – 10th Floor,
Morristown, NJ 07960 USA
Tel No. +1 973 944 5100
Fax No. +1 973 944 5110
TM Forum Web Page: www.tmforum.org

Table of Contents

List of Tables

1 Introduction

This document defines an interoperability statement template concerning the use of the TM Forum's MTNM Solution Set Error! Reference source not found.. Specifically, the interoperability statement template provides a mechanism for EMS vendors to precisely state their support for the MTNM interface and for service providers to precisely state their requirements for vendor support of the MTNM interface. The main goal of this document is to facilitate interoperability between parties representing the EMS and NMS sides of the MTNM interface. **The material in this document is not prescriptive.**

1.1 How this document will be used

It is expected that this document will be used in several ways:

(a) As a standard mechanism for service providers to request a specific subset of the MTNM interface capabilities from their vendors

(b) As a standard mechanism for a management system vendor to state the subset of the MTNM interface capabilities supported by their product(s)

(c) As a basis for an implementation agreement between 2 or more management system vendors.

1.2 Document Structure

The following is a summary of the sections of this document:

Section 1 Introduces the document

Section 2 Functional Implementation Statements (FIS) is introduced in this section. An FIS is used to detail vendor support (service provider requirements) for the functions in the MTNM interface.

Section 3 This section provides templates for Non-functional Implementation Statements (NIS). An NIS is used to describe how a function is supported rather than what functionality is supported. For example, an NIS could describe how long a client should wait for a response to an operation request.

A service provider or vendor preparing an implementation statement would use the templates in Sections 2,3. Section 4 is for informational purposes and would not be used directly in preparing an implementation statement.

Section 4 This section provides guidelines for using the MTNM interface.

Appendix A Provides terms and abbreviations used within this document.

Appendix B Lists the References.

Administrative Appendix This appendix describes the administration of the document.

1.3 Updates for Version 3.0

The following updates were made in going from v2.1 to v3.0:

1. The major set of updates in v3.0 of TMF 814A was to align with v3.0 of TMF 814.
2. Section 5 (Product Profiles) has been removed from v2.1 and a new Section 5 has been added. This new section provides a feature based summary of the capabilities in v2.1 and v3.0. The feature summary comes in the form of a spreadsheet.
3. Add some additional guidelines in Section 4 concerning usage of Network Access Domains (NADs), usage of Root Cause Alarm Indication and some FTP examples.
4. Added subsection to Section 2.27 concerning support for Proprietary LayerRates and SNC Type Transitions Related to the ModifySNC Operation.

1.4 Updates for Version 3.5

The following updates were made in going from v3.0 to v3.5:

1. The major set of updates in v3.5 of TMF 814A was to align with v3.5 of TMF 814, in particular for control plane management and Ethernet management.
2. Editorial update for Section 2.

1.5 Updates for Version 4.0

The following updates were made in going from v3.5 to v4.0:

1. Introduce two new interfaces to get history alarm and history performance by pull, which enable OSS centralized control on the sequence to get data from huge network.

2 Functional Implementation Statements (FIS)

2.1 Overview

This section provides Functional Implementation Statement (FIS) templates for the data structures (those that represent objects in the information model), operations and notifications in the MTNM version 3.5 IDL. These templates related to the functional aspects of the operations, notifications and data structures, i.e., whether or not particular capabilities are supported. The non-functional aspects of the operations, notifications and data structures are discussed in Section 3. Functional aspects relate to ***what*** an entity does, and non-functional aspects related to ***how*** an entity provides its functionality, e.g., how fast an operation request is fulfilled on average.

2.1.1 FIS Template

The following FIS templates are used for each module:

Module Name, e.g., Managed Element

Data Types

A template (in the form of a table) needs to be provided for each second-level object associated with the module. The data type template has one row for each attribute of the second-level object.

Interfaces

A template (in the form of a table) needs to be provided for each interface associated with the module. The interface template has one row for each operation comprising the interface.

Notifications

A template (in the form of a table) needs to be provided for each notification associated with the module. The notification template has one row for each parameter of the notification.

The individual components of the FIS template are explained in the following subsection.

2.1.2 Data Types

There is one table for each second-level object defined in the module.

Table 2-1. <Name of Data Type, e.g., Managed Element>

Attribute Name	Set By	Set When and How	Format	Clarification Needed
attribute1, e.g., userLabel	EMS, NMS or E to indicate either	Set When: C, L, A (see the note below for definition of these abbreviations) Set How: List operations that can be used to set the attribute's value. Typically, this field would only be included if the attribute is settable by the NMS.	FREE, FIXED, or VALUE LIST (see the note below for definition of these abbreviations)	The EMS supplier should state the maximum length for the attribute. The NMS supplier should also mention any length requirements.

attribute2				
...				

Conventions:

Set By – an indication of who can set the attribute.

- "NMS" means only the NMS can set the value,
- "EMS" means only the EMS can set the value, and
- "E" means either the NMS or EMS can set the value

Set When and How – an indication of when and how the attribute can be set.

- "C" (creation) means the attribute can only be set when the managed object is created,
- "L" (lifespan) means the attribute is only set after the managed object is created, and
- "A" (anytime) means the value can be set at creation or anytime thereafter.

Format – the format of the attribute. Possible values are

- "FREE" for Free Format (in this case the vendor or service provide ***may*** want to provide their supported/required format(s), e.g., CLLI™ codes for the nativeEMSName)
- "FIXED" for Fixed Format (the format is defined in the IDL and must be used without variance, and no additional clarification is needed by the vendor or service provider)
- "VALUE LIST" for List (the attribute takes values from a typically long set of possible values defined in the IDL, e.g., layerRate, pmParameterName). In this case, the vendor or service provider ***may*** want to provide their supported/required set of values, which is a subset of that defined in the IDL plus additional values agreed among the vendors and operators. VALUE LIST is used in only a few places in the following tables. Basically, a VALUE LIST is a data type with a set of supported values.

These three categories are meant to be mutually exclusive.

Clarification Needed – in cases where an attribute may allow for various implementation options, the EMS vendor should state how they are handling the option.

Remarks:

All attributes for second-level objects are readable by the NMS. Consequently, there is not a column to indicate whether or not an attribute is readable.

2.1.3 Interfaces

There is one table for each interface.

Table 2-2. Interface Name, e.g., EquipmentInventoryMgr

Operation	Status	Support	Exception/ Error Reason	Comments
operation1	M or O	Y, N or C	Exception_Name <error reason	

16

CLLI and CLEI are trademarks of Telcordia Technologies.

			string>	
operation2				
...				

Conventions:

Status – an indication of whether or not the operation is mandatory for a particular interface, as defined in the IDL. The status of an operation can be determined from the IDL. If the exception "EXCPT_NOT_IMPLEMENTED" is among the allowable exceptions for an operation, then the operation is optional. In this case, an "O" would be placed in the Status field. Otherwise, the operation is mandatory (this is indicated by an "M").

Support – an indication of whether or not an operation is supported by an EMS vendor (required by a service provider). "Y" is used to indicate the operation is supported (required), "N" is used to indicate the operation is not support (not required) and "C" is used if support for an operation is conditional (in this case, an explanation should be included). Note that "Support" field is different than the mandatory/optional indications derived from the IDL. If the implementation statement is prepared by a service provider, the "support" field could be used to indicate a need for capabilities that are implied to be optional in the IDL. Alternately, a vendor could use the "support" field to indicate the MTNM capabilities that they support in their product.

Exception/Error Reason – a list of the vendor-specific error reason strings on an operation/exception basis. Recall the MTNM interface provides an error reason field for each exception. It is up to the EMS vendor to decide on the contents of the error reason field. If the error reason strings are long, they can be listed and numbered elsewhere, and then cross-referenced from the table.

Comments – this field is used by the EMS vendor to state specific behavioral peculiarities of their implementation of the operation, e.g., to indicate that a particular operation parameter is not supported.

Many of the interfaces have corresponding iterator interfaces. The iterator interfaces are used for the bulk retrieval of data. All of the iterator interfaces have the same operations.

Table 2-3. General Iterator Interface

Operation	Status	Support	Exception/ Error Reason	Comments
destroy	M			
getLength	M			
next_n	M			

2.1.4 Operations

The operations template (shown in Table 2-4) is be used when an organization (that is creating a profile of TMF 814) needs to further define characteristics and/or behavior for an operation (beyond that specified in the interface template). A blank cell in the following table implies that the organization completing the template has no further information to provide with regard to the Parameter/Characteristic.

The Comments/Clarifications column is used to further clarify the characteristics and/or usage of the parameter. One can use this column (for example) to indicate the maximum supported length of a parameter, and the default value for a parameter.

It is not anticipated that this template will be needed for all operations.

Table 2-4. Operations Template Example - createAndActivateSNC

Parameter	Supported Values	Comments/ Clarifications
createData		
userLabel		Maximum length is 128 characters
forceUniqueness		Not supported by EMS
owner		Maximum length is 128 characters
direction		
staticProtectionLevel	Preemptible Unprotected Partially_Protected Fully_Protected Highly_Protected	Highly_Protected is only used to request DRI in the case of SONET/SDH
protectionEffort		
rerouteAllowed		Not supported
networkRouted		Not supported
sncType	st_simple st_add_drop_a st_add_drop_z st_interconnect st_double_ interconnect st_double_add_ drop st_open_add_drop st_explicit	
layerRate	LR_VT1_5_and_ TU11_VC11 LR_STS1_and_ AU3_High_Order_VC3	
ccInclusions		
neTpInclusions		
fullRoute		
neTpSncExclusions		Exclusions are not supported
aEnd		
zEnd		
additionalCreation Info		
tolerableImpact	GOI_Major_ Impact	
emsFreedomLevel	EMSFL_CC_AT_SNC_LAYER, EMSFL_TERMINATE_AND_ MAP	
tpsToModify		
tpName		

tpMappingMode		
transmissionParams	AlarmReporting, PotentialFuture SetupIndicator, TrailTraceActualTx, TrailTraceExpectedRx, TrailTraceMonitor, FrameFormat, LineCode, Mapping, SignalLabel ExpectedRx	
ingressTraffic DescriptorName		Not supported
egressTraffic DescriptorName		Not supported
theSNC		
errorReason		

2.1.5 Exceptions

The EMS may raise an exception in response to an MTNM operation request. The allowable exceptions are defined in the IDL. The EMS may also provide an error reason in conjunction with an exception. The allowable error reasons are not defined in the MTNM interface. It is proposed that the following table be used in cases where the EMS supplier wishes to provide further information about their error reasons.

Table 2-5. Exception Table Example - getRoute

Exception	Error Reason	Explanation
EXCPT_NOT_IMPLEMENTED		getRoute is supported in this hypothetical example. So, this exception would never by used.
EXCPT_INTERNAL_ERROR	1. EMS is processing other (higher priority requests) and cannot respond to getRoute at this time. 2. EMS undergoing system maintenance	1. In some cases, the EMS will reject requests if it is processing higher-priority activities. The NMS should try the request at a later point in time. 2. The EMS is undergoing scheduled system maintenance and cannot presently respond to the request.
EXCPT_INVALID_INPUT		No error reasons are provided with this exception
EXCPT_ENTITY_NOT_FOUND		No error reasons are provided with this exception

2.1.6 Notifications

2.1.6.1 Alarms and Probable Causes

The MTNM team has not specified a set of alarms. The team has, however, defined a set of probable causes and does allow for the definition of native probable causes (basically non-MTNM defined probable causes). An alarm type may be uniquely identified by the combination of a **probable cause**, **object type**, and **layer rate**. In

some cases, an additional parameter (i.e., the **probable cause qualifier**) is needed to uniquely identify an alarm. Table 2-6 is a template that should be used to indicate vendor support for (or service provider need of) probable causes. The table effectively implies all the supported alarms of a vendor, or the required alarms of a service provider.

Not allow of the entries in a row need to be filled. The intent is to provide sufficient information to ensure interoperability over the MTNM interface. It is also possible to cover some attributes (e.g., probableCauseQualifier) by providing a general statement before the template, describing how the attribute is used. An example usage of this table is provided in Section 4.5.

The service provider or vendor preparing an implementation statement should place their Probable Cause Template table in this section.

Table 2-6 Probable Cause Template

Probable Cause	**Native Probable Cause (Optional)**	**Associ-ated Object Type**	**Layer Rates**	**Probable Cause Qualifiers (optional)**	**Perceived Severity**	**Service Affecting**	**Additional Informa-tion (optional)**	**Comments**
Probable Cause #1, e.g., LOS	This attribute represents the equipment vendor's probable cause – this may be mapped to an MTNM probable cause.	Indicate the object types that can be associated with this probable cause	Indicate the layer rates at which this probable cause applies	Indicate the probable cause qualifiers that can be associated with the probable cause	Indicate perceived severity associated with this probable cause – this may depend on the associated object and/or layer rate	Indicate whether or not this probable cause is expected to be service affecting – this may depend on the associated object and/or layer rate	Any additional information and the associated meaning should be noted if it needs to be interpreted by the NMS. The additional information should be listed in the form Name – Value.	This column can be used to provide additional details about the probable cause, e.g., one could describe the conditions under which an alarm is generated by the EMS.
Probable Cause #2								
...								

2.1.6.1.1 Usage of the Probable Cause Qualifier

The probableCauseQualifier parameter is useful for two reasons:

It allows the set of values for the probableCause parameter to remain generic. In systems where only one probable cause field is managed, there is a never-ending succession of updates to the set of probable causes. Every time a new equipment type is introduced, new values need to be added. The practical result is that the model is never stabilized.

It allows several alarms to be sent from the same object, with the same probable cause, while still remaining identifiably different. (The alternative would be to manage unique values of the notification identifier, but some EMS vendors have problems with that.)

Note that probableCause qualifier is not necessarily human readable, as its real purpose is to satisfy the second point. For instance, if the EMS works internally with a GDMO model, then it might use a GDMO probableCause and specificProblems attribute values (which are sequences of integers, not very exciting to read). The additionalText parameter is there to tell the human operator what is going on.

Example 1:

Consider a vendor that has a line terminal NE that issues the following native alarms from the same termination point and layer rate:

- DegradedSignal
- fecUncorrectedBlocks (meaning that our Reed-Solomon Forward Error Correcting algorithm has rejected some frames)

Both these alarms are mapped to the generic probable cause BER_SD (i.e., "signal degraded"). The probableCauseQualifier, nativeEMSProbableCause and possibly the additionalText parameters distinguish between the two. Note that this saves one the trouble of going to TM Forum MTNM standards group and asking for a new probableCause value, and even better, it saves the NMS vendor the trouble of managing a new fault condition which from their point of view is equivalent to the other one ("a difference that makes no difference is no difference").

Example 2:

Just consider all the equipment alarms that can be generated by all the equipment vendors. The MTNM interface designers intentionally use a single probableCause value “EQPT” for all of them. The implication is that it is not expected for the NMS to discriminate between hundreds of equipment conditions.

2.1.6.2 Threshold Crossing Alerts (TCAs)

One TCA template should be provided for each supported (or required) PM parameter for which the vendor supports thresholding or the service provider requires thresholding. As was the case for the probable cause template, a single Threshold template can cover several types of TCA (one TCA for each layer rate associated with a PM parameter). It should be noted that for a given PM parameter some parameters in the template may vary based on the layer rate.

Table 2-7. Threshold Template

Parameter	Format	Comments/ Clarifications
notificationId	FREE	The notificationId is not guaranteed to be unique. However, if a vendor does support unique Ids, this should be noted.
objectName	FIXED	
nativeEMSName	FREE	
objectType	VALUE LIST The set of object types to which this TCA can apply	
emsTime	FIXED	In cases where the NE does not report time, a zero should be returned.
neTime	FIXED	

isClearable	FIXED	
perceivedSeverity	VALUE LIST	The EMS vendor (SP or NMS vendor) should describe the criteria that are used (required) when assigning a perceived severity to a particular alarm.
layerRate	VALUE LIST Service provider (vendor) should indicate the required (supported) layerRates to which this alarm type applies	
granularity	FIXED	
pmParameterName	VALUE LIST	The EMS vendor (SP or NMS vendor) should indicate the set support (required) PM parameters.
pmLocation	FIXED	
thresholdType	FIXED	
value	FREE	
unit	FREE	
Version 3.0 Additions		
acknowledge Indication		

2.1.6.3 AVCs, Object Creation/Deletion and Other Notifications

The second-level objects in the MTNM model are expected to emit Attribute Value Change (AVC), Object Creation, Object Deletion, and State Change notifications. Table 2-8 summarizes the notifications that are to be supported for the second-level objects. The table is based on Section 6 of the MTNM Information Agreement. A party (vendor or service provider) using this implementation statement should indicate any variances from the required MTNM notifications. A notification compliance statement is to be included for each module (see, for example, Sections 2.5.3, 2.6.3, 2.8.3). The notification formats are provided in a supporting document to the MTNM IDL Solution Set (the supporting document is called *Using the Notification and Log Services*).

Table 2-8. Notifications for Second-Level Objects

Notification Type	Notification Producers	Variations from the MTNM Interface Specification
Object Creation	Topological Link, Equipment, Equipment Holder, Subnetwork, Traffic Descriptor, PTP, TP Pool, Subnetwork Connection, Managed Element, Protection Group (v3.0) ASAP, Equipment Protection Group, Group Termination Point, Log, Transmission Descriptor	

Object Deletion	Topological Link, Equipment, Equipment Holder, Subnetwork, Traffic Descriptor, PTP, TP Pool, Subnetwork Connection, Managed Element, Protection Group (v3.0) ASAP, Equipment Protection Group, Group Termination Point, Log, Transmission Descriptor	
Attribute Value Change (AVC)	Topological Link, Equipment, Equipment Holder, Subnetwork, Traffic Descriptor, PTP, CTP, TP Pool, Subnetwork Connection, Managed Element, Protection Group, EMS (v3.0) ASAP, Equipment Protection Group, Group Termination Point, Log, Transmission Descriptor	
State Change	Equipment Holder, Equipment, PTP, CTP, Subnetwork Connection, Managed Element, Protection Group (v3.0) Equipment Protection Group, Group Termination Point, Log	

2.2 Alarm Severity Assignment Profile (v3.0)

2.2.1 Data Types

Table 2-9. ASAP

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	EMS	C	FREE	
userLabel	E	A – set by the EMS (C), or by the NMS via createASAP (C) or setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
nativeEMSName	E	A – set by the EMS (C), or by the NMS via createASAP (C) or setNativeEMSName (L)	FREE (e.g. CLEI code)	
owner	NMS	L – setOwner	FREE	
notModifiable	EMS	C	FIXED	
alarmSeverityAssignmentList	E	A – set by the EMS (C), or by the NMS via createASAP (C) or modifyASAP (L)	FREE	EMS supplier should indicate range of possible value
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo.

2.2.2 Interfaces

There is the ASAPIterator interface which is used to retrieve a large number of ASAPs (one batch at a time).

2.2.3 Notifications

To be provided by the organization preparing an interoperability statement

2.3 Call SNC Module (v3.5)

2.3.1 Data Types

Table 2-10. Call

Attribute Name	Set By	Set When and How	Format	Clarification Needed
callName	E	C	FREE	The callName could be optionally provided by the NMS. If not provided by NMS, it will be set by EMS using the same value as called.
userLabel	NMS	A – set by the NMS via establishCall (C), modifyCall (L), or setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
owner	NMS	A – set by the NMS via establishCall (C), modifyCall (L), or setOwner (L)	FREE	The value may be empty.
nativeEMSName	E	A – set by the EMS (C), or by the NMS via establishCall (C) or setNativeEMSName (L)	FREE (e.g. CLEI code)	
networkAccessDomain	NMS	C – set by the NMS via establishCall (C)	FREE	
callId	EMS	C – set by the EMS (control plane)	FREE	The value is provided by the control plane and used for signaling
callState	EMS	A – set by the EMS (C) and updated by the EMS anytime thereafter	FIXED	
aEnd	NMS	C – set by the NMS via establishCall (C)	FIXED	
zEnd	NMS	C – set by the NMS via establishCall (C)	FIXED	
callParameters	NMS	C – set by the NMS via establishCall (C)	FIXED	
callDiversity	NMS	C – set by the NMS via establishCall (C)	FIXED	
diversitySynthesis	EMS	A – set by the EMS (C) and updated by the EMS anytime thereafter	FIXED	
linkDiversityViolationsList	EMS	A – set by the EMS (C) and updated by the EMS anytime thereafter	FIXED	

nodeDiversityViolationsList	EMS	A – set by the EMS (C) and updated by the EMS anytime thereafter	FIXED	
linkPartialDiversityList	EMS	A – set by the EMS (C) and updated by the EMS anytime thereafter	FIXED	
nodePartialDiversityList	EMS	A – set by the EMS (C) and updated by the EMS anytime thereafter	FIXED	
callAdditionalInfo	NMS	C – set by the NMS via establishCall (C)	--	Name/Value list defined by the EMS vendor.

Table 2-11. Call Create Data

Attribute Name	Set By	Set When and How	Format	Clarification Needed
callName	E	C	FREE	The callName could be optionally provided by the NMS. If not provided by NMS, it will be set by EMS using the same value as called.
userLabel	NMS	A – set by the NMS via establishCall (C), modifyCall (L), or setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
forceUniqueness	NMS		FIXED	
owner	NMS	A – set by the NMS via establishCall (C), modifyCall (L), or setOwner (L)	FREE	The value may be empty.
networkAccessDomain	NMS	C – set by the NMS via establishCall (C)	FREE	
aEnd	NMS	C – set by the NMS via establishCall (C)	FIXED	
zEnd	NMS	C – set by the NMS via establishCall (C)	FIXED	
callParameters	NMS	C – set by the NMS via establishCall (C)	FIXED	
callDiversity	NMS	C – set by the NMS via establishCall (C)	FIXED	
additionalCreationInfo	NMS	C – set by the NMS via establishCall (C)	--	Name/Value list defined by the EMS vendor.

Table 2-12. Call Modify Data

Attribute Name	Set By	Set When and How	Format	Clarification Needed

userLabel	NMS	A – set by the NMS via establishCall (C), modifyCall (L), or setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
forceUniqueness	NMS		FIXED	
owner	NMS	A – set by the NMS via establishCall (C), modifyCall (L), or setOwner (L)	FREE	The value may be empty.
networkAccessDomain	NMS	C – set by the NMS via establishCall (C)	FREE	
additionalModificationInfo	NMS	C – set by the NMS via establishCall (C)	--	Name/Value list defined by the EMS vendor.

2.3.2 Interfaces

There is the CallAndTopLevelConnectionsIterator interface which is used to retrieve a large number of Calls and their associated Top level connections (one batch at a time).

There is the CallIterator interface which is used to retrieve a large number of Calls (one batch at a time).

2.3.3 Notifications

To be provided by the organization preparing an interoperability statement.

2.4 Common Module

2.4.1 Data Types

This module has no data types that represent second level objects.

2.4.2 Interfaces

Table 2-13. Common

Operation	Status	Support	Exception/ Error Reason	Comments
getCapabilities	M			The feature/capability name part is used to identify an IDL operation using the following convention: "module_name::interface_name:: operation_name." Other capabilities include Supports_CC_sharing, Supports_pending, and Supports_adjacent_termination_inclusion.
setNativeEMSName	O			
setOwner	O			
setUserLabel	O			
Version 3.0 Additions				

setAdditionalInfo	O			

2.4.3 Notifications

There are no specific notifications associated with the data types in this module.

2.5 EmsMgr Module

2.5.1 Data Types

Table 2-14. EMS

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	EMS	C	FREE	The Value string that applies to the "ManagedElement", for example, can contain any string as long as it uniquely identifies the Managed Element within the EMS. The EMS supplier should indicate the maximum length of the attribute (same comments holds true for all attributes).
userLabel	E	A – set by the EMS (C), or by the NMS via setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
nativeEMSName	E	A – set by the EMS (C), or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
owner	NMS	L – setOwner	FREE	
emsVersion	EMS	A – set at time of EMS initialization and is not expected to change very often	FREE	
type	EMS	A – set at time of EMS initialization and is not expected to change very often	FREE	
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo.
Version 3.0 Additions				
ASAPpointer	E	EMS (C), or by the NMS via assignASAP (L)	FIXED	

2.5.2 Interfaces

Table 2-15. EMSMgr

Operation	Status	Support	Exception/ Error Reason	Comments
getAllEMSAndMEActiveAlarms	M			
getAllEMSSystemActiveAlarms	M			
getAllTopLevelSubnetworkNames	M			
getAllTopLevelSubnetworks	M			
getAllTopLevelTopologicalLinkNames	O			
getAllTopLevelTopologicalLinks	O			
getTopLevelTopologicalLink	O			
getEMS	M			
Version 3.0 Additions				
acknowledgeAlarms	O			
assignASAP	O			
createASAP	O			
createTopologicalLink	O			
deassignASAP	O			
deleteASAP	O			
deleteTopologicalLink	O			
getAllASAPs	O			
getAllASAPNames	O			
getAllEMSAndMEUnacknowledged ActiveAlarms	O			
getAllEMSSystemUnacknowledged ActiveAlarms	O			
getASAP	O			
getASAPAssociatedResourceNames	O			
getASAPbyResource	O			
modifyASAP	O			
unacknowledgeAlarms	O			
Version 3.5 Additions				
getAllMLRAsandMEs	M			
getAllMLSNPPLinks	O			

getAllMLSNPPLinksWithTP	O			
getAllMLSNPPLinksWithMLSNs	O			
getAllMLSNPPLinksWithTNA	O			
getAllMLSNPPs	O			
getAllMLSNPPsWithTP	O			
getAllMLSNPPsWithTNA	O			
Version 4.0 Additions				
getHistoryAlarmDataByPull	O			

2.5.3 Notifications

To be provided by the organization preparing an interoperability statement

2.6 EMS Session Module

2.6.1 Data Types

This module has no data types that represent second level objects.

2.6.2 Interfaces

Table 2-16. EmsSession

Operation	Status	Support	Exception/ Error Reason	Comments
getEventChannel	M			
getManager	O			Attempting to gain access to the following manager interfaces may not raise EXCPT_NOT_IMPLEMENTED: EMS, ManagedElement, MultiLayerSubnetwork, GuiCutThrough
getSupportedManagers	M			

2.6.3 Notifications

There are no specific notifications associated with the data types in this module.

2.7 EmsSessionFactory

2.7.1 Data Types

This module has no data types that represent second level objects.

2.7.2 Interfaces

Table 2-17. EmsSessionFactory

Operation	Status	Support	Exception/ Error Reason	Comments
getEmsSession	M			This operation is accessed via a user name and password. If this security scheme is used by the EMS vendor, they need define a means of assigning passwords to clients (i.e., NMSs).

2.7.3 Notifications

There are no specific notifications associated with the data types in this module.

2.8 Equipment Module

2.8.1 Data Types

Table 2-18. Equipment

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	EMS	C	FREE	
userLabel	E	A – set by the EMS (C), or by the NMS via setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
nativeEMSName	E	A – set by the EMS (C), or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
owner	NMS	A – set at time of object creation via the provisionEquipment operation (C), or via setOwner (L)	FREE	
alarmReportingIndicator	E	A – set by EMS at time of equipment create (C) and may be set anytime thereafter by the NMS using the setAlarmReportingOn and setAlarmReportingOff operations (L)	FIXED	
serviceState	EMS	A	FIXED	
expectedEquipmentObjectType	EMS	C	FREE	
installedEquipmentObjectType	EMS	L	FREE	
installedPartNumber	EMS	C	FREE (a vendor	

installedVersion	EMS	C	may want to indicate specific formats used for their equipment)	
installedSerialNumber	EMS	C		
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo.
Version 3.0 Additions				
ASAPpointer	E	EMS (C), or NMS via provisionEquipment (C) or assignASAP (L)	FIXED	
Manufacturer	E	EMS (C) or NMS via provisionEquipment (C)	FREE	
ProtectionRole	E	EMS (C), or NMS via provisionEquipment (C) or setAdditionalInfo (L)	FIXED	
ProtectionSchemeState	E	EMS (C), or NMS via provisionEquipment (C) or setAdditionalInfo (L)	FIXED	

Table 2-19. Equipment Holder

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	EMS	C	FREE	
userLabel	E	A – set by the EMS (C), or by the NMS via setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
nativeEMSName	E	A – set by the EMS (C) or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
owner	NMS	L – setOwner	FREE	
alarmReportingIndicator	E	A – set by EMS at time of equipment create (C) and may be set anytime thereafter by the NMS using the setAlarmReportingOn and setAlarmReportingOff operations (L)	FIXED	
holderType	EMS	C	FIXED	
expectedOrInstalledEquipment (this attribute provides the name of the expected or installed equipment, not the type)	NMS	L – the NMS provisions the installed equipment	FIXED	

acceptableEquipmentTypeList	EMS	A – the EMS indicates the type of equipment that can be supported by the equipment holder (presumably this can change over time)	FREE	This is a list provided at run time by the equipment holder, each equipment holder may support a different set of equipment types. The NMS should be able to handle almost any list being returned in this attribute. Normally the maximum list size or maximum length of a string element within the list should be sufficient.
holderState	EMS	A	FIXED	
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo.
Version 3.0 Additions				
ASAPpointer	E	EMS (C), or by the NMS via assignASAP (L)	FIXED	

Table 2-20. Equipment Create Data

Attribute Name	Set By	Set When and How	Format	Clarification Needed
userLabel	NMS	C – provisionEquipment	FREE	
forceUniqueness	NMS	C – provisionEquipment	FIXED	
owner	NMS	C – provisionEquipment	FREE	
expectedEquipmentObjectType	NMS	C – provisionEquipment	FREE	
equipmentHolderName	NMS	C – provisionEquipment	FREE	
additionalInfo	NMS	C – provisionEquipment	FREE	
Version 3.0 Additions				
AlarmReporting	NMS	C – provisionEquipment	FIXED	
ASAPpointer	NMS	C – provisionEquipment	FIXED	
Manufacturer	NMS	C – provisionEquipment	FREE	
ProtectionRole	NMS	C – provisionEquipment	FIXED	
ProtectionSchemeState	NMS	C – provisionEquipment	FIXED	

2.8.2 Interfaces

Table 2-21. Equipment Inventory Manager

Operation	Status	Support	Exception/ Error Reason	Comments
getAllEquipment	M			
getAllEquipmentNames	M			
getAllSupportedPTPNames	M			
getAllSupportedPTPs	M			
getAllSupportingEquipment	M			
getAllSupporting EquipmentNames	M			
getContainedEquipment	M			
getEquipment	M			
provisionEquipment	O			This operation is commonly used in cases where the EMS does not support auto-provisioning (i.e., an image of the equipment is created in the EMS database when a piece of equipment is inserted into an equipment holder) and GUI Cut Through is not used to provision equipment.
setAlarmReportingOff	O			Equipment providers should indicate whether monitoring is disabled, and indicate whether this is at the NE or at the EMS.
setAlarmReportingOn	O			Equipment providers should indicate whether monitoring is enabled, and indicate whether this is at the NE or at the EMS.
unprovisionEquipment	O			This operation is commonly used in cases where the EMS does not support auto-unassignment (i.e., the image of the equipment is deleted from the EMS database when a piece of equipment is removed from an equipment holder) and GUI Cut Through is not used to unprovision equipment.
Version 3.0 Additions				
getSupportedEquipment	O			
getSupportedEquipment Names	O			
getSupportingEquipment	O			
getSupportingEquipment Names	O			

There is also an EquipmentOrHolderIterator interface. This interface is similar to all the other iterator interfaces.

2.8.3 Notifications

To be provided by the organization preparing an interoperability statement

2.9 FlowDomain Module

2.9.1 Data Types

Table 2-22. FlowDomain

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	EMS	C	FREE	
userLabel	E	A – set by the EMS (C), or by the NMS via setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
nativeEMSName	E	A – set by the EMS (C), or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
owner	NMS	L – setOwner	FREE	
transmissionParams	E	A – set by the EMS or NMS (C), or modified via modifyFlowDomain (L)	VALUE LIST	EMS vendor (SP or NMS vendor) should indicated supported (required) transmissionParams and associated values
networkAccessDomain	E	EMS (C) or NMS via setAdditionalInfo (L) or modifyFlowDomain (L)	FREE	
fDConnectivityState	EMS	A	VALUE LIST	
fdType	EMS	C	VALUE LIST	Read-only attribute
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo.

Table 2-23. FDCreateData

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	NMS	C – createFlowDomain	FREE	
userLabel	NMS	A – createFlowDomain (C), modifyFlowDomain (L)	FREE	
forceUniqueness	NMS	C – createFlowDomain	FIXED	
owner	NMS	A – createFlowDomain (C), modifyFlowDomain (L)	FREE	

networkAccess Domain	NMS	A – createFlowDomain (C), modifyFlowDomain (L)	FREE	
mfds	NMS	C – createFlowDomain	VALUE LIST	Not an attribute of the created object. Instead, this information is retrieved by operations getAllAssociatedMFDs , getAssociatingFD , and changed by operations associateMFDsWithFlowDomain, deAssociateMFDsFromFlowDomain.
transmissionParams	NMS	A – set by the EMS or NMS (C), or modified via modifyFlowDomain (L)	VALUE LIST	EMS vendor (SP or NMS vendor) should indicated supported (required) transmissionParams and associated values
additional CreationInfo	NMS	A – createFlowDomain (C), modifyFlowDomain (L)		Corresponds to additionalInfo attribute of the managed object.

Table 2-24. FlowDomain Modify Data

Attribute Name	Set By	Set When and How	Format	Clarification Needed
userLabel	NMS	A – createFlowDomain (C), modifyFlowDomain (L)	FREE	
forceUniqueness	NMS	C – modifyFlowDomain	FIXED	
owner	NMS	A – createFlowDomain (C), modifyFlowDomain (L)	FREE	
networkAccess Domain	NMS	A – createFlowDomain (C), modifyFlowDomain (L)	FREE	
transmissionParams	NMS	A – set by the EMS or NMS (C), or modified via modifyFlowDomain (L)	VALUE LIST	EMS vendor (SP or NMS vendor) should indicated supported (required) transmissionParams and associated values
additionalModificationInfo	NMS	A – createFlowDomain (C), modifyFlowDomain (L)		Corresponds to additionalInfo attribute of the managed object.

Table 2-25. MatrixFlowDomain

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	E	C	FREE	

userLabel	E	A – set by the EMS (C), or by the NMS via setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
nativeEMSName	E	A – set by the EMS (C), or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
owner	NMS	L – setOwner	FREE	
transmissionParams	E	A – set by the EMS or NMS (C), or modified via modifyMFD (L)	VALUE LIST	EMS vendor (SP or NMS vendor) should indicated supported (required) transmissionParams and associated values
networkAccessDomain	E	A – set by EMS (C) or NMS via setAdditionalInfo (L) or modifyMFD (L)	FREE	
flexible	EMS	C	FIXED	Read-only attribute
additionalInfo	NMS	A – createMFD (C), modifyMFD (L)	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo.

Table 2-26. MFDCreateData

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	NMS	C	FREE	
userLabel	NMS	A – createMFD (C), modifyMFD (L)	FREE	
forceUniqueness	NMS		FIXED	
owner	NMS	A – createMFD (C), modifyMFD (L)	FREE	
networkAccess Domain	NMS	A – createMFD (C), modifyMFD (L)	FREE	
unassignedCPTPs	NMS	C	FIXED	
transmissionParams	E	A – createMFD (C), modifyMFD (L)	FIXED	
additional CreationInfo	NMS		VALUE LIST	Corresponds to additionalInfo attribute of the managed object.

Table 2-27. MatrixFlowDomain Modify Data

Attribute Name	Set By	Set When and How	Format	Clarification Needed
userLabel	NMS	A – createMFD (C), modifyMFD (L)	FREE	
forceUniqueness	NMS	C – modifyMFD	FIXED	
owner	NMS	A – createMFD (C), modifyMFD (L)	FREE	
networkAccess Domain	NMS	A – createMFD (C), modifyMFD (L)	FREE	
transmissionParams	NMS	A – createMFD (C), modifyMFD (L)	VALUE LIST	EMS vendor (SP or NMS vendor) should indicated supported (required) transmissionParams and associated values

Note: additionalModificationInfo is an "inout" parameter of the modifyMFD operation, but not an attribute of MatrixFlowDomain Modify Data.

Table 2-28. FTP Create Data

Attribute Name	Set By	Set When and How	Format	Clarification Needed
equipmentName	NMS	C - createFTP	FREE	
userLabel	E	A – set by EMS (C), by NMS via createFTP (C) or via setUserLabel (L)	FREE	
forceUniqueness	NMS	C	FIXED	
owner	NMS	A – set by EMS (C), by NMS via createFTP (C) or via setOwner (L)	FREE	
ingressTransmissionDescriptorName	NMS	A – createFTP (C) or setTPData (L), or via side effect through activateSNC, deactivateAndDeleteSNC, deactivateSNC (C)	FIXED	
egressTransmissionDescriptorName	NMS		FIXED	
networkAccess Domain	NMS	A – createMFD (C), setTPDate (L)	FREE	In the terminationPoint object, this is part of additionalInfo
tpMappingMode	E	A – by NMS via createFTP (C) or setTPData (L), or via side effect through activateSNC, deactivateAndDeleteSNC, deactivateSNC (C) Can also be set by the EMS	FIXED	

direction	E	C – via createFTP or via side effect through any of the SNC establishment operations Can also be set by the EMS	FIXED	
transmissionParams	E	A (depends on the specific transmission parameter) – via createFTP (C), via setTPData (L), or via side effect through activateSNC, deactivateAndDeleteSNC, deactivateSNC (C)	VALUE LIST	EMS vendor (SP or NMS vendor) should indicated supported (required) transmissionParams and associated values
additional CreationInfo	E	A	VALUE LIST	Corresponds to additionalInfo attribute of the managed object.

2.9.2 Interfaces

The EMS supplier should indicate which of the 4 SNC modes of operation it can support (see Section 4.4 for a description of the SNC modes of operation). The NMS supplier should also indicate which mode(s) it can support.

Table 2-29. Flow Domain Manager

Operation	Status	Support	Exception/ Error Reason	Comments
getAllFlowDomains	O			
getFlowDomainsByUserLabel	O			
getFlowDomain	O			
getAssociatingFD	O			
getTransmissionParams	O			
createFlowDomain	O			
deleteFlowDomain	O			
modifyFlowDomain	O			
associateMFDsWithFlowDomain	O			
deAssociateMFDsFromFlowDomain	O			
associateCPTPsWithFlowDomain	O			
deAssociateCPTPsFromFlowDomain	O			
getAllAssociatedMFDs	O			

getAllSupportedMFDs	O			
getMFD	O			
getAssigningMFD	O			
createMFD	O			
deleteMFD	O			
modifyMFD	O			
assignCPTPsToMFD	O			
unassignCPTPsFromMFD	O			
createFTP	O			
deleteFTP	O			
getAllCPTPs	O			
getAllAssignedCPTPs	O			
getAllAssignableCPTPs	O			
getAllFDFrs	O			
getFDFrsWithTP	O			
getFDFrsByUserLabel	O			
getFDFr	O			
createAndActivateFDFr	O			
deactivateAndDeleteFDFr	O			
modifyFDFr	O			
getAllTopologicalLinksOfFD	O			
getFDFrRoute	O			

2.9.3 Notifications

To be provided by the organization preparing an interoperability statement

2.10 FlowDomainFragment Module

2.10.1 Data Types

Table 2-30. FlowDomainFragment

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	EMS	C	FREE	
userLabel	E	A – set by the EMS (C), or by the NMS via setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used

nativeEMSName	E	A – set by the EMS (C), or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
owner	NMS	L – setOwner	FREE	
direction	E	C	FIXED	
transmissionParams	E	A – set by the EMS or NMS (C), or modified via modifyFlowDomain (L)	VALUE LIST	EMS vendor (SP or NMS vendor) should indicated supported (required) transmissionParams and associated values
aEnd	E	C	FIXED	
zEnd	E	C	FIXED	
networkAccessDomain	E	EMS (C) or NMS via setAdditionalInfo (L) or modifyFlowDomain (L)	FREE	
flexible	EMS	C	FIXED	
administrativeState	E	A – set by the EMS or NMS (C), or modified via modifyFDFr (L)	FIXED	
fdfrState	EMS	A	FIXED	
fdfrType	E	C	VALUE LIST	Read-only attribute
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo.

Table 2-31. FDFrCreateData

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	NMS	C – createAndAcrivateFDFr	FREE	
userLabel	NMS	C – createAndAcrivateFDFr	FREE	
forceUniqueness	NMS	C – createAndAcrivateFDFr	FIXED	
owner	NMS	C – createAndAcrivateFDFr	FREE	
networkAccess Domain	NMS	C – createAndAcrivateFDFr	FREE	
direction	NMS	C – createAndAcrivateFDFr	FIXED	
administrativeState	NMS	C – createAndAcrivateFDFr	FIXED	
transmissionParams	NMS	C – createAndAcrivateFDFr	VALUE LIST	EMS vendor (SP or NMS vendor) should indicated supported (required) transmissionParams and associated values

fullRoute	NMS	C – createAndAcrivateFDFr	FIXED	
fdfrType	NMS	C – createAndAcrivateFDFr	FIXED	
additional CreationInfo	NMS	C – createAndAcrivateFDFr		Corresponds to additionalInfo attribute of the managed object.

Table 2-32. FlowDomainFragment Modify Data

Attribute Name	Set By	Set When and How	Format	Clarification Needed
userLabel	NMS	L – modifyFDFr (L)	FREE	
forceUniqueness	NMS	L – modifyFDFr (L)	FIXED	
owner	NMS	L – modifyFDFr (L)	FREE	
networkAccess Domain	NMS	L – modifyFDFr (L)	FREE	
administrativeState	NMS	L – modifyFDFr (L)	FIXED	
transmissionParams	NMS	L – modifyFDFr (L)	VALUE LIST	EMS vendor (SP or NMS vendor) should indicated supported (required) transmissionParams and associated values
tpNamesToRemove	NMS	L – modifyFDFr (L)	FIXED	
aEndTPNames	NMS	L – modifyFDFr (L)	FIXED	
zEndTPNames	NMS	L – modifyFDFr (L)	FIXED	
internalTPNames	NMS	L – modifyFDFr (L)	FIXED	
additional ModificationInfo	NMS	L – modifyFDFr (L)		Corresponds to additionalInfo attribute of the managed object.

2.10.2 Interfaces

This module has no regular interface. There is only the MFDFrInterator and FDFrIterator interfaces which are used in conjunction with the various operations that return a list of Matrix Flow Domain Fragment and Flow Domain Fragment.

2.10.3 Notifications

To be provided by the organization preparing an interoperability statement.

2.11 Globaldefs Module

2.11.1 Data Types

This module has no data types that represent second level objects.

2.11.2 Interfaces

This module has no “regular” interfaces. There is only the NamingAttributesIterator interface which is used in conjunction with the various operations that return a list of object names, e.g., getAllTopLevelSubnetworkNames.

2.11.3 Notifications

There are no specific notifications associated with the data types in this module.

2.12 GuiCutThrough Module

2.12.1 Data Types

Table 2-33. GCTProfileInfo

Attribute Name	Set By	Set When and How	Format	Clarification Needed
serverLaunchCapability	EMS	A – although unlikely, the EMS can change the value of this attribute	FIXED	
gctHostname	EMS	A – presumable the EMS can change the value of gctHostname	FREE	The information is described in the IDL but format is not.
emsGctPlatform	EMS	A – although unlikely, the EMS can change the value of this attribute	FIXED (string with possible values: "unix", "windowsNT", "local", "web-based")	
guiCutThroughDataList	EMS	A – although unlikely, the EMS can change the value of this attribute	See the GuiCutThroughData table below	

Table Error! No text of specified style in document.**-34.** GuiCutThroughData

Attribute Name	Set By	Set When and How	Format	Clarification Needed
gctScope	EMS	A – although unlikely, the EMS can change the value of these attributes	FIXED	
gctContext	EMS		FIXED	
gctCommand	EMS		FREE	
additionalInfo	E	A – set by EMS (C), or by the NMS via setAdditionalInfo	FREE	

2.12.2 Interfaces

Table 2-35. GuiCutThroughMgr

Operation	Status	Support	Exception/ Error Reason	Comments
destroyGCT	O			
getGCTProfileInfo	M			
launchGCT	O			

2.12.3 Notifications

There are no specific notifications associated with the data types in this module.

2.13 MaintenanceOperations Module

2.13.1 Data Types

Table 2-36. CurrentMaintenanceOperation

Attribute Name	Set By	Set When and How	Format	Clarification Needed
tpName	EMS	Set by the EMS when the NMS requests a persistent maintenance command for a given TP via the performMaintenanceOperation operation.	FIXED	
maintenanceOperation	EMS	Set by the EMS when the NMS requests a persistent maintenance command for a given TP via the performMaintenanceOperation operation.	FIXED	
layerRate	EMS	Set by the EMS when the NMS requests a persistent maintenance command for a given TP via the performMaintenanceOperation operation.	VALUE LIST	
additionalInfo	E	A – set by EMS (C), or by the NMS via setAdditionalInfo	FREE	Name/Value list defined by the EMS vendor

2.13.2 Interfaces

Table 2-37. MaintenanceMgr

Operation	Status	Support	Exception/ Error Reason	Comments
getActiveMaintenanceOperations	O			
performMaintenanceOperation	O			The EMS vendor (SP or NMS vendor) should indicate supported (required) maintenance operations.

There is also the CurrentMaintenanceOperationIterator, which is used to retrieve a large number of data objects associated with persistent maintenance operations.

2.13.3 Notifications

There are no specific notifications associated with the data types in this module.

2.14 Managed Element Module

2.14.1 Data Types

Table 2-38. Managed Element

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	EMS	C	FREE	
userLabel	E	A – set by the EMS (C), or by the NMS via setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
nativeEMSName	E	A – set by the EMS (C), or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
owner	NMS	L – setOwner	FREE	
location	EMS	C	FREE (e.g., CLLI code)	
version	EMS	C	FREE	
productName	EMS	C	FREE	
communicationState	EMS	L	FIXED	
emsInSyncState	EMS	L	FIXED	
supportedRates	EMS	A	VALUE LIST (e.g., LR_Low_Order_TU3_VC3, LR_STS1_and_AU3_High_Order_VC3, LR_STS3c_and_AU4_VC4)	
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo.
Version 3.0 Additions (“Set By” and “Set When And How” have the same value for all additionalInfo)				
ASAPpointer	E	EMS (C), or by the NMS via assignASAP (L)	FIXED	
Manufacturer	E	EMS (C) or NMS via setAdditionalInfo (L)	FREE	
NetworkAccess	E	EMS (C), or NMS via	FREE	

Domain		setAdditionalInfo (L)		

2.14.2 Interfaces

There are no interfaces other than the ManagedElementIterator, which is used to retrieve a large number of ME data objects.

2.14.3 Notifications

To be provided by the organization preparing an interoperability statement

2.15 ManagedElementManager Module

2.15.1 Data Types

This module has no data types that represent second level objects.

2.15.2 Interfaces

Table 2-39. ManagedElement Mgr

Operation	Status	Support	Exception/ Error Reason	Comments
getAllActiveAlarms	M			
getAllCrossConnections	M			
getAllManagedElementNames	M			
getAllManagedElements	M			
getAllPTPNames	M			
getAllPTPs	M			
getContainedCurrentTPNames	O			
getContainedCurrentTPs	O			
getContainedInUseTPNames	M			
getContainedInUseTPs	M			
getContainedPotentialTPNames	M			
getContainedPotentialTPs	M			
getContainingSubnetworkNames	M			
getContainingTPNames	M			
getContainingTPs	M			

getManagedElement	M			
getTP	M			
setTPData	M			
Version 3.0 Additions				
createGTP	O			
deleteGTP	O			
getAllFixedCrossConnections	O			
getAllFTPNames	O			
getAllFTPs	O			
getAllGTPNames	O			
getAllGTPs	O			
getAllPTPNamesWithoutFTPs	O			
getAllPTPsWithoutFTPs	O			
getAllUnacknowledgedActiveAlarms	O			
getContainingGTP	O			
getGTP	O			
getPotentialFixedCCs	O			
modifyGTP	O			
setGtpAlarmReportingOff	O			
setGtpAlarmReportingOn	O			
verifyTMDAssignment	O			

2.15.3 Notifications

There are no specific notifications associated with the data types in this module.

2.16 MLSNPP Module

2.16.1 Data Types

Table 2-40. MultiLayerSNPP

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	EMS	C	FREE	
userLabel	E	A – set by the EMS (C) or by the NMS via setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can

owner	E	A – set by the EMS (C) or by the NMS via setOwner (L)	FREE	be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
nativeEMSName	E	A – set by the EMS (C) or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
mLSNPPAliasList	EMS	C	FIXED	
direction	EMS	C	FIXED	
layeredSNPPList	EMS	C	FIXED	
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo

2.16.2 Interfaces

Table 2-41. MLSNPPMgr

Operation	Status	Support	Exception/ Error Reason	Comments
setTNANameForMLSNPP	O			

2.16.3 Notifications

There are no specific notifications associated with the data types in this module.

2.17 MLSNPP Link Module

2.17.1 Data Types

Table 2-42. MultiLayerSNPPLink

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	EMS	C	FREE	
userLabel	E	A – set by the EMS (C) or by the NMS via setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
owner	E	A – set by the EMS (C) or by the NMS via setOwner (L)	FREE	
nativeEMSName	E	A – set by the EMS (C) or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
direction	EMS	C	FIXED	
aMLRAName	EMS	C	FIXED	
zMLRAName	EMS	C	FIXED	
sNPPLinkList	EMS	C	FIXED	
interfaceType	EMS	C	FIXED	

signallingParameters	E	A – set by the EMS (C) or by the NMS via setSignallingProtocolAndParameters (L) or modifySignallingProtocolParameters (L)	VALUE LIST	NVSList
signallingControllerIdentifier	E	A – set by the EMS (C) or by the NMS via assignSignallingController (L) or deassignSignallingController (L)	FREE	
signallingProtocol	E	A – set by the EMS (C) or by the NMS via setSignallingProtocolAndParameters (L)	FIXED	
signallingEnabled	E	A – set by the EMS (C) or by the NMS via enableSignalling (L) or disableSignalling (L)	FIXED	
cost	EMS	C	VALUE LIST	NVSList
discovered	EMS	C	FIXED	Optional
availability	EMS	C	VALUE LIST	NVSList, Optional
linkSRG	EMS	C	FIXED	
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo

2.17.2 Interfaces

Table 2-43. MLSNPPLinkMgr

Operation	Status	Support	Exception/ Error Reason	Comments
getAvailableCapacity	M			
assignSignallingController	O			
deassignSignallingController	O			
setSignallingProtocolAndParameters	O			
setTNANameForMLSNPPLinkEnd	O			
modifySignallingProtocolParameters	O			
enableSignalling	O			
disableSignalling	O			

2.17.3 Notifications

There are no specific notifications associated with the data types in this module.

2.18 MTNM Version Module

2.18.1 Data Types

This module has no data types that represent second level objects.

2.18.2 Interfaces

Table 2-44. Version

Operation	Status	Support	Exception/ Error Reason	Comments
getVersion	M			

2.18.3 Notifications

There are no specific notifications associated with the data types in this module.

2.19 MultiLayerSubnetwork Module

2.19.1 Data Types

Table 2-45. MultiLayerSubnetwork

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	EMS	C	FREE	
userLabel	E	A – set by the EMS (C), or by the NMS via setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
nativeEMSName	E	A – set by the EMS (C), or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
owner	NMS	L – setOwner	FREE	
subnetworkType	EMS	C	FIXED (e.g., TOPO_SINGLETON)	
supportedRates	EMS	A	VALUE LIST (e.g., LR_Low_Order_TU3_VC3, LR_STS1_and_AU3_High_ Order_VC3, LR_STS3c_and_AU4_VC4)	
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo. For v3.5 the following elements are specified: RoutingAreaLevel, SuperiorMLRA, LayeredRoutingAreaIdList, SupportingMEName, SRGNode
Version 3.0 Additions				
NetworkAccess Domain	E	EMS (C) or NMS via setAdditionalInfo (L)	FREE	

Table 2-46. TP Pool Create Data

Attribute Name	Set By	Set When and How	Format	Clarification Needed
userLabel	NMS	C – createTPPool	FREE	
forceUniqueness	NMS	C – createTPPool	FIXED	
owner	NMS	C – createTPPool	FREE	
containingMLSN	NMS	C – createTPPool	FIXED	
containedMembers	NMS	C – createTPPool	FIXED	
transmissionParams	NMS	C – createTPPool	VALUE LIST	
descriptionOfUse	NMS	C – createTPPool	FREE	
additionalCreationInfo	NMS	C – createTPPool	FREE	

2.19.2 Interfaces

The EMS supplier should indicate which of the 4 SNC modes of operation it can support (see Section 4.4 for a description of the SNC modes of operation). The NMS supplier should also indicate which mode(s) it can support.

Table 2-47. Multi-layer Subnetwork Manager

Operation	Status	Support	Exception/ Error Reason	Comments
activateSNC	O			The activateSNC operation is idempotent. The EMS does not send cross connection commands to NE a second time however the commands may be sent for the transmission parameters (the EMS vendor should indicate if this is the case for their EMS).
checkValidSNC	O			

createAndActivateSNC	M			For this operation and the other SNC establishment operations, the EMS vendor should indicate whether it is capable of automatically establishing HO SNCs to support a new SNC request, and whether the EMS supports reconfiguration of existing SNC in support of a new SNC request.
createSNC	O			
deactivateAndDeleteSNC	M			
deactivateSNC	O			
deleteSNC	O			
getAllEdgePoints	M			
getAllEdgePointNames	M			
getAllManagedElements	M			
getAllManagedElementNames	M			
getAllSubnetworkConnections	M			
getAllSubnetworkConnectionNames	M			
getAllSubnetworkConnectionsWithTP	M			
getAllSubnetworkConnection NamesWithTP	M			
getAllTopologicalLinks	O			
getAllTopologicalLinkNames	O			
getAllTPPoolNames	O			
getAllTPPools	O			
getAssociatedTP	O			
getMultiLayerSubnetwork	M			
getRoute	O			
getSNC	O			
getSNCsByUserLabel	O			
getTopologicalLink	O			
getTPGroupingRelationships	O			
Version 3.0 Additions				
addRoute	O			
createModifiedSNC	O			

createTPPool	O			
deleteTPPool	O			
getAllFixedSubnetworkConnection Names	O			
getAllFixedSubnetworkConnection NamesWithTP	O			
getAllFixedSubnetworkConnections	O			
getAllFixedSubnetworkConnections WithTP	O			
getBackupRoutes	O			
getIntendedRoute	O			
getRouteAndTopologicalLinks	O			
getTPPool	O			
modifySNC	O			
modifyTPPool	O			
removeRoute	O			
setIntendedRoute	O			
setRoutesAdminState	O			
swapSNC	O			
switchRoute	O			
Version 3.5 Additions				
getAllSubordinateMLSNs	O			
getAllSubordinateRAidsWithConnection	O			
getMLSNPPLink	O			
getAllMLSNPPLinks	O			
getAllInternalMLSNPPLinks	O			
getAllEdgeMLSNPPLinks	O			
getAllMLSNPPs	O			
getAllCallsAndTopLevelConnections	M			
getAllCallsAndTopLevelConnectionsAndSNCs	O			
getAllCallsAndTopLevelConnectionsWithME	O			
getAllCallsAndTopLevelConnectionsAndSNCsWithME	O			

getAllCallsAndTopLevelConnectionsAndSNCsWithTP	O			
getAllCallIdsWithTP	O			
getAllCallIdsWithSNPPOrTNAName	O			
getCallAndTopLevelConnectionsAndSNCs	O			
getCallAndTopLevelConnections	O			
establishCall	O			
modifyCall	O			
releaseCall	O			
getCall	O			
addConnections	O			
removeConnections	O			
getConnectionsAndRouteDetails	O			
modifyDiversityAndCorouting	O			

2.19.3 Notifications

To be provided by the organization preparing an interoperability statement

2.20 NMS Session Module

2.20.1 Data Types

This module has no data types that represent second level objects.

2.20.2 Interfaces

Table 2-48. NmsSession

Operation	Status	Support	Exception/ Error Reason	Comments
eventLossCleared	M			These operations are mandatory in the sense that the NMS can not throw any exceptions. However, what the NMS does when these methods are called is entirely up to the NMS. The NMS could, for example, do nothing when receiving either of these operations.
eventLossOccurred	M			
Version 3.0 Additions				
alarmLossOccurred	M			

2.20.3 Notifications

There are no specific notifications associated with the data types in this module.

2.21 Notifications Module

2.21.1 Data Types

This module has no data types that represent second level objects.

2.21.2 Interfaces

There are no interfaces other than the EventIterator, which is used to retrieve a large number of events. This iterator is used in conjunction with the getAllActiveAlarms operation in the managedElementManager module, and the getAllEMSActiveAlarms operation in the emsMgr module.

2.21.3 Notifications

There are no specific notifications associated with the data types in this module.

2.22 Performance Module

2.22.1 Data Types

Table 2-49. PMData

Attribute Name	Set By	Set When and How	Format	Clarification Needed
tpName	EMS	C	FIXED	
layerRate	EMS	C	FIXED	
granularity	EMS	C	FIXED	
retrievalTime	EMS	C	FIXED	
pmMeasurementList	EMS	C	-	See PMMeasurement table below

Table 2-50. PMMeasurement

Attribute Name	Set By	Set When and How	Format	Clarification Needed
pmParameterName	EMS	C	VALUE LIST	The EMS vendor (SP) should describe their support (required) list of PM parameters.
pmLocation	EMS	C	FIXED	The EMS vendor (SP) should indicate support (need) for far-end, near-end and/or bi-directional monitoring for each PM parameter.
value	EMS	C	FIXED	
unit	EMS	C	FREE	The EMS vendor (SP) should describe the type of units they use (require) for each type of PM

				parameter.
intervalStatus	EMS	C	FIXED	

Table 2-51. PMThresholdValue

Attribute Name	Set By	Set When and How	Format	Clarification Needed
pmParameterName	EMS	C	VALUE LIST	The EMS vendor (SP) should indicate the PM parameter for which they support (require) thresholding.
pmLocation	EMS	C	FIXED	
thresholdType	EMS	C	FIXED	For each type of thresholdable PM parameter, the EMS vendor (SP) should indicate the type of TRIGGER supported (required) and whether or not there is an associated CLEAR.
value	EMS	C	FIXED	
unit	EMS	C	FREE	

Table 2-52. PMP (v3.0)

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	EMS	C	FREE	
userLabel	E	A – set by the EMS (C), or by the NMS via setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
nativeEMSName	E	A – set by the EMS (C), or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
owner	NMS	L – setOwner	FREE	
pmParameterWithThresholdsList	EMS	C	VALUE LIST	
monitoringState	E	A – set by the EMS (A), or by the NMS via enablePMData or disablePMData (L)	FIXED	
supervisionState	E	A – set by the EMS (A), or by the NMS via enableTCA or disableTCA (L)	FIXED	
additionalInfo	E	A – set by EMS (C), or by the NMS via setAdditionalInfo	FREE	Name/Value list defined by the EMS vendor

2.22.2 Interfaces

Table 2-53. PerformanceManagementMgr

Operation	Status	Support	Exception/ Error Reason	Comments
clearPMData	O			
disablePMData	M			
disableTCA	O			
enablePMData	M			
enableTCA	O			
getAllCurrentPMData	O			
getHistoryPMData	O			
getHoldingTime	O			
getMEPMcapabilities	M			
getTCATPParameter	M			
setTCATPParameter	O			
Version 3.0 Additions				
createTCAParameterProfile	O			
deleteTCAParameterProfile	O			
getAllPMPNames	O			
getAllPMPs	O			
getAllTCAParameterProfiles	O			
getAllTCAParameterProfileNames	O			
getProfileAssociatedTPs	O			
getTCAParameterProfile	O			
getTPHistoryPMData	O			
setTCAParameterProfile	O			
setTCAParameterProfilePointer	O			
Version 4.0 Additions				
getHistoryPMDataByPull	O			

There are also PMDataIterator and PMPIterator interfaces that allow for the retrieval of a large number of PM data records, and PMP objects, respectively. The PM specification also allows for the bulk retrieval of PM data via file transfer.

The TCAParameterProfileIterator allows for the bulk retrieval of TCA parameter profiles.

2.22.3 Notifications

To be provided by the organization preparing an interoperability statement

2.23 Protection Module

2.23.1 Data Types

Table 2-54. ProtectionGroup

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	EMS	C	FREE	
userLabel	E	A – set by the EMS (C), or by the NMS via setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
nativeEMSName	E	A – set by the EMS (C), or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
owner	NMS	L – setOwner	FREE	
protectionGroupType	EMS	C	FIXED	
protectionSchemeState	EMS	C	FIXED	
reversionMode	EMS	C	FIXED	
rate	EMS	C	FIXED	
pgpTPList	EMS	A	FIXED	
pgpParameters	EMS	A	FIXED	
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo.
Version 3.0 Additions				
ASAPpointer	E	EMS (C), or by the NMS via assignASAP (L)	FIXED	
G.774.3::APSfunction	E	EMS (C) or NMS via setAdditionalInfo (L)	FIXED	

Table 2-55. EprotectionGroup (v3.0)

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	EMS	C	FREE	
userLabel	E	A – set by the EMS (C), or by the NMS via setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
nativeEMSName	E	A – set by the EMS (C), or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
owner	NMS	L – setOwner	FREE	
eProtectionGroupType	EMS	C	FIXED	
protectionSchemeState	EMS	C	FIXED	
reversionMode	EMS	C	FIXED	
protectedList	EMS	A	FIXED	
protectingList	EMS	A	FIXED	
ePgpParameters	EMS	A	FIXED	
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo.
Version 3.0 Addition				
ASAPpointer	E	EMS (C), or by the NMS via assignASAP (L)	FIXED	
G.774.3::APSfunction	E	EMS (C) or NMS via setAdditionalInfo (L)	FIXED	

2.23.2 Interfaces

Table 2-56. ProtectionMgr

Operation	Status	Support	Exception/ Error Reason	Comments
getAllNUTTPNames	O			
getAllPreemptibleTPNames	O			
getAllProtectedTPNames	O			
getAllProtectionGroups	M			
getProtectionGroup	M			
performProtectionCommand	O			

retrieveSwitchData	O			
Version 3.0 Additions				
getAllEProtectionGroups	O			
getEProtectionGroup	O			
retrieveESwitchData	O			

Also available are the ProtectionGroupIterator and EprotectionGroupIterator interfaces that allow the NMS to retrieve a large number of protection groups and eprotection groups, respectively (one batch at a time).

2.23.3 Notifications

To be provided by the organization preparing an interoperability statement

2.24 Session Module

2.24.1 Data Types

This module has no data types that represent second level objects.

2.24.2 Interfaces

Table 2-57. Session

Operation	Status	Support	Exception/ Error Reason	Comments
endSession	M			
ping	M			
getAssociatedSession	M			This operation is automatically generated by the CORBA language mappings

2.24.3 Notifications

There are no specific notifications associated with the data types in this module.

2.25 SoftwareAndDataManager Module (v3.0)

2.25.1 Data Types

This module has no data types that represent second level objects.

2.25.2 Interfaces

Table 2-58. SoftwareAndDataManager

Operation	Status	Support	Exception/ Error Reason	Comments
abortMEBackup	O			
backupME	O			
getBackupList	O			
getMEBackupStatus	O			

There is also a BackupIdIterator interface that allows the NMS to retrieve a large number of BackupId structs.

2.25.3 Notifications

To be provided by the organization preparing an interoperability statement

2.26 Subnetwork Connection Module

2.26.1 Data Types

Table 2--59. Cross Connection

Attribute Name	Set By	Set When and How	Format	Clarification Needed
active	EMS		FIXED	
aEndNameList	E	C – these attributes are set when the associated SNC is created (i.e., via activateSNC, createSNC or createAndActivateSNC) If the CC is created outside of the interface, these attributes are set by the EMS.	FIXED	
zEndNameList	E		FIXED	
direction	E		FIXED	
ccType	E		FIXED	
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo.
Version 3.0 Additions				
Fixed	EMS	C	FIXED	

RouteActualState	E	A	FIXED	
RouteAdminState	E	EMS (C), or NMS via createAndActivateSNC, createSNC (C), or setAdditionalInfo (L)	FIXED	
RouteExclusive	E	EMS (C), or NMS via createAndActivateSNC, createSNC (C), or setAdditionalInfo (L)	FIXED	
RouteId	E	EMS (C), or NMS via createAndActivateSNC, createSNC (C)	FREE	
RouteIntended	E	EMS (C), or NMS via createAndActivateSNC, createSNC (C)	FIXED	
RouteInUseBy	E	EMS (C), or NMS via createAndActivateSNC, createSNC (C)	FIXED	
Version 3.5 additions for control plane				
RouteId	EMS	C	FREE	Part of additionalInfo
RouteIntended	EMS	C	FIXED	Part of additionalInfo
RouteActualState	EMS	A	FIXED	Part of additionalInfo
RouteAdminState	EMS	A	FIXED	Part of additionalInfo
RouteInUse	EMS	A	FIXED	Part of additionalInfo
RouteExclusive	EMS	C	FIXED	Part of additionalInfo
Supported CP connection	EMS	C	FREE	Part of additionalInfo
End points	EMS	C	FIXED	Part of additionalInfo
Routing Node information	EMS	C	FREE	Part of additionalInfo

Table 2-60. SNCCreateData

Attribute Name	Set By	Set When and How	Format	Clarification Needed
userLabel	NMS	C – createAndActivateSNC or createSNC	FREE	
forceUniqueness	NMS	C – createAndActivateSNC or createSNC	FIXED	
owner	NMS	C – createAndActivateSNC or createSNC	FREE	

direction	NMS	C – createAndActivateSNC or createSNC	FIXED	
staticProtection Level	NMS	C – createAndActivateSNC or createSNC	FIXED	
protectionEffort	NMS	C – createAndActivateSNC or createSNC	FIXED	
rerouteAllowed	NMS	C – createAndActivateSNC or createSNC	FIXED	
networkRouted	NMS	C – createAndActivateSNC or createSNC	FIXED	
sncType	NMS	C – createAndActivateSNC or createSNC	FIXED	
layerRate	NMS	C – createAndActivateSNC or createSNC	VALUE LIST	
ccInclusions	NMS	C – createAndActivateSNC or createSNC	FIXED	For Control plane connection, this represents the Routing Node connections
neTpInclusions	NMS	C – createAndActivateSNC or createSNC	FIXED	MLRAs/MEs/TLs/MLSNPPLinks/SNPPList/TPs/SNPs/GTPs/SNCs
fullRoute	NMS	C – createAndActivateSNC or createSNC	FIXED	
neTpSnc Exclusions	NMS	C – createAndActivateSNC or createSNC	FIXED	MLRAs/MEs/TLs/MLSNPPLinks/SNPPList/TPs/SNPs/GTPs/SNCs
aEnd	NMS	C – createAndActivateSNC or createSNC	FIXED	For Control plane connection, this identifies an SNP or an SNPP. The optional TNA/FTP/CTP values are provided by additional info.
zEnd	NMS	C – createAndActivateSNC or createSNC	FIXED	
additional CreationInfo	NMS	C – createAndActivateSNC or createSNC		Version 3.5 allows adding additional Control plane specific attributes. See the list of these attributes below.
Version 3.0 Additions				
A<*n*>Role	NMS	C – createAndActivateSNC or createSNC	FIXED	Indicates the end point role of an aEnd TP of an SNC, where <n> refers to the index of the TP within the aEnd list.

AlarmReporting	NMS	C – createAndActivateSNC or createSNC	FIXED	
ASAPpointer	NMS	C – createAndActivateSNC or createSNC	FIXED	
BLSRDirection	NMS	C – createAndActivateSNC or createSNC	FIXED	
BundledSNC Indicator	NMS	C – createAndActivateSNC or createSNC	FIXED	
MustRemoveGTPs	NMS	C – createAndActivateSNC or createSNC	FIXED	
NetworkAccess Domain	NMS	C – createAndActivateSNC or createSNC	FREE	
NetworkReroute	NMS	C – createAndActivateSNC or createSNC	FIXED	
SNC_INTENDED_ ROUTE_ EXCLUSIVE	NMS	C – createAndActivateSNC or createSNC	FIXED	
SNC_PRIORITY	NMS	C – createAndActivateSNC or createSNC	FIXED	
SNC_REVERTIVE	NMS	C – createAndActivateSNC or createSNC	FIXED	
Timeslot	NMS	C – createAndActivateSNC or createSNC	FIXED	
Z<*n*>Role	NMS	C – createAndActivateSNC or createSNC	FIXED	Indicates the end point role of a zEnd TP of an SNC, where <n> refers to the index of the TP within the zEnd list.
Version 3.5 Additions				
ConnectionName	NMS	C – by NMS (C) via establishCall	FIXED	
aEndTPList	NMS	C – by NMS (C) via establishCall	FIXED	
zEndTPList	NMS	C – by NMS (C) via establishCall	FIXED	
tNANameOrGroupTNAName	NMS	C – by NMS (C) via establishCall	FIXED	
tNANameOrGroupTNAName	NMS	C – by NMS (C) via establishCall	FIXED	

maximumCost	NMS	C – by NMS (C) via establishCall	FIXED	
routingConstraintEffort	NMS	C – by NMS (C) via establishCall	FIXED	
routeGroupLabel	NMS	C – by NMS (C) via establishCall	FIXED	

Table 2-61. SNC Modify Data

Attribute Name	Set By	Set When and How	Format	Clarification Needed
userLabel	NMS	C - createModifiedSNC	FREE	
forceUniqueness	NMS	C - createModifiedSNC	FIXED	
owner	NMS	C - createModifiedSNC	FREE	
direction	NMS	C - createModifiedSNC	FIXED	
modify_type	NMS	C - createModifiedSNC	FIXED	
retainOldSNC	NMS	C - createModifiedSNC	FIXED	
modify_servers_allowed	NMS	C - createModifiedSNC	FIXED	
staticProtection Level	NMS	C - createModifiedSNC	FIXED	
protectionEffort	NMS	C - createModifiedSNC	FIXED	
rerouteAllowed	NMS	C - createModifiedSNC	FIXED	
networkRouted	NMS	C - createModifiedSNC	FIXED	
sncType	NMS	C - createModifiedSNC	FIXED	
layerRate	NMS	C - createModifiedSNC	VALUE LIST	
addedOrNewRoute	NMS	C - createModifiedSNC	FIXED	
removedRoute	NMS	C - createModifiedSNC	FIXED	
neTpInclusions	NMS	C - createModifiedSNC	FIXED	
fullRoute	NMS	C - createModifiedSNC	FIXED	
neTpSnc Exclusions	NMS	C - createModifiedSNC	FIXED	
aEnd	NMS	C - createModifiedSNC	FIXED	
zEnd	NMS	C - createModifiedSNC	FIXED	
additionalCreationInfo	NMS	C - createModifiedSNC		
Version 3.0 Additions				

AlarmReporting	NMS	C - createModifiedSNC	FIXED	
ASAPpointer	NMS	C - createModifiedSNC	FIXED	
MustRemoveGTPs	NMS	C - createModifiedSNC	FIXED	
NetworkAccess Domain	NMS	C - createModifiedSNC	FREE	
NetworkReroute	NMS	C - createModifiedSNC	FIXED	
SNC_PRIORITY	NMS	C - createModifiedSNC	FIXED	

Table 2-62. Subnetwork Connection

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	E	C – set by the EMS (C) or by the NMS (C) via establishCall	FREE	Version 3.5 allows NMS to set the name
userLabel	E	A – set by the EMS (C), or by the NMS (L) via setUserLabel, createAndActivateSNC, createSNC	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
nativeEMSName	E	A – set by the EMS (C), or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
owner	NMS	L – setOwner, createAndActivateSNC, createSNC	FREE	
sncState	EMS	A	FIXED	EMS vendor, Service Provider (SP) or NMS vendor should indicate whether or not they support the *pending* state. In cases where the *pending* state is supported, it should indicate whether "Soft reservation with multiple pending" or "Hard Reservation with single pending" is supported. For Control plane SNC, always fixed to ACTIVE
direction	E	C – any of the SNC establishment operations	FIXED	
rate	E		FIXED	

staticProtection Level	E	If the SNC is created outside of the interface, these attributes are set by the EMS.	FIXED NMS requests protection level and EMS may provide a lower level if the Protection Effort is set to “bestEffort” in the SNC request	EMS vendor (SP or NMS vendor) should indicate the protection levels that they support (require). Any dependencies on MEs should be noted.
sncType	E		FIXED	
aEnd	E		FIXED	For control plane SNC, shall identify an SNP or an SNPP. The optional TNA/FTP/CTP values are provided by additional info.
zEnd	E		FIXED	
rerouteAllowed	E		FIXED	
networkRouted	E		FIXED	
additionalInfo	E	A – set by EMS (C), or by the NMS via setAdditionalInfo	FREE	Name/Value list defined by the EMS vendor In version 3.5, it allows additional attributes for Control plane connection to be added via (1) the additionalCreationInfo field of SNCCreateDataList_T in the establishCall operation or (2) the additionalModificationInfo field of CallModifyData_T in the modifyCall operation
route (The SNC’s route is retrieved by getRoute operation and is characterized by the subtending cross connections. The SNC struct does not have a “route” attribute.)	All or part of the route can be specified by the NMS or the NMS can let the EMS design the entire route	C	FIXED	
Version 3.0 Additions				
A<*n*>Role	E	EMS (C) or set by the NMS using the createAndActivateSNC or createSNC operation	FIXED	Indicates the end point role of an aEnd TP of an SNC, where <*n*> refers to the index of the TP within the aEnd list.
AlarmReporting	E	EMS (A), or NMS via createAndActivateSNC, createSNC (C), or setAdditionalInfo (L)	FIXED	

ASAPpointer	E	EMS (C), or NMS via createAndActivateSNC, createSNC (C), or assignASAP (L)	FIXED	
BundledSNC Indicator	E	EMS (A), or NMS via createAndActivateSNC, createSNC (C), or setAdditionalInfo (L)	FIXED	
CorrelationID	EMS	L	FREE	
Fixed	EMS	C	FIXED	
MustRemoveGTPs	E	EMS (A), or NMS via createAndActivateSNC, createSNC (C), or setAdditionalInfo (L)	FIXED	
NetworkAccess Domain	E	EMS (A), or NMS via createAndActivateSNC, createSNC (C), or setAdditionalInfo (L)	FREE	
NetworkReroute	E	EMS (A), or NMS via createAndActivateSNC, createSNC (C), or setAdditionalInfo (L)	FIXED	
SNC_PRIORITY	E	EMS (A), or NMS via createAndActivateSNC, createSNC (C), or setAdditionalInfo (L)	FIXED	
SNC_REVERTIVE	E	EMS (A), or NMS via createAndActivateSNC, createSNC (C), or setAdditionalInfo (L)	FIXED	
Z<*n*>Role	E	EMS (C) or set by the NMS using the createAndActivateSNC or createSNC operation	FIXED	Indicates the end point role of a zEnd TP of an SNC, where <*n*> refers to the index of the TP within the zEnd list.
Version 3.5 additions				
connectionSetUpType	EMS	EMS (C)	FIXED	
CallId	EMS	EMS (C)	FIXED	
CallName	EMS	EMS (C)	FIXED	
ConnectionId	EMS	EMS (C)	FIXED	
aEndTPList	EMS	EMS (C)	FIXED	
zEndTPList	EMS	EMS (C)	FIXED	
aEndTNANameList	EMS	EMS (C)	FIXED	
zEndTNANameList	EMS	EMS (C)	FIXED	
connectionState	EMS	EMS (C)	FIXED	

usingHomeRoute	EMS	EMS (C)	FIXED	
routeGroupLabel	EMS	EMS (C)	FIXED	

Table 2-63. Route Create Data (v3.0)

Attribute Name	Set By	Set When and How	Format	Clarification Needed
intended	NMS	C – addRoute	FIXED	
exclusive	NMS	C – addRoute	FIXED	
ccInclusions	NMS	C – addRoute	FIXED	
neTpInclusions	NMS	C – addRoute	FIXED	MLRAs/MEs/TLs/MLSNPPLinks/SNPPList/TPs/SNPs/GTPs
fullRoute	NMS	C – addRoute	FIXED	
neTpSncExclusions	NMS	C – addRoute	FIXED	MLRAs/MEs/TLs/MLSNPPLinks/SNPPList/TPs/SNPs/GTPs
additionalCreationInfo	NMS	C – addRoute	-	

Table 2-64. Route Descriptor (v3.0)

Attribute Name	Set By	Set When and How	Format	Clarification Needed
id	EMS	C- by EMS in response to addRoute	FREE	
intended	EMS	C- by EMS in response to addRoute	FIXED	
actualState	EMS	C- by EMS in response to addRoute	FIXED	
administrativeState	EMS	C- by EMS in response to addRoute	FIXED	
inUseBy	EMS	C- by EMS in response to addRoute	FIXED	
exclusive	EMS	C- by EMS in response to addRoute	FIXED	
routeXCs	EMS	C- by EMS in response to addRoute	FIXED	
additionalInfo	EMS	C- by EMS in response to addRoute	-	

2.26.2 Interfaces

There are no interfaces in the subnetwork connection module other than cross connection iterator (i.e., CCIterator) and the SNC iterator (i.e., SNCIterator).

2.26.3 Notifications

To be provided by the organization preparing an interoperability statement

2.27 Termination Point Module

2.27.1 Data Types

Table 2-65. Termination Point

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	EMS	C	FREE	The CTP names are fully specified in TMF 814. However, any an example PTP name is given. It is recommended that the EMS supplier state the format used for their PTP names.
userLabel	E	A – set by the EMS (C), or by the NMS via setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes. In version 3.5, FTPs may be created by the NMS, entailing different behaviour for these attributes. See section 2.9.
nativeEMSName	E	A – set by the EMS (C), or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
owner	NMS	L – setOwner	FREE	
ingressTrafficDescriptorName	NMS	A – setTPData (L), or via side effect through activateSNC, deactivateAndDeleteSNC, deactivateSNC (C)	FIXED	The assignment of a Transmission Descriptor (TMD) to a Termination Point (TP) as egress or ingress TMD by using the TMD's name amounts to an overwriting of the layered transmission parameters of the TP by the layered transmission parameters of the TMD, and to an overwriting of the additional info parameters of the TP by the "additional TP information" parameters of the TMD.
egressTrafficDescriptorName	NMS	A – setTPData (L), or via side effect through activateSNC, deactivateAndDeleteSNC, deactivateSNC (C)	FIXED	
type	EMS	C	FIXED	

connectionState	EMS	A	FIXED	
tpMappingMode	E	A – setTPData (L), or via side effect through activateSNC, deactivateAndDeleteSNC, deactivateSNC (C) Can also be set by the EMS	FIXED	
direction	E	C – via side effect through any of the SNC establishment operations Can also be set by the EMS	FIXED	
transmissionParams	E	A (depends on the specific transmission parameter) – setTPData (L), or via side effect through activateSNC, deactivateAndDeleteSNC, deactivateSNC (C)	VALUE LIST	EMS vendor (SP or NMS vendor) should indicated supported (required) transmissionParams and associated values
tpProtectionAssociation	EMS	A	FIXED	
edgePoint	EMS	A	FIXED	
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo.
Version 3.0 Additions				
ClientConnectivity	EMS	C	FIXED	
ConformanceDefinition	E	EMS (A) or NMS via setAdditionalInfo (L)	VALUE LIST	
ContainedMember	E		FIXED	Used for TP Pools only
DescriptionOfUser	E	EMS (A) or NMS via setAdditionalInfo (L)	FREE	Used for TP Pools only
EgressTMDstate	EMS	A	FIXED	
EquipmentProtected	EMS	A	FIXED	
IngressTMDstate	EMS	A	FIXED	
MemberContainingME<n>	E	EMS (A) or NMS via modifyTPPool	FIXED	Used for TP Pools only
MemberContainingTP<n>	E	EMS (A) or NMS via modifyTPPool	FIXED	Used for TP Pools only
NetworkAccessDomain	E	EMS (C) or NMS via setAdditionalInfo (L)	FREE	
NumberOfMembers	E	EMS (A) or NMS via modifyTPPool	FIXED	Used for TP Pools only

NumberOfIdleMembers	E	EMS (A) or NMS via modifyTPPool	FIXED	Used for TP Pools only
ServerConnectivity	EMS	A	FIXED	
ServiceCategory	E	EMS (C) or NMS via setAdditionalInfo (L)	FIXED	

Table 2-66. Group Termination Point (v3.0)

Attribute Name	Set By	Set When and How	Format	
name	EMS	C	FREE	
userLabel	E	A – set by the EMS (C), or by the NMS via setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
nativeEMSName	E	A – set by the EMS (C), or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
owner	NMS	L – setOwner	FREE	
alarmReportingIndicator	E	A – set by the EMS (C), or by the NMS via or setGtpAlarmReportingOn (L)	FIXED	
listOfTPs	E	A – this attribute can be set at time of creation or modified by either the EMS, or the NMS. The NMS uses the createGTP and modifyGTP operations to set and modify the listOfTPs, respectively.	FIXED	
gtpConnectionState	EMS	A	FIXED	
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo.
Version 3.0 Additions				
ASAPpointer	E	EMS (C), or by the NMS via assignASAP (L)	FIXED	
NetworkAccessDomain	E	EMS (C), or NMS via createGTP(C) or setAdditionalInfo (L)	FREE	

2.27.2 Interfaces

There are no interfaces in the subnetwork connection module other than Termination Point iterator and the GTP iterator. The TP iterator is used to retrieve a large number of TPs, a batch at a time. The GTP iterator is used to retrieve a large number of GTPs, a batch at a time.

2.27.3 Notifications

To be provided by the organization preparing an interoperability statement

2.28 Topological Link Module

2.28.1 Data Types

Table 2-67. Topological Link

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	EMS	C	FREE	
userLabel	E	A – set by the EMS (C), or by the NMS via setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
nativeEMSName	E	A – set by the EMS (C), or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
owner	NMS	L – setOwner	FREE	
direction	EMS	C	FIXED	
rate	EMS	C	FIXED	
aEndTP	EMS	C	FIXED	
zEndTP	EMS	C	FIXED	
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo.
Version 3.0 Additions				
AlarmReporting	E	EMS (C) or NMS via setAdditionalInfo (L)	FIXED	
ASAPpointer	E	EMS (C), or by the NMS via assignASAP (L)	FIXED	
AllocatedNumber	E	EMS (C) or NMS via setAdditionalInfo (L)	FIXED	
FragmentServer Layer	E	EMS (C) or NMS via setAdditionalInfo (L)	FIXED	
NetworkAccess Domain	E	EMS (C) or NMS via setAdditionalInfo (L)	FREE	

Table 2-68. TL Create Data (v3.0)

Attribute Name	Set By	Set When and How	Format	Clarification Needed
userLabel	NMS	C - createTopologicalLink	FREE	
forceUniqueness	NMS	C - createTopologicalLink	FIXED	
owner	NMS	C - createTopologicalLink	FREE	
direction	NMS	C - createTopologicalLink	FIXED	
rate	NMS	C - createTopologicalLink	FIXED	
aEndTP	NMS	C - createTopologicalLink	FIXED	
zEndTP	NMS	C - createTopologicalLink	FIXED	
additional CreationInfo	NMS	C - createTopologicalLink	FREE	
Version 3.0 Additions				
AlarmReporting	NMS	C - createTopologicalLink	FIXED	
ASAPpointer	NMS	C - createTopologicalLink	FIXED	
NetworkAccess Domain	NMS	C - createTopologicalLink	FREE	

2.28.2 Interfaces

There are no interfaces in the subnetwork connection module other than the Topological Link iterator.

2.28.3 Notifications

To be provided by the organization preparing an interoperability statement

2.29 Traffic Conditioning Profile Module (v3.5)

2.29.1 Data Types

Table 2-69. TCProfile

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	EMS	C	FREE	

userLabel	E	A – set by the EMS (C), or by the NMS via setUserLabel (L) or modifyTransmissionDescriptor (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
nativeEMSName	E	A – set by the EMS (C), or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
owner	NMS	L – setOwner or modifyTransmissionDescriptor	FREE	
defaultProfile	EMS	C	FIXED	Defines the TC Profile as a non-deleteable profile (modification may be possible or rejected). An EMS may contain more than one default profile which condition different TPs. Default profiles are created by EMS only.
transmissionParams	NMS	A – set by the NMS when requesting the creation (C) of a Transmission Descriptor, or by modifyTransmissionDescriptor (L)	VALUE LIST	EMS vendor (SP or NMS vendor) should indicated supported (required) transmissionParams and associated values
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo.

Table 2-70. TCProfile Create Data

Attribute Name	Set By	Set When and How	Format	Clarification Needed
userLabel	NMS	C – createTransmission Descriptor	FREE	
forceUniqueness	NMS	C – createTransmission Descriptor	FIXED	
owner	NMS	C – createTransmission Descriptor	FREE	
transmissionParams	NMS	C – createTransmission Descriptor	VALUE LIST	
additionalCreationInfo	NMS	C – createTransmission Descriptor	FREE	

Note: TCProfile Create Data is also used to modify a TC Profile.

There is also the TCProfileIterator interface that allows the NMS to retrieve a large number of Traffic Conditioning Profiles (a batch at a time).

2.29.2 Interfaces

Table 2-71. TCProfileMgr

Operation	Status	Support	Exception/ Error Reason	Comments
createTCProfile	O			
deleteTCProfile	O			
getAllTCProfiles	O			
getTCProfileAssociatedTPs	O			
getTCProfile	O			
modifyTCProfile	O			

2.29.3 Notifications

To be provided by the organization preparing an interoperability statement

2.30 TrafficDescriptor Module

Traffic Descriptors have been included in release 3 of the NML-EML interface for reasons of backward/forward compatibility. For a release 3 implementation of this interface it is recommended that Traffic Descriptors are not used but instead Transmission Descriptors are used in their place.

2.30.1 Data Types

Traffic descriptors can be created by either the EMS or the NMS. They are presently used in conjunction with ATM. The NMS uses the createTrafficDescriptor operation to create a traffic descriptor.

Table 2-72. Traffic Descriptor

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	EMS	C	FREE	
userLabel	E	A – set by the EMS (C), or by the NMS via setUserLabel (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
nativeEMSName	E	A – set by the EMS (C), or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
owner	NMS	L – setOwner	FREE	
serviceCategory	E	C	FIXED	
trafficParameters	E	C	FIXED	
conformanceDefinition	E	C	FIXED	
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo.

Table 2-73. TD Create Data

Attribute Name	Set By	Set When and How	Format	Clarification Needed
userLabel	NMS	C – createTraffic Descriptor	FREE	
forceUniqueness	NMS	C – createTraffic Descriptor	FIXED	
owner	NMS	C – createTraffic Descriptor	FREE	
serviceCategory	NMS	C – createTraffic Descriptor	FIXED	
trafficParameters	NMS	C – createTraffic Descriptor	FIXED	
conformance Definition	NMS	C – createTraffic Descriptor	FIXED	
additionalInfo	NMS	C – createTraffic Descriptor	FREE	

2.30.2 Interfaces

Table 2-74. TrafficDescriptorMgr

Operation	Status	Support	Exception/ Error Reason	Comments
createTrafficDescriptor	O			
deleteTrafficDescriptor	O			
getAllTrafficDescriptorNames	M			
getAllTrafficDescriptors	M			
getAssociatedCTPs	M			
getTrafficDescriptor	M			

There is also the TrafficDescriptorIterator interface that allows the NMS to retrieve a large number of Traffic Descriptors (a batch at a time).

2.30.3 Notifications

To be provided by the organization preparing an interoperability statement

2.31 Transmission Descriptor Module (v3.0)

2.31.1 Data Types

Table 2-75. Transmission Descriptor

Attribute Name	Set By	Set When and How	Format	Clarification Needed
name	EMS	C	FREE	
userLabel	E	A – set by the EMS (C), or by the NMS via setUserLabel (L) or modifyTransmissionDescriptor (L)	FREE	As noted in Section 4.3, userLabel, owner, nativeEMSName can be used in a variety of ways. The EMS vendor (SP) should describe their particular usage of these attributes.
nativeEMSName	E	A – set by the EMS (C), or by the NMS via setNativeEMSName (L)	FREE (e.g. CLEI code)	
owner	NMS	L – setOwner or modifyTransmissionDescriptor	FREE	
transmissionParams	NMS	A – set by the NMS when requesting the creation (C) of a Transmission Descriptor, or by modifyTransmissionDescriptor (L)	VALUE LIST	EMS vendor (SP or NMS vendor) should indicated supported (required) transmissionParams and associated values
additionalTPInfo	NMS	A – set by the NMS when requesting the creation of a Transmission Descriptor, or by modifyTransmissionDescriptor (L)	FREE	
ConformanceDefinition (in additionalTPInfo)	E	EMS (C) or NMS via setAdditionalInfo (L)	VALUE LIST	
ServiceCategory (in additionalTPInfo)	E	EMS (C) or NMS via setAdditionalInfo (L)	FIXED	
containingTMDName	NMS	A – set by the NMS when requesting the creation of a Transmission Descriptor (C), or by modifyTransmissionDescriptor (L)	FIXED	
externalRepresentationReference	NMS	A – set by the NMS when requesting the creation of a Transmission Descriptor(C), or by modifyTransmissionDescriptor (L)	FREE	
additionalInfo	-	-	-	Name/Value list defined by the EMS vendor. Also, extensions are included in additionalInfo.

Table 2-76. TMD Create Data

Attribute Name	Set By	Set When and How	Format	Clarification Needed
userLabel	NMS	C – createTransmission Descriptor	FREE	
forceUniqueness	NMS	C – createTransmission Descriptor	FIXED	
owner	NMS	C – createTransmission Descriptor	FREE	
transmissionParams	NMS	C – createTransmission Descriptor	VALUE LIST	
additionalObjectInfo	NMS	C – createTransmission Descriptor	FREE	
containingTMDName	NMS	C – createTransmission Descriptor	FIXED	
externalRepresentation Reference	NMS	C – createTransmission Descriptor	FREE	
additionalCreationInfo	NMS	C – createTransmission Descriptor	FREE	

Table 2-77. TMD Modification Data

Attribute Name	Set By	Set When and How	Format	Clarification Needed
userLabel	NMS	C – createTransmission Descriptor	FREE	
forceUniqueness	NMS	C – createTransmission Descriptor	FIXED	
owner	NMS	C – createTransmission Descriptor	FREE	
transmissionParams	NMS	C – createTransmission Descriptor	VALUE LIST	
additionalObjectInfo	NMS	C – createTransmission Descriptor	FREE	
containingTMDName	NMS	C – createTransmission Descriptor	FIXED	
externalRepresentation Reference	NMS	C – createTransmission Descriptor	FREE	
additionalCreationInfo	NMS	C – createTransmission Descriptor	FREE	

There is also the TransmissionDescriptorIterator interface that allows the NMS to retrieve a large number of Traffic Descriptors (a batch at a time).

2.31.2 Interfaces

Table 2-78. TransmissionDescriptorMgr

Operation	Status	Support	Exception/ Error Reason	Comments
createTransmissionDescriptor	O			
deleteTransmissionDescriptor	O			
getAllTransmissionDescriptorNames	O			
getAllTransmissionDescriptors	O			
getAssociatedTPs	M			
getTransmissionDescriptor	O			
modifyTransmissionDescriptor	O			Added in version 3.5.
validateTMDAssignmentToObject	O			
setTMDAssociation	O			

2.31.3 Notifications

To be provided by the organization preparing an interoperability statement

2.32 Transmission Parameters Module

The MTNM transmission parameters are defined in two supporting documents, i.e., *Layered Parameters* and *Layer Rates*.

2.32.1 Data Types

This module has no data types that represent second level objects.

2.32.2 Interfaces

There are no interfaces associated with the TransmissionParameters module.

2.32.3 Notifications

There are no specific notifications associated with the data types in this module.

2.33 Miscellaneous

This section covers functional implementation items that do not fit easily in the templates presented earlier in Section 2.

2.33.1 Version of CORBA and CORBA Services

The vendor should indicate the version of CORBA and the version of the CORBA services that they are using in their product.

Table 2-79. CORBA Version

	OMG Version	Version of Vendor Product	Comments
CORBA	e.g., CORBA 2.2		
Naming Service	e.g., Version 2.2	e.g., Vendor X Version 5.1 (of OMG Version 2.2 Naming Service)	
Notification Service			Note that a vendor could use an OMG compliant Telecom Log Service to support the MTNM notifications requirements.
Telecom Log Service			

2.33.2 Usage of the OMG Notification Service

A number of parameters are associated with the usage of the Notification Service. The supporting document to the MTNM IDL, entitled Notification Service Usage, recommends values for the various Notification Service parameters. Table 2-80 is used by the vendor or service provider to indicate any variances from the recommendations in the Notification Service Usage document.

Table 2-80. Usage of the Notification Service

Notification Service Characteristic	Vendor Support (Service Provider Requirement)	Comment
Push or Pull		The MTNM Notification Service usage guidelines stipulate that Push should be used
Number of Notifications Channel		The MTNM Notification Service usage guidelines recommend one channel per EMS
Instances of Notification Service		The MTNM Notification Service usage guidelines recommend one per EMS
EventReliability		The use of Persistency is recommended in both cases
ConnectionReliability		
Usage of Event Priority		The same priority for all events is recommended
Discard Policy		FIFO is recommended in both cases
Delivery Order Policy		
Start Time Supported		The recommended value is “FALSE.”
Maximum Batch Size		Batch delivery is not recommended. Thus, the recommended value is “0.”
Pacing Interval		The recommended value is “0.”
RejectNewEvents		The recommended value is “TRUE.”

TimeOut		The recommended value is 30 minutes for NT_TCA and NT_ALARM events, and 24 hours for all other events.

2.33.3 Proprietary LayerRates

The EML vendor should publish a list of proprietary layerRates that are used along with their definition. When a layerRate has been subsumed into the standard this list should be augmented with the equivalence and obsolescence statement.

The proprietary layerRate should be used where:

- A transport technology within a product is as yet to be incorporated in the MTNM model
    - This layerRate should be marked as "**being standardized**" in its definition in the conformance statement
    - When the transport technology is incorporated the vendor should identify the equivalence and obsolescence in their conformance statement
- A proprietary transport technology has been used which is necessary to expose over the interface for monitoring and/or configuration purposes
    - This layerRate should be marked as "**proprietary**" in its definition in the conformance statement
- Where an NE only provides partial information on the layerRate, e.g., the NE only indicates that the layerRate is SONET or ATM
    - This layerRate should be marked as "**partial**" in its definition and should (where possible) be related to layerRates that it covers

Where a proprietary layerRate is used it may be necessary to name a TP (CTP or FTP) from that layerRate. In this case a name-string similar in structure to that used for a standardize layerRate (in the layerRate list) should be allocated by the vendor and recorded in the conformance statement. To conform to the specification this name-string will start with "PROP" (e.g. "PROP_sts_291c").

Table 2-81. Proprietary LayerRate Table

Layer Rate Number (integer greater than or equal to 10,000)	**Layer Rate Name** (the names of all proprietary layer rates start with "PROP")	**Status (either "Being Standardized", "Proprietary", "Partial" or "Obsolete"**
10,000	PROP_DSR_OC3072_and_STM1024	Being Standardized
10,001	PROP_STS_291c	Proprietary
10,002	PROP_SONET	Partial

10,003	PROP_STS12c_and_VC4_4c	Obsolete In v3.0, a layerRate equivalent to PROP_STS12c_and_VC4_4c has been defined, i.e., LR_STS12c_and_VC4_4c. So, PROP_STS12c_and_VC4_4c will no longer be used.

2.33.4 SNC Type Transitions Related to the ModifySNC Operation

The following table indicates the EMS support of SNC Type transitions that can be invoked through the modifySNC() operation. Although the EMS can provide support for transitions among all SNC Types, the MTNM interface only defines the transitions between ST_SIMPLE and ST_ADD_DROP_A or ST_ADD_DROP_Z to be mandatory, when modifySNC() is supported. Note that ST_ADD_DROP_Z bi-directional is new for Phase 3.

The modification of the SNC's Static Protection Level does not impact the SNC type. The EMS may support the modification of an unprotected SNC to being a partially, fully or highly protected SNC while maintaining the same SNC type.

In order to complete the following table, the EMS supplier should do the following:

1. Indicate the supported transitions by placing the word "supported" in the appropriate cells (this should, at least, include the mandatory transitions). The other cells should be blank.

2. For each supported transition, indicate the GOI.

3. For ST_EXPLICIT to ST_EXPLICIT or other SNC type to/from ST_EXPLICIT, the GOI needs to be described for different ST_EXPLCIT traffic flows (e.g. for ST_EXPLICIT to ST_SIMPLE the SNC modification is supported as Hitless whenever only inactive protection legs are removed).

Table 2-82. EMS support for SNC Type Transitions

New SNC type / Previous SNC Type	**ST_ SIMPLE**	**ST_ ADD_DROP_A**	**ST_ ADD_DROP_Z**	**ST_ INTERCONNECT**	**ST_ DOUBLE_ INTERCONNECT**	**ST_ DOUBLE_ ADD_DROP**	**ST_ OPEN_ ADD_DROP**	**ST_EXPLICIT (Describe connectivity transitions that apply to each GOI category)**
ST_ SIMPLE	N/A	Mandatory GOI:	Mandatory GOI:	Optional GOI:	Optional GOI:	Optional GOI:	Optional GOI:	HITLESS: MINOR: MAJOR:
ST_ ADD_DROP_A	Mandatory GOI:	N/A	Optional GOI:	Optional GOI:	Optional GOI:	Optional GOI:	Optional GOI:	HITLESS: MINOR: MAJOR:
ST_ ADD_DROP_Z	Mandatory GOI:	Optional GOI:	N/A	Optional GOI:	Optional GOI:	Optional GOI:	Optional GOI:	HITLESS: MINOR: MAJOR:
ST_ INTERCONNECT	Optional GOI:	Optional GOI:	Optional GOI:	N/A	Optional GOI:	Optional GOI:	Optional GOI:	HITLESS: MINOR: MAJOR:
ST_ DOUBLE_ INTERCONNECT	Optional GOI:	Optional GOI:	Optional GOI:	Optional GOI:	N/A	Optional GOI:	Optional GOI:	HITLESS: MINOR: MAJOR:
ST_ DOUBLE_ ADD_DROP	Optional GOI:	Optional GOI:	Optional GOI:	Optional GOI:	Optional GOI:	N/A		HITLESS: MINOR: MAJOR:
ST_ OPEN_ ADD_DROP	Optional GOI:	Optional GOI:	Optional GOI:	Optional GOI:	Optional GOI:	Optional GOI:	N/A	HITLESS: MINOR: MAJOR:

ST_EXPLICIT	HITLESS: MINOR: MAJOR:	HITLESS: MINOR: MAJOR:	HITLESS: MINOR: MAJOR:	HITLESS: MINOR: MAJOR:	HITLESS: MINOR: MAJOR:	HITLESS: MINOR: MAJOR:	HITLESS: MINOR: MAJOR:	N/A

3 Non-Functional Interoperability Statements (NIS)

This section covers interoperability issues related to the non-functional aspects of the MTNM interface.

3.1 Iterator Implementation Issues

The EMS vendor should state how many iterators can be open at any one point in time. The MTNM team has agreed that 10 is a reasonable number. Any timer behavior associated with an iterator should be stated, e.g., the EMS automatically deletes an iterator if it has not been used in the last 5 minutes.

The EMS may have trouble fulfilling a getLength request when the data is scattered across a number of systems and its volume is not readily measurable and/or where the volume of data available for transfer may change after the request and during the iteration process. Under these circumstances it is acceptable for the EMS to throw a CAPACITY_EXCEEDED exception. In order to assist the NMS, the EMS vendor shall state clearly in which cases the getLength operation may throw the CAPACITY_EXCEEDED exception.

3.2 Timing Issues

None of the MTNM documents address timing. An EMS and NMS could follow all the MTNM standards as well as an agreed to interoperability statement (such as the one suggested in Section 2 of this document) and still fail to interoperate because of timing issues.

For example, if the NMS uses are large value of how_many in the next_n operation, the EMS may take too long to prepare the result, leading to a CORBA message timeout. So, the NMS and EMS need to agree on a reasonable range of values for how_many. Another example of a potential timeout would be SNC establishment.

In general, the maximum time an NMS is designed to wait for a response to a particular type operation request should be greater than expected maximum time the EMS takes to fulfill the operation request. It is suggested that the EMS vendor provide the following table as guidance:

Table 3-1. Operations Response Times and Timeouts

Operation Type	Maximum Expected EMS Response Time	Suggested Timeout Setting for NMS	Comments
Iterators (if various iterators have different characteristics, then multiple entries in this table are needed)			A Maximum Value for how_many should be provided
Data Retrieval of a Data Structure for a Second-Level Object			
Other operations can be listed on a case-by-case basis (createAndActivate), or groups of operations that			

have similar response time characteristics can be grouped into a single entry in this table (the group of SNC establishment operations)			

A similar table is needed for each notification type. There may be delays in delivering a notification. Once a notification has reached a particular age without being delivered to the NMS, it may make no sense for the Notification service to deliver the notification. The parties representing the NMS and EMS need to agree on a timeout period for each notification type.

The following table is needed to specify the heartbeat time constraints for the session supervision from NMS to EMS, and vice versa. It should be noted that Table refers to pings that are sent from the NMS to the EMS and vice versa, in order to make sure the connection between the NMS and EMS is still up. Table , on the other hand, refers to NT_HEARTBEAT notifications sent from the notification service to the NMS (this allows the NMS to know that its connection to the notification service is up.

Table 3-2 Heartbeat Time Constraints

Supervising Partner	Time in seconds between two Heartbeat Messages	Number of consecutively failed Heartbeats after which the session is closed	Comments
NMS			The NMS may use requests/responses for supervision instead of heartbeats.
EMS			

In Version 3.0, the EMS may send heartbeat notifications (NT_HEARTBEAT) when it has not sent any notifications for a period of time. The EMS supplier should indicate how often heartbeat notifications are sent (in the absence of other notifications). The EMS supplier should also indicate the number of consecutively missed heartbeat notifications after which the NMS should consider its association with the notification service has been lost.

Table 3-3 Heartbeat Notification Time Constraints

Time in seconds between two heartbeat notifications	Number of consecutively missed heartbeat notifications after which the NMS should consider its association with the notification service has been lost	Comments

4 Guidelines

This section covers guidelines concerning usage of the MTNM specifications. The intent is to describe, in further detail, the more complicated and/or less constrained aspects of the MTNM specifications. For example, the MTNM specifications state that subnetwork configuration is the responsibility of the EMS. Without further information, the NMS vendor needs to design their NMS to anticipate any of a number of possible subnetwork configuration schemes employed by various EMS vendors.

4.1 Implementation-specific Use Cases

The MTNM Business Agreement document provides use cases for many management activities, e.g., *NMS Provisions Alarm Monitoring Off*, *The NMS Creates a Subnetwork Connection*, and *Retrieval of Protection Group by the NMS*. These use cases are not prescriptive and allow for many options. For a particular implementation of the NMS-EMS interface, it is recommended that the use cases be specialized and detailed in order to ensure greater interoperability between the NMS and EMS. The use cases in the MTNM BA could server as a basis for any product-specific use cases.

4.2 EMS Subnetwork Configuration

In the MTNM model, the EMS decides how to package MEs into subnetworks. An understanding of how an EMS packages MEs into subnetwork will facilitate inventory discovery on the part of an NMS. Some examples concerning the packaging of the MEs within rings are as follows:

- The EMS packages each ring (SONET, SDH, or DWDM) into a separate subnetwork. Open rings are packaged into separate subnetworks.
    - In this case, the NMS will need to configure ring interconnection.
- The EMS packages interconnected rings (having protected interconnection) into separate subnetworks (this could lead large mesh subnetworks), and all other rings into individual subnetworks. Open rings are packaged into separate subnetworks.
    - In this case, the EMS (or the EMS GUI users) handles most of the configuration concerning ring interconnection.

In the first case above, each ring (or open ring) is in a separate subnetwork. For example, consider the case where a DSC is used to interconnect rings (say ring 1 and ring 2). The DSC is managed by one EMS and the rest of each ring is managed by another EMS. In this case, the NMS needs to setup three SNCs to get an end-to-end connection between a CTP in ring 1 and a CTP in ring 2. The SNCs in ring 1 and 2 would be of type ST_SIMPLE and the SNC in the DSC would be of type ST_DOUBLE_INTERCONNECT.

In the second case, the NMS only needs to request one SNC. In this case, the SNC type would be ST_DOUBLE_INTERCONNECT. This case is easier for the NMS since it does not need to know much about the internal structure of the mesh subnetwork (only needs to know that the mesh support DRI).

In the event subnetwork configuration is done via the EMS GUI, the EMS vendor may simply provide guidelines, suggestions and constraints concerning the configuration of subnetworks. In the event subnetwork configuration is performed via fixed logic in the EMS, the EMS vendor would an explanation of how the EMS decides on subnetwork packaging.

4.3 Usage of the Various Resource Names

The MTNM interface allows several identifiers to be assigned to a managed resource, i.e., name, userLabel, and nativeEMSName.

- The ***name*** is assigned by the EMS and it is the unique identifier of resource to be used across the NMS-EMS interface. The NMS has no control over a resource's name.
- The ***userLabel*** is typically assigned by the NMS via the setUserLabel operation or a create operation for resources that can be created by an NMS request, e.g., SNCs. If a resource is created by the EMS (actually the interface object representing a resource), then the EMS sets the userLabel to the nativeEMSName.
    - Further, the createSNC, createAndActivateSNC, checkValidSNC, and setUserLabel operations accept a parameter that allows the NMS to request that the EMS check the supplied userLabel for uniqueness. If the supplied userLabel is not unique, the request is rejected.
- The ***nativeEMSName*** was originally intended to be the name of a resource as shown on an EMS display. This usage is not mandatory and other uses are possible. The nativeEMSName can be set by the EMS (typically through the EMS GUI) or it can be set by the NMS (if allowed by the EMS vendor) using the setNativeEMSName operation.

As an example use of the three resource names, consider the end-to-end connection shown in Figure 1-1 (as represented by the solid horizontal line). In this example, SNC AbcInc_3837 crosses three subnetworks, two of which are in one EMS domain (i.e., Subnetworks A and B are in EMS Domain 1) and the other subnetwork (i.e., C) is in another EMS domain (i.e., Domain 2). The EMS uses the Name attribute to uniquely identify an SNC within its domain (this is always true for EMSs that follow the MTNM interface specifications). The network provider uses the NativeEMSName to store its identifier for an SNC. The NativeEMSName can be assigned via the EMS GUI or through the setNativeEMSName operations. The service provider uses the userLabel to associate the three SNCs that comprise the end-to-end connection. In this case, the NMS does not want to force uniqueness of the userLabel. Of course, the NMS could use its own internal mechanism to associate connections that cross several subnetworks. The advantages of using the userLabel in this case are as follows:

- If the NMS fails, the EMS user can still determine the association among related SNCs in different subnetworks (the EMS GUI would need to have appropriate search functions). If the connection crosses multiple EMS domains, correlation becomes more complicated and is likely to require more analysis on the part of the EMS user.
- If the NMS fails, the EMS GUI could be used to create connections crossing multiple subnetworks. After it has recovered, the NMS could discover the SNC associations by looking at the userLabels of the newly created SNCs.

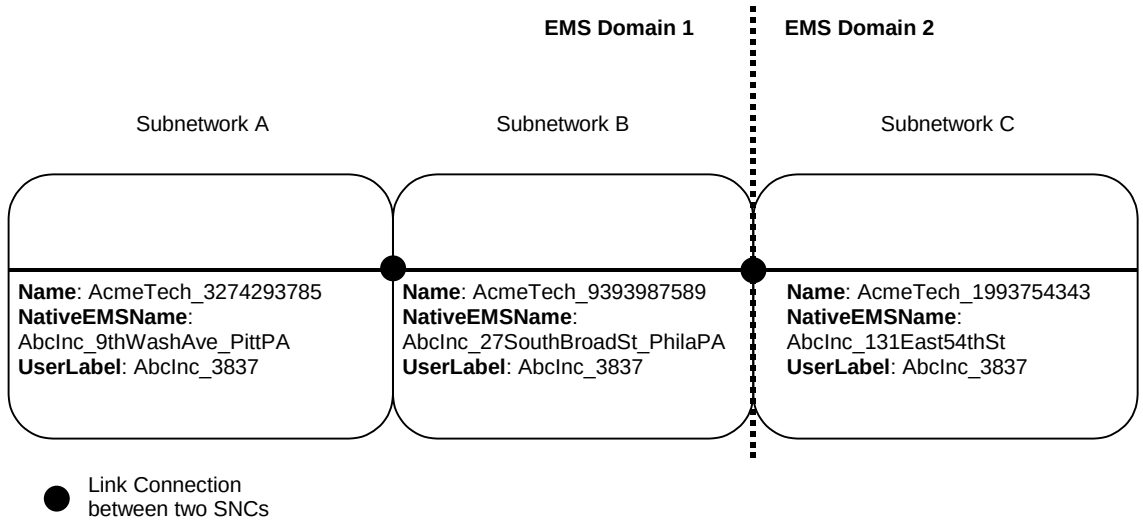

Figure 1-1. SNC Naming Example 1: Non-unique UserLabels

As a second example, consider the end-to-end connection shown in Figure 1-2. The end-to-end connection and component SNCs are the same as in the previous example, only the naming scheme is different. In this approach, the service provider wants the EMS to ensure that unique userLabels are used for each SNC within an EMS's management domain. If the NMS attempts to setup an SNC using a userLabel that has already been assigned, the EMS will reject the request. This prevents the mistaken re-use of the SNC names selected by the service provider (this could happen during manual entry of SNC names via the NMS GUI, for example). Related SNCs can still be associated. However, in this case, the NativeEMSName is used to associate related SNCs.

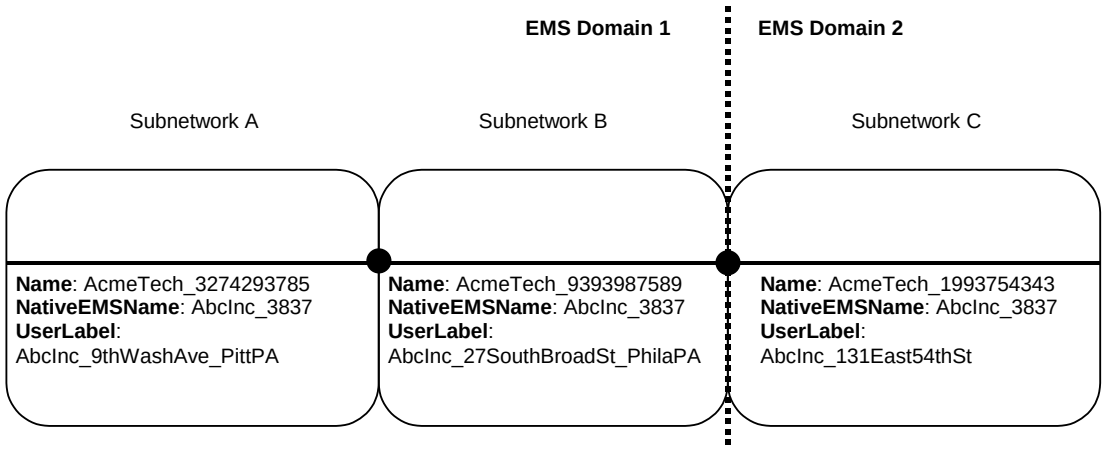

Figure 1-2. SNC Naming Example 2: Unique UserLabels

In another approach, each EMS domain consists of a single subnetwork. The userLabel and nativeEMSName of an SNC are same. Further, the component SNCs of an end-to-end connection have the same userLabel and nativeEMSName, as shown in Figure 1-3. This example has the same advantages as Example 1 and also allows the NMS to request userLabel uniqueness from the EMS as in Example 2. The main disadvantage is that the EMS domain cannot be subdivided into multiple subnetworks.

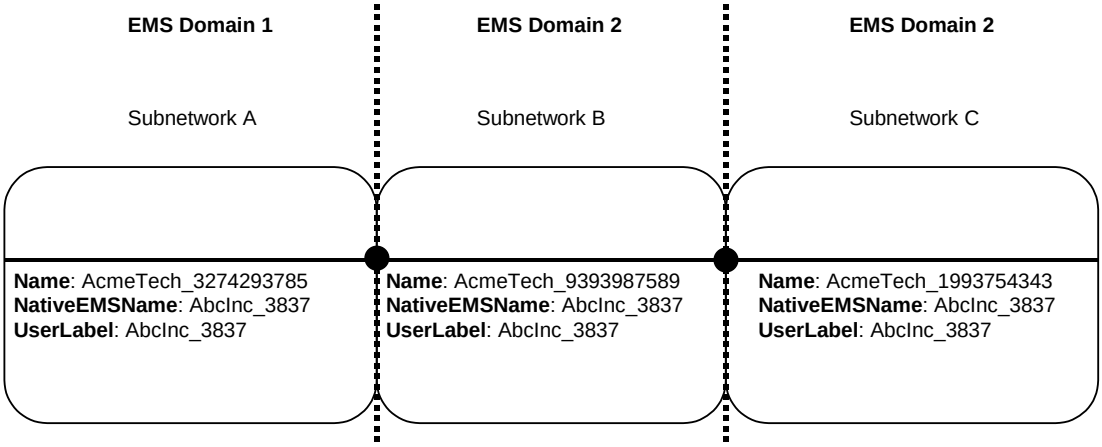

Figure 1-3. SNC Naming Example 3: Same name for userLabel and nativeEMSName

4.4 SNC State Representations – Modes

The member companies comprising the MTNM team have varying opinions concerning the SNC state model, and agreement could not be reached on a single state representation model. Consequently, it was agreed to allow four different SNC state modes:

Table 4-1. SNC State Representation Modes

Mode	Support for Pending state, and allow SNC conflicts on creation, i.e., shared Cross Connections (CCs)	Allow for sharing of Active CCs
A		
B		
C		
D		

The various modes in the above table are explained in the following subsections.

4.4.1 Mode A

In Mode A, the goal is for the EMS to represent only the current network configuration, to limit as much as possible sharing of resources (i.e., cross connections) among SNCs, and to attempt to have a one-to-one correspondence between the network configurations and the SNC configurations. This mode does not support the pending state. An SNC that does not have any non-shared, active CCs in the network is considered non-existent. Consequently, when the last non-shared cross-connect of an SNC is deactivated in the network, the SNC is deleted by the EMS.

In Mode A, a CC, active or not, can only belong to at most one SNC. In the case of non-singleton subnetworks, an exception to the non-sharing rule is made for SNCs of a broadcast system, i.e., a multipoint connection. The legs of a multipoint connection are represented as individual SNCs. The legs of a broadcast system may share CCs and always share the same source CTP.

Mode A is best for EMSs that only support singleton subnetworks (i.e., subnetwork consisting of a single managed element) and that do not keep a database of pending SNCs. Mode A might be popular with vendors having lightweight switch-bound EMSs.

4.4.2 Mode B

In Mode B, the goal is for the EMS to represent the current network configuration as well as potential “future” SNCs that have been prepared by the NMS, but not yet activated. The Pending state can also be used in situations where SNC share CCs at different times. For example, SNC1 is used by customer A from 8am to 8pm every day. SNC2 shares many CCs with SNC1 but is only used by Customer B from 9pm to 7am every day. When SNC1 is in the Active state, SNC2 is put in the Pending state, and vice versa.

An SNC’s entry and exit to and from the Pending state is controlled solely by the NMS. Neither the EMS or craft intervention can put an SNC into (or remove an SNC from) the Pending state.

4.4.3 Mode C

In Mode C, the goal is for the EMS to represent only the current network configuration (as was the case with Mode A). Contrary to Mode A, it does not limit sharing of CCs among SNCs, and does not attempt to have a one-to-one correspondence between the network configurations and the SNC configurations.

Mode C is best for EMSs that support non-singleton subnetworks and that do not keep a database of pending SNCs. This mode has an advantage over Mode A, because it allows SNC reorganizations without traffic interruption (only useful in non-singleton subnetworks). For example, if the EMS currently has two “consecutive” SNCs that the NMS wants to merge into one “larger” SNC, this can be done without interrupting traffic by creating and activating the larger SNC, then deactivating and deleting the two consecutive SNCs.

4.4.4 Mode D

Mode D is basically a combination of Modes B and C. This Mode is favored by vendors (or service providers) that have a feature rich NMS that can take advantage of the Pending state (related to scheduling features) and the sharing of Active CCs.

4.5 Example Probable Cause Template

The following example illustrates the use of the probable cause template defined in Section 2.1.6.1. The example is based on actual SDH and WDM equipment that are managed according to ITU principles. The "Transmission" alarms (from TPs) are from a submarine equipment type that uses Forward Error Correction, or FEC: a method based on Reed-Solomon encoding that enables errors to be not only detected, but corrected.

The following conventions are used in Table .

Identifiers

The “official” MTNM identifiers are used, such as “OT_PHYSICAL_TERMINATION_POINT” instead of abbreviations, e.g., “PTP”. The latter would make for a more legible table of course, but a set of abbreviations is not currently available. The worst problem is with the layer rates: if one just gives the value (e.g., "22") it will not be understood. Unfortunately, the complete strings are fairly long, e.g., "22 = LR_Section_OC48_STS48_and_RS_STM16".

Different kinds of Probable Causes:

The example shows the kind of relationships that can exist between (MTNM) Probable Cause, Native Probable Cause, and Probable Cause Qualifier:

- One (MTNM) Probable Cause value may correspond to several Native Probable Cause values
- One Native Probable Cause Value may correspond to several Probable Cause Qualifier values - or to none at all.

Additional Text:

The strings shown in Probable Cause Qualifier (which is not displayed to NMS operator) can also used as Additional Text (which is displayed). This is not a general rule, but it is allowed by the MTNM model. The example itself does not include Additional Text.

Additional Info and ITU alarm coding:

The X.733 Event Type is used an example of Additional Info.

- The rules for doing this are set out in the "Notification Service Usage" supporting document to the IDL.
- Note that this is optional both for EMS to produce and for NMS to take into account. This only makes sense if both EMS and NMS actually use X.733 alarm format internally.
- The table gives examples of the four X.733 Event Type values applicable for MTNM Alarms (Communications Alarm, Equipment Alarm, Environmental Alarm and Processing Error Alarm). The last Event Type value is Quality Of Service Alarm, which is the same thing in practice as a TCA, even if they are defined differently.

One difference between MTNM and ITU alarm management principles is in "software" or "processing error" faults:

- ITU "Processing Error Alarm" Event Type corresponds to software problems in general, wherever they occur.
- The assumption here is "software" type problems in NEs are given a (MTNM) Probable Cause value "EQPT". Such problems can occur, since NEs perform information processing. However, the same kind of error in the EMS is given a (MTNM) Probable Cause value "EMS".

Table 4-2 Example Probable Cause Template

Prob-able Cause	Native Probable Cause	Associated Object Type	Layer Rates	Probable Cause Qualifiers	Per-ceived Severity	Service Affecting	Additional Informa-tion (name)	Additional Informa-tion (value)
BER_SD	Degraded Signal	OT_PHYSICAL_TERMINATION_POINT	76 = LR_DSR_OC48_and_STM16	B1 LER detected	PS_MINOR	SA_NON_SERVICE_AFFECTING	"X.721::EventType"	"communicationsAlarm"
BER_SD	Degraded Signal	OT_CONNECTION_TERMINATION_POINT	76 = LR_DSR_OC48_and_STM16	FEC error low alarms (LBER)	PS_MINOR	SA_NON_SERVICE_AFFECTING	"X.721::EventType"	"communicationsAlarm"

BER_SD	Fec Uncorrected Blocks	OT_CONNECTION_TERMINATION_POINT	76 = LR_DSR_OC48_and_STM16	FEC Uncorrected Blocks	PS_MINOR	SA_NON_SERVICE_AFFECTING	"X.721::EventType"	"communicationsAlarm"
BER_SD	Optical Signal Degraded	OT_CONNECTION_TERMINATION_POINT	40 = LR_Optical_Channel	Optical Signal Degraded	PS_MINOR	SA_NON_SERVICE_AFFECTING	"X.721::EventType"	"communicationsAlarm"
EQPT	Replaceable Unit Missing	OT_EQUIPMENT	1 = LR_Not_Applicable	Replaceable Unit Missing	PS_MAJOR	SA_SERVICE_AFFECTING	"X.721::EventType"	"equipmentAlarm"
EQPT	Replaceable Unit Type Mismatch	OT_EQUIPMENT	1 = LR_Not_Applicable	Replaceable Unit Type Mismatch	PS_MAJOR	SA_SERVICE_AFFECTING	"X.721::EventType"	"equipmentAlarm"
EQPT	Internal Communication Problem	OT_EQUIPMENT	1 = LR_Not_Applicable	Internal Communication Problem	PS_MAJOR	SA_SERVICE_AFFECTING	"X.721::EventType"	"equipmentAlarm"
EQPT	Unit Failed	OT_EQUIPMENT	1 = LR_Not_Applicable	unitFail	PS_MAJOR	SA_SERVICE_AFFECTING	"X.721::EventType"	"equipmentAlarm"
EQPT	Unit Failed	OT_EQUIPMENT	1 = LR_Not_Applicable	converterFail	PS_MAJOR	SA_SERVICE_AFFECTING	"X.721::EventType"	"equipmentAlarm"
EQPT	Unit Failed	OT_EQUIPMENT	1 = LR_Not_Applicable	overCurrent	PS_MAJOR	SA_SERVICE_AFFECTING	"X.721::EventType"	"equipmentAlarm"
ENV	Air Conditioning Failure	OT_MANAGED_ELEMENT	1 = LR_Not_Applicable		PS_MINOR	SA_NON_SERVICE_AFFECTING	"X.721::EventType"	"environmentalAlarm"
EMS	Database Inconsistency	OT_EMS	1 = LR_Not_Applicable		PS_WARNING	SA_NON_SERVICE_AFFECTING	"X.721::EventType"	"processingErrorAlarm"
EQPT	Reinitialized	OT_MANAGED_ELEMENT	1 = LR_Not_Applicable		PS_WARNING	SA_NON_SERVICE_AFFECTING	"X.721::EventType"	"processingErrorAlarm"

4.6 Mapping of MTNM Services States to Other Models

The Telcordia state model in GR-1093-CORE and the ITU-T state model defined in ITU-T Recommendation X.731 are commonly used in network management interfaces. The MTNM team has defined a state that is different from both the Telcordia and ITU-T model. Mappings between the MTNM state model, and the Telcordia and ITU-T state models are provided in the following two subsections.

4.6.1 Mapping to Telcordia State Model (GR-1093-CORE)

Table 4-3 MTNM to Telcordia State Model Mapping

Telcordia State	MTNM State
IS-NR (In service-normal)	IN_SERVICE
IS-ANR (In service-abnormal)	
IS-RST (In service-restricted)	
IS-ANRST (In service-abnormal & restricted)	
OOS-AU (Out of Service-autonomous - meaning failed, equivalent of disabled)	OUT_OF-SERVICE

OOS-MA (Out of Service-management - meaning administratively placed OOS, equivalent of locked)	OUT_OF_SERVICE_BY_MAINTENANCE
OOS-AUMA (Out of Service-autonomous & management)	OUT_OF_SERVICE_BY_MAINTENANCE
OOS-AURST (Out of Service-autonomous & restricted)	OUT_OF-SERVICE
OOS-MAANR (Out of Service-management & abnormal)	OUT_OF_SERVICE_BY_MAINTENANCE

4.6.2 Mapping to ITU-T States

Table 4-4 ITU-T to MTNM State Mapping

ITU Operational State	ITU Administrative State	MTNM Service State
enabled	unlocked	IN_SERVICE
	locked	OUT_OF_SERVICE_BY_MAINTENANCE
	shutting down	SERV_NA
disabled	unlocked	OUT_OF_SERVICE
	locked	OUT_OF_SERVICE_BY_MAINTENANCE

4.7 Usage of Network Access Domains (NADs)

A given NAD represents a domain to which a set of transmission network resources (e.g., PTPs, CTPs, SNCs) can be assigned and a given Functional Access Domain (FAD) or set of FADs). The FAD or FADs determines the functions which can be applied to the NAD. A network resource can be assigned to only one NAD, or be unassigned (or free). In terms of the MTNM interface, the NMS may assign a NAD to a set of resources. The assignment of the FADs is outside the scope of the interface. This section provides an explanation of how the NAD/FAD concept can be used.

The network administrator can

a) assign network resources to NADs (i.e. specify owners of various resources). This is done over the NMS-EMS interface by setting the NAD parameter (a v3.0 parameter within additionalInfo) for a set of resources, e.g., TPs and SNCs.

b) assign users (i.e., operators at NMS and EMS user interface) functional profiles (FAD) thereby allowing the users to perform specific functions on designated resources (as determined by the NAD previously assigned to the resource). One or more FADs can be associated to a given NAD. Thus according to type of user, different types of operations are permitted on the specific NAD. Again, management of the FADs is outside the scope of the MTNM interface.

A given resource with its NetworkAccessDomain parameter set to the empty string is intended to be a “free” resource, i.e., any users can request only MTNM operation on the resource. A given FAD can be defined to permit or refuse access to “free” resources.

Example use of NADs and FADs:

- some resources are marked for the NetworkAccessDomain = ”owner23”
- user44 is defined by the network administrator at the EMS to be associated with FAD = “readonly” and to be given access to the “owner23” NAD. So, on the EMS side, “user44” is granted readonly access to the TPs and SNCs belonging to NAD = “owner23”.

As another example, it is noted that the NAD/FAD mechanism can be used to define VPNs. In fact, a given VPN could be seen as NAD, where the VPN operator can create/activate, and deactivate/delete SNCs using only the routing resources belonging to such NAD/VPN (e.g., the NMS could include the NAD identifier in the additionalCreationInfo parameter of the createAndActivateSNC operation). At the EMS level, an SNC can be

routed only on resources (CTPs) which NetworkAccessDomain value is equal to such NAD/VPN (or is “free”, if the FAD allows it).

4.8 Root Cause Alarm Indication

The Root Cause Alarm Indication (RCAI) feature allows the EMS to indicate a distinction between raw (i.e., un-correlated) alarms, and root cause alarm indications.

The NMS can use the filtering capability of the Notification Service in conjunction with the RCAI field to

- receive only raw alarms, i.e., alarms with the root cause alarm field set to FALSE,
- receive only RCAIs, i.e., alarms with the root cause alarm field set to TRUE, or
- receive both raw alarms and RCAIs.

The following suggestions are noted concerning usage of the RCAI:

1. If a single alarm persists but does not get correlated with any other alarm, two notifications are expected (the raw alarm and the RCAI associated with the uncorrelated raw alarm).
2. There is no field in the RCAI or the raw alarms that points to the other correlated alarms. The idea is to keep the interface as simple as possible and still allow for the reporting of RCAIs.
3. The raw alarms result in notifications immediately. The RCAI is generated after the correlation window closes (the length of the correlation window is an implementation issue for the EMS/NE vendor).
4. RCAIs usually clear after all related raw alarms clear and stay cleared for a persistence interval.

The above points are just suggestions concerning how the RCAI feature could work. Since these suggestions relate to EMS implementation, they are not included in the requirements of the Business Agreement document.

ITU-T Recommendation M.2140, Transport Network Event Correlation, provides additional detail concerning how RCAI might be supported.

Some common questions and answers concerning the proposed model:

Doesn’t a notification need to be sent when a raw alarm latter gets declared as a root cause?

> In the proposed model, raw alarms do not "change" to root causes. If the root cause of several alarms is identified to be the same as a previous raw alarm, a new root cause alarm indication could be sent to the NMS. Both the original raw alarm and the root cause would be treated as separate alarms in terms of notifications sent over the EMS-NMS interface. Each would be cleared separately.

What notification is sent when a previously declared root cause is now related to another root cause?

> If the EMS decides that it has found a better root cause, the EMS could clear the original root cause and declare a new root cause.

4.9 FTP Examples

In order to represent managed element with several layers of flexibility, the MTNM team has added support for Floating TPs in v3.0.

An example managed element with several layers of flexibility is shown in Figure 1-4 and Figure 1-5. (Figure 1-4 is the usual "elevation" figure where layers are shown stacked vertically one above the other - server layers are always beneath client layers. Figure 1-5 is a "plane" figure where the layout is "horizontal", as if all the lines in Figure 1-4 were twisted to make them parallel. This allows multiplexing to be shown, and is easier to relate to hardware. [Note that Figure 1-5 shows four VC4 ports whereas Figure 1-4 shows only VC4 port.] The figures use the graphical conventions of ITU-T documents such as G.805. Figure 1-4 shows, from right to left:

- An SDH port (layers are modeled according to TMF MTNM, to facilitate comparison with the following figures. These layers are: Physical, Optical Section, Digital Signal Rate, Regeneration Section, Multiplex Section);
- A High Order Matrix with a flexible cross-connection at AU4 / VC4 (SONET: STS3c) level;
- The VC4 TTP, comprising termination function and adaptation function, with mapping to 3 TU3's;
- A Low Order Matrix with a flexible cross-connection at TU3 / VC3 (SONET: STS1) level;
- A fixed connection at 45Mbit (DS3) tributary level;
- A PDH port (with layers: Phy, DSR, 45M).

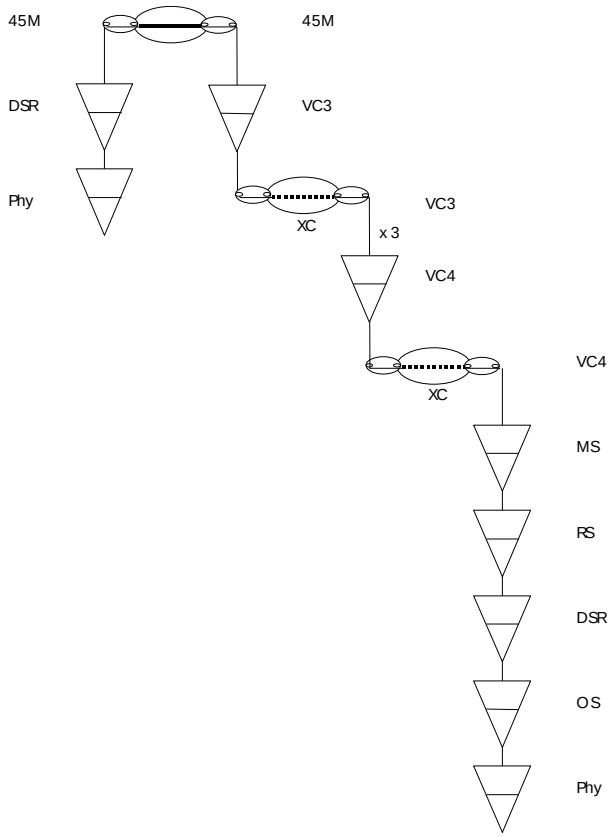

Figure 1-4. G. 805 modeling example ("elevation" view)

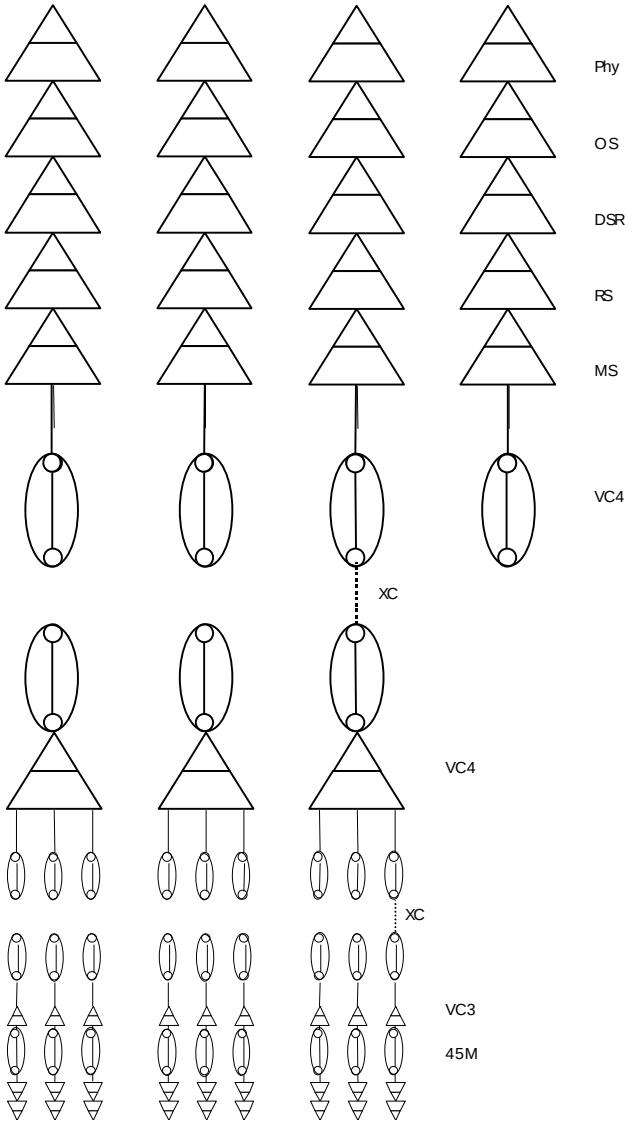

Figure 1-5. G.805 modeling example ("plan" view)

Figure 1-6 (elevation), Figure 1-7 (plan, generic figure), and Figure 1-8 (plan, example of cross-connect) show how FTPs can be used to model managed elements with several layers of flexibility. PTPs are blue, CTPs are green, cross-connections are yellow, and containment relationships are shown by red arrows.

- This is a straightforward mapping from the G.805 model: two cross-connections across physically different matrices are modeled as two different cross-connect objects.
- As can be seen, we find ourselves with a "floating CTP" in the middle that cannot conveniently be attached to the port on either side. This is a VC4 CTP that contains cross-connectable (non-terminated) Low Order CTPs, in our example VC3 CTPs.

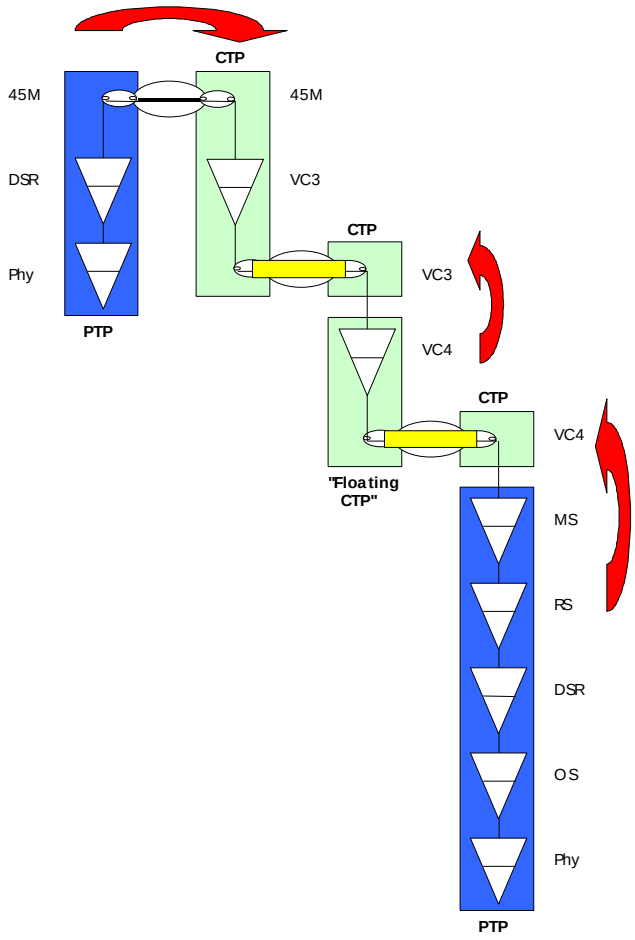

Figure 1-6. Modeling with Floating CTPs (elevation)

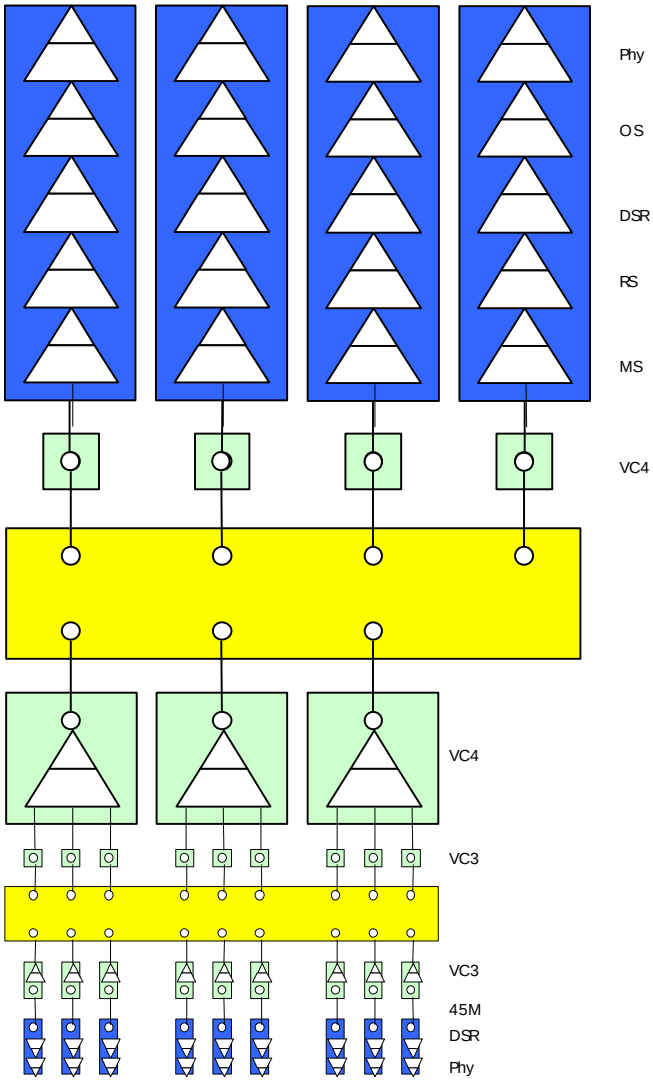

Figure 1-7. Modeling with Floating CTPs (plan, generic figure)

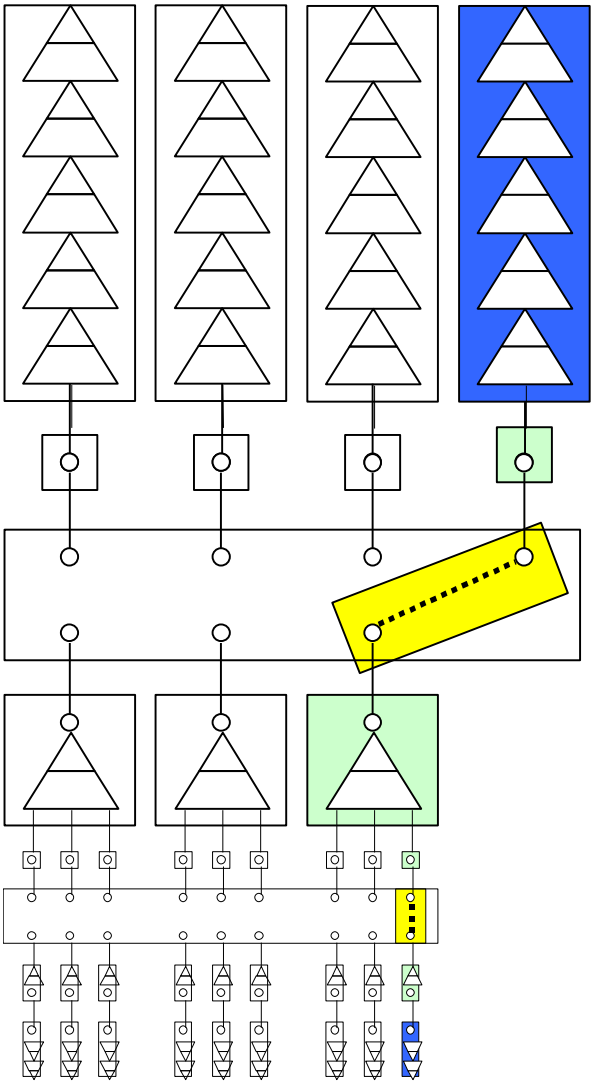

Figure 1-8. Modeling with Floating CTPs (plan): Example of cross-connects

Another important application of the FTP concept is the end-to end SNCP protection of a server SNC. We can define a server SNC as a SNC which does not support service, but does provide infrastructure for client SNCs. In the SDH hierarchy, this is the case of a VC4 SNC supporting lower order (TU3/12) SNCs. The end points of the VC4 SNC are terminated and mapped. For example, a VC12 SNC can be supported by a sequence (1 or more) of VC4 SNCs.

Figure 1-9 depicts SNCP protection of a server SNC (VC4 layer), where there is no use of FTPs at aEnd or zEnd of the SNC. The protection is partial, i.e., the signals flowing through the topological links at each end of the SNC are not protected.

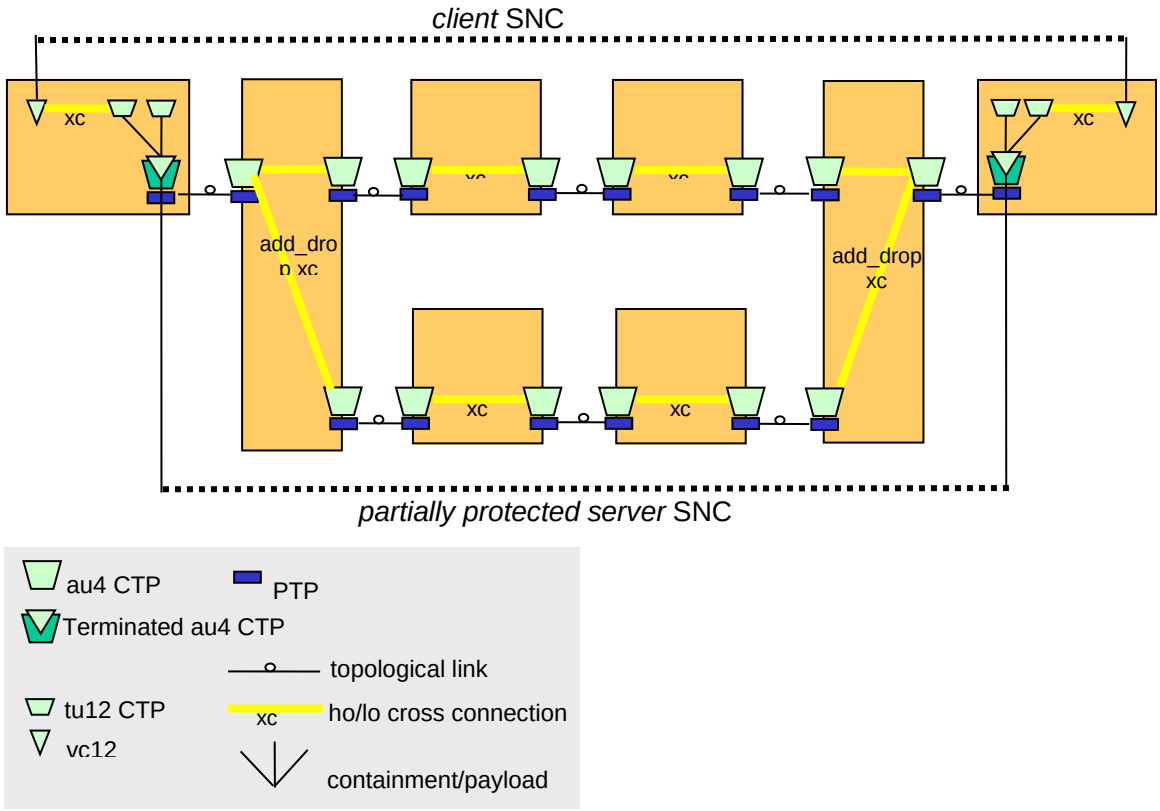

Figure 1-9. Partial SNC Protection without FTPs

Figure 1-10 shows the case where FTPs are used to provide protection at each end of the trail. In particular, the MEs at each end have 2 unreliable au4 CTPs which protect the (reliable) VC4 CTP. The trail is terminated by an FTP at each end.

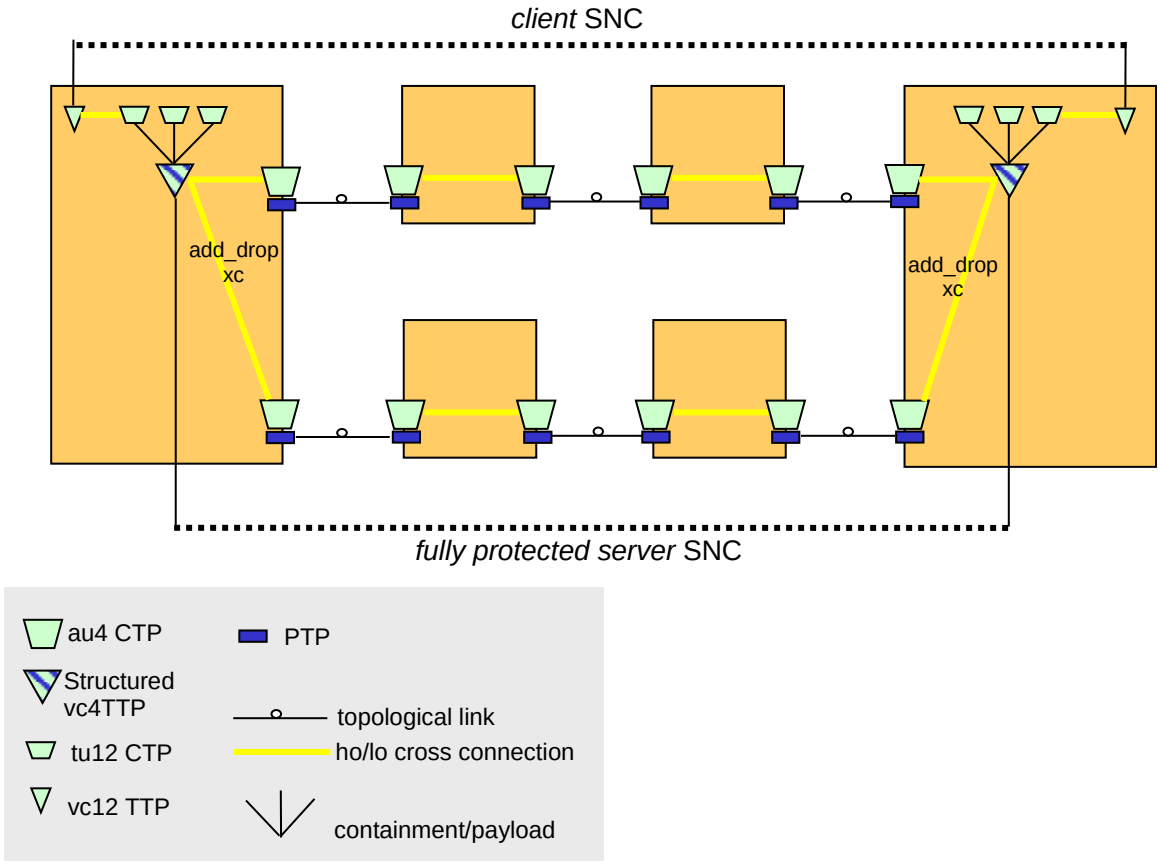

Figure 1-10. Protected Trail using FTPs

Appendix A:Terms and Abbreviations Used within this Document

Terminology

Please refer to
TMF 513 - Multi Technology Network Management (MTNM) Business Agreement,
and
TMF 608 - Multi Technology Network Management (MTNM) Information Agreement.

Abbreviations and Acronyms

Please refer to
TMF 513 - Multi Technology Network Management (MTNM) Business Agreement,
and
TMF 608 - Multi Technology Network Management (MTNM) Information Agreement.

Appendix B:References

References

Reference	Description	Brief Use Summary
Project Charter	MTNM Project Charter v3.0	Used for objectives and scope and for the project benefits.
TMF 513	MTNM Business Agreement. The basic requirements and use cases.	Used for the requirements and use cases references.
TMF 608	MTNM Information Agreement TheUML specification.	Used for information model.
TMF 814	MTNM Solution Set The IDL specification	

Specific Analysis or Comments on a Source or Use

None

IPR Releases and Patent Disclosures

None

Administrative Appendix

This Appendix provides additional background material about the TeleManagement Forum and this document.

About this document

The *Multi-Technology Network Management (MTNM) Implementation Statement Templates and Guidelines* specification is being issued as Member Evaluation Version 3.1. When released, it will be considered valid until further notice by the TeleManagement Forum. At which time the TeleManagement Forum expects to update it to reflect comments from implementation experience, as well as to reflect additional member comment. This version 3.1 MTNM Solution Set will supersede the TMF814 version 3.0 in its entirety.

The purpose of an Evaluation Version is to encourage input based on experience of members and the public as they begin to use the document. Following the Evaluation Period, documents that are seen to deliver value are candidates for formal approval by the TM Forum. All documents approved by the TM Forum (as well as those previously approved by the NMF) undergo a formal review and approval process.

This document will continue under formal change control. Supporting work will be issued as revisions to this document. A document of this type is a "living document," capture and communicating current knowledge and practices. Further inputs will be made because of detailed work ongoing in the TM Forum and the industry.

This document should be read in conjunction with the following documents:

- TMF513 – MTNM Business Agreement
- TMF608 – MTNM Information Agreement
- TMF814 – MTNM Solution Set
- SD1 series – MTNM Supporting documents

Document Life Cycle

Use and Extension of a TM Forum Implementation Statement Template and Guidelines

This document entails an implementation statement template and set of interoperability guidelines for an NML-EML interface used to enable management of SONET, SDH, WDM and ATM networks. Given that the Multi-Technology Network Management (MTNM) interface defined in TMF814 can be packaged and realized in different ways, this document is intended to provide guidance to those implementing the MTNM interface, with the goal of ensuring interoperability among various vendor implementations.

The TMF814A Implementation Statement Template and Guidelines should be used in conjunction with the TMF513, TMF608, the TMF814 and the SD1 series of supporting documents.

It is expected that this document will be used:

- To facilitate interface agreements between Service Providers and vendors
- As input to a service Provider's Request for Information / Request for Proposal (RFx)
- To allow operational support managers to compare their interface requirements with those identified by the project
- As input for companies developing COTS products.
- As a source of requirements for other bodies working in this area.

Document History

Revision History

ISSUE	DATE	COMMENTS
0.0	18 October 2000	Initial contribution (entitled *Proposed Product Profiles for Phase II*) submitted to the November MTNM meeting in Lisle, IL by Telcordia and SBC.
1.5	August 2001	Formatted for Member Evaluation Release
2.0	October 2001	Formatted for TM Forum and Approved after Corporate Vote
2.1	August 2002	Issued v2.1
3.0	August 2003	Submitted v3.0 to TM Forum for subsequently release to the TM Forum membership for evaluation and comment.
3.1	May 2006	Contribution for Ethernet part of the MTNM 3.5 model. The following new IDL modules have been introduced:: • flowDomain.idl, • flowDomainFragment.idl, • trafficConditioningProfile.idl. In addition, the following IDL modules have been modified: • equipment.idl (no change in this document) • managedElementManager.idl: change to IDL comments => Termination Point section: change to "Clarification Needed" column for ingressTrafficDescriptorName and egressTrafficDescriptorName attributes. • subnetworkConnection.idl (mainly Control Plane material, no change to this document) • transmissionDescriptor.idl: new operation modifyTransmissionDescriptor => Transmission Descriptor section: impact on data and interface tables. **Note**: This document "TMF814A-Ethernet Contribution 20060430.doc" does not contain updates for Control Plane model.

3.1	End May 2006	Updates after bug fixes in Ethernet model: • FlowDomain data types: removed fdEdgeCPTPs, also known as assignedCPTPs, from FDCreateData. (The assigned CPTPs can be retrieved via getAllAssignedCPTPs.) • Alarm Severity Assignment Profile data types: attribute fixed is now notModifiable. • New module trafficConditioningProfile.
3.1	July 2006	Contribution for Control plane part of the MTNM 3.5 model. The following new IDL modules have been introduced: • Call SNC module (3 data types: Call, Call Create Data, Call Modify Data) • MLSNPP module (1 data type: MultiLayerSNPP) (1 interface: MLSNPPMgr) • MLSNPP Link module (1 data type: MultiLayerSNPPLink) (1 interface: MLSNPPLinkMgr) In addition, the following IDL modules have been modified: • MultiLayerSubnetwork module (new operations added to the Multi-layer Subnetwork Manager interface) • MultiLayerSubnetwork module (update Data type: SNCCreateData [Note: In version 3.0 SNCCreateData & SNCModifyData appear in the MultiLayerSubnetwork module of TMF 814A. But the IDL are defined in the SubnetworkConnectin module]) Why? Should this be corrected. • SubnetworkConnection module (update to data types: subnetworkConnection, Cross Connection, Route Create Data) • EmsMgr module (new operations added to the EMSMgr_I interface) Editorial update to Sections 2 up to 2.1.3
3.1	September 2006	Clean up open questions based on definitive answers from the group. Move the data types SNCCreateData & SNCModifyData back to the SubnetworkConnectin module section so that to be consistent with the IDL. Update front material
3.1	2 October 2006	Change all the operations of the “Multi-Layer Subnetwork Manager” in the FlowDomain Module from Mandatory to Optional. In MLSN module, additionalInfo, change the attribute “LayeredRoutingArea” to “LayeredRoutingAreaIdList”.
3.1	17 October 2006	Changes because of the change for set TNA on MLSNPP & MLSNPPLink • In MLSNPP module, delet tNAName and groupTNAName from data type MultilayerSNPP • In MLSNPPLink module MultiLayer SNPPLink data type, replace aEnd and zEnd with aMLRAName and zMLRAName.
3.1	08 January 2007	Update to align with the changes in the 30/12/2006 and 7/1/2007 IDL • For the Flow Domain Module – In FDCreateData: add name and transmissionParams – In FlowDomain Modify Data: add transmissionParams

		– In MatrixFlowDomain: for name, change Set by EMS to Set by E – In MFDCreateData: add name – In MatrixFlowDomain Modify Data: add transmissionParams – In FTP Create Data: delete networkAccessDomain – In Interface: change the heading of Table 2-29 from 'Multi-layer Subnetwork Manager" to "Flow Domain Manager"; change operation name "getAllTopologicalLinks" to "getAllTopologicalLinksOfFD"; delete the operations addFPsToFDFr, removeFPsFromFDFr, and validateTMDAssignmentToMFD • For the Call SNC Module – In Call Create Data: change callAdditionalInfo to additionalCreationInfo – In Call Modify Data: change callAdditionalInfo to additionalModificationInfo • For the Multi-Layer Subnetwork Module – Change the operation name "getAllSubordinateMLSNWithConnection" to "getAllSubordinateRAidsWithConnection – Add mandatory operation "getCallAndTopLevelConnections" – Delete operation "getMLSNPP" – Delete operation "getAllCallsAndSNCs" – Add optional operation "getAllCallsAndTopLevelConnectionsAndSNCs" – Add optional operation "getAllCallsAndTopLevelConnectionsAndSNCsWithME" • For the Traffic Conditioning Profile Module – In the TCProfile datatype, change the attribute name "default" to "defaultProfile" – In the interface, change the following operations from Mandatory to Optional: getAllTCProfiles, getTCProfile, and getTCProfileAssociatedTPs • For the Transmission Descriptor Module – In the TMD Create datatype, change the attribute name "additionalTPInfo" to "additionalObjectInfo" – In the TMD Modification datatype, change the attribute name "additionalTPInfo" to "additionalObjectInfo" – In the Transmission Descriptor Mgr interface: change the operations getAllTransmissionDescriptors, getAllTransmissionDescriptorNames, and getTransmissionDescriptor from mandatory to optional. – In the Transmission Descriptor Mgr interface, add new optional operations validateTMDAssignmentToObject and setTMDAssociation • Add new Flow Domain Fragment Module
3.1	February 4, 2007	• Align with 2007.01.28 IDL changes to add networkAccessDomain attribute to FTPCreateData_T and remove connectionState attribute from FTPCreateData_T. • Update front matter
3.1	February 28, 2007	• In MultiLayerSubnetwork module, – change the operation getCallAndTopLevelConnections from mandatory to optional

		• In Call SNC Module – Change linkDiversityViolationsList to linkDiversityViolations – Change nodeDiversityViolationsList to nodeDiversityViolations
3.1	March 8, 2007	Align with the changes in the IDL distributed on March 7, 2007 (Word Files_2007-03-07) • In Flow Domain Module FTP Create Data – Change IngressTrafficDescriptorName to ingressTransmissionDescriptorName – Change egressTrafficDescriptorName to egressTransmissionDescriptorName
3.1	March 9, 2007	Align with the changes in the IDL as agreed on March 8 (Main open issues_bz) • In Flow Domain Module FTP Create Data – Delete tpProtectionAssociation and edgePoint
3.2	April, 2014	Two new interface for OSS to control the sequence and time of collection data from for huge network • getHistoryAlarmDataByPull • getHistoryPMDataByPull
3.3	Nov, 2014	Enhance support for L3VPN object, VRF, Static route , FRR & VRRP add operation getAllStaticRoutings, getAllVRFs, getFDFrVRFs Add new struct: StaticRoutingTableEntry_T, StaticRoutingTable_T, VRF_T

Contacts

MTNM Team Leader
Keith Dorking, Ciena
Phone: +1-678-867-5007
Fax: +1-678-867-5010
email: Kdorking@ciena.com

MTNM Business Agreement Editor
Keith Dorking, Ciena
Phone: +1-678-867-5007

Fax: +1-678-867-5010
email: Kdorking@ciena.com

MTNM Information Agreement Editor
Bernd Zeuner, T-Systems
Phone: +49 6151 937 3709
Fax: +49 6151 937 19875
email: bernd.zeuner@t-systems.com

MTNM CORBA IDL Solution Set Editor
Gerard Vila, Alcatel-Lucent
Phone : +33 1 30 77 78 68
Fax :
email : Gerard.Vila@alcatel-lucent.fr

MTNM Implementation Statement Editor
Kam Lam, Alcatel-Lucent
Phone: +1 732 949 8338
Fax: +1 732 615 4610
email: hklam@alcatel-lucent.com

TMF Senior Technical Manager
Stephen Gaito, TM Forum
Phone : +44 2476 693 633
Fax :
email : sgaito@tmforum.org

MTNM email address: mtnm@lists.tmforum.org

Acknowledgements

The TMF 814A Solution Set, Version 3.5 is a genuinely collaborative effort. The TeleManagement Forum would like to thank the following people for contributing their time and expertise to the production of this document. It is just not possible to recognize all the organizations and individuals that have contributed or influenced the MTNM Implementation Statement Guidelines and Template. We apologize to any person or organization we inadvertently missed in these acknowledgments.

Key individuals that reviewed, provided input, managed, edited and determined how to utilize inputs coming from all over the world, and really made this document happen were:

- Steve Arnott, Ericsson
- Beauford Atwater, Telcordia Technologies
- François Bédard, Nortel Networks
- Ian Best, TM Forum
- Malcolm Betts, Nortel Networks
- Teddy Bezoza, at&t

➢	Alex	Bojanic,	Tellabs
➢	Harrie	Buiting,	Alcatel-Lucent
➢	Debbie	Burkett	TM Forum
➢	Henri	Campagno,	France Telecom
➢	Lily	Chen,	Verizon
➢	Lee	Chow,	Tellabs
➢	Merav	Cohen-Sarussi,	ECI Telecom
➢	Nigel	Davis,	Nortel Networks
➢	Keith	Dorking,	Ciena
➢	Shane	Edmonds,	Alcatel-Lucent
➢	Felix	Flemisch,	Siemens
➢	Marianne	Franklin,	Alcatel-Lucent
➢	Steve	Fratini,	Telcordia Technologies
➢	Einat	Gabo-Rafaeli,	ECI Telecom
➢	Stephen	Gaito,	TM Forum
➢	Salim	Galou,	Fujitsu
➢	Elisabetta	Gardelli,	Siemens
➢	CB	Genrich,	Hitachi
➢	Noam	Goldberg,	LightScape Networks
➢	Floyd	Goldstein,	Alcatel-Lucent
➢	Alison	Greig,	Nortel Networks
➢	Garry	Grimes,	Ericsson
➢	Milton	Hall,	Telcordia Technologies
➢	Dragan	Jovanovic,	Acterna
➢	Roshan	Joyce,	Fujitsu
➢	Nathan	Kohn,	Alcatel-Lucent
➢	Hing-Kam	Lam,	Alcatel-Lucent
➢	Bertrand	LeBorgne,	Alcatel-Lucent
➢	François	Leroux,	Fujitsu
➢	Tom	Ligda,	Cisco Systems
➢	Peter	Litz,	Siemens
➢	Astrid	Lozano,	Nortel Networks
➢	Menachem	Malkosh,	ECI Telecom

- Andrea Mazzini, Alcatel-Lucent
- Srini Nangavaram, Alcatel-Lucent
- Frank O'Rourke, Ericsson
- Sandip Patel, Alcatel-Lucent
- Noga Peled, ECI Telecom
- Edgar Riemann, Siemens
- Jeffrey Ross, Fujitsu
- Victor Santos, Acterna
- Horst Schukat, Siemens
- Eric Seldner, Telcordia Technologies
- Charles Sheridan, Nortel Networks
- Sidney Shiba, Fujitsu
- Wendy Teller, Tellabs
- Niranjan Tripathy, Fujitsu
- Alexander Tschersich, Siemens
- Gerard Vila, Alcatel-Lucent
- Xuli Wu, Tellabs
- Bernd Zeuner, Deutsche Telekom
- Yakov Zimmerman, ECI Telecom

Although not directly used within this document, access to documentation and work from standards bodies and other forums have contributed to the evolution of the Multi-Technology Network Management (MTNM) NML-EML Interface. This access was via public information or TM Forum member knowledge. This list of standards bodies and forums is not exhaustive and does not imply review and concurrence by these organizations or their representatives. It is important however to acknowledge the work and their influence on the TeleManagement Forum work:

- American National Standards Institute (ANSI)
- ATM Forum
- DSL Forum
- European Telecommunications Standards Institute (ETSI)
- Institute of Electrical and Electronics Engineers (IEEE)
- International Telecommunications Union - Telecommunication Standardization Sector (ITU-T)
- Internet Engineering Task Force (IETF)
- Metro Ethernet Forum (MEF)
- MPLS and Frame Relay Alliance
- Object Management Group (OMG)
- Optical Internetworking Forum (OIF)