INTERNATIONAL TELECOMMUNICATION UNION

CCITT

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

M.3100
(10/92)

MAINTENANCE: TELECOMMUNICATIONS MANAGEMENT NETWORK

GENERIC NETWORK INFORMATION MODEL

Recommendation M.3100

FOREWORD

The CCITT (the International Telegraph and Telephone Consultative Committee) is a permanent organ of the International Telecommunication Union (ITU). CCITT is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The Plenary Assembly of CCITT which meets every four years, establishes the topics for study and approves Recommendations prepared by its Study Groups. The approval of Recommendations by the members of CCITT between Plenary Assemblies is covered by the procedure laid down in CCITT Resolution No. 2 (Melbourne, 1988).

Recommendation M.3100 was prepared by Study Group IV and was approved under the Resolution No. 2 procedure on the 5th of October 1992.

———————————

CCITT NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized private operating agency.

CONTENTS

Recommendation M.3100

GENERIC NETWORK INFORMATION MODEL

(1992)

Abstract

This Recommendation provides a generic network information model. The model describes managed object classes and their properties that are generic and useful to describe information exchanged across all interfaces defined in Recommendation M.3010 TMN architecture. These generic managed object classes are intended to be applicable across different technologies, architectures and services. The managed object classes in this Recommendations may be specialized to support the management of various telecommunications networks.

Keywords

– Actions;

– ASN.1;

– Attributes;

– Generic Network Information Model;

– Managed Object Class;

– Notifications.

1   Scope, purpose, and field of application

1.1   *Scope*

This Recommendation provides a generic network information model. It identifies TMN object classes that are common to managed telecommunications networks; or are of a generic type that can be used to manage a network at a technology-independent level; or are super-classes of technology-specific managed objects in a telecommuni-cations network; or management support objects that are required for the management of the telecommunications network. These objects are relevant to information exchanged across standardized interfaces defined in Recommendation M.3010 [1], Principles for a Telecommunications Management Network.

Recommendation M.3100 addresses generically the abstractions of those aspects of telecommunication resources (e.g. equipment, telecommunication services) required to manage the network. It also includes the abstractions related to the management services.

Recommendation M.3100 does not address abstractions relevant to technology-specific areas or implemen-tation-specific details.

1.2   *Purpose*

1.2.1   *Interoperability*

There will be a variety of TMN conformant management systems and managed systems concerning many technology-specific areas, such as switching and transmission. One purpose of this Recommendation is to provide a vehicle for management interoperability between such systems.

1.2.2 *Technology-independent-management*

By introducing the concept of technology-independent-management, it is possible to perform management of diverse equipment using common communications interfaces. In this manner, an "abstract" view over a set of Network Elements can be achieved.

1.2.3 *Facilitating information model development*

This Recommendation also provides a framework from which technology-specific information models may be developed using the modeling principles defined in Recommendation X.720 [2].

1.3 *Field of application*

This Recommendation captures the generally applicable requirements of the technology-independent and technology-specific information models as well as information relating to TMN management services.

Through specialization, this Recommendation is applicable to technology-specific TMN information models. The mechanism for specialization is inheritance.

Even though technology-specific models may be derived from this Recommendation, some of the generic managed object classes in this Recommendation are instantiable (i.e. instances of the classes may be created) in order to provide interoperability between equipment supporting information models derived from this Recommendation and equipment that only supports the information model in this Recommendation.

1.4 *Structure of this Recommendation*

Section 2 (of this Recommendation) provides an overview of the Generic Network Information Model. The definition of management information in §§ 3 to 8, describing information model is documented using the notational mechanisms defined in Recommendation X.722 [3]. The relationships between the managed object classes for the different fragments of the model in § 3 are depicted using entity relationship diagrams. Section 9 contains the syntax definitions of the information carried in the protocol. The notation used is Abstract Syntax Notation One (ASN.1) defined in Recommendation X.208 [4].

When referencing the definitions for the templates in this Recommendation by other documents, the prefix "Recommendation M.3100": is recommended to identify the sources for the definitions.

2 Overview of the Generic Network Information Model

A Generic Network Information Model is essential to the generation of uniform fault, configuration, performance, security, and accounting management standards. A common network model, identifying the generic resources that exist in a network and their associated attribute types, events, actions, and behaviour, provides a foundation for understanding the interrelationships between these resources and attributes, and may, in turn, promote uniformity in dealing with the various aspects of managing these resources and attributes.

Network resources may be customer- or provider-owned; the latter includes portions that may be assigned for exclusive use by specific customers. Resources may be physical or logical in nature. Physical resources include customer (e.g. PBXs) or provider (e.g. digital cross connect systems) systems, their associated subsystems (e.g. a line card within a PBX) and also the links that interconnect these systems. Such systems are generally known as Network Elements (NEs). Logical resources include communication protocols, application programs, logs, and network services.

There may also exist (separate or integrated) TMN resources involved in operating a telecommunication network. These resources include the OSs closely associated with managing specific NEs, and OSs that have network-wide responsibilities.

Resources have attributes which allow the user to control and/or observe the behaviour of the resource. Attributes may also allow the user to control and/or observe the relationships between resources.

There is a need to represent the way resources, or entities can be combined and interrelated (relationships). In this version, Entity-Relationship (E-R) diagram techniques have been used to represent inter-object relationships. As these tools are improved, newer ones may be used in future issues.

These E-R diagrams result in a high-level view (schema) of the Generic Network Information Model. This view can be used to derive information related to naming, to verify consistency, and to ensure completeness. For example, it ensures that sufficient information (i.e. relationships) is provided from a physical resource to identify the services that are dependent on that resource.

The information exchanged at the management interface is modelled using design principles outlined in Recommendation X.720 [2], Management Information Model. Resources are modelled as objects, and the manage-ment view of a resource is called a managed object. Additional objects, called support managed objects, are defined to support the functions of managing a telecommunication network.

Objects with similar attributes and behaviours may be grouped into object classes. An object is characterized by its object class and object instance, and may possess multiple attribute types and associated values. Similarly, the terms managed object class and managed object instance apply specifically to objects that are being managed. This Recommendation specifies the properties of the resource (i.e. managed object) visible for management.

An object class may be a subclass of another object class. A subclass inherits attribute types and behaviours of its super-class, in addition to possessing its own specific attributes and properties.

Object classes and attribute types are defined only for the purpose of communicating network management messages between systems, and need not be related to the structuring of data within these systems. Some object classes defined in this issues (and future issues) of the model apply to many management functional areas, while others support specific functional areas.

This version of the Generic Network Information Model contains common object classes and attribute types as well as those particular to alarm surveillance. Subsequent issues of this Recommendation will augment the list of object classes, attribute types, and operations in order to further accommodate other functional areas.

Annex A contains an index of managed object classes, packages, attributes, notifications and actions defined in this Recommendation.

There are several different viewpoints of management information which may be defined for management purposes, with the NE level viewpoint, the network level viewpoint and the service level viewpoint defined below. These viewpoints are not restrictive but define the levels of abstraction of particular types of interfaces. That is, object class definitions are not forced into this categorization but are constructed to meet the needs of exchanging management information across TMN interfaces. Objects defined for a given viewpoint may be used in others, and any object may be used by any interface which requires it. The definition of viewpoint is a means of generating requirements, hence there is no implicit definition of interfaces or storage requirements. This information is defined for the purpose of management via an open interface.

The NE level viewpoint is concerned with the information that is required to manage an NE. This refers to the information required to manage the Network Element Function (NEF) and the physical aspects of the NE. The information may be derived from open systems other than the NE.

The network level viewpoint is concerned with the information representing the network, both physically and logically. It is concerned with how network element entities are related, topographically interconnected, and configured to provide and maintain end-to-end connectivity.

The service level viewpoint is concerned with how network level aspects (such as an end-to-end path) are utilized to provide a network service, and as such is concerned with the requirements of a network service (e.g. availability, cost, etc.) and how these requirements are met through the use of the network, and all related customer information.

3   Object Classes

This section contains the definitions of the object classes that form the basis for the Generic Network Information Model. These object classes are grouped into 6 fragments and they are depicted in Figures 1/M.3100 to 6/M.3100. These fragments show all related object classes from different perspectives. Additional fragments and object classes for each fragment are for further study. The inheritance hierarchy of this model is presented in Figure 7/M.3100.
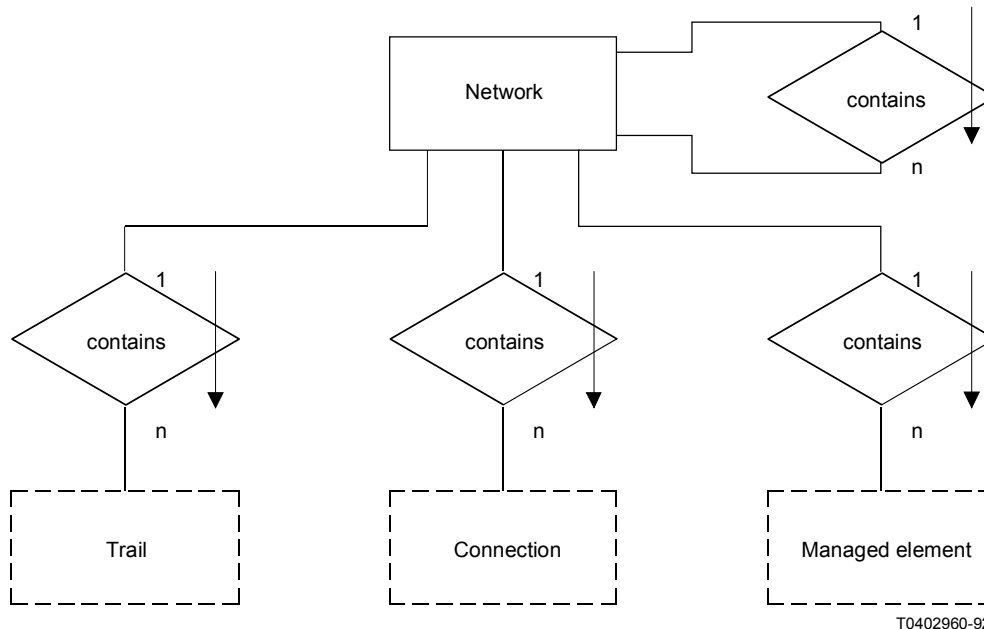
Note − Object classes contained in dotted boxes may be found in other views.

FIGURE 1/M.3100

Entity-relationship depiction of the Network Fragment

Network

contains

1

n

Managed element

contains

1

n

Equipment

contains

1

n

contains

1

n

n

n

Fabric

n

Trail termination
point (Note 1)

Software

contains

n

1

T0402970-92
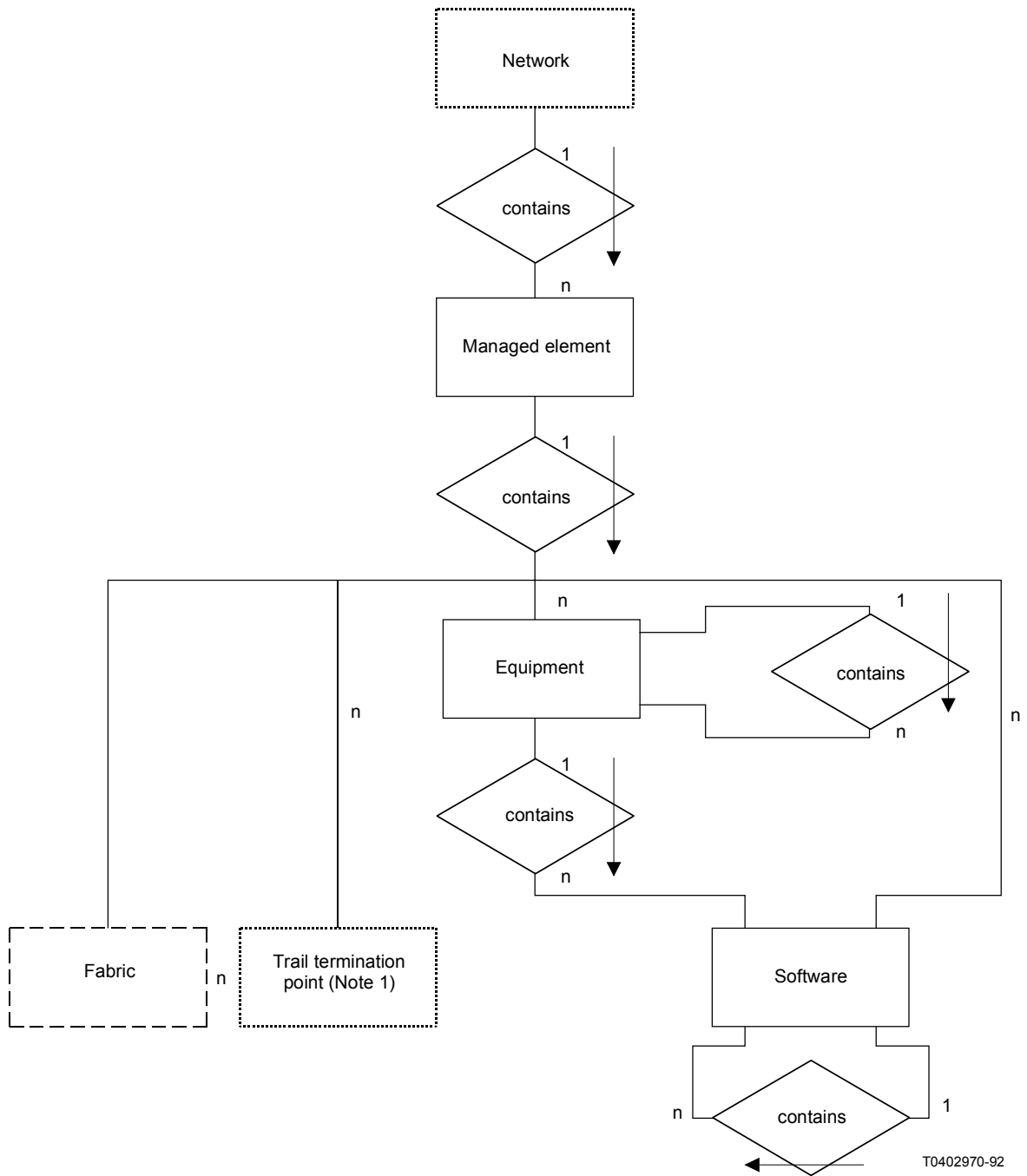
Note 1 – Represents source, sink and bidirectional object classes.

Note 2 – Object classes contained in dotted boxes may be found in other views.

FIGURE 2/M.3100

Entity-relationship depiction of the Managed Element Fragment

T0402980-92
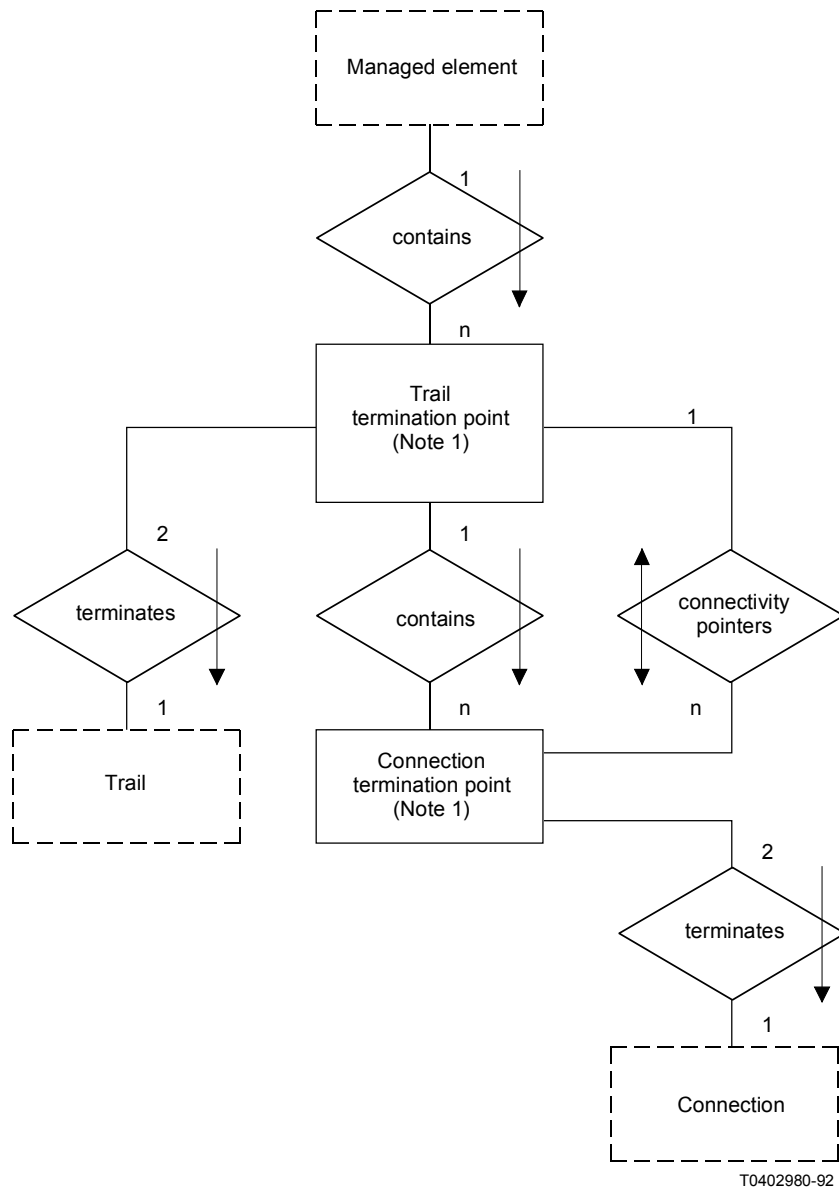
Note 1 – Represents source, sink and bidirectional classes.

Note 2 – Object classes contained in dotted boxes may be found in other views.

FIGURE 3/M.3100

Entity-relationship depiction of Termination Point Fragment
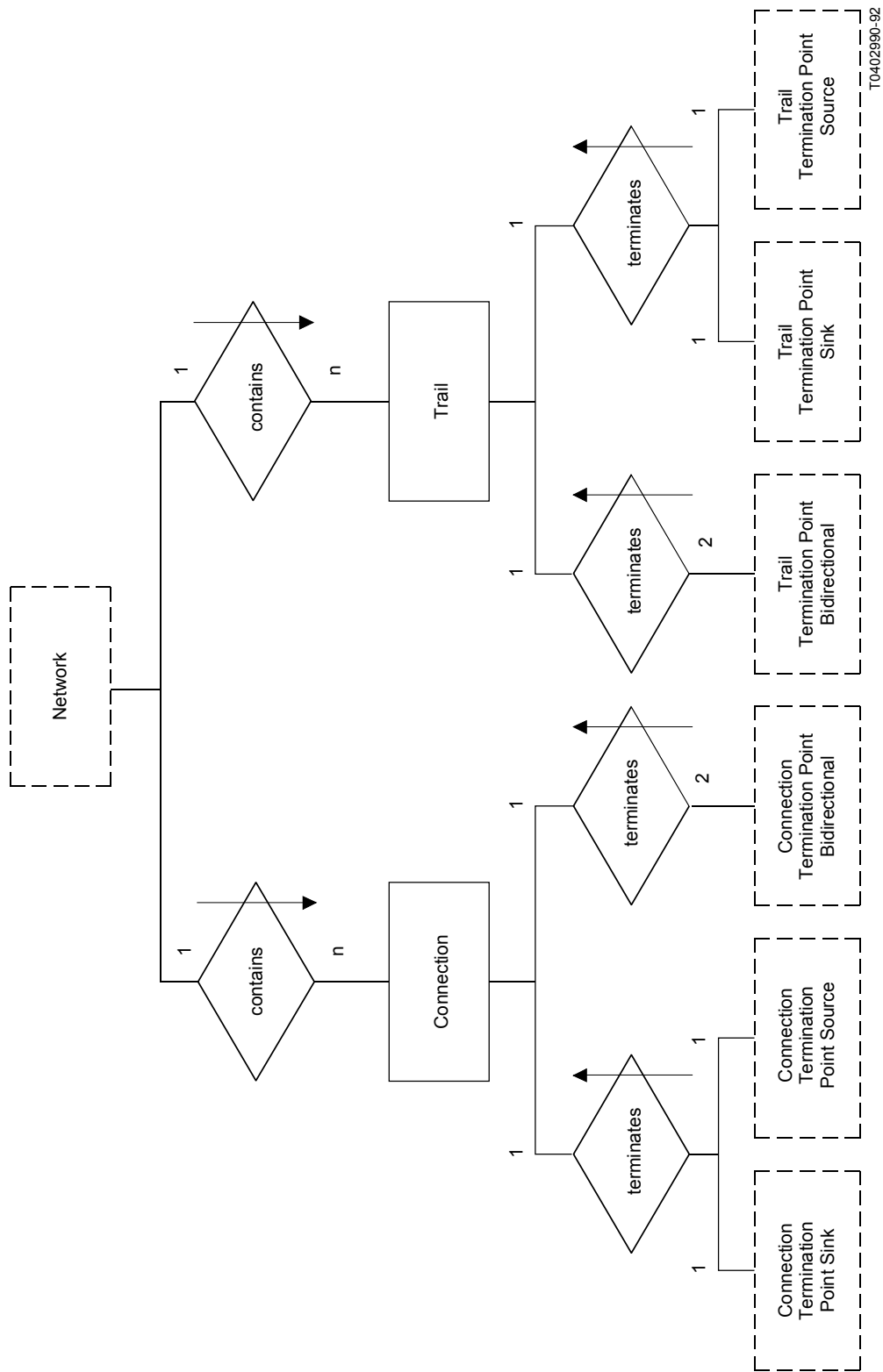
T0402990-92

FIGURE 4/M.3100

Entity-relationship depiction of Transmission Fragment

Note – Object classes contained in dotted boxes may be found in other views.
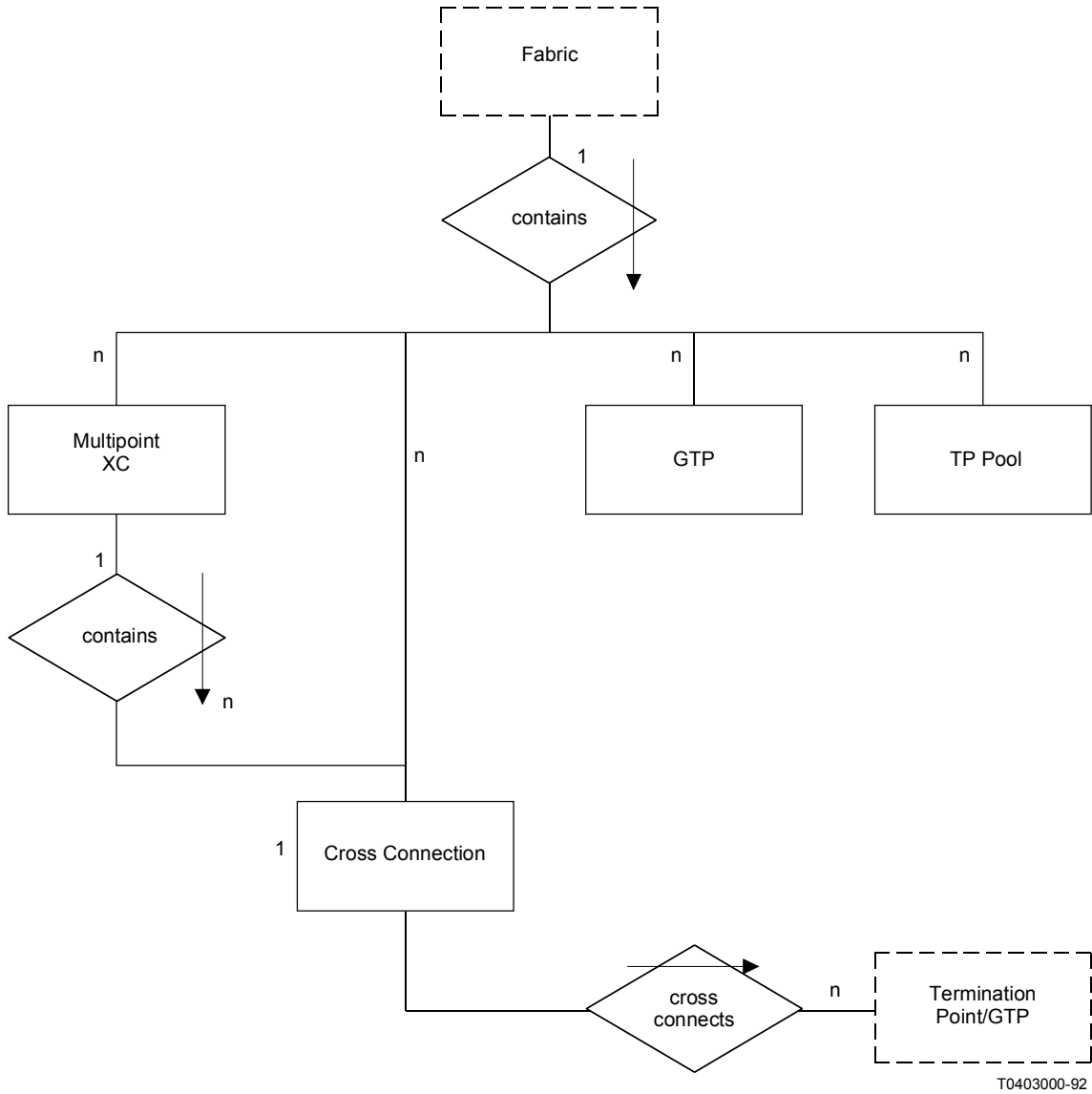
Note – Object classes (Termination point subclasses or GTP) contained in dotted boxes may be found in other views.

FIGURE 5/M.3100

Entity-Relationship depiction of Cross Connection Fragment
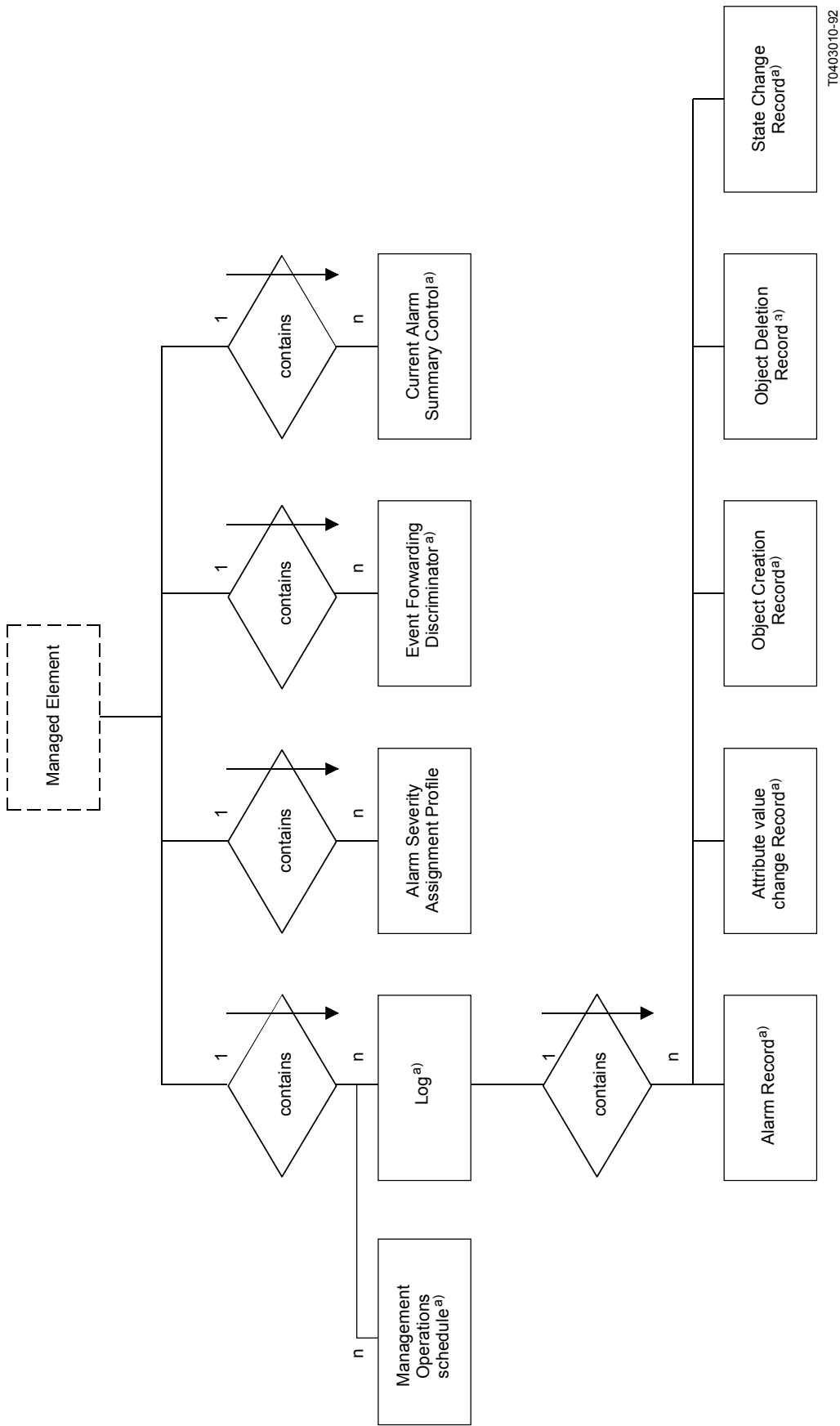
FIGURE 6/M.3100

Entity–Relationship depiction of the Functional Area Fragment
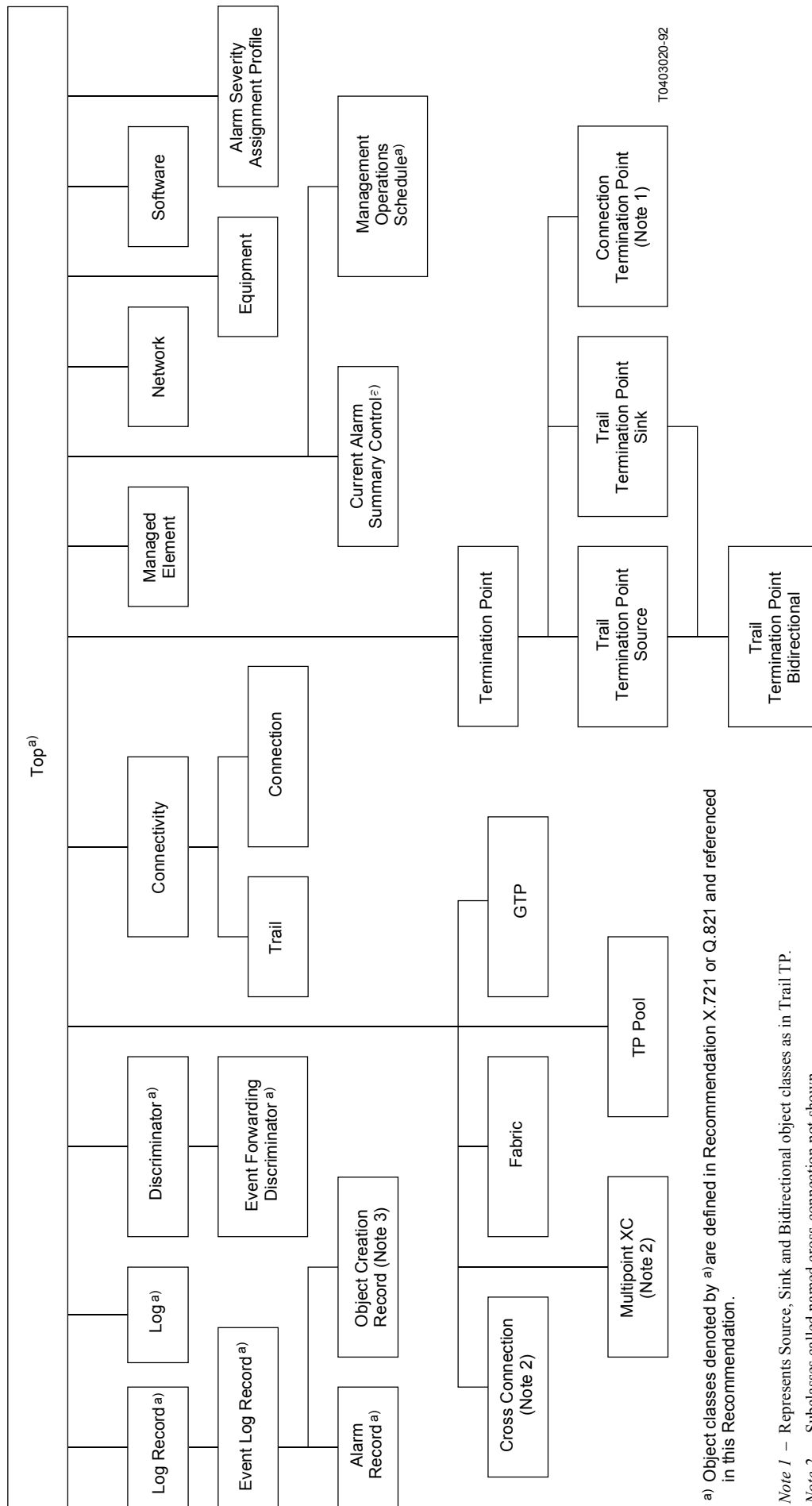
a) Object classes denoted by a) are defined in Recommendation X.721 or Q.821 and referenced in this Recommendation.

Note – Object classes contained in dotted boxes may be found in other views.

T0403010-92

FIGURE 7/M.3100

Inheritance hierarchy

a) Object classes denoted by a) are defined in Recommendation X.721 or Q.821 and referenced in this Recommendation.

Note 1 – Represents Source, Sink and Bidirectional object classes as in Trail TP.

Note 2 – Subclasses called named cross-connection not shown.

Note 3 – Other Records not shown.

The purpose of defining fragments is only to have a document that is easier to read by grouping a limited number of object class definitions. Each fragment deals with a particular subject (e.g. network, managed element, transmission, support objects) but object classes of each fragment will be usable in various models depending on the functional area managed and/or on the level viewpoint considered.

Table 1/M.3100 lists the object classes defined or referenced in this Recommendation.

TABLE 1/M.3100

Managed Object Classes

Object Classes
Alarm Record[a]
Alarm Severity Assignment Profile
Attribute Value Change Record[a]
Connection
Connection Termination Point Bidirectional
Connection Termination Point Sink
Connection Termination Point Source
Connectivity
Cross-Connection
Current Alarm Summary Control[a]
Discriminator[a]
Equipment
Event Forwarding Discriminator[a]
Event Log Record[a]
Fabric
Group Termination Point
Log[a]
Log Record[a]
Managed Element
Management Operations Scheduler[a]
Multipoint Cross-Connection
Named Cross-Connection
Named Multipoint Cross-ConnectionNetwork
Network
Object Creation Record[a]
Object Delection Record[a]
Software
State Change Record[a]
Termination Point
TP Pool
Trail
Trail Termination Point Bidirectional
Trail Termination Point Sink
Trail Termination Point Source

Note – Object classes denoted by [a] are defined in other Recommendations and referenced in this Recommendation.

3.1 *Network Fragment*

Managed object classes in network fragment are presented in Figure 1/M.3100. The definition(s) of the managed object class(es) are specified as follows:

3.1.1 *Network*

network MANAGED OBJECT CLASS

    DERIVED FROM   "Recommendation X.721: 1992":top;

    CHARACTERIZED BY

    networkPackage PACKAGE

        BEHAVIOUR

            networkDefinition;

        ATTRIBUTES

            networkId GET;;;

    CONDITIONAL PACKAGES

        userLabelPackage PRESENT IF "an instance supports it";

REGISTERED AS { m3100ObjectClass 1 };

networkDefinition BEHAVIOUR

    DEFINED AS

"The Network object class is a class of managed objects that are collections of interconnected telecommunications and management objects (logical or physical) capable of exchanging information. These objects have one or more common characteristics, for example they may be owned by a single customer or provider, or associated with a specific service network. A network may be nested within another (larger) network, thereby forming a containment relationship. An example of a network that is contained in another network is a transmission sub-network. It is owned by a single administration and can only perform trans-mission functions.";

3.2 *Managed Element Fragment*

Managed object classes in managed element fragment are presented in Figure 2/M.3100. The definition(s) of the managed object class(es) are specified as follows:

3.2.1 *Equipment*

equipment MANAGED OBJECT CLASS

    DERIVED FROM   "Recommendation X.721: 1992":top;

    CHARACTERIZED BY

    equipmentPackage PACKAGE

        BEHAVIOUR

        equipmentBehaviour BEHAVIOUR
        DEFINED AS

"The Equipment object class is a class of managed objects that represents physical components of a managed element, including replaceable components. An instance of this object class is present in a single geographic location. An equipment may be nested within another equipment, thereby creating a containment relationship. The equipment type shall be identified by sub-classing this object class. Either the name of the sub-class or an attribute may be used for identifying the equipment type.

When the attribute value change notification package is present, the attributeValueChange notification defined in Recommendation X.721 shall be emitted when the value of one of the following atrributes changes: alarm status, affected object list, user label, version, location name and current problem list. Because the above attributes are all in conditional packages, the behaviour for emitting the attribute

value change notification applies only when the corresponding conditional packages are present in the managed object. When the state change notification package is present, the stateChangeNotification defined in Recommendation X.721 shall be emitted if the value of administrative state or operational state changes (when the administrativeOperationalStates conditional package is present)."",

ATTRIBUTES

        equipmentId GET,

        replaceable GET;

    ;;

CONDITIONAL PACKAGES

        createDeleteNotificationsPackage PRESENT IF "the objectCreation and objectDeletion

            notifications defined in Recommendation X.721 are supported by an instance of this class.",

        attributeValueChangeNotificationPackage PRESENT IF "the attributeValueChange

            notification defined in Recommendation X.721 is supported by an instance of this class.",

        stateChangeNotificationPackage PRESENT IF "the stateChange notification defined in

            Recommendation X.721 is supported by an instance of this class.",

        administrativeOperationalStatesPackage PRESENT IF "an instance supports it.",

        affectedObjectListPackage PRESENT IF "an instance supports it.",

        equipmentEquipmentAlarmPackage PRESENT IF "the equipmentAlarm notification defined in

            Recommendation X.721 is supported by an instance of this class.",

        environmentalAlarmPackage PRESENT IF "the environmentalAlarm notification defined in

            Recommendation X.721 is supported by an instance of this class.",

        tmnCommunicationsAlarmInformationPackage PRESENT IF "the communicationsAlarm notification defined in Recommendation X.721 is supported by an instance of this class.",

        processingErrorAlarmPackage PRESENT IF "the processingErrorAlarm notification defined in

            Recommendation X.721 is supported by an instance of this class.",

        userLabelPackage PRESENT IF "an instance supports it",

        vendorNamePackage PRESENT IF "an instance supports it",

        versionPackage PRESENT IF "an instance supports it",

        locationNamePackage PRESENT IF "an instance supports it",

        currentProblemListPackage PRESENT IF "an instance supports it";

REGISTERED AS { m3100ObjectClass 2 };

3.2.2   *Managed Element*

managedElement MANAGED OBJECT CLASS

    DERIVED FROM   "Recommendation X.721: 1992":top;

    CHARACTERIZED BY

    managedElementPackage PACKAGE

        BEHAVIOUR

            managedElementBehaviour BEHAVIOUR

            DEFINED AS

            "The Managed Element object class is a class of managed objects representing telecommunications equipment or TMN entities (either groups or parts) within the telecommunications network that performs managed element functions, i.e. provides support and/or service to the subscriber. Managed elements may or may not additionally perform mediation/OS functions. A managed element communicates with the manager over one or more standard Q-interfaces for the purpose of being monitored and/or controlled. A managed element contains equipment that may or may not be geographically distributed.

            When the attribute value change notification package is present, the attributeValueChange notification defined in Recommendation X.721 shall be emitted when the value of one of the following attributes changes: alarm status, user label, version, location name and current problem list. For the above attributes that are in conditional packages, the behaviour for emitting the

attribute value change notification applies only when the corresponding conditional packages are present in the managed object. When the state change notification package is present, the stateChangeNotification defined in Recommendation X.721 shall be emitted if the value of administrative state or operational state or usage state changes."

; ;

ATTRIBUTES

managedElementId GET,

"Recommendation X.721: 1992":systemTitle GET-REPLACE,

alarmStatus GET,

"Recommendation X.721: 1992":administrativeState GET-REPLACE,

"Recommendation X.721: 1992":operationalState GET,

"Recommendation X.721: 1992":usageState GET;

NOTIFICATIONS

"Recommendation X.721: 1992":environmentalAlarm,

"Recommendation X.721: 1992":equipmentAlarm,

"Recommendation X.721: 1992":communicationsAlarm,

"Recommendation X.721: 1992":processingErrorAlarm;;;

CONDITIONAL PACKAGES

createDeleteNotificationsPackage PRESENT IF "the objectCreation and objectDeletion

notifications defined in Recommendation X.721 are supported by an instance of this class.",

attributeValueChangeNotificationPackage PRESENT IF "the attributeValueChange

notification defined in Recommendation X.721 is supported by an instance of this class.",

stateChangeNotificationPackage PRESENT IF "the stateChange notification defined in

Recommendation X.721 is supported by an instance of this class.",

audibleVisualLocalAlarmPackage PRESENT IF "an instance supports it",

resetAudibleAlarmPackage PRESENT IF "an instance supports it",

userLabelPackage PRESENT IF "an instance supports it",

vendorNamePackage PRESENT IF "an instance supports it",

versionPackage PRESENT IF "an instance supports it",

locationNamePackage PRESENT IF "an instance supports it",

currentProblemListPackage PRESENT IF "an instance supports it",

externalTimePackage PRESENT IF "an instance supports it",

systemTimingSourcePackage PRESENT IF "an instance supports it";

REGISTERED AS { m3100ObjectClass 3 };

3.2.3   *Software*

software MANAGED OBJECT CLASS

DERIVED FROM   "Recommendation X.721: 1992":top;

CHARACTERIZED BY

softwarePackage PACKAGE

BEHAVIOUR

softwareBehaviour BEHAVIOUR

DEFINED AS

"The Software object class is a class of managed objects that represent logical information stored in equipment, including programs and data tables. Software may be nested within other software, thereby creating a containment relationship.

When the attribute value change notification package is present, the attributeValueChange notification defined in Recommendation X.721 shall be emitted when the value of one of the following attributes changes: alarm status, affected object list, user label, version, and current problem list. Because the above attributes are all in conditional packages, the behaviour for emitting the attribute value change notification applies only when the corresponding conditional

packages are present in the managed object. When the state change notification package is present, the stateChangeNotification defined in Recommendation X.721 shall be emitted if the value of administrative state or operational state changes (when the administrativeOpera-tionalStates conditional package is present)."

 ;;

ATTRIBUTES

 softwareId GET;

;;

CONDITIONAL PACKAGES

 createDeleteNotificationsPackage PRESENT IF "the objectCreation and objectDeletion

  notifications defined in Recommendation X.721 are supported by an instance of this class.",

 attributeValueChangeNotificationPackage PRESENT IF "the attributeValueChange

  notification defined in Recommendation X.721 is supported by an instance of this class.",

 stateChangeNotificationPackage PRESENT IF "the stateChange notification defined in

  Recommendation X.721 is supported by an instance of this class.",

 administrativeOperationalStatesPackage PRESENT IF "an instance supports it.",

 affectedObjectListPackage PRESENT IF "an instance supports it.",

 softwareProcessingErrorAlarmPackage PRESENT IF "an instance supports it.",

 userLabelPackage PRESENT IF "an instance supports it",

 vendorNamePackage PRESENT IF "an instance supports it",

 versionPackage PRESENT IF "an instance supports it",

 currentProblemListPackage PRESENT IF "an instance supports it";

REGISTERED AS { m3100ObjectClass 4 };

3.3 *Termination Point Fragment*

Managed object classes in termination point fragment are presented in Figure 3/M.3100. The behaviour definition(s) of the managed object class(es) are specified as follows:

3.3.1 *Connection Termination Point Bidirectional*

The Connection Termination Point Bidirectional object class is a class of managed objects that originates a link connection and terminates a link connection. Technology specific connection termination bidirectional sub-classes are derived by multiple inheritance from this object class and corresponding technology specific source and sink object classes.

connectionTerminationPointBidirectional MANAGED OBJECT CLASS

 DERIVED FROM

  connectionTerminationPointSource,

  connectionTerminationPointSink;

REGISTERED AS { m3100ObjectClass 5 };

3.3.2 *Connection Termination Point Sink*

The Connection Termination Point Sink object class is a class of managed objects that terminates a link connection.

connectionTerminationPointSink MANAGED OBJECT CLASS

 DERIVED FROM terminationPoint;

 CHARACTERIZED BY

 connectionTerminationPointSinkPackage PACKAGE

  BEHAVIOUR

   connectionTerminationPointSinkBehaviour BEHAVIOUR

   DEFINED AS

   "This managed object terminates a link connection. The downstream connectivity pointer attribute points to the termination point managed object, within the same managed element, that receives information (traffic) from this termination point at the same layer, or is null. The referenced object shall be an instance of one of the following classes or its sub-classes: Trail

Termination Point Sink, Trail Termination Point Bidirectional, Connection Termination Point Source, Connection Termination Point Bidirectional. The downstream connectivity pointer may identify one or more objects depending on whether the signal is connected to one or more termination point objects."

;;

ATTRIBUTES

    downstreamConnectivityPointer PERMITTED VALUES
    -- The allowed choices for the syntax of this attribute are restricted in the subtype

    -- CTPDownstreamPointer

        ASN1DefinedTypesModule.CTPDownstreamPointer GET;;;

CONDITIONAL PACKAGES

    ctpInstancePackage PRESENT IF "the name binding used to create an instance of this object

    class requires this attribute.",

    channelNumberPackage PRESENT IF "an instance supports it";

REGISTERED AS { m3100ObjectClass 6 };

3.3.3 *Connection Termination Point Source*

The Connection Termination Point Source object class is a class of managed objects that originates a link connection.

connectionTerminationPointSource MANAGED OBJECT CLASS

    DERIVED FROM   terminationPoint;

    CHARACTERIZED BY

    connectionTerminationPointSourcePackage PACKAGE

        BEHAVIOUR

            connectionTerminationPointSourceBehaviour BEHAVIOUR

            DEFINED AS

            "This managed object originates a link connection. The upstream connectivity pointer attribute points to the termination point managed object, within the same managed element, that sends information (traffic) to this termination point at the same layer, or is null. The referenced object shall be an instance of one of the following classes or its sub-classes: Trail Termination Point Source, Trail Termination Point Bidirectional, Connection Termination Point Sink, Connection Termination Point Bidirectional."

    ;;

    ATTRIBUTES

        upstreamConnectivityPointer   PERMITTED VALUES
        -- The allowed choices for the syntax of this attribute are restricted in the subtype

        -- CTPUpstreamPointer

            ASN1DefinedTypesModule.CTPUpstreamPointer   GET

    ;;;

    CONDITIONAL PACKAGES

        ctpInstancePackage PRESENT IF "the name binding used to create an instance of this object

        class requires this attribute.",

        channelNumberPackage PRESENT IF "an instance supports it";

REGISTERED AS { m3100ObjectClass 7 };

3.3.4 *Termination Point*

The Termination Point object class is a class of managed objects that terminates transport entities, such as trails and connections. This object class is a basic object class from which sub-classes, such as trail termination and connection termination point are derived. The use of operational state is further refined in sub-classes of this class. This is an uninstantiable managed object class.

terminationPoint MANAGED OBJECT CLASS

    DERIVED FROM "Recommendation X.721: 1992":top;

    CHARACTERIZED BY

    terminationPointPackage PACKAGE

        BEHAVIOUR

            terminationPointBehaviour BEHAVIOUR

            DEFINED AS

            "This managed object represents the termination of a transport entity, such as a trail or a connection. The characteristic information attribute is used to identify equivalence between sub-classes of termination points in order to determine whether cross connection or connectivity is possible. The operational state reflects the perceived ability to generate and/or receive a valid signal. Sub-classes of termination point shall specify the attributes and states for which attribute value change and state change notifications will be generated."

            ;;

        ATTRIBUTES

            supportedByObjectList GET;;;

    CONDITIONAL PACKAGES

        createDeleteNotificationsPackage PRESENT IF

            "the objectCreation and objectDeletion notifications defined in Recommendation X.721 are supported by an instance of this managed object class",

        attributeValueChangeNotificationPackage PRESENT IF

            "the attributeValueChange notification defined in Recommendation X.721 is supported by an instance of this managed object class",

        stateChangeNotificationPackage PRESENT IF

            "the stateChange notification defined in Recommendation X.721 is supported by an instance of this managed object class",

        operationalStatePackage PRESENT IF

            "the resource represented by this managed object is capable of assessing the ability to generate and/or receive a valid signal.",

        crossConnectionPointerPackage PRESENT IF

            "the termination point can be flexibly assigned, (i.e. cross connected).",

        characteristicInformationPackage PRESENT IF

            "an instance supports it.",

        networkLevelPackage PRESENT IF "an instance supports it",

        tmnCommunicationsAlarmInformationPackage PRESENT IF

            "the communicationsAlarm notification (as defined in Recommendation X.721) is supported by this managed object",

        alarmSeverityAssignmentPointerPackage PRESENT IF

            "the tmnCommunicationsAlarmInformationPackage package is present AND the managed object supports configuration of alarm severities";

REGISTERED AS { m3100ObjectClass 8 };

3.3.5   *Trail Termination Point Bidirectional*

       The Trail Termination Point Bidirectional object class is a class of managed objects representing a termination point where one trail is originated and another trail is terminated. It represents the access point in a layer network which is a focus for both the trail relationship and the client/server relationship. These concepts are defined in Recommendation G.803 [6]. Sub-classes of this generic super-class include in addition the overhead interpretation aspects of trail termination function described in Recommendation G.803 [6]. Technology specific (e.g. PDH, SDH) trail termination bidirectional object classes may be defined directly as sub-classes of this class and corresponding source and sink object classes using multiple inheritance.

trailTerminationPointBidirectional MANAGED OBJECT CLASS

    DERIVED FROM   trailTerminationPointSource, trailTerminationPointSink;

    CHARACTERIZED BY

    trailTerminationPointBidirectionalPackage PACKAGE

        BEHAVIOUR

            trailTerminationPointBidirectionalBehaviour BEHAVIOUR

            DEFINED AS

            "The operational state is disabled if either the sink or source part of the termination point is disabled."

            ;;;;

REGISTERED AS { m3100ObjectClass 9 };

3.3.6   *Trail Termination Point Sink*

       The Trail Termination Point Sink object class is a class of managed objects representing a termination point where a trail is terminated. It represents the access point in a layer network which is a focus for both the trail relationship and the client/server relationship. These concepts are defined in Recommendation G.803 [6]. Sub-classes of this generic superclass include in addition the overhead interpretation aspects of the trail termination function described in Recommendation G.803 [6].

trailTerminationPointSink MANAGED OBJECT CLASS

    DERIVED FROM   terminationPoint;

    CHARACTERIZED BY

    operationalStatePackage,

    trailTerminationPointSinkPackage PACKAGE

        BEHAVIOUR

            trailTerminationPointSinkBehaviour BEHAVIOUR

            DEFINED AS

            "This managed object represents a termination point where a trail is terminated. It represents the access point in a layer network which is a focus for both the trail relationship and the client/server relationship.

            The operational state reflects the perceived ability to receive a valid signal. If the termination point detects that a signal received has failed or it is unable to process the incoming signal, then the operational state will have the value disabled.

            When the administrative state is locked, the termination point is administratively removed from service. When the administrative state is unlocked, the termination point is administratively in service. Changes to administrative state have no effect on the connectivity pointer.

            A change in the operational state shall cause a state change notification. If administrative state is present in an instance of trail termination point sink class, it shall not emit a state change notification. However, sub-classes of trail termination point sink class may modify this behaviour to require this notification. Sub-classes of trail termination point sink shall specify the attributes for which attribute value change notifications should be generated.

The upstream connectivity pointer attribute points to the termination point managed object, within the same managed element, that sends information (traffic) to this termination point at the same layer, or is null. The referenced object shall be an instance of one of the following classes or its sub-classes: Connection Termination Point Sink or Bidirectional (single or a concatenated sequence) or Trail Termination Point Source or Bidirectional."

;;

ATTRIBUTES

upstreamConnectivityPointer GET ;;;

CONDITIONAL PACKAGES

"Recommendation X.721: 1992":administrativeStatePackage PRESENT IF

"the resource represented by the managed object is capable of being administratively placed in and out of service",

supportableClientListPackage PRESENT IF

"the object class can support more than one type of client",

ttpInstancePackage PRESENT IF

"the name binding used to create an instance of this object class requires this attribute.";

REGISTERED AS { m3100ObjectClass 10 };

3.3.7   *Trail Termination Point Source*

The Trail Termination Point Source object class is a class of managed objects representing a termination point where a trail is originated. It represents the access point in a layer network which is a focus for both the trail relationship and the client/server relationship. These concepts are defined in Recommendation G.803 [6].

trailTerminationPointSource MANAGED OBJECT CLASS

DERIVED FROM   terminationPoint;

CHARACTERIZED BY

operationalStatePackage,

trailTerminationPointSourcePackage PACKAGE

BEHAVIOUR

trailTerminationPointSourceBehaviour BEHAVIOUR

DEFINED AS

"This managed object represents a termination point where a trail is originated. It represents the access point in a layer network which is a focus for both the trail relationship and the client/server relationship.

The operational state reflects the perceived ability to generate a valid signal. If the termination point detects that a valid signal can not be generated, then the operational state will have the value disabled.

When the administrative state is locked, the termination point is administratively removed from service. When the administrative state is unlocked, the termination point is administratively in service. Changes to administrative state have no effect on the connectivity pointer.

A change in the operational state shall cause a state change notification. If administrative state is present in an instance of trail termination point source class, it shall not emit a state change notification. However, sub-classes of trail termination point source class may modify this behaviour to require this notification. Sub-classes of trail termination point source shall specify the attributes for which attribute value change notifications should be generated.

The downstream connectivity pointer attribute points to the termination point managed object, within the same managed element, that receives information (traffic) from this termination point at the same layer, or is null. The referenced object shall be an instance of one of the following

classes or its sub-classes: Connection Termination Point Source or Bidirectional (single or a concatenated sequence or a set if connected to more than one connection termination point source objects) or Trail Termination Point Sink or Bidirectional (single or a set if connected to more than one trail termination point sink objects)."

;;

ATTRIBUTES

 downstreamConnectivityPointer GET;;;

CONDITIONAL PACKAGES

 "Recommendation X.721: 1992":administrativeStatePackage PRESENT IF

  "the resource represented by the managed object is capable of being administratively placed in and out of service",

 supportableClientListPackage PRESENT IF

  "the object class can support more than one type of client",

 ttpInstancePackage PRESENT IF

  "the name binding used to create an instance of this object class requires this attribute.";

REGISTERED AS { m3100ObjectClass 11 };

3.4 *Transmission Fragment*

Object classes in transmission fragment are presented in Figure 4/M.3100. The behavior definition(s) of the object class(es) are specified as follows:

3.4.1 *Connection*

connection MANAGED OBJECT CLASS

 DERIVED FROM connectivity;

 CHARACTERIZED BY

 connectionPackage PACKAGE

  BEHAVIOUR

  connectionBehaviour BEHAVIOUR

   DEFINED AS

   "The Connection object class is a class of managed objects responsible for the transparent transfer of information between connection termination points. A connection is a component of a trail.

   Several connections can be bundled into a higher rate trail. A sequence of one or more connections are linked together to form a trail. A connection may be either uni- or bi-directional."

 ;;

 ATTRIBUTES

  connectionId GET;;;

 CONDITIONAL PACKAGES

 serverTrailListPackage PRESENT IF "an instance supports it",

 clientTrailPackage PRESENT IF "an instance supports it";

REGISTERED AS { m3100ObjectClass 12 };

3.4.2 *Connectivity*

connectivity MANAGED OBJECT CLASS

 DERIVED FROM "Recommendation X.721: 1992":top;

 CHARACTERIZED BY

 connectivityPackage PACKAGE

  BEHAVIOUR

  connectivityBehaviour BEHAVIOUR

  DEFINED AS

  "The Connectivity object class is a class of managed objects which ensures the transfer of information between two termination points. It is not instantiable because the transfer is effected via the client-server relationship of trail and connection. Connectivity direction is determined by the directionality of the a and z termination points.

If an instance of this class is bidirectional, the a and z termination points shall also be bidirectional. If an instance of this class is unidirectional, the a point shall be the source TP and the z-termination point shall be the sink TP.

The operational state indicates the capability to carry a signal."

;;

    ATTRIBUTES

        directionality GET,

        "Recommendation X.721: 1992":administrativeState GET-REPLACE,

        "Recommendation X.721: 1992":operationalState GET,

        a-TPInstance GET,

        z-TPInstance GET

    ;;;

CONDITIONAL PACKAGES

    createDeleteNotificationsPackage PRESENT IF

    "the objectCreation and objectDeletion notifications defined in Recommendation X.721 are supported by an instance of this managed object class",

    attributeValueChangeNotificationPackage PRESENT IF

    "the attributeValueChange notification defined in Recommendation X.721 is supported by an instance of this managed object class",

    stateChangeNotificationPackage PRESENT IF

    "the stateChange notification defined in Recommendation X.721 is supported by an instance of this managed object class",

    characteristicInformationPackage PRESENT IF

    "an instance supports it.",

    protectedPackage PRESENT IF

    "an instance supports it.",

    tmnCommunicationsAlarmInformationPackage PRESENT IF

    "the communicationsAlarm notification (as defined in Recommendation X.721) is supported by this managed object",

    alarmSeverityAssignmentPointerPackage PRESENT IF

    "the tmnCommunicationsAlarmInformationPackage package is present AND the managed object supports configuration of alarm severities";

REGISTERED AS { m3100ObjectClass 13 };

3.4.3   *Trail*

trail MANAGED OBJECT CLASS

    DERIVED FROM   connectivity;

    CHARACTERIZED BY

    trailPackage PACKAGE

        BEHAVIOUR

        trailBehaviour BEHAVIOUR

        DEFINED AS

    "Trail is a class of managed objects in layer networks which is responsible for the integrity of transfer of characteristic information from one or more other layer networks. A trail is composed of two or more Trail Termination Points and one or more connection and associated connection termination points."

    ;;

    ATTRIBUTES

        trailId GET;;;

    CONDITIONAL PACKAGES

        serverConnectionListPackage PRESENT IF "an instance supports it",

        clientConnectionPackage PRESENT IF "an instance supports it";

REGISTERED AS { m3100ObjectClass 14 };

3.5   *Cross Connection Fragment*

Object classes in Cross Connection Fragment are presented in Figure 5/M.3100. The behaviour definition(s) of that object class(es) are specified as follows:

3.5.1   *Cross-Connection*

crossConnection MANAGED OBJECT CLASS
> **DERIVED FROM "Recommendation X.721: 1992":top;**
> **CHARACTERIZED BY**
>> **crossConnectionPackage PACKAGE**
>>> **BEHAVIOUR**
>>>> **crossConnectionBehaviour BEHAVIOUR**
>>>> **DEFINED AS**

"A managed object of this class represents an assignment relationship between the termination point or GTP object listed in the From Termination attribute and the termination point or GTP objects listed in the To Termination attribute of this managed object.

The To Termination attribute will always be non-NULL. The From Termination attribute will only be NULL in the case of point-to-multipoint configurations. If the From Termination attribute has a value of NULL, the assignment relationship is between the termination point object or the GTP object listed in the From Termination attribute of the containing Multipoint Cross-Connection managed object and the termination point object or GTP object listed in the To Termination attribute of this managed object.

A point to point cross-connection can be established between: one of CTP Sink, CTP Bidirectional, TTP Source, TTP Bidirectional, or GTP; and one of CTP Source, CTP Bidirectional, TTP Sink, TTP Bidirectional, or GTP.

In a unidirectional cross-connection, the termination or GTP object pointed to by the From Termination and the termination point or GTP object pointed to by the To Termination attribute (in this object or the containing mpCrossConnection) are related in such a way that traffic can flow between the termination points represented by these managed objects. In a bidirectional cross-connection, information flows in both directions.

If the objects listed in the From Termination and To Termination attributes are GTPs, the nth element of the From Termination GTP is related to the nth element of the To Termination GTP (for every n).

If the fromTermination attribute has a value of NULL, the directionality attribute must have the value "unidirectional".

The total rate of the From Terminations must be equal to the total rate of To Terminations.

The attribute Signal Type describes the signal that is cross-connected. The termination points or GTPs that are cross-connected must have signal types that are compatible.

If an instance of this object class is contained in a multipoint cross-connection and the operational state of the containing multipoint cross-connection is "disabled", the operational state of this object will also be "disabled".

The following are the definitions of the administrative state and the operational state attributes:

Administrative State:

– Unlocked: The Cross-Connection object is administratively unlocked. Traffic is allowed to pass through the connection.

– Locked: No traffic is allowed to pass through the Cross-Connection. The connectivity pointers in the cross-connected termination points is NULL.

Operational State:

– Enabled: The Cross-Connection is performing its normal function.
– Disabled: The Cross-Connection is incapable of performing its normal cross-connection function."

    ;;
    ATTRIBUTES
        crossConnectionId   GET,
        "Recommendation X.721: 1992":administrativeState GET-REPLACE,
        "Recommendation X.721: 1992":operationalState   GET,
        signalType   GET,
        fromTermination   GET,
        toTermination   GET,
        directionality   GET;
        ;;

REGISTERED AS { m3100ObjectClass 15 };

3.5.2   *Fabric*

fabric MANAGED OBJECT CLASS
        DERIVED FROM "Recommendation X.721: 1992":top;
        CHARACTERIZED BY
                fabricPackage PACKAGE
                        BEHAVIOUR
                                fabricBehaviour BEHAVIOUR
                                DEFINED AS
                                "The Fabric object represents the function of managing the establishment and release of cross-connections. It also manages the assignment of termination points to TP Pools and GTPs.

                                Administrative State:

                                – Unlocked: The Fabric is allowed to perform its normal functions. ACTIONS will be accepted to setup or remove cross-connections, to rearrange TP Pools, to add/remove termination points to/from GTPs.
                                – Locked: The Fabric is not allowed to perform its normal functions. No ACTIONS will be accepted. No new cross-connection can be setup or removed, no TP Pool can be rearranged, and no termination points can be added/removed to/from GTPs.

                                Operational State:

                                – Enabled: When the Fabric is in the enabled operational state, it may be fully operational or partially operational (partially operational is indicated by the availability status attribute).
                                – Disabled: The Fabric is incapable of performing its normal function. For instance, the managing system will not be able to
                                  1) setup or remove any cross-connection,
                                  2) rearrange TP Pools, and
                                  3) add/remove termination points to/from GTPs.

                                Availability Status:

                                The supported values for this attribute are:

                                – Degraded: The Fabric is degraded in some respect. For instance, the Fabric cannot perform the function of establishing new cross-connections while it can still accept ACTIONS to re-arrange TP Pools. The Fabric remains available for service (i.e. its operational state is enabled) while it is degraded.
                                – Empty SET."

```
                ;;
                ATTRIBUTES
                        fabricId                                GET,
                        "Recommendation X.721: 1992":administrativeState  GET-REPLACE,
                        "Recommendation X.721: 1992":operationalState      GET,
                        "Recommendation X.721: 1992":availabilityStatus    GET,
                        listOfCharacteristicInfo                GET,
                        supportedByObjectList                   GET-REPLACE  ADD-REMOVE;
                ACTIONS
                        addTpsToGTP,
                        removeTpsFromGTP,
                        addTpsToTpPool,
                        removeTpsFromTpPool,
                        connect,
                        disconnect;
        ;;

REGISTERED AS { m3100ObjectClass 16 };
```

3.5.3 *Group Termination Point*

```
gtp MANAGED OBJECT CLASS
        DERIVED FROM   "Recommendation X.721: 1992": top;
        CHARACTERIZED BY
                gtpPackage PACKAGE
                        BEHAVIOUR
                                gtpBehaviour BEHAVIOUR
                                DEFINED AS
                                "This object class represents a group of termination points treated as a single unit for
                                management purposes such as cross-connections. The signalType attribute describes the
                                composition of the GTP. When a termination point is involved in a GTP, it cannot be cross-
                                connected independently of that GTP."
                        ;;
                        ATTRIBUTES
                                gtpId                         GET,
                                crossConnectionObjectPointer  GET,
                                signalType                    GET,
                                tpsInGtpList                  GET;
                ;;

REGISTERED AS { m3100ObjectClass 17 };
```

3.5.4 *Multipoint Cross-Connection*

```
mpCrossConnection MANAGED OBJECT CLASS
        DERIVED FROM "Recommendation X.721: 1992":top;
        CHARACTERIZED BY
                mpCrossConnectionPackage PACKAGE
                BEHAVIOUR
                        mpCrossConnectionBehaviour BEHAVIOUR
                        DEFINED AS
                        "This class represents an assignment relationship between the termination point or GTP object
                        listed in the From Termination attribute and the termination point or GTP objects listed in the To
                        Termination attributes of the contained crossConnection managed objects.

                        A multipoint cross-connection can be established between one of CTP Sink, CTP Bidirectional,
                        TTP Source, TTP Bidirectional, or GTP; and a set whose members are CTP Source, CTP
                        Bidirectional, TTP Sink, TTP Bidirectional, or GTP.
```

The From Termination attribute will always be non-NULL. The termination point or GTP object pointed to by the From Termination attribute is related to all the termination point or GTP objects pointed to by the To Termination attribute of the contained crossConnection managed objects in such a way that traffic can flow between the termination points represented by these managed objects.

Information flows from the From Termination to the To Termination of the contained cross-connection objects.

If the objects listed in the From Termination attribute and in the To Termination attribute of the contained crossConnection objects are GTPs, the *n*th element of the From Termination GTP is related to the *n*th element of the To Termination GTP (for every n).

The total rate of the From Terminations must be equal to the total rate of To Terminations in each contained crossConnection object.

The attribute Signal Type describes the signal that is cross-connected. The termination points or GTPs that are cross-connected must have signal types that are compatible.

The following are the definitions of the administrative state and the operational state attributes:

Administrative State:

– Unlocked: The mpCrossConnection object is administratively unlocked. It allows traffic to pass through each contained connection depending on its administrative state.
– Locked: No traffic is allowed to pass through the Cross-Connection between the cross-connected termination points. The effect of this value overrides the effect of the administrative state of each contained cross-connection.

Operational State:

The operational state of a Multipoint Cross-Connection object reflects the overall health of the cross-connection including all the cross-connection objects contained in the Multipoint Cross-Connection.

– Enabled: The Cross-Connection is performing its normal function. Note that some (but not all) of the cross-connection objects contained in the Multipoint Cross-Connection may be disabled.
– Disabled: The Cross-Connection is incapable of performing its normal cross-connection function. All the cross-connection objects contained in the Multipoint Cross-Connection are disabled.

Availability Status:

The supported values for this attribute are:

– In test
– Degraded: The Multipoint Cross-Connection is degraded in some respect. For instance, if one or more (but not all) cross-connection objects contained in the Multipoint Cross-Connection are disabled, the Multipoint Cross-Connection will be considered as degraded. The Multipoint Cross-Connection remains available for service (i.e. its operational state is enabled) while it is degraded.
– Empty SET."

```
;;
ATTRIBUTES
        mpCrossConnectionId                          GET,
        "Recommendation X.721: 1992":administrativeState        GET-REPLACE,
        "Recommendation X.721: 1992":operationalState    GET,
        "Recommendation X.721: 1992":availabilityStatus   GET,
        signalType                                   GET,
        fromTermination                              GET;
;;
REGISTERED AS { m3100ObjectClass 18 };
```

3.5.5   *Named Cross-Connection*

namedCrossConnection MANAGED OBJECT CLASS
      DERIVED FROM crossConnection;
      CHARACTERIZED BY
          namedCrossConnectionPackage;

REGISTERED AS { m3100ObjectClass 19 };

3.5.6   *Named Multi-Point Cross-Connection*

namedMpCrossConnection MANAGED OBJECT CLASS
      DERIVED FROM mpCrossConnection;
      CHARACTERIZED BY
          namedCrossConnectionPackage;

REGISTERED AS { m3100ObjectClass 20 };

3.5.7   *TP Pool*

tpPool MANAGED OBJECT CLASS
      DERIVED FROM "Recommendation X.721: 1992":top;
      CHARACTERIZED BY
          tpPoolPackage PACKAGE
              BEHAVIOUR
                  tpPoolBehaviour BEHAVIOUR
                      DEFINED AS

                      "The tpPool object represents a set of termination points or GTPs that are used for some management purpose, such as routing. A termination point that is a member of a GTP cannot be a member of a tpPool independent of the remainder of the GTP."

              ;;
              ATTRIBUTES

tpPoolId	**GET,**
tpsInTpPoolList	**GET,**
totalTpCount	**GET,**
connectedTpCount	**GET,**
idleTpCount	**GET;**

              ;;

REGISTERED AS { m3100ObjectClass 21 };

3.6   *Functional Area Fragment*

Object classes in functional area fragment are presented in Figure 6/M.3100. The references/definitions of the object classes are given below:

3.6.1   *Alarm Record*

The Alarm Record object class is defined in Recommendation X.721 [5].

3.6.2   *Alarm Severity Assignment Profile*

alarmSeverityAssignmentProfile MANAGED OBJECT CLASS
      DERIVED FROM   "Recommendation X.721: 1992":top;
      CHARACTERIZED BY
      alarmSeverityAssignmentProfilePackage PACKAGE
          BEHAVIOUR
          alarmSeverityAssignmentProfileBehaviour BEHAVIOUR
              DEFINED AS

"The alarm severity assignment profile object class is a class of management support object that specifies the alarm severity assignment for managed objects. Instances of this object are referenced by the alarmSeverityAssignmentProfilePointer attribute in the managed objects."

;;

 ATTRIBUTES

  alarmSeverityAssignmentProfileId GET,

  alarmSeverityAssignmentList GET-REPLACE ADD-REMOVE ;

;;

 CONDITIONAL PACKAGES

  objectManagementNotificationsPackage PRESENT IF "an instance supports it";

REGISTERED AS { m3100ObjectClass 22 };

3.6.3 *Attribute Value Change Record*

The Attribute Value Change Record object class is defined in Recommendation X.721 [5].

3.6.4 *Current Alarm Summary Control*

The Current Alarm Summary Control object class is defined in Recommendation Q.821 [7].

3.6.5 *Discriminator*

The Discriminator object class is defined in Recommendation X.721 [5].

3.6.6 *Event Forwarding Discriminator*

The Event Forwarding Discriminator object class is defined in Recommendation X.721 [5].

3.6.7 *Event Log Record*

The Event Log Record object class is defined in Recommendation X.721 [5].

3.6.8 *Log*

The Log object class is defined in Recommendation X.721 [5].

3.6.9 *Log Record*

The Log Record object class is defined in Recommendation X.721 [5].

3.6.10 *Management Operations Schedule*

The Management Operations Schedule object class is defined in Recommendation Q.821 [7].

3.6.11 *Object Creation Record*

The Object Creation Record object class is defined in Recommendation X.721 [5].

3.6.12 *Object Deletion Record*

The Object Deletion Record object class is defined in Recommendation X.721 [5].

3.6.13 *State Change Record*

The State Change Record object class is defined in Recommendation X.721 [5].

4 Packages

4.1 *Administrative Operational States*

administrativeOperationalStatesPackage PACKAGE
    ATTRIBUTES
        "Recommendation X.721: 1992":administrativeState GET-REPLACE,
        "Recommendation X.721: 1992":operationalState   GET;

REGISTERED AS { m3100Package 1 };

4.2 *Affected Object List*

affectedObjectListPackage   PACKAGE
    ATTRIBUTES
        affectedObjectList GET;

REGISTERED AS { m3100Package 2 };

4.3 *Alarm Severity Assignment Pointer*

alarmSeverityAssignmentPointerPackage   PACKAGE
    BEHAVIOUR
        alarmSeverityAssignmentPointerPackageBehaviour BEHAVIOUR
        DEFINED AS
        "If the alarm severity assignment profile pointer is NULL, then one of the following two choices applies
         when reporting alarms:
        a) agent assigns the severity; or
        b) the value "indeterminate" is used."
        ;;
    ATTRIBUTES
        alarmSeverityAssignmentProfilePointer GET-REPLACE ;

REGISTERED AS { m3100Package 3 };

4.4 *Attribute Value Change Notification*

attributeValueChangeNotificationPackage PACKAGE
    NOTIFICATIONS
        "Recommendation X.721: 1992":attributeValueChange;

REGISTERED AS { m3100Package 4 };

4.5 *Audible Visual Local Alarm*

audibleVisualLocalAlarmPackage PACKAGE
    ACTIONS
        allowAudibleVisualLocalAlarm,
        inhibitAudibleVisualLocalAlarm;

REGISTERED AS { m3100Package 5 };

4.6 *Channel Number*

channelNumberPackage **PACKAGE**
  ATTRIBUTES
    channelNumber **GET;**

REGISTERED AS { m3100Package 6 };

4.7 *Characteristic Information*

characteristicInformationPackage **PACKAGE**
  ATTRIBUTES
    characteristicInformation GET;

REGISTERED AS { m3100Package 7 };

4.8 *Client Connection*

clientConnectionPackage **PACKAGE**
  ATTRIBUTES
    clientConnection GET;

REGISTERED AS { m3100Package 8 };

4.9 *Client Trail*

clientTrailPackage **PACKAGE**
  ATTRIBUTES
    clientTrail GET;

REGISTERED AS { m3100Package 9 };

4.10 *Create Delete Notifications*

createDeleteNotificationsPackage PACKAGE
  NOTIFICATIONS
    "Recommendation X.721: 1992":objectCreation,
    "Recommendation X.721: 1992":objectDeletion;

REGISTERED AS { m3100Package 10 };

4.11 *Cross Connection Pointer*

crossConnectionPointerPackage PACKAGE
  ATTRIBUTES
  crossConnectionObjectPointer GET;

REGISTERED AS { m3100Package 11 };

4.12 *CTP Instance*

ctpInstancePackage **PACKAGE**
  ATTRIBUTES
    cTPId **GET;**

REGISTERED AS { m3100Package 12 };

4.13   *Current Problem List*

currentProblemListPackage PACKAGE
    ATTRIBUTES
        currentProblemList GET;

REGISTERED AS { m3100Package 13 };

4.14   *Environmental Alarm*

environmentalAlarmPackage PACKAGE
    NOTIFICATIONS
        "Recommendation X.721: 1992":environmentalAlarm;

REGISTERED AS { m3100Package 14 };

4.15   *Equipment Equipment Alarm*

equipmentEquipmentAlarmPackage PACKAGE
    ATTRIBUTES
        alarmStatus   GET;
    NOTIFICATIONS
        "Recommendation X.721: 1992":equipmentAlarm;

REGISTERED AS { m3100Package 15 };

4.16   *External Time*

externalTimePackage PACKAGE
    ATTRIBUTES
        externalTime   GET-REPLACE;

REGISTERED AS { m3100Package 16 };

4.17   *Location Name*

locationNamePackage PACKAGE
    ATTRIBUTES
        locationName GET-REPLACE;

REGISTERED AS { m3100Package 17 };

4.18   *Named Cross-Connection*

namedCrossConnectionPackage PACKAGE
    ATTRIBUTES
        redline   GET-REPLACE,
        crossConnectionName   GET-REPLACE;;
-- *The above package is not registered because it is used as a mandatory package in the current*
-- *document.*

4.19   *Network Level*

networkLevelPackage   PACKAGE
    BEHAVIOUR
        networkLevelPackageBehaviour   BEHAVIOUR
        DEFINED AS
        "The network level pointer identifies a network level object. The value of the network level pointer shall only be modified by the managing system."
        ;;
    ATTRIBUTES
        networkLevelPointer   GET-REPLACE;

REGISTERED AS { m3100Package 18 };

4.20 *Operational State*

operationalStatePackage PACKAGE
    ATTRIBUTES
        "Recommendation X.721: 1992":operationalState GET;

REGISTERED AS { m3100Package 19 };

4.21 *Object Management Notifications*

objectManagementNotificationsPackage   PACKAGE
    NOTIFICATIONS
        "Recommendation X.721: 1992":objectCreation,
        "Recommendation X.721: 1992":objectDeletion,
        "Recommendation X.721: 1992":attributeValueChange;

REGISTERED AS { m3100Package 20 };

4.22 *Processing Error Alarm*

processingErrorAlarmPackage PACKAGE
    NOTIFICATIONS
        "Recommendation X.721: 1992":processingErrorAlarm;

REGISTERED AS { m3100Package 21 };

4.23 *Protected*

protectedPackage   PACKAGE
    ATTRIBUTES
        protected   GET;

REGISTERED AS { m3100Package 22 };

4.24 *Reset Audible Alarm*

resetAudibleAlarmPackage PACKAGE
    ACTIONS
        "Recommendation Q.821: 1992":resetAudibleAlarm;

REGISTERED AS { m3100Package 23 };

4.25 *Server Connection List*

serverConnectionListPackage   PACKAGE
    ATTRIBUTES
        serverConnectionList   GET;

REGISTERED AS { m3100Package 24 };

4.26 *Server Trail List*

serverTrailListPackage   PACKAGE
    ATTRIBUTES
        serverTrailList   GET;

REGISTERED AS { m3100Package 25 };

4.27   *Software Processing Error Alarm*

softwareProcessingErrorAlarmPackage PACKAGE
    ATTRIBUTES
        alarmStatus   GET;
    NOTIFICATIONS
        "Recommendation X.721: 1992":processingErrorAlarm;

REGISTERED AS { m3100Package 26 };

4.28   *Supportable Client List*

supportableClientListPackage   PACKAGE
    ATTRIBUTES
        supportableClientList   GET;

REGISTERED AS { m3100Package 27 };

4.29   *State Change Notification*

stateChangeNotificationPackage PACKAGE
    NOTIFICATIONS
        "Recommendation X.721: 1992":stateChange;

REGISTERED AS { m3100Package 28 };

4.30   *System Timing Source*

systemTimingSourcePackage   PACKAGE
    ATTRIBUTES
        systemTimingSource   GET-REPLACE;

REGISTERED AS { m3100Package 29 };

4.31   *TMN Communications Alarm Information*

tmnCommunicationsAlarmInformationPackage   PACKAGE
    BEHAVIOUR
        tmnCommunicationsAlarmInformationBehaviour;
    ATTRIBUTES
        alarmStatus   GET,
        currentProblemList GET;
    NOTIFICATIONS
        "Recommendation X.721: 1992":communicationsAlarm
            "Recommendation Q.821:1992":logRecordIdParameter
            "Recommendation Q.821:1992":correlatedRecordNameParameter
            "Recommendation Q.821:1992":suspectObjectListParameter;

REGISTERED AS { m3100Package 30 };

tmnCommunicationsAlarmInformationBehaviour   BEHAVIOUR

    -- The following behaviour text is taken directly from Recommendation Q.821, § 5.3.1.1
    DEFINED AS
    "An alarm report which contains a Perceived Severity parameter with a value of "cleared" and a Correlated Notifications parameter shall only indicate the clearing of those alarms whose Notification Identifiers are included in the set of Correlated Notifications. An alarm report which contains a Perceived Severity parameter with a value of "cleared", but no Correlated Notifications parameter, shall indicate the clearing of alarms based on the value of the Alarm Type, Probable Cause, and Specific Problems parameters.

    The parameters that are associated with the communications alarm, if present, are placed in individual elements of the SET OF ManagementExtension in the additionalInformation field of the notification.";

4.32   *TTP Instance*

ttpInstancePackage   PACKAGE

ATTRIBUTES

tTPId GET;

REGISTERED AS { m3100Package 31 };

4.33   *User Label*

userLabelPackage PACKAGE

ATTRIBUTES

userLabel GET-REPLACE;

REGISTERED AS { m3100Package 32 };

4.34   *Vendor Name*

vendorNamePackage PACKAGE

ATTRIBUTES

vendorName GET-REPLACE;

REGISTERED AS { m3100Package 33 };

4.35   *Version*

versionPackage PACKAGE

ATTRIBUTES

version GET-REPLACE;

REGISTERED AS { m3100Package 34 };

5   Attributes

5.1   *A-Termination Point Instance*

a-TPInstance ATTRIBUTE

WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.RelatedObjectInstance;

MATCHES FOR EQUALITY;

BEHAVIOUR

aTPInstanceBehaviour   BEHAVIOUR

DEFINED AS

"The A-Termination Point Instance attribute type identifies one of the two termination points of an instance of the connectivity object class or one of its sub-classes.";;

REGISTERED AS { m3100Attribute 1 };

5.2   *Administrative State*

The Administrative State attribute is defined in Recommendation X.721 [5].

5.3 *Affected Object List*

affectedObjectList ATTRIBUTE

 WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.ObjectList;

 MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;

 BEHAVIOUR

  affectedObjectListBehaviour BEHAVIOUR

  DEFINED AS

  "The Affected Object List attribute type specifies the object instances which can be directly affected by a change in state or deletion of a given managed object. The attribute does not force internal details to be specified, but only the necessary level of detail required for management.";;

REGISTERED AS { m3100Attribute 2 };

5.4 *Alarm Severity Assignment List*

alarmSeverityAssignmentList ATTRIBUTE

 WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.AlarmSeverityAssignmentList;

 BEHAVIOUR

  alarmSeverityAssignmentListBehaviourBEHAVIOUR

  DEFINED AS

  "The Alarm Severity Assignment List is an attribute type whose value provides a listing of all abnormal conditions that may exist in instances of an object class, and shows the assigned alarm severity information (minor, major etc.) for each condition.";;

REGISTERED AS { m3100Attribute 3 };

5.5 *Alarm Severity Assignment Profile Id*

alarmSeverityAssignmentProfileId ATTRIBUTE

 WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.NameType;

 MATCHES FOR EQUALITY;

 BEHAVIOUR

  alarmSeverityAssignmentProfileIdBehaviour BEHAVIOUR

  DEFINED AS

  "The Alarm Severity AssignmentProfile Id is an attribute type whose distinguished value can be used as an RDN when naming an instance of the Alarm SeverityAssignment Profile object class.";;

REGISTERED AS { m3100Attribute 4 };

5.6 *Alarm Severity Assignment Profile Pointer*

alarmSeverityAssignmentProfilePointer ATTRIBUTE

 WITH ATTRIBUTE SYNTAX   ASN1DefinedTypesModule.PointerOrNull;

 MATCHES FOR EQUALITY;

 BEHAVIOUR

  alarmSeverityAssignmentProfilePointerBehaviour BEHAVIOUR

  DEFINED AS

  "This attribute identifies an Alarm Severity Assignment Profile object.";;

REGISTERED AS { m3100Attribute 5 };

5.7   *Alarm Status*

alarmStatus ATTRIBUTE
 WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.AlarmStatus;
 MATCHES FOR EQUALITY;
 BEHAVIOUR
  alarmStatusBehaviour BEHAVIOUR
  DEFINED AS
  "The Alarm Status attribute type indicates the occurrence of an abnormal condition relating to an object. This attribute may also function as a summary indicator of alarm conditions associated with a specific resource. It is used to indicate the existence of an alarm condition, a pending alarm condition such as threshold situations, or (when used as a summary indicator) the highest severity of active alarm conditions. When used as a summary indicator, the order of severity (from highest to lowest) is:

  activeReportable-Critical

  activeReportable-Major

  activeReportable-Minor

  activeReportable-Indeterminate

  activeReportable-Warning

  activePending

  cleared";;

REGISTERED AS { m3100Attribute 6 };

5.8   *Channel Number*

channelNumber ATTRIBUTE
 WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.ChannelNumber;
 MATCHES FOR EQUALITY, ORDERING;

REGISTERED AS { m3100Attribute 7 };

5.9   *Characteristic Information*

characteristicInformation ATTRIBUTE
 WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.CharacteristicInformation;
 MATCHES FOR EQUALITY;
 BEHAVIOUR
  characteristicInformationBehaviour BEHAVIOUR
  DEFINED AS
  "The value of this attribute is used to verify the connectability of instances of the termination point sub-classes";;

REGISTERED AS { m3100Attribute 8 };

5.10   *Client Connection*

clientConnection ATTRIBUTE
 WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.ObjectInstance;
 MATCHES FOR EQUALITY;
 BEHAVIOUR
  clientConnectionBehaviour BEHAVIOUR
  DEFINED AS
  "The value of this attribute identifies the client object instance that is served by a trail at a higher order network layer.";;

REGISTERED AS { m3100Attribute 9 };

5.11   *Client Trail*

clientTrail ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.ObjectInstance;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
        clientTrailBehaviour   BEHAVIOUR
        DEFINED AS
        "The value of this attribute identifies the trail object instance in the same network layer as the connection served by a connection object.";;

REGISTERED AS { m3100Attribute 10 };

5.12   *Connected Termination Point Count*

connectedTpCount ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   ASN1DefinedTypesModule.Count;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR
        connectedTpCountBehaviour BEHAVIOUR
        DEFINED AS
        "This attribute indicates the total number of termination points associated with a tpPool that have been connected.";;

REGISTERED AS { m3100Attribute 11 };

5.13   *Connection Id*

connectionId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.NameType;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
        connectionIdBehaviour   BEHAVIOUR
        DEFINED AS
        "The Connection Id is an attribute type whose distinguished value can be used as an RDN when naming an instance of the Connection object class.";;

REGISTERED AS { m3100Attribute 12 };

5.14   *Connection Termination Point Id*

cTPId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.NameType;
    MATCHES FOR EQUALITY;

REGISTERED AS { m3100Attribute 13 };

5.15   *Cross-Connection Id*

crossConnectionId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   ASN1DefinedTypesModule.NameType;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
        crossConnectionIdBehaviour BEHAVIOUR
        DEFINED AS
        "The Cross-Connection Id is an attribute type whose distinguished value can be used as an RDN when naming an instance of the crossConnection object class.";;

REGISTERED AS { m3100Attribute 14 };

5.16   *Cross-Connection Name*

crossConnectionName ATTRIBUTE

    WITH ATTRIBUTE SYNTAX   ASN1DefinedTypesModule.CrossConnectionName;

    MATCHES FOR   EQUALITY, SUBSTRINGS;

    BEHAVIOUR

        crossConnectionNameBehaviour BEHAVIOUR

        DEFINED AS

        "This attribute is a descriptive name for a cross-connection or multipoint cross-connection managed object.";;

REGISTERED AS { m3100Attribute 15 };

5.17   *Cross-Connection Object Pointer*

crossConnectionObjectPointer ATTRIBUTE

        WITH ATTRIBUTE SYNTAX

            ASN1DefinedTypesModule.CrossConnectionObjectPointer;

        MATCHES FOR EQUALITY;

        BEHAVIOUR

            crossConnectionObjectPointerBehaviour BEHAVIOUR

            DEFINED AS

        "This attribute points to a managed object such as a Cross-connection, a GTP or a Fabric.

        When a termination point is neither connected nor reserved for connection, its

        crossConnectionObjectPointer points to the Fabric object responsible for its connection.";;

REGISTERED AS { m3100Attribute 16 };

5.18   *Current Problem List*

currentProblemList ATTRIBUTE

    WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.CurrentProblemList;

    BEHAVIOUR

        currentProblemListBehaviour   BEHAVIOUR

        DEFINED AS

        "The Current Problem List attribute type identifies the current existing problems, with severity, associated with the managed object.";;

REGISTERED AS { m3100Attribute 17 };

5.19   *Directionality*

directionality ATTRIBUTE

    WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.Directionality;

    MATCHES FOR EQUALITY;

    BEHAVIOUR

        directionalityBehaviour   BEHAVIOUR

        DEFINED AS

        "The Directionality attribute type specifies whether the associated managed object is uni- or bi-directional.";;

REGISTERED AS { m3100Attribute 18 };

5.20   *Downstream Connectivity Pointer*

downstreamConnectivityPointer ATTRIBUTE

    WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.DownstreamConnectivityPointer;

    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;

    BEHAVIOUR

        downstreamConnectivityPointerBehaviour BEHAVIOUR

        DEFINED AS

        "The matching for equality is applicable for all choices of the syntax. The set operations are permitted only when the choice of the syntax correspond to either broadcast or concatenated broadcast.";;

REGISTERED AS { m3100Attribute 19 };

5.21   *Equipment Id*

equipmentId ATTRIBUTE

    WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.NameType;

    MATCHES FOR EQUALITY;

    BEHAVIOUR

        equipmentIdBehaviour   BEHAVIOUR

        DEFINED AS

        "The Equipment Id is an attribute type whose distinguished value can be used as an RDN when naming an instance of the Equipment object class.";;

REGISTERED AS { m3100Attribute 20 };

5.22   *External Time*

externalTime ATTRIBUTE

    WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.ExternalTime;

    MATCHES FOR EQUALITY;

    BEHAVIOUR

        externalTimeBehaviour   BEHAVIOUR

        DEFINED AS

        "The External time attribute provides time-of-day system time. The attribute functions as a reference for all time stamp activities in the managed element.";;

REGISTERED AS { m3100Attribute 21 };

5.23   *Fabric Id*

fabricId ATTRIBUTE

    WITH ATTRIBUTE SYNTAX   ASN1DefinedTypesModule.NameType;

    MATCHES FOR EQUALITY;

    BEHAVIOUR

        fabricIdBehaviour BEHAVIOUR

        DEFINED AS

        "The Fabric Id is an attribute type whose distinguished value can be used as an RDN when naming an instance of the Fabric object class.";;

REGISTERED AS { m3100Attribute 22 };

5.24   *From Termination*

fromTermination ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   ASN1DefinedTypesModule.PointerOrNull;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
    fromTerminationBehaviour BEHAVIOUR
    DEFINED AS
        "This attribute identifies a TTP (source or bidirectional), a CTP (sink or bidirectional) or a GTP composed of members of one of these categories.";;

REGISTERED AS { m3100Attribute 23 };

5.25   *Group Termination Point Id*

gtpId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   ASN1DefinedTypesModule.NameType;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
        gtpIdBehaviour BEHAVIOUR
        DEFINED AS
        "The gtp Id is an attribute type whose distinguished value can be used as an RDN when naming an instance of the gtp object class.";;

REGISTERED AS { m3100Attribute 24 };

5.26   *Idle TP Count*

idleTpCount ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.Count;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR
        idleTpCountBehaviour BEHAVIOUR
        DEFINED AS
        "This attribute indicates the total number of termination points associated with a tpPool that are in an operational state of enabled and that are available for Cross-Connection.";;

REGISTERED AS { m3100Attribute 25 };

5.27   *List of Characteristic Info*

listOfCharacteristicInfo ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.ListOfCharacteristicInformation;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
        listOfCharacteristicInfoBehaviour BEHAVIOUR
        DEFINED AS
        "This attribute lists the characteristic information types that can be cross-connected by a Fabric.";;

REGISTERED AS { m3100Attribute 26 };

5.28   *Location Name*

locationName ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.LocationName;
    MATCHES FOR EQUALITY, SUBSTRINGS;
    BEHAVIOUR
        locationNameBehaviour   BEHAVIOUR
        DEFINED AS
        "The Location Name attribute type identifies a location.";;

REGISTERED AS { m3100Attribute 27 };

5.29   *Managed Element Id*

managedElementId ATTRIBUTE

 WITH ATTRIBUTE SYNTAX   ASN1DefinedTypesModule.NameType;

 MATCHES FOR EQUALITY;

 BEHAVIOUR

  managedElementIdBehaviour   BEHAVIOUR

  DEFINED AS

  "The Managed Element Id is an attribute type whose distinguished value can be used as an RDN when naming an instance of the Managed Element object class.";;

REGISTERED AS { m3100Attribute 28 };

5.30   *Multi-Point Cross-Connection Id*

mpCrossConnectionId ATTRIBUTE

 WITH ATTRIBUTE SYNTAX   ASN1DefinedTypesModule.NameType;

 MATCHES FOR EQUALITY;

 BEHAVIOUR

  mpCrossConnectionIdBehaviour   BEHAVIOUR

  DEFINED AS

  "The mp Cross-Connection Id is an attribute type whose distinguished value can be used as an RDN when naming an instance of the mpCrossConnection object class.";;

REGISTERED AS { m3100Attribute 29 };

5.31   *Network Id*

networkId ATTRIBUTE

 WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.NameType;

 MATCHES FOR EQUALITY;

 BEHAVIOUR

  networkIdBehaviour   BEHAVIOUR

  DEFINED AS

  "The Network Id is an attribute type whose distinguished value can be used as an RDN when naming an instance of the Network object class.";;

REGISTERED AS { m3100Attribute 30 };

5.32   *Network Level Pointer*

networkLevelPointer ATTRIBUTE

 WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.ObjectInstance;

 MATCHES FOR EQUALITY;

REGISTERED AS { m3100Attribute 31 };

5.33   *Operational State*

 The Operational State attribute is defined in Recommendation X.721 [5].

5.34   *Protected*

protected ATTRIBUTE

  WITH ATTRIBUTE SYNTAX   ASN1DefinedTypesModule.Boolean;

  MATCHES FOR EQUALITY;

  BEHAVIOUR

    protectedBehaviour BEHAVIOUR

    DEFINED AS

    "This attribute identifies whether the associated managed object is protected or not. The value TRUE implies it is protected.";;

REGISTERED AS { m3100Attribute 32 };

5.35   *Redline*

redline ATTRIBUTE

  WITH ATTRIBUTE SYNTAX   ASN1DefinedTypesModule.Boolean;

  MATCHES FOR EQUALITY;

  BEHAVIOUR

    redlineBehaviour BEHAVIOUR

    DEFINED AS

    "This attribute identifies whether the associated managed object is red lined, e.g. identified as being part of a sensitive circuit.";;

REGISTERED AS { m3100Attribute 33 };

5.36   *Replaceable*

replaceable ATTRIBUTE

  WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.Replaceable;

  MATCHES FOR EQUALITY;

  BEHAVIOUR

    replaceableBehaviour   BEHAVIOUR

    DEFINED AS

    "The Replaceable attribute type indicates whether the associated managed object is replaceable or non-replaceable.";;

REGISTERED AS { m3100Attribute 34 };

5.37   *Server Connection List*

serverConnectionList ATTRIBUTE

  WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.SequenceOfObjectInstance;

  MATCHES FOR EQUALITY;

  BEHAVIOUR

    serverConnectionListBehaviour   BEHAVIOUR

    DEFINED AS

    "The value of this attribute identifies one or more connection objects within the same network layer as the trail that are connected in series to constitute the trail.";;

REGISTERED AS { m3100Attribute 35 };

5.38   *Server Trail List*

serverTrailList ATTRIBUTE
>**WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.ObjectList;**
>**MATCHES FOR EQUALITY;**
>**BEHAVIOUR**
>>**serverTrailListBehaviour   BEHAVIOUR**
>>**DEFINED AS**
>>"The value of this attribute identifies the trail objects (in most cases one) in a lower order network layer which may be used in parallel to serve a conection object.";;

REGISTERED AS { m3100Attribute 36 };

5.39   *Signal Type*

signalType ATTRIBUTE
>**WITH ATTRIBUTE SYNTAX   ASN1DefinedTypesModule.SignalType;**
>**MATCHES FOR EQUALITY;**
>**BEHAVIOUR**
>>**signalTypeBehaviour BEHAVIOUR**
>>**DEFINED AS**
>>"This attribute uniquely identifies the signal type of a cross-connection, TP Pool or GTP. The signal type can either be simple, bundle, or complex. If the signal type is simple, it consists of a single type of characteristic information. If the signal type is bundle, it is made up of a number of signal types all of the same characteristic information. If the signal type is complex, it consists of a sequence of bundle signal type. The order in the complex signal type represents the actual composition of the signal.";;

REGISTERED AS { m3100Attribute 37 };

5.40   *Software Id*

softwareId ATTRIBUTE
>**WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.NameType;**
>**MATCHES FOR EQUALITY;**
>**BEHAVIOUR**
>>**softwareIdBehaviour   BEHAVIOUR**
>>**DEFINED AS**
>>"The Software Id is an attribute type whose distinguished value can be used as an RDN when naming an instance of the Software object class.";;

REGISTERED AS { m3100Attribute 38 };

5.41   *Supportable Client List*

supportableClientList ATTRIBUTE
>**WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.SupportableClientList;**
>**MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;**
>**BEHAVIOUR**
>>**supportableClientListBehaviour BEHAVIOUR**
>>**DEFINED AS**
>>"The value of this attribute is the list of object classes representing the clients which the particular managed object is capable of supporting. This may be a subset of the client layers identified in Recommendation G.803 by the particular server layer managed object.";;

REGISTERED AS { m3100Attribute 39 };

5.42   *Supported By Object List*

supportedByObjectList ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.ObjectList;
  MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
  BEHAVIOUR
    supportedByListBehaviour   **BEHAVIOUR**
    DEFINED AS
    "The Supported By List is an attribute type whose value identifies a set of object instances which are capable of directly affecting a given managed object. The object instances include both physical and logical objects. This attribute does not force internal details to be specified, but only the necessary level of detail required for management. If the object instances supporting the managed object are unknown to that object, then this attribute is an empty set.";;

REGISTERED AS { m3100Attribute 40 };

5.43   *System Timing Source*

systemTimingSource ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.SystemTimingSource;
  MATCHES FOR EQUALITY;
  BEHAVIOUR
    systemTimingSourceBehaviour **BEHAVIOUR**
    DEFINED AS
    "The System Timing Source attribute is used to specify the primary and secondary managed element timing source for synchronization.";;

REGISTERED AS { m3100Attribute 41 };

5.44   *System Title*

This attribute is defined in Recommendation X.721 [5].

5.45   *Total TP Count*

totalTpCount ATTRIBUTE
  WITH ATTRIBUTE SYNTAX   ASN1DefinedTypesModule.Count;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR
    totalTpCountBehaviour **BEHAVIOUR**
    DEFINED AS
    "This attribute indicates the total number of termination points associated with a tpPool.";;

REGISTERED AS { m3100Attribute 42 };

5.46   *To Termination*

toTermination ATTRIBUTE
  WITH ATTRIBUTE SYNTAX   ASN1DefinedTypesModule.Pointer;
  MATCHES FOR EQUALITY;
  BEHAVIOUR
    toTerminationBehaviour BEHAVIOUR
    DEFINED AS
    "This attribute identifies a CTP (source or bidirectional), a TTP (sink or bidirectional) or a GTP composed of members of one of these categories.";;

REGISTERED AS { m3100Attribute 43 };

5.47   *TP Pool Id*

tpPoolId ATTRIBUTE

    WITH ATTRIBUTE SYNTAX   ASN1DefinedTypesModule.NameType;
    MATCHES FOR EQUALITY;

    BEHAVIOUR

        tpPoolIdBehaviour BEHAVIOUR

        DEFINED AS

        "The TP Pool Id is an attribute type whose distinguished value can be used as an RDN when naming an instance of the tpPool object class.";;

REGISTERED AS { m3100Attribute 44 };

5.48   *TPs In GTP List*

tpsInGtpList ATTRIBUTE

    WITH ATTRIBUTE SYNTAX   ASN1DefinedTypesModule.TpsInGtpList;

    MATCHES FOR EQUALITY;

    BEHAVIOUR

        tpsInGtpListBehaviour BEHAVIOUR

        DEFINED AS

        "This attribute list the termination points that are represented by a GTP.";;

REGISTERED AS { m3100Attribute 45 };

5.49   *TPs In TP Pool List*

tpsInTpPoolList ATTRIBUTE

    WITH ATTRIBUTE SYNTAX   ASN1DefinedTypesModule.ListOfTPs;

    MATCHES FOR EQUALITY;

    BEHAVIOUR

        tpsInTpPoolListBehaviour BEHAVIOUR

        DEFINED AS

        "This attribute list the termination points that are represented by a TP Pool.";;

REGISTERED AS { m3100Attribute 46 };

5.50   *Trail Id*

trailId ATTRIBUTE

        WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.NameType;

        MATCHES FOR EQUALITY;

        BEHAVIOUR

        trailIdBehaviour   BEHAVIOUR

        DEFINED AS

        "The Trail Id is an attribute type whose distinguished value can be used as an RDN when naming an instance of the Trail object class.";;

REGISTERED AS { m3100Attribute 47 };

5.51 *Trail Termination Point Id*

tTPId ATTRIBUTE

      WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.NameType;

      MATCHES FOR EQUALITY;

      BEHAVIOUR

      ttpIdBehaviour   BEHAVIOUR

      DEFINED AS

      "The Trail Termination Point Id is an attribute type whose distinguished value can be used as an RDN when naming an instance of the Trail Termination Point object class.";;

REGISTERED AS { m3100Attribute 48 };

5.52 *Upstream Connectivity Pointer*

upstreamConnectivityPointer ATTRIBUTE

      WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.ConnectivityPointer;

      MATCHES FOR EQUALITY;

      BEHAVIOUR

      upstreamConnectivityPointerBehaviour BEHAVIOUR

      DEFINED AS

      "The matching for equality is applicable for all the choices of the syntax.";;

REGISTERED AS { m3100Attribute 49 };

5.53 *Usage State*

The Usage State attribute is defined in Recommendation X.721 [5].

5.54 *User Label*

userLabel ATTRIBUTE

      WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.UserLabel;

      MATCHES FOR EQUALITY, SUBSTRINGS;

      BEHAVIOUR

      userLabelBehaviour   BEHAVIOUR

      DEFINED AS

      "The User Label attribute type assigns a user friendly name to the associated object.";;

REGISTERED AS { m3100Attribute 50 };

5.55 *Vendor Name*

vendorName ATTRIBUTE

      WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.VendorName;

      MATCHES FOR EQUALITY, SUBSTRINGS;

      BEHAVIOUR

      vendorNameBehaviour   BEHAVIOUR

      DEFINED AS

      "The Vendor Name attribute type identifies the vendor of the associated managed object.";;

REGISTERED AS { m3100Attribute 51 };

5.56   *Version*

version ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.Version;
    MATCHES FOR EQUALITY, SUBSTRINGS;
    BEHAVIOUR
    versionBehaviour   BEHAVIOUR
    DEFINED AS
    "The Version attribute type identifies the version of the associated managed object.";;

REGISTERED AS { m3100Attribute 52 };

5.57   *Z-Termination Point Instance*

z-TPInstance ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.RelatedObjectInstance;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
    z-TPInstanceBehaviour   BEHAVIOUR
    DEFINED AS
    "The Z-Termination Point Instance attribute type identifies one of the two termination points of an instance of the connectivity object class.";;

REGISTERED AS { m3100Attribute 55 };

6   Name Bindings

    The naming hierarchy of the model is presented in Figure 8/M.3100. The arrows are used to point from the subordinate object classes to the superior object classes. Each arrow represents a name binding defined in this Recommendation.

6.1   *Alarm Record*

    The name binding for alarm record is the same as that defined for log record in Recommendation X.721.

6.2   *Alarm Severity Assignment Profile*

alarmSeverityAssignment-managedElement NAME BINDING
  SUBORDINATE OBJECT CLASS alarmSeverityAssignmentProfile AND SUBCLASSES;
  NAMED BY
  SUPERIOR OBJECT CLASS   managedElement AND SUBCLASSES;
  WITH ATTRIBUTE   alarmSeverityAssignmentProfileId;
  CREATE
    WITH-REFERENCE-OBJECT,
    WITH-AUTOMATIC-INSTANCE-NAMING;
 **DELETE**
    ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1 };

6.3   *Connection*

connection-network NAME BINDING
  SUBORDINATE OBJECT CLASS connection;
  NAMED BY
  SUPERIOR OBJECT CLASS   network;
  WITH ATTRIBUTE   connectionId;
  CREATE;
  DELETE;

REGISTERED AS { m3100NameBinding 2 };
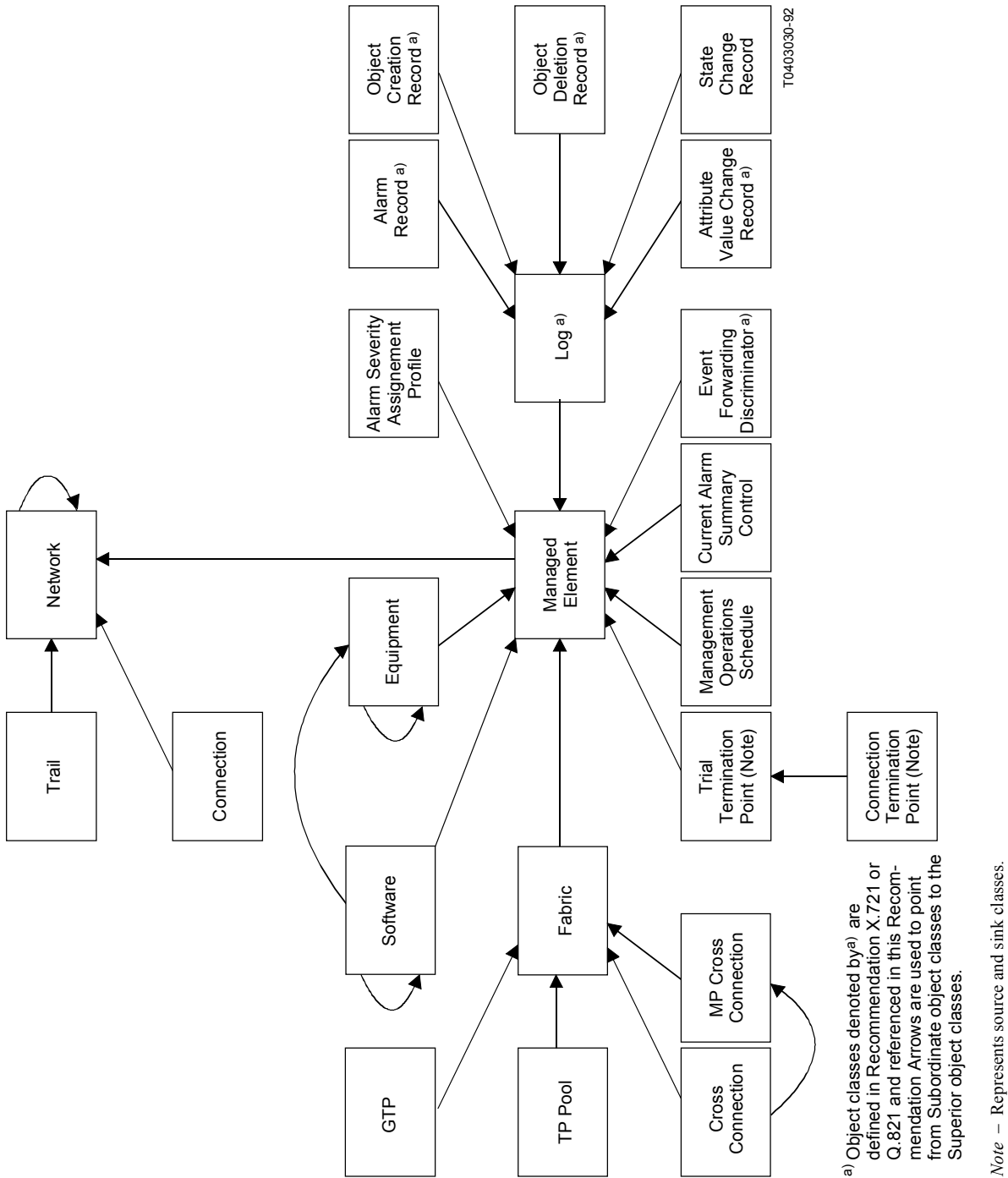
FIGURE 8/M.3100
Naming hierarchy

T0403030-92

a) Object classes denoted by a) are defined in Recommendation X.721 or Q.821 and referenced in this Recommendation Arrows are used to point from Subordinate object classes to the Superior object classes.

Note – Represents source and sink classes.

6.4 *Connection Termination Point Source*

connectionTerminationPointSource-trailTerminationPointSource NAME BINDING
> **SUBORDINATE OBJECT CLASS**
> > **connectionTerminationPointSource;**
>
> **NAMED BY**
> **SUPERIOR OBJECT CLASS trailTerminationPointSource AND SUBCLASSES;**
> **WITH ATTRIBUTE cTPId;**
> **BEHAVIOUR**
> > **cTPSource-TTPBehaviour;**
>
> **CREATE**
> > **WITH-REFERENCE-OBJECT,**
> > **WITH-AUTOMATIC-INSTANCE-NAMING;**
>
> **DELETE**
> > **ONLY-IF-NO-CONTAINED-OBJECTS;**

REGISTERED AS { m3100NameBinding 3 };

connectionTerminationPointSource-trailTerminationPointBidirectional NAME BINDING
> **SUBORDINATE OBJECT CLASS**
> > **connectionTerminationPointSource;**
>
> **NAMED BY**
> **SUPERIOR OBJECT CLASS trailTerminationPointBidirectional**
> > **AND SUBCLASSES;**
>
> **WITH ATTRIBUTE cTPId;**
> **BEHAVIOUR**
> > **cTPSource-TTPBehaviour;**
>
> **CREATE**
> > **WITH-REFERENCE-OBJECT,**
> > **WITH-AUTOMATIC-INSTANCE-NAMING;**
>
> **DELETE**
> > **ONLY-IF-NO-CONTAINED-OBJECTS;**

REGISTERED AS { m3100NameBinding 4 };

cTPSource-TTPBehaviour BEHAVIOUR
> > **DEFINED AS**
> >
> > "The name binding represents a relationship in which a TTP receives information (traffic) from a source CTP.
> >
> > When automatic instance naming is used, the choice of name binding is left as a local matter.";

6.5 *Connection Termination Point Sink*

connectionTerminationPointSink-trailTerminationPointSink
NAME BINDING
> **SUBORDINATE OBJECT CLASS connectionTerminationPointSink;**
> **NAMED BY**
> **SUPERIOR OBJECT CLASS trailTerminationPointSink AND SUBCLASSES;**
> **WITH ATTRIBUTE cTPId;**
> **BEHAVIOUR**
> > **cTPSink-TTPBehaviour;**
>
> **CREATE**
> > **WITH-REFERENCE-OBJECT,**
> > **WITH-AUTOMATIC-INSTANCE-NAMING;**
>
> **DELETE**
> **ONLY-IF-NO-CONTAINED-OBJECTS;**

REGISTERED AS { m3100NameBinding 5 };

connectionTerminationPointSink-trailTerminationPointBidirectional NAME BINDING
    SUBORDINATE OBJECT CLASS connectionTerminationPointSink;
    NAMED BY
    SUPERIOR OBJECT CLASS trailTerminationPointBidirectional
        AND SUBCLASSES;
    WITH ATTRIBUTE cTPId;
    BEHAVIOUR
        cTPSink-TTPBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS { m3100NameBinding 6 };

cTPSink-TTPBehaviour BEHAVIOUR
        DEFINED AS

        "The name binding represents a relationship in which a TTP sends information (traffic) to a sink CTP.

        When automatic instance naming is used, the choice of name binding is left as a local matter.";

6.6   *Cross-Connection*

crossConnection-fabric NAME BINDING
    SUBORDINATE OBJECT CLASS crossConnection
        AND SUBCLASSES;
    NAMED BY
    SUPERIOR OBJECT CLASS   fabric
        AND SUBCLASSES;
    WITH ATTRIBUTE   crossConnectionId;
    BEHAVIOUR
        crossConnection-fabricBehaviour BEHAVIOUR
            DEFINED AS

            "The value of the FromTermination attribute in the Cross-Connection object shall not be NULL.

            When an instance of cross-connection is deleted, the following attributes will be affected. The crossConnectionObjectPointer attribute in the termination point objects or in the GTP objects that were pointing to the deleted cross-connection instance shall be set to point to the Fabric responsible for the connection of the termination points.

            The counters in the appropriate TP Pool objects (if applicable) shall be updated. The connectivityPointer attributes in the disconnected termination points shall be set to NULL. Deleting a cross-connection object instance has no effect on the composition of any GTP."
    ;;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 7 };

crossConnection-mpCrossConnection NAME BINDING
    SUBORDINATE OBJECT CLASS crossConnection
        AND SUBCLASSES;
    NAMED BY
    SUPERIOR OBJECT CLASS   mpCrossConnection
        AND SUBCLASSES;
    WITH ATTRIBUTE   crossConnectionId;
    BEHAVIOUR
        crossConnection-mpCrossConnectionBehaviour BEHAVIOUR
            DEFINED AS

"The value of the FromTermination attribute in the Cross-Connection object must be NULL.

When an instance of cross-connection is deleted, the following attributes will be affected. The crossConnectionObjectPointer attribute in the termination point object or in the GTP object that was pointing to the deleted cross-connection instance shall be set to point to the Fabric responsible for the connection of the termination points. The counters in the appropriate TP Pool objects (if applicable) shall be updated. The connectivity pointers in the disconnected termination point shall be set to NULL.

Deleting the last cross-connection contained in a multipoint cross-connection object instance has the effect of also deleting the multipoint cross-connect object instance (and updating the appropriate pointers). Deleting a cross-connection object instance has no effect on the composition of any GTP.";;

DELETE

ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 8 };

6.7   *Equipment*

equipment-managedElement NAME BINDING

SUBORDINATE OBJECT CLASS equipment AND SUBCLASSES;

NAMED BY

SUPERIOR OBJECT CLASS   managedElement AND SUBCLASSES;

WITH ATTRIBUTE   equipmentId;

BEHAVIOUR

equipmentNameBindingBehaviour;

CREATE

WITH-REFERENCE-OBJECT,

WITH-AUTOMATIC-INSTANCE-NAMING;

DELETE

ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 9 };

equipment-equipment NAME BINDING

SUBORDINATE OBJECT CLASS equipment AND SUBCLASSES;

NAMED BY

SUPERIOR OBJECT CLASS   equipment AND SUBCLASSES;

WITH ATTRIBUTE   equipmentId;

BEHAVIOUR

equipmentNameBindingBehaviour;

CREATE

WITH-REFERENCE-OBJECT,

WITH-AUTOMATIC-INSTANCE-NAMING;

DELETE

ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 10 };

equipmentNameBindingBehaviour

BEHAVIOUR

DEFINED AS

"When automatic instance naming is used, the choice of name binding is left as a local matter.";

6.8   *Event Forwarding Discriminator*

eventForwardingDiscriminator-managedElement NAME BINDING
    SUBORDINATE OBJECT CLASS "Recommendation X.721: 1992":eventForwardingDiscriminator;
    NAMED BY
    SUPERIOR OBJECT CLASS   managedElement AND SUBCLASSES;
    WITH ATTRIBUTE   "Recommendation X.721: 1992":discriminatorId;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 11 };

6.9   *Fabric*

fabric-managedElement NAME BINDING
    SUBORDINATE OBJECT CLASS fabric
        AND SUBCLASSES;
    NAMED BY
    SUPERIOR OBJECT CLASS   managedElement AND SUBCLASSES;
    WITH ATTRIBUTE   fabricId;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 12 };

6.10   *GTP*

gtp-fabric NAME BINDING
    SUBORDINATE OBJECT CLASS gtp;
    NAMED BY
    SUPERIOR OBJECT CLASS   fabric
        AND SUBCLASSES;
    WITH ATTRIBUTE   gtpId;

REGISTERED AS { m3100NameBinding 13 };

6.11   *Log*

log-managedElement NAME BINDING
    SUBORDINATE OBJECT CLASS "Recommendation X.721: 1992":log;
    NAMED BY
    SUPERIOR OBJECT CLASS   managedElement AND SUBCLASSES;
    WITH ATTRIBUTE   "Recommendation X.721: 1992":logId;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 14 };

6.12 *Managed Element*

managedElement-network NAME BINDING
    SUBORDINATE OBJECT CLASS managedElement AND SUBCLASSES;
    NAMED BY
    SUPERIOR OBJECT CLASS   network;
    WITH ATTRIBUTE   managedElementId;
    BEHAVIOUR
        managedElementCreateBehaviour   BEHAVIOUR
        DEFINED AS
        "Managed Element object is not created or deleted by system management protocol. The object is created when initializing the managed element.";;

REGISTERED AS { m3100NameBinding 15 };

6.13 *Multi-Point Cross-Connection*

mpCrossConnection-fabric NAME BINDING
    SUBORDINATE OBJECT CLASS mpCrossConnection
        AND SUBCLASSES;
    NAMED BY
    SUPERIOR OBJECT CLASS   fabric
        AND SUBCLASSES;
    WITH ATTRIBUTE   mpCrossConnectionId;

REGISTERED AS { m3100NameBinding 16 };

6.14 *Network*

network-network NAME BINDING
    SUBORDINATE OBJECT CLASS network AND SUBCLASSES;
    NAMED BY
    SUPERIOR OBJECT CLASS   network AND SUBCLASSES;
    WITH ATTRIBUTE   networkId;
    BEHAVIOUR
        networkCreateBehaviour BEHAVIOUR
        DEFINED AS
        "Network object is not created or deleted by system management protocol. The object is created when initializing the network.";;

REGISTERED AS { m3100NameBinding 17 };

6.15 *Software*

software-equipment NAME BINDING
    SUBORDINATE OBJECT CLASS software AND SUBCLASSES;
    NAMED BY
    SUPERIOR OBJECT CLASS   equipment AND SUBCLASSES;
    WITH ATTRIBUTE   softwareId;
    BEHAVIOUR
        softwareNameBindingBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 18 };

software-software NAME BINDING
    SUBORDINATE OBJECT CLASS software AND SUBCLASSES;
    NAMED BY
    SUPERIOR OBJECT CLASS   software AND SUBCLASSES;
    WITH ATTRIBUTE   softwareId;
    BEHAVIOUR
        softwareNameBindingBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 19 };

software-managedElement NAME BINDING
    SUBORDINATE OBJECT CLASS software AND SUBCLASSES;
    NAMED BY
    SUPERIOR OBJECT CLASS   managedElement AND SUBCLASSES;
    WITH ATTRIBUTE   softwareId;
    BEHAVIOUR
        softwareNameBindingBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 20 };

softwareNameBindingBehaviour
    BEHAVIOUR
    DEFINED AS
    "When automatic instance naming is used, the choice of name binding is left as a local matter.";

6.16   *TP Pool*

tpPool-fabric NAME BINDING
    SUBORDINATE OBJECT CLASS tpPool;
    NAMED BY
    SUPERIOR OBJECT CLASS   fabric
        AND SUBCLASSES;
    WITH ATTRIBUTE   tpPoolId;

REGISTERED AS { m3100NameBinding 21 };

6.17   *Trail*

trail-network NAME BINDING
    SUBORDINATE OBJECT CLASS trail;
    NAMED BY
    SUPERIOR OBJECT CLASS   network;
    WITH ATTRIBUTE   trailId;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 22 };

6.18 *Trail Termination Point Source*

trailTerminationPointSource-managedElement NAME BINDING

    SUBORDINATE OBJECT CLASS trailTerminationPointSource AND SUBCLASSES;

    NAMED BY

    SUPERIOR OBJECT CLASS   managedElement AND SUBCLASSES;

    WITH ATTRIBUTE   tTPId;

    BEHAVIOUR

    trailTerminationPointNameBindingBehaviour;

    CREATE

        WITH-REFERENCE-OBJECT,

        WITH-AUTOMATIC-INSTANCE-NAMING;

    DELETE

        ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 23 };

trailTerminationPointNameBindingBehaviour

        BEHAVIOUR

        DEFINED AS

        "When automatic instance naming is used, the choice of name binding is left as a local matter.";

6.19 *Trail Termination Point Sink*

trailTerminationPointSink-managedElement NAME BINDING

    SUBORDINATE OBJECT CLASS trailTerminationPointSink AND SUBCLASSES;

    NAMED BY

    SUPERIOR OBJECT CLASS   managedElement AND SUBCLASSES;

    WITH ATTRIBUTE   tTPId;

    BEHAVIOUR

    trailTerminationPointNameBindingBehaviour;

    CREATE

        WITH-REFERENCE-OBJECT,

        WITH-AUTOMATIC-INSTANCE-NAMING;

    DELETE

        ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 24 };

7   Actions

7.1   *Add TPs To GTP*

addTpsToGTP ACTION

  BEHAVIOUR

   addTpsToGtpBehaviour BEHAVIOUR

    DEFINED AS

"This action is used to arrange termination points into GTPs. If the group termination point instance does not exist then a new one is automatically created and its identity returned in the result. Otherwise the termination points are added to those already in the GTP. Termination points may be members of zero or one GTP. This action will fail if the GTP is involved in a cross-connection, if the GTP is a member of a TP pool or if the termination point is already a member of a GTP. A bidirectional termination point that can provide independent unidirectional connectivity may be a member of zero or one GTP for each direction of connectivity.";;

  MODE   CONFIRMED;

  WITH INFORMATION SYNTAX   ASN1DefinedTypesModule.AddTpsToGtpInformation;

  WITH REPLY SYNTAX   ASN1DefinedTypesModule.AddTpsToGtpResult;

REGISTERED AS { m3100Action 1 };

7.2   *Add TPs To TP Pool*

addTpsToTpPool ACTION

  BEHAVIOUR

   addTpsToTpPoolBehaviour BEHAVIOUR

    DEFINED AS

"This action is used to arrange termination points or GTPs into pools of termination points or GTPs that are all equivalent for some management purpose such as routing. If the tpPool instance does not exist then a new one is automatically created and its identity returned in the result. Otherwise the termination points or GTPs are added to those already in the tpPool. If an Indirect Adaptor is specified, a GTP representing the CTPs contained from the Indirect Adaptor will be created and it will be added to the tpPool.";;

  MODE   CONFIRMED;

  WITH INFORMATION SYNTAX   ASN1DefinedTypesModule.AddTpsToTpPoolInformation;

  WITH REPLY SYNTAX   ASN1DefinedTypesModule.AddTpsToTpPoolResult;

REGISTERED AS { m3100Action 2 };

7.3   *Allow Audible Visual Local Alarm*

allowAudibleVisualLocalAlarm ACTION

    BEHAVIOUR   allowAudibleVisualLocalAlarmBehaviour;

REGISTERED AS { m3100Action 3 };

allowAudibleVisualLocalAlarmBehaviour BEHAVIOUR

    DEFINED AS "The allow Audible Visual Local Alarm action allows a managed system to present audible and/or visual indications.",

connect ACTION

  BEHAVIOUR

   connectBehaviour BEHAVIOUR

     DEFINED AS

     "This action is used to establish a connection between termination points or GTPs. The termination points to be connected can be specified in one of two ways: 1) by explicitly identifying the two termination points or GTPs, 2) by specifying one termination point or GTP, and specifying a tpPool from which any idle termination point/GTP may be used. The result, if successful, always returns an explicit list of termination points or GTP.

     There are two basic forms of cross-connection arrangement: point to point and point to multipoint (broadcast). A single cross-connection is created if either the explicitPtoP or ptoTpPool option is selected in this action. This cross-connection object points to the termination points or GTPs involved in the cross-connection. Connections are indicated in termination points by the connectivityPointer attribute. If the administrativeState in the crossConnection object is unlocked, this attribute is set, as a result of this action, to the local name of the termination point to which it is connected. Also, the crossConnectionObjectPointer in the termination points or GTPs points to the cross-connection object.

     For point to multipoint cross-connection (indicated by choosing the explicitPtoMp or ptoMPools option), one multi-point cross-connection object will be created containing one crossConnection object for each termination point specified in the toTps parameter. In the source TP the crossConnectionObjectPointer will point at the newly created Multi-point cross-connect object. In each Tp named in the toTPs list (possibly selected from a specified tpPool), the CrossConnectionObject pointer will point at the corresponding cross-connection object. The connectivity pointers in the connected termination points will be updated to reflect the new connectivity.

     The idleTPcount and the connectedTPcount attributes in the tpPool object (if any) are updated as a result of the action. If a GTP is implicitly defined by specifying several termination points to be connected together, the GTP object will be automatically created and its id will be returned in the action reply.

     If an Indirect Adaptor is specified, a GTP representing the CTPs contained from the Indirect Adaptor will be created and it will be connected.

     The administrative state of the created cross-connection or multipoint cross-connection objects is specified as an optional parameter of this action. If this parameter is omitted, the administrative state will be set to "unlocked" (unless the addLegs parameter is specified).

     This action will fail if any of the termination points specified are already involved in a cross-connection or if part of an existing GTP is specified.

     If the addLeg parameter is specified, one or more Legs will be added to an existing multipoint cross-connection arrangement. Selected termination points or GTPs must support a similar signal type to that of the termination points already connected to the arrangement. The result, if successful, always returns the termination points or GTPs involved in the multipoint cross-connection. A cross-connection object is created as a result of this action. This object will be named from the specified mpCrossConnection object instance.

     The administrative state of the created cross-connection object will be the same as that of the containing multipoint cross-connection object unless otherwise specified in the action parameters." ;;

  MODE   CONFIRMED;

  WITH INFORMATION SYNTAX   ASN1DefinedTypesModule.ConnectInformation;

  WITH REPLY SYNTAX   ASN1DefinedTypesModule.ConnectResult;

REGISTERED AS { m3100Action 4 };

7.5   *Disconnect*

disconnect ACTION
  BEHAVIOUR
   disconnectBehaviour BEHAVIOUR
     DEFINED AS

"This action is used to take down a cross-connection. The connection to be taken down is specified by identifying termination point(s) (or GTP(s)) of the connection. If the connection was point to point then the other termination point or GTP is implicitly disconnected as well and the cross-connection object is deleted. If the connection was point to multipoint and the action referred to the master, all the termination points or GTPs that are legs are implicitly disconnected as well and the multipoint cross-connection and cross-connection objects are deleted.

If the connection was point to multipoint and the action referred to a leg, then just that leg is disconnected, unless it is the last leg, in which case the master termination point is also implicitly disconnected and the multipoint cross-connection and cross-connection objects are deleted. The idleTPcount and the connectedTPcount attributes in the tpPool objects (if any) are updated as a result of the action. The connectivity pointers in the disconnected termination points will be set to NULL as a result of this action.

This action has no effect on the composition of GTPs and GTPs are not deleted as a result of this action. This action will fail if part of a GTP is specified.";;

  MODE   CONFIRMED;
  WITH INFORMATION SYNTAX   ASN1DefinedTypesModule.DisconnectInformation;
  WITH REPLY SYNTAX   ASN1DefinedTypesModule.DisconnectResult;

REGISTERED AS { m3100Action 5 };

7.6   *Inhibit Audible Visual Local Alarm*

inhibitAudibleVisualLocalAlarm ACTION
     BEHAVIOUR   inhibitAudibleVisualLocalAlarmBehaviour;

REGISTERED AS { m3100Action 6 };

inhibitAudibleVisualLocalAlarmBehaviour BEHAVIOUR
     DEFINED AS
     "The Inhibit Audible/Visual Local Alarm action inhibits a managed system from presenting audible and/or visual indications";

7.7   *Remove TPs From GTP*

removeTpsFromGTP ACTION
  BEHAVIOUR
   removeTpsFromGtpBehaviour BEHAVIOUR
     DEFINED AS

"This action is used to remove termination points from GTPs. This action will fail if the GTP is involved in a cross-connection or if it is a member of a TP pool. Removing the last termination point from a GTP has the effect of deleting the GTP object. If the GTP is deleted, the name of the GTP will be sent back in the ACTION reply.";;

  MODE   CONFIRMED;
  WITH INFORMATION SYNTAX   ASN1DefinedTypesModule.RemoveTpsFromGtpInformation;
  WITH REPLY SYNTAX   ASN1DefinedTypesModule.RemoveTpsFromGtpResult;

REGISTERED AS { m3100Action 7 };

7.8 *Remove TPs From TP Pool*

removeTpsFromTpPool ACTION
  BEHAVIOUR
   removeTpsFromTpPoolBehaviour BEHAVIOUR
     DEFINED AS
     "This action is used to remove termination points from termination point pools. Removing the last termination point from a pool has the effect of deleting the TP Pool object. If the TP pool is deleted, the name of the TP Pool will be sent back in the ACTION reply.";;

  MODE CONFIRMED;
  WITH INFORMATION SYNTAX   ASN1DefinedTypesModule.RemoveTpsFromTpPoolInformation;
  WITH REPLY SYNTAX   ASN1DefinedTypesModule.RemoveTpsFromTpPoolResult;

REGISTERED AS { m3100Action 8 };

7.9 *Reset Audible Alarm*

This action is defined in Recommendation Q.821 [7].

8 Notifications

8.1 *Attribute Value Change*

This notification type is used to report when there is a change in some of the attribute values of a managed object. It is defined in Recommendation X.721 [5].

8.2 *Communications Alarm*

This notification type is used to report when the managed object detects a communication error. It is defined in Recommendation X.721 [5].

8.3 *Environmental Alarm*

This notification type is used to report when the managed object detects a problem in the environment. It is defined in Recommendation X.721 [5].

8.4 *Equipment Alarm*

This notification type is used to report a failure in the equipment. It is defined in Recommendation X.721 [5].

8.5 *Object Creation*

This notification type is used to report the creation of a managed object if defined in the managed object class specification. It is defined in Recommendation X.721 [5].

8.6 *Object Deletion*

This notification type is used to report the deletion of a managed object if defined in the managed object class specification. It is defined in Recommendation X.721 [5].

8.7 *Processing Error Alarm*

This notification type is used to report a processing failure in a managed object. It is defined in Recommendation X.721 [5].

8.8 *State Change*

This notification type is used to report when there is a change in some of the state values of a managed object. It is defined in Recommendation X.721 [5].

9 ASN.1 Defined Types Module

ASN1DefinedTypesModule { ccitt recommendation m gnm(3100) informationModel(0) asn1Modules(2) asn1DefinedTypesModule(0) }

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

-- EXPORTS everything

IMPORTS
RDNSequence
      FROM InformationFramework { joint-iso-ccitt ds(5) modules(1) informationFramework(1) }
ObjectInstance,ObjectClass FROM CMIP-1 { joint-iso-ccitt ms(9) cmip(1) modules(0) protocol(3) }
ProbableCause, AdministrativeState FROM Attribute-ASN1Module { joint-iso-ccitt ms(9) smi(3) part2(2) asn1Module (2) 1 } ;

m3100InformationModel OBJECT IDENTIFIER ::= { ccitt recommendation m gnm(3100) informationModel(0) }
m3100standardSpecificExtension OBJECT IDENTIFIER ::= { m3100InformationModel standardSpecificExtension(0) }
m3100ObjectClass OBJECT IDENTIFIER ::= { m3100InformationModel managedObjectClass(3) }
m3100Package OBJECT IDENTIFIER ::= { m3100InformationModel package(4) }
m3100Attribute OBJECT IDENTIFIER ::= { m3100InformationModel attribute(7) }
m3100NameBinding OBJECT IDENTIFIER ::= { m3100InformationModel nameBinding(6) }
m3100Action OBJECT IDENTIFIER ::= { m3100InformationModel action(9) }
m3100Notification OBJECT IDENTIFIER ::= { m3100InformationModel notification(10) }
-- *Reserved arcs below m3100InformationModel are (5) for parameters, (8) for attribute groups*

characteristicInfo OBJECT IDENTIFIER ::= { m3100standardSpecificExtension 0 }

opticalSTM1SPICI   CharacteristicInformation ::= { characteristicInfo 1 }
-- *opticalSPITTP* object instances with stmLevel attribute = 1*

opticalSTM4SPICI   CharacteristicInformation ::= { characteristicInfo 2 }
-- *opticalSPITTP* object instances with stmLevel attribute = 4*

opticalSTM16SPICI   CharacteristicInformation ::= { characteristicInfo 3 }
-- *opticalSPITTP*object instances with stmLevel attribute = 16*

electricalSTM1SPICI CharacteristicInformation ::= { characteristicInfo 4 }
-- *opticalSPITTP* object instances with stmLevel attribute = 1*

rsSTM1SPICI CharacteristicInformation ::= { characteristicInfo 5 }
-- *rsCTP* object instances with stmLevel attribute = 1*

rsSTM4SPICI CharacteristicInformation ::= { characteristicInfo 6 }
-- *rsCTP* object instances with stmLevel attribute = 4*

rsSTM16SPICI CharacteristicInformation ::= { characteristicInfo 7 }
-- *rsCTP* object instances with stmLevel attribute = 16*

msSTM1SPICI CharacteristicInformation ::= { characteristicInfo 8 }
-- *rsCTP* object instances with stmLevel attribute = 1*

msSTM4SPICI CharacteristicInformation ::= { characteristicInfo 9 }
-- *rsCTP* object instances with stmLevel attribute = 4*

msSTM16SPICI CharacteristicInformation ::= { characteristicInfo 10 }
-- *rsCTP* object instances with stmLevel attribute = 16*

au3TU3VC3CI CharacteristicInformation ::= { characteristicInfo 11 }

au4VC4CI CharacteristicInformation ::= { characteristicInfo 12 }

tu11VC11CI CharacteristicInformation ::= { characteristicInfo 13 }

tu12VC12CI CharacteristicInformation ::= { characteristicInfo 14 }

tu2VC2CI CharacteristicInformation ::= { characteristicInfo 15 }

tu12VC11CI CharacteristicInformation ::= { characteristicInfo 16 }
-- The following value assignments are for the Probable Cause when Integer Choice is used within the
-- TMN application context. These values shall always be assigned by this Recommendation in the context
-- of TMN.
indeterminate ProbableCause ::= localValue : 0
-- The following are used with communications alarm.
aIS ProbableCause ::= localValue : 1
callSetUpFailure ProbableCause ::= localValue : 2
degradedSignal ProbableCause ::= localValue : 3
farEndReceiverFailure ProbableCause ::= localValue : 4
framingError ProbableCause ::= localValue : 5
lossOfFrame ProbableCause ::= localValue : 6
lossOfPointer ProbableCause ::= localValue : 7
lossOfSignal ProbableCause ::= localValue : 8
payloadTypeMismatch ProbableCause ::= localValue : 9
transmissionError ProbableCause ::= localValue : 10
remoteAlarmInterface ProbableCause ::= localValue : 11
excessiveBER ProbableCause ::= localValue : 12
pathTraceMismatch ProbabaleCause ::= localValue13
-- Values 14 to 50 are reserved for communications alarm related probable causes

-- The following are used with equipment alarm.

backplaneFailure ProbableCause ::= localValue : 51
dataSetProblem ProbableCause ::= localValue : 52
equipmentIdentifierDuplication ProbableCause ::= localValue : 53
externalIFDeviceProblem ProbableCause ::= localValue : 54
lineCardProblem ProbableCause ::= localValue : 55
multiplexerProblem ProbableCause ::= localValue : 56
nEIdentifierDuplication ProbableCause ::= localValue : 57
powerProblem ProbableCause ::= localValue : 58
processorProblem ProbableCause ::= localValue : 59
protectionPathFailure ProbableCause ::= localValue : 60
receiverFailure ProbableCause ::= localValue : 61
replaceableUnitMissing ProbableCause ::= localValue : 62
replaceableUnitTypeMismatch ProbableCause ::= localValue : 63
synchronizationSourceMismatch ProbableCause ::= localValue : 64
terminalProblem ProbableCause ::= localValue : 65
timingProblem ProbableCause ::= localValue : 66
transmitterFailure ProbableCause ::= localValue : 67
trunkCardProblem ProbableCause ::= localValue : 68
replaceableUnitProblem ProbableCause ::= localValue : 69
-- Values 70 to 100 are reserved for equipment alarm related probable causes

-- The following are used with environmental alarm.

airCompressorFailure ProbableCause ::= localValue : 101
airConditioningFailure ProbableCause ::= localValue : 102
airDryerFailure ProbableCause ::= localValue : 103
batteryDischarging ProbableCause ::= localValue : 104
batteryFailure ProbableCause ::= localValue : 105
commercialPowerFailure ProbableCause ::= localValue : 106
coolingFanFailure ProbableCause ::= localValue : 107
engineFailure ProbableCause ::= localValue : 108
fireDetectorFailure ProbableCause ::= localValue : 109
fuseFailure ProbableCause ::= localValue : 110
generatorFailure ProbableCause ::= localValue : 111
lowBatteryThreshold ProbableCause ::= localValue : 112
pumpFailure ProbableCause ::= localValue : 113
rectifierFailure ProbableCause ::= localValue : 114
rectifierHighVoltage ProbableCause ::= localValue : 115

rectifierLowFVoltage ProbableCause ::= localValue : 116
ventilationsSystemFailure ProbableCause ::= localValue : 117
enclosureDoorOpen ProbableCause ::= localValue : 118
explosiveGas ProbableCause ::= localValue : 119
fire ProbableCause ::= localValue : 120
flood ProbableCause ::= localValue : 121
highHumidity ProbableCause ::= localValue : 122
highTemperature ProbableCause ::= localValue : 123
highWind ProbableCause ::= localValue : 124
iceBuildUp ProbableCause ::= localValue : 125
intrusionDetection ProbableCause ::= localValue : 126
lowFuel ProbableCause ::= localValue : 127
lowHumidity ProbableCause ::= localValue : 128
lowCablePressure ProbableCause ::= localValue : 129
lowTemperature ProbableCause ::= localValue : 130
lowWater ProbableCause ::= localValue : 131
smoke ProbableCause ::= localValue : 132
toxicGas ProbableCause ::= localValue : 133
-- *Values 134 to 150 are reserved for environmental alarm related probable causes*

-- *The following are used with Processing error alarm.*

storageCapacityProblem ProbableCause ::= localValue : 151
memoryMismatch ProbableCause ::= localValue : 152
corruptData ProbableCause ::= localValue : 153
outOfCPUCycles ProbableCause ::= localValue : 154
sfwrEnvironmentProblem ProbableCause ::= localValue : 155
sfwrDownloadFailure ProbableCause ::= localValue : 156

-- *Service ProblemType is for further study*

AddedTps ::= SEQUENCE {
    gtp   ObjectInstance,
    tpsAdded   SEQUENCE OF ObjectInstance
    }

AddLeg ::= SEQUENCE {
    mpCrossConnection ObjectInstance,
    legs   SET OF ToTermSpecifier
    }

AddTpsToGtpInformation ::= SEQUENCE OF SEQUENCE {
    tpsAdded   SEQUENCE OF TerminationPointInformation,
    gtp   ObjectInstance   OPTIONAL
    }

AddTpsToGtpResult ::= SEQUENCE OF CHOICE {
    failed   [0] Failed,
    addedTps [1] AddedTps
    }
-- *the nth element in the "SEQUENCE OF" is related to the nth element in the "SEQUENCE OF" of the*
-- *"AddTpsToGtpInformation" type.*

AddTpsToTpPoolInformation ::= SEQUENCE OF SEQUENCE {
    tps   SET OF TerminationPointInformation,
    toTpPool   ObjectInstance   OPTIONAL
    }

AddTpsToTpPoolResult ::= SEQUENCE OF CHOICE {
    failed   [0]   Failed,
    tpsAddedToTpPool [1]   TpsAddedToTpPool
    }
 -- *the nth element in the "SEQUENCE OF" is related to the nth element in the "SEQUENCE OF" of the*
-- *"AddTPsToTpPoolInformation" type.*

AlarmSeverityAssignment ::= SEQUENCE {

    problem   ProbableCause,

    severityAssignedServiceAffecting   **[0]** AlarmSeverityCode OPTIONAL,

    severityAssignedNonServiceAffecting **[1]** AlarmSeverityCode OPTIONAL,

    severityAssigned/ServiceIndependent **[2]** AlarmSeverityCode OPTIONAL }

AlarmSeverityAssignmentList ::= SET OF AlarmSeverityAssignment

AlarmSeverityCode ::= ENUMERATED {

    non-alarmed (0),

    minor (1),

    major (2),

    critical (3),

    warning (4) }

AlarmStatus ::=   ENUMERATED {

    cleared (0),

    activeReportable-Indeterminate (1),

    activeReportable-Warning (2),

    activeReportable-Minor (3),

    activeReportable-Major (4),

    activeReportable-Critical (5),

    activePending (6)

    }

Boolean ::= BOOLEAN

Bundle ::= SEQUENCE {

   characteristicInfoType   CharacteristicInformation,

   bundlingFactor   INTEGER

}

ChannelNumber ::= INTEGER

CharacteristicInformation ::= OBJECT IDENTIFIER

Connected ::= CHOICE {

   pointToPoint   **[0]**   PointToPoint,

   pointToMultipoint   **[1]**   PointToMultipoint

   }

ConnectInformation ::= SEQUENCE OF SEQUENCE {

   CHOICE {

      unidirectional   **[0]**   ConnectionType,

      bidirectional   **[1]**   ConnectionTypeBi,

      addleg   **[2]**   AddLeg

   }

   administrativeState   AdministrativeState OPTIONAL

}

ConnectivityPointer ::= CHOICE { none   NULL,

                       single   ObjectInstance,

                       concatenated   SEQUENCE OF ObjectInstance }

ConnectResult ::= SEQUENCE OF CHOICE {

   failed   Failed,

   connected   Connected

   }

-- the nth element in the "SEQUENCE OF" is related to the n-th element in the "SEQUENCE OF" of the

-- "ConnectInformation" type.

ConnectionType ::= CHOICE {

   explicitPToP   **[0]**   ExplicitPtoP,

   ptoTpPool   **[1]**   PtoTPPool,

   explicitPtoMP   **[2]**   ExplicitPtoMP,

   ptoMPools   **[3]**   PtoMPools

   }

ConnectionTypeBi ::= CHOICE {
   explicitPToP   [0] ExplicitPtoP,
   ptoTpPool   [1] PtoTPPool
   }

Count ::= INTEGER

CrossConnectionName ::= GraphicString

CrossConnectionObjectPointer ::= CHOICE {
                         notConnected   [0]   ObjectInstance,   *-- Fabric object --*
                         connected   [1]   ObjectInstance,   *-- Cross-connection object --*
                         multipleConnections MultipleConnections
                         }

CTPUpstreamPointer ::= ConnectivityPointer(WITH COMPONENTS { ...,
                                   -- the other two choices are present
                                           concatenated   ABSENT })
CTPDownstreamPointer ::= DownstreamConnectivityPointer (WITH COMPONENTS
                                   {...,
                                       concatenated   ABSENT,
                             broadcastConcatenated   ABSENT
                               -- other choices are present
                               })

CurrentProblem ::= SEQUENCE {
      problem   [0]   ProbableCause,
      alarmStatus   [1]   AlarmStatus
      }

CurrentProblemList ::= SET OF CurrentProblem

Directionality ::= ENUMERATED { unidirectional(0),
      bidirectional(1) }

DisconnectInformation ::= SEQUENCE OF ObjectInstance *-- tps*

DisconnectResult ::= SEQUENCE OF CHOICE {
   failed   Failed,
   disconnected   ObjectInstance
   }
-- the nth element in the "SEQUENCE OF" is related to the nth element in the "SEQUENCE OF" of the
-- "DisconnectInformation" type.

DownstreamConnectivityPointer ::= CHOICE {
                       none   NULL,
                       single   ObjectInstance,
                       concatenated   SEQUENCE OF ObjectInstance,
                       broadcast   SET OF ObjectInstance,
                       broadcastConcatenated [1] SET OF SEQUENCE OF ObjectInstance }

ExplicitPtoMP ::= SEQUENCE {
   fromTp   ExplicitTP,
   toTPs   SET OF ExplicitTP
   }

ExplicitPtoP ::= SEQUENCE {
   fromTp   ExplicitTP,
   toTp   ExplicitTP
   }

ExplicitTP ::= CHOICE {
   oneTPorGTP ObjectInstance,
   listofTPs SEQUENCE OF ObjectInstance
   }

ExternalTime ::= GeneralizedTime

Failed ::= CHOICE {
          logicalProblem   LogicalProblem,
          resourceProblem   ResourceProblem }

ListOfCharacteristicInformation ::= SET OF CharacteristicInformation

ListOfTPs ::= SET OF ObjectInstance

LocationName ::= GraphicString

LogicalProblem ::= SEQUENCE {
          problemCause   ProblemCause,
          incorrectInstances   SET OF ObjectInstance   OPTIONAL }

MultipleConnections ::= SET OF CHOICE {
          downstreamNotConnected [0] ObjectInstance,
          downstreamConnected   [1] ObjectInstance,
          upstreamNotConnected   [2] ObjectInstance,
          upstreamConnected   [3] ObjectInstance }

NameType ::=   CHOICE {
                numericName   INTEGER,
                pString   GraphicString
                }

ObjectList ::=   SET OF ObjectInstance

Pointer ::= ObjectInstance

PointerOrNull ::= CHOICE {
          pointer ObjectInstance,
          null   NULL }

PointToPoint ::= SEQUENCE {
  fromTp   ObjectInstance,
  to   TpObjectInstance,
  xCon   ObjectInstance
  }

PointToMultipoint ::= SEQUENCE {
  fromTp   ObjectInstance,
  toTps   SET OF SEQUENCE {
    tps   ObjectInstance,
    xConnections ObjectInstance
    },
  mpXCon   ObjectInstance
  }

ProblemCause ::= CHOICE {
          unknown   NULL,
          integerValue   INTEGER }
-- The values of integer value for ProblemCause and integerValue for ResourceProblem shall always be
-- assigned by this Recommendation. No values of integerValue for ResourceProblem have been assigned.
-- The following values are used for integerValue of ProblemCause.

noSuchTpInstance   ProblemCause ::= integerValue : 0
noSuchGtpInstance   ProblemCause ::= integerValue : 1
noSuchTpPoolInstance   ProblemCause ::= integerValue : 2
mismatchingTpInstance   ProblemCause ::= integerValue : 3
mismatchingGtpInstance   ProblemCause ::= integerValue : 4
partOfGtp   ProblemCause ::= integerValue : 5
involvedInCrossConnection   ProblemCause ::= integerValue : 6

```
memberOfTpPool                ProblemCause ::= integerValue : 7
alreadyMemberOfGtp            ProblemCause ::= integerValue : 8
noTpInTpPool                  ProblemCause ::= integerValue : 9
noMoreThanOneTpIsAllowed      ProblemCause ::= integerValue : 10
noMoreThanTwoTpsAreAllowed ProblemCause ::= integerValue : 11

PtoMPools ::= SEQUENCE {
   fromTp       ExplicitTP,
   toTPPools   ToTPPools
   }

PtoTPPool  ::= SEQUENCE {
   fromTp       ExplicitTP,
   toTpPool     ObjectInstance
   }

RelatedObjectInstance ::= CHOICE {
                          notAvailable  NULL,
                          relatedObject ObjectInstance
                          }

RemoveTpsFromGtpInformation ::= SEQUENCE OF SEQUENCE {
   fromGtp     ObjectInstance,
   tps       SET OF ObjectInstance
   }

RemoveTpsFromGtpResult ::= SEQUENCE OF CHOICE {
   failed      [0]  Failed,
   removed     [1]  RemoveTpsResultInformation
   }
-- the nth element in the "SEQUENCE OF" is related to the nth element in the "SEQUENCE OF" of the
-- "RemoveTPsFromGtpInformation" type.

RemoveTpsFromTpPoolInformation  ::= SEQUENCE OF SEQUENCE {
   fromTpPool            ObjectInstance,
   tps                   SET OF ObjectInstance
   }

RemoveTpsFromTpPoolResult ::= SEQUENCE OF CHOICE {
   failed                [0]  Failed,
   removed               [1]  RemoveTpsResultInformation
   }
-- the nth element in the "SEQUENCE OF" is related to the nth element in the "SEQUENCE OF" of the
-- "RemoveTPsFromTpPoolInformation" type.

RemoveTpsResultInformation ::= SEQUENCE {
             deletedTpPoolOrGTP    ObjectInstance    OPTIONAL,
             tps                   SET OF ObjectInstance }
-- If the TP Pool or GTP is deleted, the deleted TP Pool or GTP should be provided in the
-- RemoveTpsResultInformation

Replaceable ::=       ENUMERATED {
                          yes (0),
                          no (1),
                          notapplicable (2)
                          }

ResourceProblem::= CHOICE {
             unknown       NULL,
             integerValue   INTEGER }

SequenceOfObjectInstance ::= SEQUENCE OF ObjectInstance

SignalType ::= CHOICE {
   simple     CharacteristicInformation,
   bundle     Bundle,
   complex    [0] SEQUENCE OF Bundle
}
```

SupportableClientList ::= SET OF ObjectClass

SystemTiming ::= SEQUENCE {
              sourceType   ENUMERATED { internalTimingSource(0),
                              remoteTimingSource(1), slavedTimingTerminationSignal(2) },
              sourceID   ObjectInstance   OPTIONAL -- *not needed for internal source*
              }

SystemTimingSource ::= SEQUENCE {
              primaryTimingSource   SystemTiming,
              secondaryTimingSource   SystemTiming OPTIONAL }

TerminationPointInformation ::= CHOICE {
                  tPOrGTP   [0]   ObjectInstance,
                  sourceTP   [1]   ObjectInstance,
                  sinkTP   [2]   ObjectInstance }

ToTermSpecifier ::= CHOICE {
   toTpOrGTP   [0]   ExplicitTP,
   toPool   [1]   ObjectInstance
}

ToTPPools ::= SET OF SEQUENCE {
   tpPoolId   ObjectInstance,
   numberOfTPs INTEGER
   }

TpsAddedToTpPool ::= SEQUENCE {
   tpPool   ObjectInstance,
   tps   SET OF ObjectInstance
   }

TpsInGtpList ::= SEQUENCE OF ObjectInstance

UserLabel ::= GraphicString

VendorName ::= GraphicString

Version ::= GraphicString

END -- *end of ASN1DefinedTypesModule*

10   TMN application context

The object identifier value

{ ccitt recommendation m(13) gnm(3100) protocolSupport(1) applicationContext(0) tmnApplicationContextOne(1) }

is assigned to the application context that has the same capabilities as the systems management application context defined in Recommendation X.701, but also supports the integer values for ProbableCause. These integer value assignments are specified in this Recommendation.

11   Entity – relationship diagrams

Figures 1/M.3100 to 8/M.3100 depict the various relationships between the managed object classes specified in this Recommendation. These figures are representative of the relationships and do not contain all the relationships. All the relationships can be determined from the templates in § 3.

ANNEX A

(to Recommendation M.3100)

Index

A.1   *Managed Objects*

A.2   *Packages*

A.3   *Attributes*

A.4   *Name Bindings*

A.5   *Actions*

A.6   *Notifications*

ANNEX B

(to Recommendation M.3100)

B.1   *Introduction*

This annex contains additional object classes that may be useful for technology independent management.

B.2   *Object classes*

B.2.1   *Circuit Trail Termination Point Bidirectional*

The Circuit Trail Termination Point Bidirectional object class is a class of managed objects that terminates a circuit in one direction and originates a circuit in the opposite direction. This object class is instantiated[1] when technology-independent-management is required. Technology-specific subclasses should not be derived from this class.

circuitTrailTerminationPointBidirectional MANAGED OBJECT CLASS
  DERIVED FROM
    trailTerminationPointBidirectional,
    circuitTrailTerminationPointSource,
    circuitTrailTerminationPointSink;

REGISTERED AS { m3100ObjectClass 1001 };

_____

[1] i.e. instance(s) of the object class(es) is created.

B.2.2   *Circuit Trail Termination Point Sink*

The Circuit Trail Termination Point Sink object class is a class of managed objects that terminates a circuit which is concerned with the transfer of payload in direct support of telecommunication services. This object class is instantiated[2] when technology-independent-management is required. Technology-specific subclasses should not be derived from this class.

```
circuitTrailTerminationPointSink MANAGED OBJECT CLASS
    DERIVED FROM        trailTerminationPointSink;
    CHARACTERIZED BY
    circuitTrailTerminationPointSinkPackage PACKAGE
        BEHAVIOUR
            circuitTrailTerminationPointSinkBehaviour;
        ATTRIBUTES
            circuitTTPId GET;;;

REGISTERED AS { m3100ObjectClass 1002 };

circuitTrailTerminationPointSinkBehaviour BEHAVIOUR
        DEFINED AS
        "This managed object terminates a circuit which is concerned with the transfer of payload in direct
        support of telecommunication services.";
```

B.2.3   *Circuit Trail Termination Point Source*

The Circuit Trail Termination Point Source object class is a class of managed objects that originates a circuit which is concerned with the transfer of payload in direct support of telecommunication services. This object class is instantiated[2] when technology-independent-management is required. Technology-specific subclasses should not be derived from this class.

```
circuitTrailTerminationPointSource MANAGED OBJECT CLASS
    DERIVED FROM        trailTerminationPointSource;
    CHARACTERIZED BY
    circuitTrailTerminationPointSourcePackage PACKAGE
        BEHAVIOUR
            circuitTrailTerminationPointSourceBehaviour;
        ATTRIBUTES
            circuitTTPId GET;;;

REGISTERED AS { m3100ObjectClass 1003 };

circuitTrailTerminationPointSourceBehaviour BEHAVIOUR
        DEFINED AS
        "This managed object originates a circuit which is concerned with the transfer of payload in direct
        support of telecommunication services.";
```

B.2.4   *Media Trail Termination Point Bidirectional*

The Media Trail Termination Point Bidirectional object class is a class of managed objects that generates and detects the carrier mechanism which is dependent on the physical media (e.g. for radio, the carrier wave is generated in one direction and detected in the opposite direction). This object class is instantiated[2] when management at a technology-independent level is required. Technology-specific subclasses should not be derived from this class.

```
mediaTrailTerminationPointBidirectional MANAGED OBJECT CLASS
    DERIVED FROM
        trailTerminationPointBidirectional,
        mediaTrailTerminationPointSource,
        mediaTrailTerminationPointSink;

REGISTERED AS { m3100ObjectClass 1004 };
```

---

[2]   i.e. instance(s) of the object class(es) is created.

B.2.5 *Media Trail Termination Point Sink*

The Media Trail Termination Point Sink object class is a class of managed objects that detects the carrier mechanism which is dependent on the physical media (e.g. for radio, the carrier wave is detected). This object class is instantiated[3] when management at a technology-independent level is required. Technology-specific subclasses should not be derived from this class.

```
mediaTrailTerminationPointSink MANAGED OBJECT CLASS
    DERIVED FROM      trailTerminationPointSink;
    CHARACTERIZED BY
    mediaTrailTerminationSinkPackage PACKAGE
        BEHAVIOUR
            mediaTrailTerminationSinkBehaviour;
        ATTRIBUTES
            mediaTTPId GET,
            mediaType GET;;;

REGISTERED AS { m3100ObjectClass 1005 };

mediaTrailTerminationSinkBehaviour BEHAVIOUR
        DEFINED AS
        "This managed object detects the carrier mechanism which is dependent on the physical media (e.g. for
        radio, the carrier wave is detected).";
```

B.2.6 *Media Trail Termination Point Source*

The Media Trail Termination Point Source object class is a class of managed objects that originates the carrier mechanism which is dependent on the physical media (e.g. for radio, the carrier wave is generated). This object class is instantiated[3] when management at a technology-independent level is required. Technology-specific subclasses should not be derived from this class.

```
mediaTrailTerminationPointSource MANAGED OBJECT CLASS
    DERIVED FROM      trailTerminationPointSource;
    CHARACTERIZED BY
    mediaTrailTerminationSourcePackage PACKAGE
        BEHAVIOUR
            mediaTrailTerminationSourceBehaviour;
        ATTRIBUTES
            mediaTTPId GET,
            mediaType GET;;;

REGISTERED AS { m3100ObjectClass 1006 };

mediaTrailTerminationSourceBehaviour BEHAVIOUR
        DEFINED AS
        "This managed object originates the carrier mechanism which is dependent on the physical media (e.g.
        for radio the carrier wave is generated).";
```

B.2.7 *Overhead and Adaptation Trail Termination Point Bidirectional*

The Overhead and Adaptation Trail Termination Point Bidirectional object class is a class of managed objects that terminates an overhead and adaptation trail in one direction and originates an overhead and adaptation trail in the opposite direction. This object class is instantiated[3] when management at a technology-independent level is required. Technology-specific subclasses should not be derived from this class.

---

[3] i.e. instance (s) of the object class(es) is created.

overheadAndAdaptationTrailTerminationPointBidirectional MANAGED OBJECT CLASS
   DERIVED FROM
      trailTerminationPointBidirectional,
      overheadAndAdaptationTrailTerminationPointSource,
      overheadAndAdaptationTrailTerminationPointSink;

REGISTERED AS { m3100ObjectClass 1007 };

B.2.8   *Overhead and Adaptation Trail Termination Point Sink*

     The Overhead and Adaptation Trail Termination Point Sink object class is a class of managed objects that terminates an overhead and adaptation trail. At this point the overhead is extracted from the payload. The extracted information may be used to collect end-to-end error information, perform error correction and end-to-end protection switching. This object class is instantiated[4] when management at a technology-independent level is required. Technology-specific subclasses should not be derived from this class.

overheadAndAdaptationTrailTerminationPointSink MANAGED OBJECT CLASS
   DERIVED FROM   **trailTerminationPointSink;**
   CHARACTERIZED BY
   overheadAndAdaptationTrailTerminationPointSinkPackage PACKAGE
      BEHAVIOUR
         overheadAndAdaptationTrailTerminationPointSinkBehaviour;
      ATTRIBUTES
         oAndATTPId GET,
         informationRate GET;;;

REGISTERED AS { m3100ObjectClass 1008 };

overheadAndAdaptationTrailTerminationPointSinkBehaviour BEHAVIOUR
      DEFINED AS
      "This managed object terminates an overhead and adaptation trail. At this point the overhead is extracted from the payload.";

B.2.9   *Overhead and Adaptation Trail Termination Point Source*

     The Overhead and Adaptation Trail Termination Point Source object class is a class of managed objects that originates an overhead and adaptation trail. At this point the overhead is generated and added to the payload. This object class is instantiated[4] when management at a technology-independent level is required. Technology-specific subclasses should not be derived from this class.

overheadAndAdaptationTrailTerminationPointSource MANAGED OBJECT CLASS
   DERIVED FROM   **trailTerminationPointSource;**
   CHARACTERIZED BY
   overheadAndAdaptationTrailTerminationPointSourcePackage PACKAGE
      BEHAVIOUR
         overheadAndAdaptationTrailTerminationPointSourceBehaviour;
      ATTRIBUTES
         oAndATTPId GET,
         informationRate GET;;;

REGISTERED AS { m3100ObjectClass 1009 };

overheadAndAdaptationTrailTerminationPointSourceBehaviour BEHAVIOUR
      DEFINED AS
      "This managed object originates an overhead and adaptation trail. At this point the overhead is generated and added to the payload.";

---

[4] i.e. instance(s) of the object class(es) is created.

B.2.10   *Signal Trail Termination Point Bidirectional*

The Signal Trail Termination Point Bidirectional object class is a class of managed objects that terminates a modulated/encoded signal in one direction and generates a modulated/encoded signal in the opposite direction. This object class is instantiated[5] when management at a technology-independent level is required. Technology-specific subclasses should not be derived from this class.

signalTrailTerminationPointBidirectional MANAGED OBJECT CLASS
  DERIVED FROM
      trailTerminationPointBidirectional,
      signalTrailTerminationPointSource,
      signalTrailTerminationPointSink;

REGISTERED AS { m3100ObjectClass 1010 };

B.2.11   *Signal Trail Termination Point Sink*

The Signal Trail Termination Point Sink object class is a class of managed objects that terminates a modulated/encoded signal. This object class is instantiated[5] when management at a technology-independent level is required. Technology-specific subclasses should not be derived from this class.

signalTrailTerminationPointSink MANAGED OBJECT CLASS
  DERIVED FROM   trailTerminationPointSink;
  CHARACTERIZED BY
  signalTrailTerminationPointSinkPackage PACKAGE
      BEHAVIOUR
          signalTrailTerminationPointSinkBehaviour;
      ATTRIBUTES
          signalTTPId GET,
          lineCoding GET;;;

REGISTERED AS { cm3100ObjectClass 1011 };

signalTrailTerminationPointSinkBehaviour BEHAVIOUR
          DEFINED AS
          "This managed object terminates a modulated/encoded signal.";

B.2.12   *Signal Trail Termination Point Source*

The Signal Trail Termination Point Source object class is a class of managed objects that creates a modulated/encoded signal ready for transmission. This object class is instantiated[5] when management at a technology-independent level is required. Technology-specific subclasses should not be derived from this class.

signalTrailTerminationPointSource MANAGED OBJECT CLASS
  DERIVED FROM   trailTerminationPointSource;
  CHARACTERIZED BY
  signalTrailTerminationPointSourcePackage PACKAGE
      BEHAVIOUR
          signalTrailTerminationPointSourceBehaviour;
      ATTRIBUTES
          signalTTPId GET,
          lineCoding GET;;;

REGISTERED AS { m3100ObjectClass 1012 };

signalTrailTerminationPointSourceBehaviour BEHAVIOUR
          DEFINED AS
          "This managed object creates a modulated/encoded signal ready for transmission.";

_____
[5] i.e. instance(s) of the object class(es) is created.

B.3 *Definition of attributes*

B.3.1 *Circuit TTP Id*

circuitTTPId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.NameType;
    MATCHES FOR EQUALITY;

REGISTERED AS { m3100Attribute 1001 };

B.3.2 *Information Rate*

informationRate ATTRIBUTE
    WITH ATTRIBUTE SYNTAX MODULE.InformationRate;
    MATCHES FOR EQUALITY;

REGISTERED AS { m3100Attribute 1002 };

B.3.3 *Line Coding*

lineCoding ATTRIBUTE
    WITH ATTRIBUTE SYNTAX MODULE.LineCoding;
    MATCHES FOR EQUALITY;

REGISTERED AS { m3100Attribute 1003 };

B.3.4 *Media TTP Id*

mediaTTPId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.NameType;
    MATCHES FOR EQUALITY;

REGISTERED AS { m3100Attribute 1004 };

B.3.5 *Media Type*

mediaType ATTRIBUTE
    WITH ATTRIBUTE SYNTAX MODULE.MediaType;
    MATCHES FOR EQUALITY;

REGISTERED AS { m3100Attribute 1005 };

B.3.6 *O and A TTP Id*

oAndATTPId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.NameType;
    MATCHES FOR EQUALITY;

REGISTERED AS { m3100Attribute 1006 };

B.3.7 *Signal TTP Id*

signalTTPId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.NameType;
    MATCHES FOR EQUALITY;

REGISTERED AS { m3100Attribute 1007 };

B.4 *Name Bindings*

B.4.1 *Circuit Termination Point Bidirectional*

circuitTerminationPointBidirectional-managedElement NAME BINDING
    SUBORDINATE OBJECT CLASS circuitTrailTerminationPointBidirectional;
    NAMED BY
    SUPERIOR OBJECT CLASS managedElement AND SUBCLASSES;
    WITH ATTRIBUTE circuitTTPId;
    BEHAVIOUR
        automaticInstanceNamingBehaviour;

```
        CREATE
            WITH-REFERENCE-OBJECT,
            WITH-AUTOMATIC-INSTANCE-NAMING;
        DELETE
            ONLY-IF-NO-CONTAINED-OBJECTS;


REGISTERED AS { m3100NameBinding 1001 };

circuitTerminationPointBidirectional-equipment NAME BINDING
    SUBORDINATE OBJECT CLASS circuitTrailTerminationPointBidirectional;
    NAMED BY
    SUPERIOR OBJECT CLASS equipment  AND SUBCLASSES;
    WITH ATTRIBUTE circuitTTPId;
    BEHAVIOUR
        automaticInstanceNamingBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;


REGISTERED AS { m3100NameBinding 1002 };
```

B.4.2   *Circuit Termination Point Sink*

```
circuitTerminationPointSink-managedElement NAME BINDING
    SUBORDINATE OBJECT CLASS circuitTrailTerminationPointSink;
    NAMED BY
    SUPERIOR OBJECT CLASS managedElement AND SUBCLASSES;
    WITH ATTRIBUTE circuitTTPId;
    BEHAVIOUR
        automaticInstanceNamingBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;


REGISTERED AS { m3100NameBinding 1003 };

circuitTerminationPointSink-equipment NAME BINDING
    SUBORDINATE OBJECT CLASS circuitTrailTerminationPointSink;
    NAMED BY
    SUPERIOR OBJECT CLASS equipment AND SUBCLASSES;
    WITH ATTRIBUTE circuitTTPId;
    BEHAVIOUR
        automaticInstanceNamingBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;


REGISTERED AS { m3100NameBinding 1004 };
```

B.4.3   *Circuit Termination Point Source*

```
circuitTerminationPointSource-managedElement NAME BINDING
    SUBORDINATE OBJECT CLASS circuitTrailTerminationPointSource;
    NAMED BY
    SUPERIOR OBJECT CLASS managedElement AND SUBCLASSES;
    WITH ATTRIBUTE circuitTTPId;
    BEHAVIOUR
        automaticInstanceNamingBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
```

DELETE
            ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1005 };

circuitTerminationPointSource-equipment NAME BINDING
    SUBORDINATE OBJECT CLASS circuitTrailTerminationPointSource;
    NAMED BY
    SUPERIOR OBJECT CLASS equipment AND SUBCLASSES;
    WITH ATTRIBUTE   circuitTTPId;
    BEHAVIOUR
        automaticInstanceNamingBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1006 };

B.4.4   *Media Termination Point Bidirectional*

mediaTerminationPointBidirectional-managedElement NAME BINDING
    SUBORDINATE OBJECT CLASS mediaTrailTerminationPointBidirectional;
    NAMED BY
    SUPERIOR OBJECT CLASS managedElement AND SUBCLASSES;
    WITH ATTRIBUTE mediaTTPId;
    BEHAVIOUR
        automaticInstanceNamingBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1007 };

mediaTerminationPointBidirectional-equipment NAME BINDING
    SUBORDINATE OBJECT CLASS   mediaTrailTerminationPointBidirectional;
    NAMED BY
    SUPERIOR OBJECT CLASS   equipment AND SUBCLASSES;
    WITH ATTRIBUTE   mediaTTPId;
    BEHAVIOUR
        automaticInstanceNamingBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1008 };

B.4.5   *Media Termination Point Sink*

mediaTerminationPointSink-managedElement NAME BINDING
    SUBORDINATE OBJECT CLASS mediaTrailTerminationPointSink;
    NAMED BY
    SUPERIOR OBJECT CLASS managedElement AND SUBCLASSES;
    WITH ATTRIBUTE mediaTTPId;
    BEHAVIOUR
        automaticInstanceNamingBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1009 };

mediaTerminationPointSink-equipment NAME BINDING
   SUBORDINATE OBJECT CLASS mediaTrailTerminationPointSink;
   NAMED BY
   SUPERIOR OBJECT CLASS equipment AND SUBCLASSES;
   WITH ATTRIBUTE mediaTTPId;
   BEHAVIOUR
      automaticInstanceNamingBehaviour;
   CREATE
      WITH-REFERENCE-OBJECT,
      WITH-AUTOMATIC-INSTANCE-NAMING;
   DELETE
      ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1010 };

B.4.6 *Media Termination Point Source*

mediaTerminationPointSource-managedElement NAME BINDING
   SUBORDINATE OBJECT CLASS mediaTrailTerminationPointSource;
   NAMED BY
   SUPERIOR OBJECT CLASS managedElement AND SUBCLASSES;
   WITH ATTRIBUTE mediaTTPId;
   BEHAVIOUR
      automaticInstanceNamingBehaviour;
   CREATE
      WITH-REFERENCE-OBJECT,
      WITH-AUTOMATIC-INSTANCE-NAMING;
   DELETE
      ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1011 };

mediaTerminationPointSource-equipment NAME BINDING
   SUBORDINATE OBJECT CLASS mediaTrailTerminationPointSource;
   NAMED BY
   SUPERIOR OBJECT CLASS equipment AND SUBCLASSES;
   WITH ATTRIBUTE mediaTTPId;
   BEHAVIOUR
      automaticInstanceNamingBehaviour;
   CREATE
      WITH-REFERENCE-OBJECT,
      WITH-AUTOMATIC-INSTANCE-NAMING;
   DELETE
      ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1012 };

B.4.7 *Overhead and Adaptation Termination Point Bidirectional*

overheadAndAdaptationTTPBidirectional-managedElement NAME BINDING
   SUBORDINATE OBJECT CLASS
      overheadAndAdaptationTrailTerminationPointBidirectional;
   NAMED BY
   SUPERIOR OBJECT CLASS   managedElement AND SUBCLASSES;
   WITH ATTRIBUTE   oAndATTPId;
   BEHAVIOUR
      automaticInstanceNamingBehaviour;
   CREATE
      WITH-REFERENCE-OBJECT,
      WITH-AUTOMATIC-INSTANCE-NAMING;
   DELETE
      ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1013 };

overheadAndAdaptationTTPBidirectional-equipment NAME BINDING
   SUBORDINATE OBJECT CLASS
      overheadAndAdaptationTrailTerminationPointBidirectional;
   NAMED BY
   SUPERIOR OBJECT CLASS   equipment AND SUBCLASSES;
   WITH ATTRIBUTE   oAndATTPId;
   BEHAVIOUR
      automaticInstanceNamingBehaviour;
   CREATE
      WITH-REFERENCE-OBJECT,
      WITH-AUTOMATIC-INSTANCE-NAMING;
   DELETE
      ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1014 };

B.4.8   *Overhead and Adaptation Termination Point Sink*

overheadAndAdaptationTTPSink-managedElement NAME BINDING
   SUBORDINATE OBJECT CLASS
      overheadAndAdaptationTrailTerminationPointSink;
   NAMED BY
   SUPERIOR OBJECT CLASS   managedElement AND SUBCLASSES;
   WITH ATTRIBUTE   oAndATTPId;
   BEHAVIOUR
      automaticInstanceNamingBehaviour;
   CREATE
      WITH-REFERENCE-OBJECT,
      WITH-AUTOMATIC-INSTANCE-NAMING;
   DELETE
      ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1015 };

overheadAndAdaptationTTPSink-equipment NAME BINDING
   SUBORDINATE OBJECT CLASS
      overheadAndAdaptationTrailTerminationPointSink ;
   NAMED BY
   SUPERIOR OBJECT CLASS   equipment AND SUBCLASSES;
   WITH ATTRIBUTE   oAndATTPId;
   BEHAVIOUR
      automaticInstanceNamingBehaviour;
   CREATE
      WITH-REFERENCE-OBJECT,
      WITH-AUTOMATIC-INSTANCE-NAMING;
   DELETE
      ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1016 };

B.4.9   *Overhead and Adaptation Termination Point Source*

overheadAndAdaptationTTPSource-managedElement NAME BINDING
   SUBORDINATE OBJECT CLASS
      overheadAndAdaptationTrailTerminationPointSource;
   NAMED BY
   SUPERIOR OBJECT CLASS   managedElement AND SUBCLASSES;
   WITH ATTRIBUTE   oAndATTPId;
   BEHAVIOUR
      automaticInstanceNamingBehaviour;
   CREATE
      WITH-REFERENCE-OBJECT,
      WITH-AUTOMATIC-INSTANCE-NAMING;
   DELETE
      ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1017 };

overheadAndAdaptationTTPSource-equipment NAME BINDING
      SUBORDINATE OBJECT CLASS
       overheadAndAdaptationTrailTerminationPointSource;
   NAMED BY
   SUPERIOR OBJECT CLASS   **equipment AND SUBCLASSES;**
   WITH ATTRIBUTE   **oAndATTPId;**
   BEHAVIOUR
     automaticInstanceNamingBehaviour;
   CREATE
     WITH-REFERENCE-OBJECT,
     WITH-AUTOMATIC-INSTANCE-NAMING;
   DELETE
     ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1018 };

B.4.10   *Signal Termination Point Bidirectional*

signalTerminationPointBidirectional-managedElement NAME BINDING
   SUBORDINATE OBJECT CLASS   **signalTrailTerminationPointBidirectional;**
   NAMED BY
   SUPERIOR OBJECT CLASS   **managedElement AND SUBCLASSES;**
   WITH ATTRIBUTE   **signalTTPId;**
   BEHAVIOUR
     automaticInstanceNamingBehaviour;
   CREATE
     WITH-REFERENCE-OBJECT,
     WITH-AUTOMATIC-INSTANCE-NAMING;
   DELETE
     ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1019 };

signalTerminationPointBidirectional-equipment NAME BINDING
   SUBORDINATE OBJECT CLASS signalTrailTerminationPointBidirectional;
   NAMED BY
   SUPERIOR OBJECT CLASS   **equipment AND SUBCLASSES;**
   WITH ATTRIBUTE   **signalTTPId;**
   BEHAVIOUR
     automaticInstanceNamingBehaviour;
   CREATE
     WITH-REFERENCE-OBJECT,
     WITH-AUTOMATIC-INSTANCE-NAMING;
   DELETE
     ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1020 };

B.4.11   *Signal Termination Point Sink*

signalTerminationPointSink-managedElement NAME BINDING
   SUBORDINATE OBJECT CLASS   **signalTrailTerminationPointSink;**
   NAMED BY
   SUPERIOR OBJECT CLASS   **managedElement AND SUBCLASSES;**
   WITH ATTRIBUTE   **signalTTPId;**
   BEHAVIOUR
     automaticInstanceNamingBehaviour;
   CREATE
     WITH-REFERENCE-OBJECT,
     WITH-AUTOMATIC-INSTANCE-NAMING;
   DELETE
     ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1021 };

signalTerminationPointSink-equipment NAME BINDING
    SUBORDINATE OBJECT CLASS   signalTrailTerminationPointSink;
    NAMED BY
    SUPERIOR OBJECT CLASS   equipment AND SUBCLASSES;
    WITH ATTRIBUTE   signalTTPId;
    BEHAVIOUR
        automaticInstanceNamingBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1022 };

B.4.12 *Signal Termination Point Source*

signalTerminationPointSource-managedElement NAME BINDING
    SUBORDINATE OBJECT CLASS   signalTrailTerminationPointSource;
    NAMED BY
    SUPERIOR OBJECT CLASS   managedElement AND SUBCLASSES;
    WITH ATTRIBUTE   signalTTPId;
    BEHAVIOUR
        automaticInstanceNamingBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1023 };

signalTerminationPointSource-equipment NAME BINDING
    SUBORDINATE OBJECT CLASS   signalTrailTerminationPointSource;
    NAMED BY
    SUPERIOR OBJECT CLASS   equipment AND SUBCLASSES;
    WITH ATTRIBUTE   signalTTPId;
    BEHAVIOUR
        automaticInstanceNamingBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS { m3100NameBinding 1024 };

B.5 *Supporting Productions*

```
MediaType ::= INTEGER {
        twistedPairCopper    (0),
        coaxial              (1),
        singleModeFiber      (2),
        multiModeFiber       (3),
        radio                (4),
        satellite            (5)
}
```

```
LineCoding ::= INTEGER {
        nRZ        (0),
        rZ         (1),
        diphase    (2),
        bipolar    (3),
        b6ZS       (4),
        b8ZS       (5),
        b3ZS       (6),
        ami        (7),
        amizcs     (8),
        hDB2       (9),
        hDB3      (10),
        cchan     (11)
}

InformationRate ::= INTEGER {
        dS1sf(10), dS1esf(11), zbtsi(12), tidm(14),
        cept1(20),
        dS1C(25),
        dS2(30),
        cept2(40),
        dS3async(50), dS3sync(51), dS3cbit(52), dS3pbit(53),
        dS4(60),
        dS4e(65),
        cept3(70),
        vC11(80), vC12(85), vC2(90), vC3(95), vC4(100),
        sTM1(110), sTM4(120), sTM16(130)

}
```

APPENDIX I

(to Recommendation M.3100)

Candidates for Management Information

I.1   *Introduction*

        This appendix proposes candidate object classes for further study.

I.2   *Object Classes*

I.2.1   *Organization*

        The Organization object class is a class of managed objects characterized by a specific group of people or organizations (e.g. a vendor, a customer, an owner) as viewed from the agent.

```
organization    MANAGED OBJECT CLASS
        DERIVED FROM "Recommendation X.721": top;
        CHARACTERIZED BY
        organizationPackage    PACKAGE
          BEHAVIOUR
                organization Behaviour BEHAVIOUR
                DEFINED AS
                "An organization may be instantiated by the agent system as the result of the internal processing
                (initialization, request, etc.,) or by managing system issuing a CREATE operation to the agent
                system. The relationshipChange notification is used when the contactNames and/or
                locationNames are changed.";
          ATTRIBUTES
                organization Id              GET,
                contactNames                 GET-REPLACE ADD-REMOVE,
                locationNames                GET-REPLACE ADD-REMOVE;
```

ATTRIBUTE GROUP
            "Recommendation X.721":relationship;
            -- consists of contactNames and locationNames --
        NOTIFICATIONS
            "Recommendation X.721":relationshipChange,
            "Recommendation X.721":objectCreation,
            "Recommendation X.721":objectDeletion;;;

REGISTERED AS {ccittObjectClass x };

I.2.2   *Location*

The Location object class is a class of managed objects characterized by a place that may be occupied by managed objects and has some address (e.g. geographic, postal, or telecommunications address).

Location   MANAGED OBJECT CLASS
        DERIVED FROM "Recommendation X.721":top;
        CHARACTERIZED BY
            BEHAVIOUR
                locationBehaviour BEHAVIOUR
                DEFINED AS
                "A location may be instantiated by the agent system as the result of the internal processing (initialization, request, etc.) or by managing system issuing a CREATE operation to agent system.";
            ATTRIBUTES
                locationId   GET,
                organizationNames   GET-REPLACE ADD-REMOVE,
                objectAtLocation   GET-REPLACE ADD-REMOVE;
            ATTRIBUTE GROUP
                "Recommendation X.721":relationshipChange,
                -- contains organizationNames and objectsAtLocation --
            NOTIFICATIONS
                "Recommendation X.721":relationshipChange,
                "Recommendation X.721" objectCreation,
                "Recommendation X.721":objectDeletion;;;

REGISTERED AS { ccittObjectClass x };

I.2.3   *Managed Function*

The Managed Function object class is a class of managed objects that characterize partitions of functions such as access side, truck side within a Network Element.

managedFunction   MANAGED OBJECT CLASS
        DERIVED FROM "Recommendation X.721":top;
        CHARACTERIZED BY
            BEHAVIOUR
                managedFunction BEHAVIOUR
                DEFINED AS
                "The managed function object class is a class of managed objects that are contained within a managed element. Instances of this object class can be used to partition functions of a managed element.";
            ATTRIBUTES
                managedFunctionId   GET,
                affectedObjectList   GET;;;
        CONDITIONAL PACKAGES
            userLabelPackage   PRESENT IF "an instance supports it",
            locationNamePackage   PRESENT IF "an instance supports it",

REGISTERED AS { ccittObjectClass x };

I.3 *Attributes*

I.3.1 *Location Names*

This attribute provides the list of locations at which this object instance is located.

locationNames   ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ASN1Module.LocationNames;
  MATCHES FOR Set Comparison Set Intersection;

REGISTERED AS { };

I.3.2 *Organization Names*

This attribute specifies the organizations that are located in the Location.

organizationNames   ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ASN1Module.ObjectInstanceList;
  MATCHES FOR Set Comparison Set Intersection;

REGISTERED AS { };

I.3.3 *Object At Locations*

This attribute specifies the List of managed objects (e.g. equipment, circuits) that are located in the Location.

objectAtLocations   ATTRIBUTE
  WITH ATTRIBUTE SYNTAX   ASN1Module.ObjectInstanceList;
  MATCHES FOR Set Comparison Set Intersection;

REGISTERED AS { };

I.3.4 *Vendor Name (to be replaced with the current definition)*

This attribute specifies the name of the vendor (i.e. an organization) which supplied this managed object.

vendorName   ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ASN1Module.VendorName;
  MATCHES FOR Equality;

REGISTERED AS { };

I.3.5 *Organization Id*

This attribute identifies the organization managed object instance.

organizationId   ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ASN1Module.Pstring;
  MATCHES FOR Equality;

REGISTERED AS { };

I.3.6 *Location Id*

This attribute identifies the geographic location managed object instance.

locationId   ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ASN1Module.Pstring;
  MATCHES FOR Equality;

REGISTERED AS { };

I.3.7   *Contact Names*

This attribute provides the list of names (e.g. a person, an office, or a department) that are main contacts of this organization.

contactNames   ATTRIBUTE
WITH ATTRIBUTE SYNTAX   ASN1Module.ObjectInstanceList;
MATCHES FOR Set Comparison Set Intersection;

REGISTERED AS { };

I.4   *ASN.1 Module*

LocationNames ::= SET OF {
CHOICE { PrintableString,
ObjectInstance } }

VendorName ::= SET OF {
CHOICE { PrintableString,
ObjectInstance } }

Pstring ::= PrintableString

References

[1]   CCITT Recommendation M.3010 *Principles for a Telecommunications Management Network, 1992.*

[2]   CCITT Recommendation X.720 *Management Information Model, 1992.*

[3]   CCITT Recommendation X.722 *Guidelines for Definition of Managed Objects, 1992.*

[4]   CCITT Recommendation *Specification of abstract syntax notation one, (ASN.1), BLUE Book, Vol. VIII.4, Rec. X.208, ITU, Geneva 1989.*

[5]   CCITT Recommendation X.721 *Definition of Management Information, 1992.*

[6]   CCITT Recommendation G.803 (temporary number was G.sna1 *Architectures of transport network based on the synchronous digital hierarchy (SDH), 1992.*

[7]   CCITT Recommendation Q.821 *Stage 2 and Stage 3 Description for the Q3 Interface, 1992.*