ATIS-0300003.2017(R2022)

XML Schema Interface for Fault Management (Trouble Administration)

AMERICAN NATIONAL STANDARD FOR TELECOMMUNICATIONS

As a leading technology and solutions development organization, the Alliance for Telecommunications Industry Solutions (ATIS) brings together the top global ICT companies to advance the industry's most pressing business priorities. ATIS' nearly 200 member companies are currently working to address the All-IP transition, 5G, network functions virtualization, big data analytics, cloud services, device solutions, emergency services, M2M, cyber security, network evolution, quality of service, billing support, operations, and much more. These priorities follow a fast-track development lifecycle — from design and innovation through standards, specifications, requirements, business use cases, software toolkits, open source solutions, and interoperability testing.

ATIS is accredited by the American National Standards Institute (ANSI). The organization is the North American Organizational Partner for the 3rd Generation Partnership Project (3GPP), a founding Partner of the oneM2M global initiative, a member of and major U.S. contributor to the International Telecommunication Union (ITU), as well as a member of the Inter-American Telecommunication Commission (CITEL). For more information, visit www.atis.org.

AMERICAN NATIONAL STANDARD

Approval of an American National Standard requires review by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made towards their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give an interpretation of any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

Notice of Disclaimer & Limitation of Liability

The information provided in this document is directed solely to professionals who have the appropriate degree of experience to understand and interpret its contents in accordance with generally accepted engineering or other professional standards and applicable regulations. No recommendation as to products or vendors is made or should be implied.

NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS TECHNICALLY ACCURATE OR SUFFICIENT OR CONFORMS TO ANY STATUTE, GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO REPRESENTATION OR WARRANTY IS MADE OFMERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. ATIS SHALL NOT BE LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN PAYMENT BY ATIS FOR THIS DOCUMENT, AND IN NO EVENT SHALL ATIS BE LIABLE FOR LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. ATIS EXPRESSLY ADVISES THAT ANY AND ALL USE OF OR RELIANCE UPON THE INFORMATION PROVIDED IN THIS DOCUMENT IS AT THE RISK OF THE USER.

NOTE - The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights. By publication of this standard, no position is taken with respect to whether use of an invention covered by patent rights will be required, and if any such use is required no position is taken regarding the validity of this claim or any patent rights in connection therewith. Please refer to [http://www.atis.org/legal/patentinfo.asp] to determine if any statement has been filed by a patent holder indicating a willingness to grant a license either without compensation or on reasonable and non-discriminatory terms and conditions to applicants desiring to obtain a license.

ATIS-0300003.2017(R2022), *XML Schema Interface for Fault Management (Trouble Administration)*

Is an American National Standard developed by the Telecom Management and Operations Committee (TMOC).

Published by

Alliance for Telecommunications Industry Solutions
1200 G Street, NW, Suite 500
Washington, DC 20005

ATIS-0300003.2017(R2022)

[Revision of ATIS-0300003.2015]

American National Standard for Telecommunications

XML Schema Interface for Fault Management (Trouble Administration)

Alliance for Telecommunications Industry Solutions

Approved November 16, 2016

American National Standards Institute, Inc.

Abstract

This standard provides an XML schema information model for Trouble Administration and an XML schema interface for Trouble Administration functions and services. Additional information from the original CMIP-based Trouble Administration standards ATIS-0300227.2008 and ATIS-0300228.2011 can be found in informative annexes to this document. It is the intention that this document be used for current implementations of Electronic Bonding, which to this Committee's knowledge are all implemented using XML. The informative annexes have been included here so as not to lose context information that may still prove to be of value.

Foreword

The information contained in this Foreword is not part of this American National Standard (ANS) and has not been processed in accordance with ANSI's requirements for an ANS. As such, this Foreword may contain material that has not been subjected to public review or a consensus process. In addition, it does not contain requirements necessary for conformance to the Standard.

The Alliance for Telecommunication Industry Solutions (ATIS) serves the public through improved understanding between carriers, customers, and manufacturers. The Telecom Management and Operations Committee (TMOC) -- formerly T1M1 -- develops operations, administration, maintenance and provisioning standards, and other documentation related to Operations Support System (OSS) and Network Element (NE) functions and interfaces for communications networks – with an emphasis on standards development related to U.S.A. communication networks in coordination with the development of international standards.

ANSI guidelines specify two categories of requirements: mandatory and recommendation. The mandatory requirements are designated by the word *shall* and recommendations by the word *should*. Where both a mandatory requirement and a recommendation are specified for the same criterion, the recommendation represents a goal currently identifiable as having distinct compatibility or performance advantages.

Suggestions for improvement of this document are welcome. They should be sent to the Alliance for Telecommunications Industry Solutions, TMOC, 1200 G Street NW, Suite 500, Washington, DC 20005.

At the time it approved this document, TMOC, which is responsible for the development of this Standard, had the following leadership:

M. Usry, TMOC Chair (iconectiv)
T. Barrett, Technical Editor (AT&T)

Table of Contents

Table of Figures

Table of Tables

American National Standard for Telecommunications –

XML Schema Interface for Fault Management (Trouble Administration)

1 Scope, Purpose, & Application

The *tML Framework Document* (ITU-T Recommendation M.3030) identifies potential applications for using XML in the interface design specification. Fault Management (Trouble Administration) is one such application. ITU-T Recommendation M.3020 defines a three phase methodology for developing interface specifications related to management information exchanges. The requirements phase has not been formally defined but is derived from former ANSI standard ATIS-0300227.2008 (included here as Annexes A through H) and former ANSI standard ATIS-0300228.2011 (included here as Annex I). The formal analysis phase from M.3020 is not included in this document. This document presents the design phase using XML schemas (tML TA schemas) for Trouble Administration functions and services specified in Annexes A through I. The design phase is a result of analyzing (in the form of UML model) ATIS-0300227/ATIS-0300228 data definitions. The model does not include enhanced features such as multiple object operations available based on the protocol design in Annexes A through I. A Trouble Administration architecture diagram based on Annexes A through I is shown in Figure 1.1 below.

The approach used in this standard defines XML application protocol data units that are specific to a subset of TA as defined in Annexes A through H and Annex I (ANSI version of ITU-T X.790). The XML messages specified in this standard are defined via an XML schema (tML-TA schema) and there is no coupling to a transport protocol. While mapping to a transport protocol (such as tML Transport Protocol, see Ref [15]) is required for interoperability, it is outside the scope of this document, and should be addressed by each pair of bonded companies as a joint implementation issue.

Informative Annexes A through H define an information model using GDMO (ITU-T Recommendation X.722) and ASN.1 (ITU-T Recommendation X.680) for the TMN X-interface (ITU-T Recommendation M.3010) to support Fault Management. These clauses are applicable when Common Management Information Service Element (CMISE) is used for conveying inquiry information across an OS-OS interactive interface.

Informative Annex J specifies a functionally equivalent interface using the Common Object Request Broker Architecture (CORBA)/Interface Definition Language (IDL), as defined in *The Common Object Request Broker Architecture and Specification, Revision 2.2, Object Management Group, Feb 1998.*

Since this Committee is no longer aware of CMISE or CORBA based implementations supporting tML framework, these references have been moved to the informative annexes.

Trouble Administration Architecture

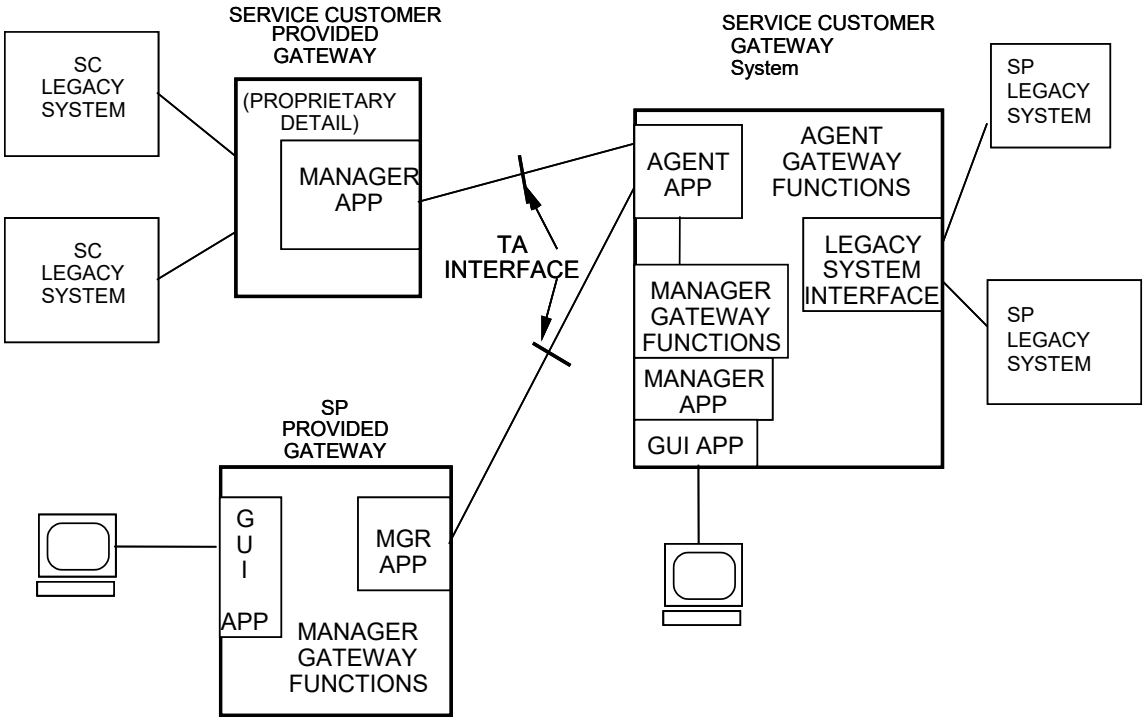


Figure 1.1 – Trouble Administration Architecture Diagram

1.1 Purpose

The purpose of this document is to provide the XML schemas (tML TA schemas) for trouble administration functions and services specified in Annexes A through I.

1.2 Abbreviations & Acronyms

For the purposes of this standard, the following abbreviations are used:

ANSI	American National Standards Institute
ASN.1	Abstract Syntax Notation One
ATIS	Alliance for Telecommunications Industry Solutions
CMIP	Common Management Information Protocol
CMIS	Common Management Information Service
CMISE	Common Management Information Service Element
CNM	Customer Network Management
DN	Distinguished Name
FM	Fault Management
GDMO	Guidelines for the Definition of Managed Objects
GNIM	General Network Information Model
GNM	General Network Model
HTTP (S)	Hyper Text Transport Protocol (Secure Sockets)
PDU	Protocol Data Unit
RDN	Relative Distinguished Name

SOAP	Simple Object Access Model
TA	Trouble Administration (part of FM)
tML	telecommunications Markup Language
TMN	Telecommunications Management Network
TR	Trouble Report
W3C	World Wide Web Consortium
WSDL	Web Service Description Language
WS-I	Web Services Interoperability Organization
XML	Extensible Markup Language

1.3 Conventions Used in This Document

The XML Schema version used is < http://www.w3.org/2001/XMLSchema >, for the W3C's May 2nd, 2001 XML Schema Recommendation.

2 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this American National Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this American National Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

[1] *Extensible Markup Language (XML) 1.0, Second Edition*, Tim Bray *et al*., eds,. W3C, latest version.[1]

[2] *Namespaces in XML*, Tim Bray *et al*., eds,. W3C, 14 January 1999.[2]

[3] *XML Schema Part 0: Primer*, David C. Fallside, *et al*., eds,. W3C Recommendation, latest version.[3]

[4] *XML Schema Part 1: Structures*, Henry Thompson *et al*., eds,. W3C Recommendation, latest version[4].

[5] *XML Schema Part 2: Datatypes*, Paul Biron *et al*., eds,. W3C Recommendation, latest version.[5]

[6] ATIS-0300240.2014, *Operations, Administration, Maintenance, and Provisioning (OAM&P) –Generic Network Information Model for Interface between Operations Systems and Network Element*.[6]

[7] ISO/IEC 9595: 1998, *Information technology – Open systems interconnection – Common management information service definition*.[7]

[8] ITU-T Recommendation M.3020 (2011), *Telecommunications management network Management interface specification methodology*.[8]

[9] ITU-T Recommendation M.3030 (2002), *tML Framework Document*.[9]

[10] ITU-T Recommendation X.790 (1995), *Data Network and Open System Communications – OSI Management – Trouble Management Function for ITU-T Applications*. [9]

[11] ITU-T Recommendation X.680 (2008) | ISO/IEC 8824-1:2008, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation (plus Amendments and Corrigendum)*. [9]

[1] See < http://www.w3.org/TR/REC-xml >.

[2] See < http://www.w3.org/TR/REC-xml-names/ >.

[3] See < http://www.w3.org/TR/xmlschema-0/ >.

[4] See < http://www.w3.org/TR/xmlschema-1/ >.

[5] See < http://www.w3.org/TR/xmlschema-2/ >.

[6] This document is available from the Alliance for Telecommunications Industry Solutions (ATIS), 1200 G Street N.W., Suite 500, Washington, DC 20005. < https://www.atis.org/docstore/product.aspx?id=24712 >

[7] This document is available from the International Electrotechnical Commission. < http://www.iec.ch />

[8] This document is available from the International Telecommunications Union. < http://www.itu.int/ITU-T/ >

[12] ATIS-0300094.2014, *Trouble Codes in Support of ATIS Trouble Administration Standards*.[9]

3 Trouble Administration: Design Phase for UML & XML Representations

This document presents a design using an XML model derived from the model in Annexes A through H and Annex I (ANSI version of ITU-T X.790). In developing this standard, the UML representation of data type is provided to facilitate future migration to a protocol neutral model conforming to the methodology in M.3020. This clause contains two steps for the management information: 1) using UML for mapping the data types from ASN.1, and 2) defining the XML schema.

Because the management information required for the analysis has been included in Annexes A through H using ASN.1 notation, the approach taken here is to develop the UML representation for the management information in ASN.1 notation and define using XML, application protocol specific design. The XML message definition is exchanged on the interface between service customer and service provider using a request/response protocol data unit.

The explanation of the approach for developing the tML TA interface is detailed in different clauses of this document. Clause 3.1 defines the mapping rules between ASN.1 and UML for representing the management information in a protocol independent approach. Clause 3.2 specifies the rules for generating XML schema for trouble administration functions and services. The detailed class diagram for trouble administration services including Request, Response, and Notification, as well as attribute types will be presented in clauses 3.3 and 3.4.

The tML TA interface consists of three categories based on the message exchange between service customer and service provider: *Request*, *Response*, and *Notification*. A *request* represents the message transmitted from service customer to service provider initiated by service customer. A *response* corresponds to the message conveyed from service provider to service customer resulting from a Request. A *notification*, on the other hand, is the unsolicited message emitted by service provider for service customer. Users of this standard should consult Annexes A through I and the generic implementation guidelines for Trouble Administration (see Ref [13]). For example, for a certain modification of an attribute value, the type of modification shall be consistent with those defined for the attribute in Annexes A through H.

This proposed standard uses upper camel case (i.e., UpperCamelCase) for XML tag definition throughout the implementation of tML TA schemas.

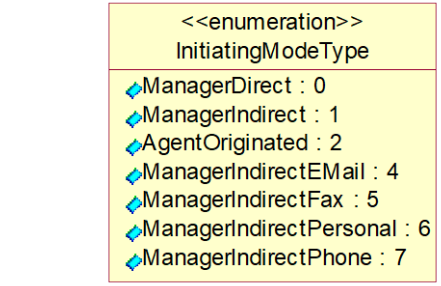
3.1 Attribute Types Mapping Rules

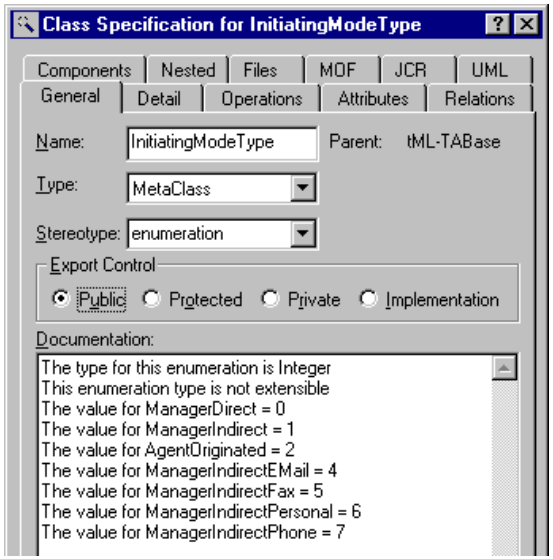
This clause provides explanations on mapping rules for a given ASN.1 definition to a corresponding XML UML notation and XML schema. The syntaxes defined are for the types in Annexes A through I. In cases where multiple type definitions exist for the same base ASN.1 type, only one XML mapping rule is provided. The data structures are applicable to trouble administration even though other applications may find them useful.

[9] This document is available from the Alliance for Telecommunications Industry Solutions (ATIS), 1200 G Street N.W., Suite 500, Washington, DC 20005. < https://www.atis.org/docstore/product.aspx?id=26770 >

Table 3.1 – INTEGER "Group"

<table>
<tr><th>ASN.1 Definition</th><th>UML Model</th><th>Schema</th></tr>
<tr>
<td>
InitiatingMode::=INTEGER{
managerDirect (0),
managerIndirect (1),
agentOriginated (2),
managerIndirectEMail (4),
managerIndirectFax (5),
managerIndirectPersonal (6),
managerIndirectPhone (7)
}

Excerpt from X.680
3.8.40 integer type: A simple type with distinguished values which are positive and negative whole numbers, including zero (as a single value).
</td>
<td>




</td>
<td>
<pre>
<simpleType name="InitiatingModeType">
  <restriction base="integer">
    <annotation>
      <documentation>
        ManagerDirect = 0
        ManagerIndirect = 1
        AgentOriginated = 2
        ManagerIndirectEMail = 4
        ManagerIndirectFax = 5
        ManagerIndirectPersonal = 6
        ManagerIndirectPhone = 7
      </documentation>
    </annotation>
    <enumeration value="0"/>
    <enumeration value="1"/>
    <enumeration value="2"/>
    <enumeration value="4"/>
    <enumeration value="5"/>
    <enumeration value="6"/>
    <enumeration value="7"/>
  </restriction>
</simpleType>
</pre>
</td>
</tr>
</table>

Table 3.2 – ENUMERATION

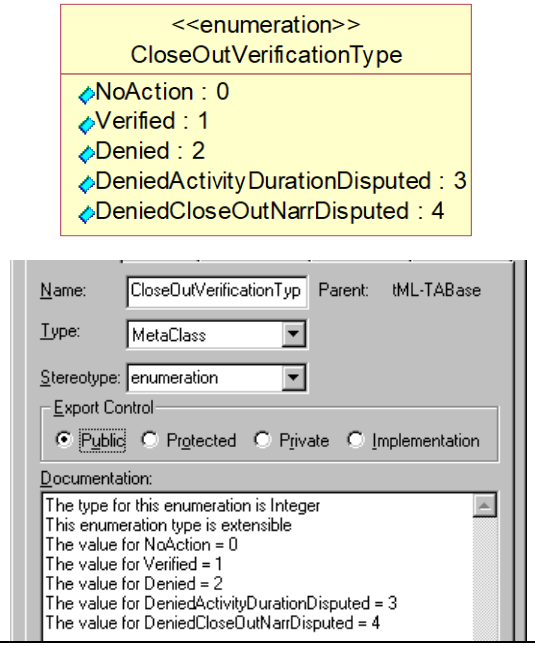
ASN.1 Definition

```
CloseOutVerification::=ENUMERATED{
        noAction                        (0),
        verified                        (1),
        denied                          (2),
        deniedActivityDurationDisputed  (3),
        deniedCloseOutNarrDisputed      (4),
        ...
        }
```

Excerpt from X.680

3.8.24 enumerated types: Simple types whose values are given distinct identifiers as part of the type notation.

UML Model



Schema

```
<simpleType
  name="CloseOutVerificationType">
    <restriction base="integer">
    <annotation>
            <documentation>
                NoAction = 0
                Verified = 1
                Denied = 2
                DeniedActivityDurationDisputed = 3
                DeniedCloseOutNarrDisputed = 4
            </documentation>
        </annotation>
        <enumeration value="0"/>
        <enumeration value="1"/>
        <enumeration value="2"/>
        <enumeration value="3"/>
        <enumeration value="4"/>
    </restriction>
</simpleType>
```

Table 3.3 – OBJECT IDENTIFIER

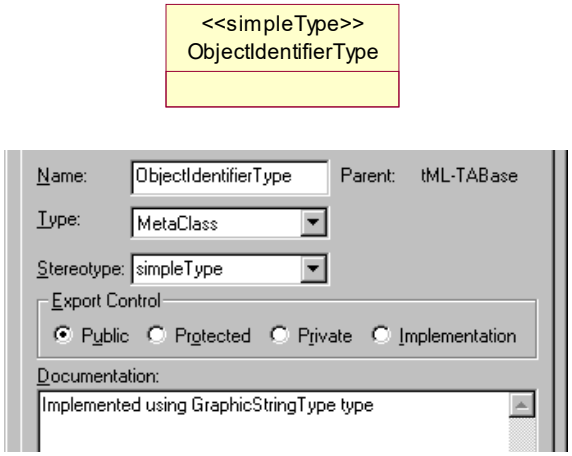
Excerpt from X.680	UML Model	Schema
3.8.46 object identifier: A value (distinguishable from all other such values) which is associated with an object. **3.8.47 object identifier type**: A simple type whose distinguished values are the set of all object identifiers allocated in accordance with the rules of ITU-T Rec. X.660 | ISO/IEC 9834-1. **31.1** The object identifier type (see 3.8.47) shall be referenced by the notation "ObjectIdentifierType": **ObjectIdentifierType ::= OBJECT IDENTIFIER**		<simpleType name="ObjectIdentifierType"> <annotation> <documentation> When an ObjectIdentifier is being used for creating a company specific extension (seen as Identifier tag), use a company's ECC or ACNA as the prefix, followed by an underscore ("_") then the company own definition obeying W3C XML tag rules to ensure uniqueness. </documentation> </annotation> <restriction base="tML-TABase:GraphicStringType"/> </simpleType>

Table 3.4 – CHOICE: Case 1

ASN.1 Definition

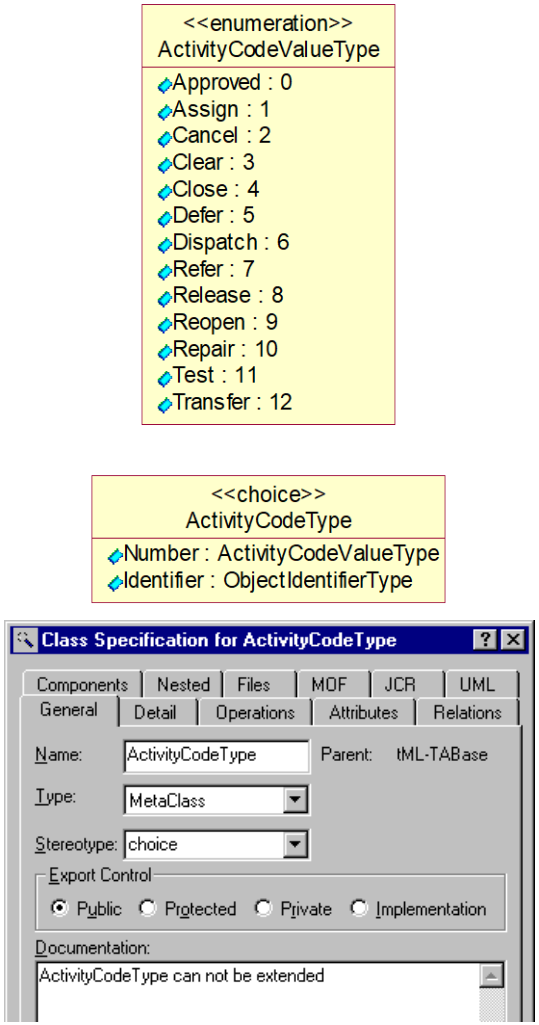
```
ActivityCode::=CHOICE{
    number INTEGER{
        Approved        (0),
        assign          (1),
        cancel          (2),
        clear           (3),
        close           (4),
        defer           (5),
        dispatch        (6),
        refer           (7),
        release         (8),
        re-open         (9),
        repair      (10),
        test            (11),
        transfer        (12)
    },
    identifier OBJECT IDENTIFIER
}
```

Excerpt from X.680

3.8.13 choice types: Types defined by referencing a list of distinct types; each value of the choice type is derived from the value of one of the component types.

3.8.14 component type: One of the types referenced when defining a CHOICE, SET, SEQUENCE, SET OF, or SEQUENCE OF.

UML Model



Schema

```
<simpleType name="ActivityCodeValueType">
    <restriction base="integer">
        <enumeration value="0"/>
        <enumeration value="1"/>
        <enumeration value="2"/>
        <enumeration value="3"/>
        <enumeration value="4"/>
        <enumeration value="5"/>
        <enumeration value="6"/>
        <enumeration value="7"/>
        <enumeration value="8"/>
        <enumeration value="9"/>
        <enumeration value="10"/>
        <enumeration value="11"/>
        <enumeration value="12"/>
    </restriction>
</simpleType>

<complexType name="ActivityCodeType" final="#all">
    <choice>
        <element name="Number"
          type="tML-TABase:ActivityCodeValueType"/>
        <element name="Identifier"
          type="tML-TABase:ObjectIdentifierType"/>
    </choice>
</complexType>
```

Schema Data Structure Diagram

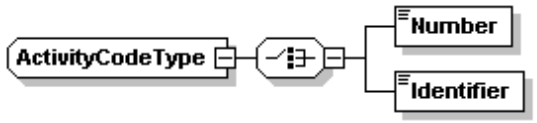


Table 3.5 – CHOICE: Case 2

ASN.1 Definition	UML Model	Schema

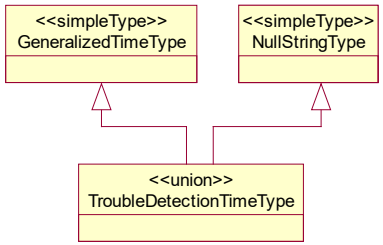
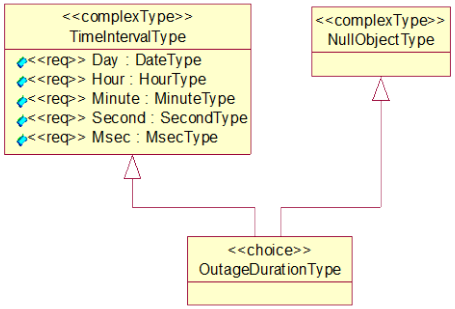
TroubleDetectionTime::=CHOICE{
NULL,
GeneralizedTime,
…
}
		<simpleType name="TroubleDetectionTimeType">
		<union>
		<simpleType>
		<restriction base="tML-TABase:NullStringType"/>
		</simpleType>
		<simpleType>
		<restriction base="tML-TABase:GeneralizedTimeType"/>
		</simpleType>
		</union>
		</simpleType>
		<simpleType name="GeneralizedTimeType">
		<restriction base="dateTime"/>
		</simpleType>

Table 3.6 – CHOICE: Case 3

<table>
<tr><th>ASN.1 Definition</th><th>UML Model</th><th>Schema</th></tr>
<tr>
<td>OutageDuration::=CHOICE{
NULL,
TimeInterval,
...
}</td>
<td> 
</td>
<td><pre>
<complexType name="NullObjectType" final="#all"/>

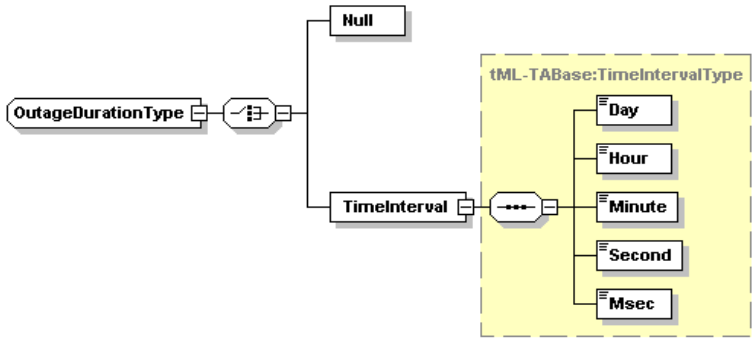
<complexType name="OutageDurationType">
  <choice>
    <element name="Null" type="tML-TABase:NullObjectType"/>
    <element name="TimeInterval"
      type="tML-TABase:TimeIntervalType"/>
  </choice>
</complexType>
</pre></td>
</tr>
<tr><th colspan="3">Schema Data Structure Diagram</th></tr>
<tr><td colspan="3"> 
</td></tr>
</table>

Table 3.7 – Artificial "CHOICE" Creation for Accommodating Two Definitions

ASN.1 Definition

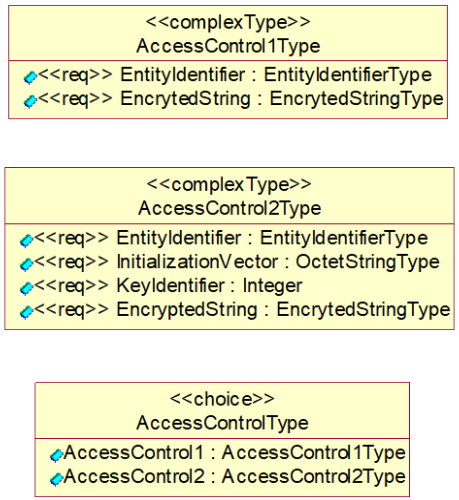
(a) Abstract Syntax 1

```
AccessControl ::= SEQUENCE {
  entityIdentifier [0] IMPLICIT
        GraphicString (SIZE (1..64)),
  encryptedString [1] IMPLICIT
        OCTET STRING (SIZE (1..64))
}
```

(b) Abstract Syntax 2

```
AccessControl ::= SEQUENCE {
  entityIdentifier [0] IMPLICIT
        GraphicString (SIZE 1..64)),
  initializationVector[1]IMPLICIT
        OCTET STRING (SIZE (8)),
  keyIdentifier [2]
        IMPLICIT INTEGER,
  encryptedString [3] IMPLICIT
        OCTET STRING (SIZE (8..64))
}
```

UML Model



Schema

```
<complexType name="AccessControl2Type" final="#all">
    <sequence>
        <element name="EntityIdentifier">
            <simpleType>
                <restriction
          base="tML-TABase:GraphicStringType">
                    <minLength value="1"/>
                    <maxLength value="64"/>
                </restriction>
            </simpleType>
        </element>
        <element name="InitializationVector">
            <simpleType>
                <restriction
          base="tML-TABase:OctetStringType">
                    <length value="8"/>
                </restriction>
            </simpleType>
        </element>
        <element name="KeyIdentifier" type="integer"/>
        <element name="EncrytedString">
            <simpleType>
                <restriction
          base="tML-TABase:OctetStringType">
                    <minLength value="8"/>
                    <maxLength value="64"/>
                </restriction>
            </simpleType>
        </element>
    </sequence>
</complexType>

<complexType name="AccessControlType">
    <choice>
        <element name="AccessControl1"
          type="tML-TABase:AccessControl1Type"/>
        <element name="AccessControl2"
          type="tML-TABase:AccessControl2Type"/>
    </choice>
</complexType>
```

Schema Data Structure Diagram

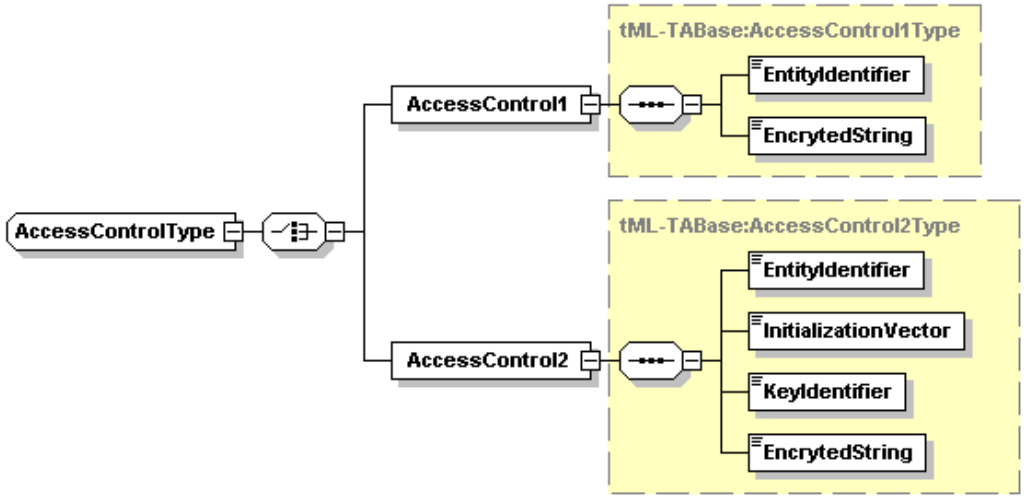


Table 3.8 – BIT STRING

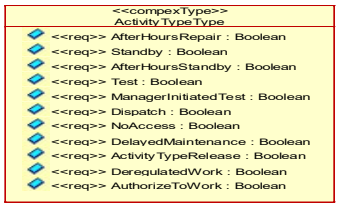
ASN.1 Definition

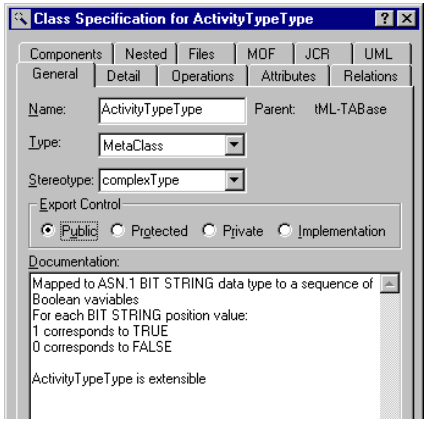
```
ActivityType::= BIT STRING {
    after-hours-repair          (0),
    standby                     (1),
    after-hours-standby         (2),
    test                        (3),
    manager-initiated-test(4),
    dispatch                    (5),
    no-access
    (6),
    delayed-maintenance         (7),
    release                     (8),
    deregulatedWork             (9),
    authorize-to-work          (10),
    …
    }
```

Excerpt from X.680

3.8.6 bitstring type: A simple type whose distinguished values are an ordered sequence of zero, one, or more bits.

UML Model





Schema

```
<complexType name="ActivityTypeType">
    <sequence>
        <element name="AfterHoursRepair" type="boolean"/>
        <element name="Standby" type="boolean"/>
        <element name="AfterHoursStandby" type="boolean"/>
        <element name="Test" type="boolean"/>
        <element name="ManagerInitiatedTest" type="boolean"/>
        <element name="Dispatch" type="boolean"/>
        <element name="NoAccess" type="boolean"/>
        <element name="DelayedMaintenance" type="boolean"/>
        <element name="Release" type="boolean"/>
        <element name="DeregulatedWork" type="boolean"/>
        <element name="AuthorizeToWork" type="boolean"/>    </sequence>
</complexType>
```

Schema Data Structure Diagram

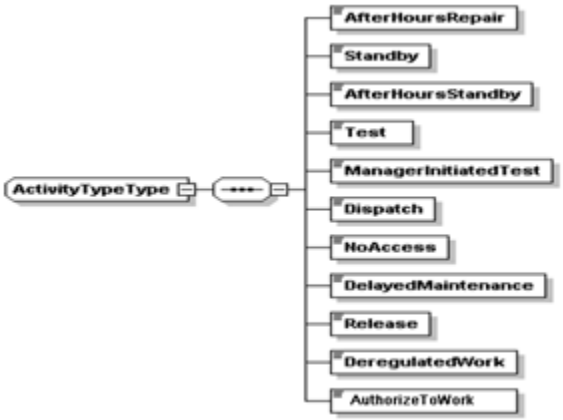


Table 3.9 – SEQUENCE: Case 1

ASN.1 Definition

```
PersonReach::= SEQUENCE{
      number    [0] PersonNumber DEFAULT "",
      name            [1] PersonName DEFAULT "",
      phone           [2] PersonPhone OPTIONAL,
      loc             [3] PersonLocation OPTIONAL,
      email           [4] PersonEmail OPTIONAL,
      fax             [5] PersonFax OPTIONAL,
      respon          [6] PersonRespon OPTIONAL,
      pager           [7] PersonPhone OPTIONAL,
      …
}
```

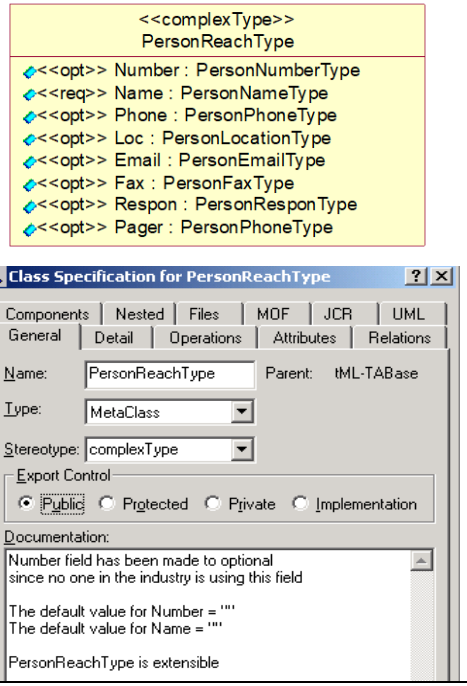
Excerpt from X.680

3.8.56 sequence types: Types defined by referencing an ordered list of types (some of which may be declared to be optional); each value of the sequence type is an ordered list of values, one from each component type.

Schema

```
<complexType name="PersonReachType">
  <annotation>
      <documentation>The reason for mapping Number field as optional is that
          no one in the industry is using this field.
      </documentation>
    </annotation>
    <sequence>
      <element name="Number" minOccurs="0" default=""
        type="tML-TABase:PersonNumberType"/>
      <element name="Name" default="" type="tML-TABase:PersonNameType"/>
      <element name="Phone" minOccurs="0" type="tML-TABase:PersonPhoneType"/>
      <element name="Loc" minOccurs="0" type="tML-TABase:PersonLocationType"/>
      <element name="Email" minOccurs="0" type="tML-TABase:PersonEmailType"/>
      <element name="Fax" minOccurs="0" type="tML-TABase:PersonFaxType"/>
      <element name="Respon" minOccurs="0" type="tML-TABase:PersonResponType"/>
      <element name="Pager" minOccurs="0" type="tML-TABase:PersonPhoneType"/>
    </sequence>
</complexType>
```

UML Model	Schema Data Structure Diagram



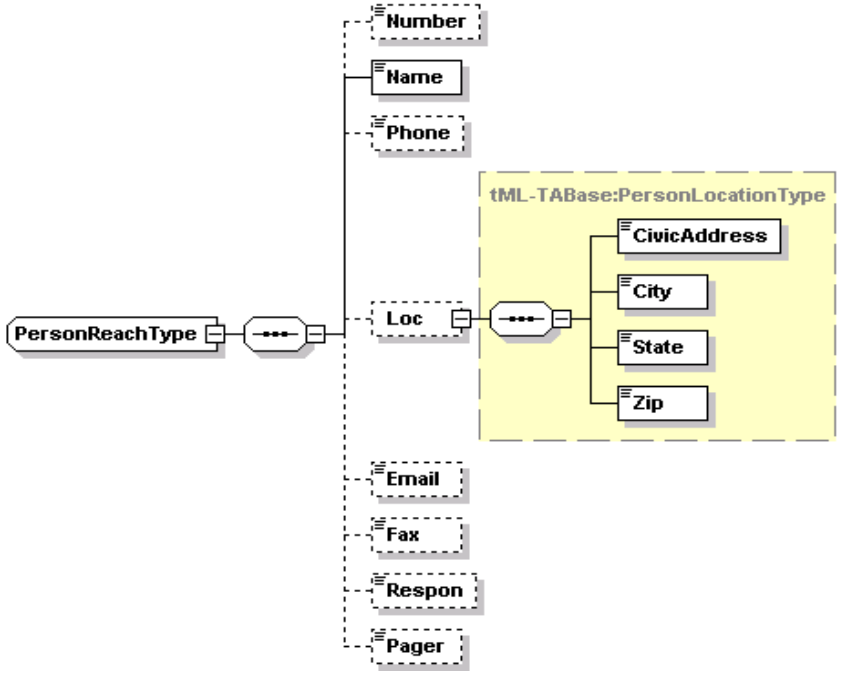
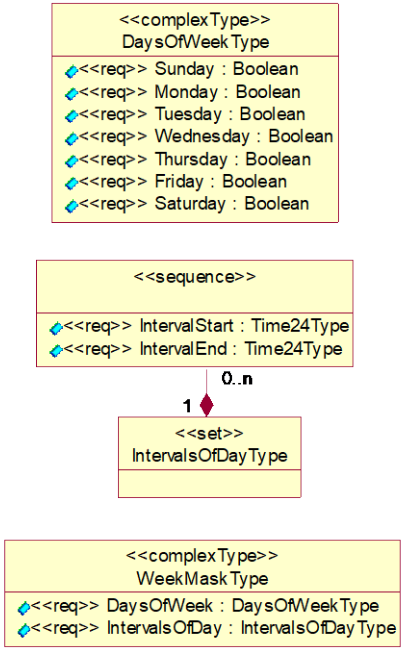


Table 3.10 – SEQUENCE: Case 2

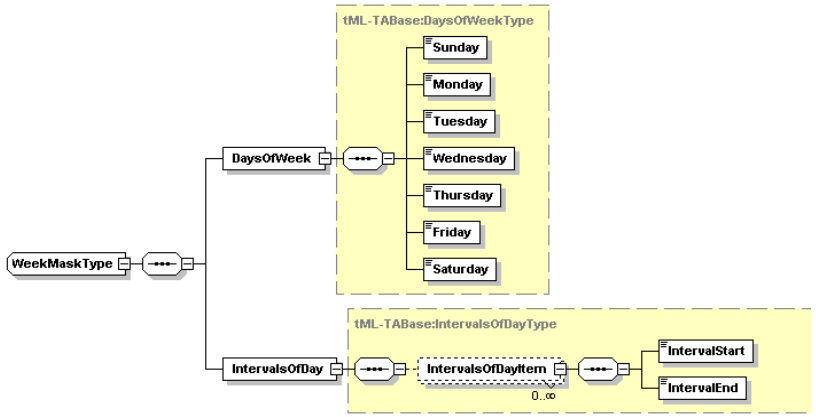
ASN.1 Definition

```
WeekMask::=SEQUENCE {
  daysOfWeek BIT STRING {
      sunday          (0),
      monday          (1),
      tuesday         (2),
      wednesday       (3),
      thursday        (4),
      friday          (5),
      saturday        (6),
      ...
    }
 DEFAULT '1111111'B,

      intervalsOfDay      SET      OF
SEQUENCE {
            intervalStart       Time24,
            intervalEnd
Time24,
            ...
            }
      DEFAULT{{ {0,0},{23,59}}},
      ...
}
```

UML Model



Schema Data Structure Diagram



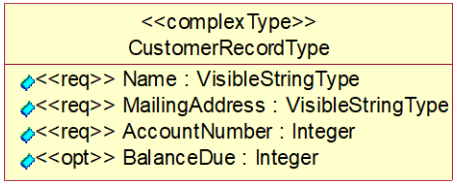
Schema

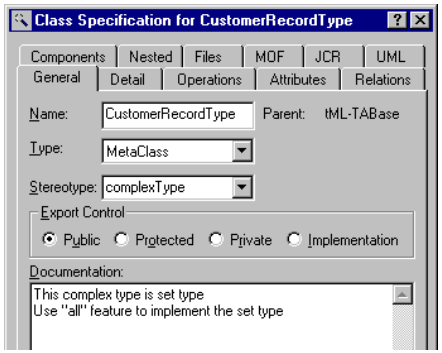
```
<complexType name="DaysOfWeekType">
    <sequence>
        <element name="Sunday" type="boolean" default="true"/>
        <element name="Monday" type="boolean" default="true"/>
        <element name="Tuesday" type="boolean" default="true"/>
        <element name="Wednesday" type="boolean" default="true"/>
        <element name="Thursday" type="boolean" default="true"/>
        <element name="Friday" type="boolean" default="true"/>
        <element name="Saturday" type="boolean" default="true"/>
    </sequence>
</complexType>

<complexType name="IntervalsOfDayType">
    <sequence>
        <element name="IntervalsOfDayItem" minOccurs="0"
          maxOccurs="unbounded">
            <complexType>
                <sequence>
                    <element name="IntervalStart"
                type="tML-TABase:Time24Type"
                default="00:00:00"/>
                    <element name="IntervalEnd"
                type="tML-TABase:Time24Type"
                default="23:59:00"/>
                </sequence>
            </complexType>
        </element>
    </sequence>
</complexType>

<complexType name="WeekMaskType">
    <sequence>
        <element name="DaysOfWeek" type="tML-TABase:DaysOfWeekType"/>
        <element name="IntervalsOfDay"
          type="tML-TABase:IntervalsOfDayType"/>
    </sequence>
</complexType>


<simpleType name="Time24Type">
    <restriction base="time"/>
</simpleType>
```

Table 3.11 – SET

<table>
<tr><th>ASN.1 Definition (X.680)</th><th>UML Model</th><th>Schema</th></tr>
<tr>
<td>

```
CustomerRecord ::= SET {
      name                [0] VisibleString,
      mailingAddress            [1] VisibleString,
      accountNumber       [2] INTEGER,
      balanceDue          [3] INTEGER
OPTIONAL,
      …
}
```

</td>
<td rowspan="2">




</td>
<td rowspan="2">

```
<complexType name="CustomerRecordType">
    <all>
        <element name="Name"
          type="tML-TABase:VisibleStringType"/>
        <element name="MailingAddress"
          type="tML-TABase:VisibleStringType"/>
        <element name="AccountNumber" type="integer"/>
        <element name="BalanceDue"
          type="integer" minOccurs="0"/>
    </all>
</complexType>
```

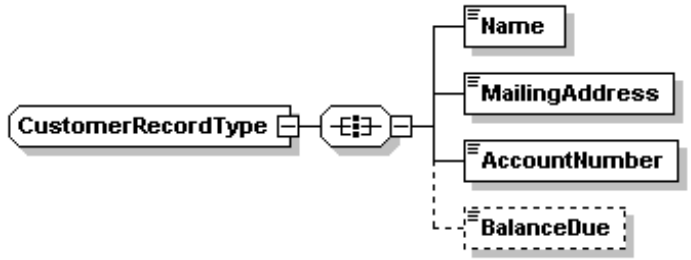
</td>
</tr>
<tr>
<td>

Excerpt from X.680

3.8.58 set types: Types defined by referencing a fixed, unordered, list of distinct types (some of which may be declared to be optional); each value in the set type is an unordered list of values, one from each of the component types.

</td>
</tr>
<tr>
<td colspan="3">

Schema Data Structure Diagram



</td>
</tr>
</table>

Table 3.12 – SET OF SEQUENCE

<table>
<tr><th>ASN.1 Definition</th><th>UML Model</th><th>Schema</th></tr>
<tr>
<td>ActivityDuration::=SET OF SEQUENCE{
duration [0] TimeInterval,
billable [1] BOOLEAN DEFAULT TRUE,
type [2] ActivityType OPTIONAL,
…
}

Excerpt from X.680

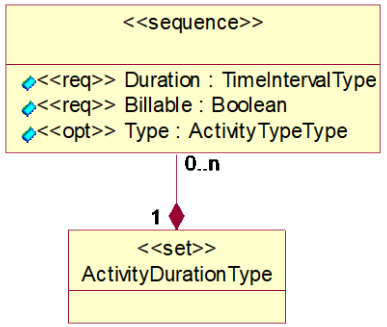
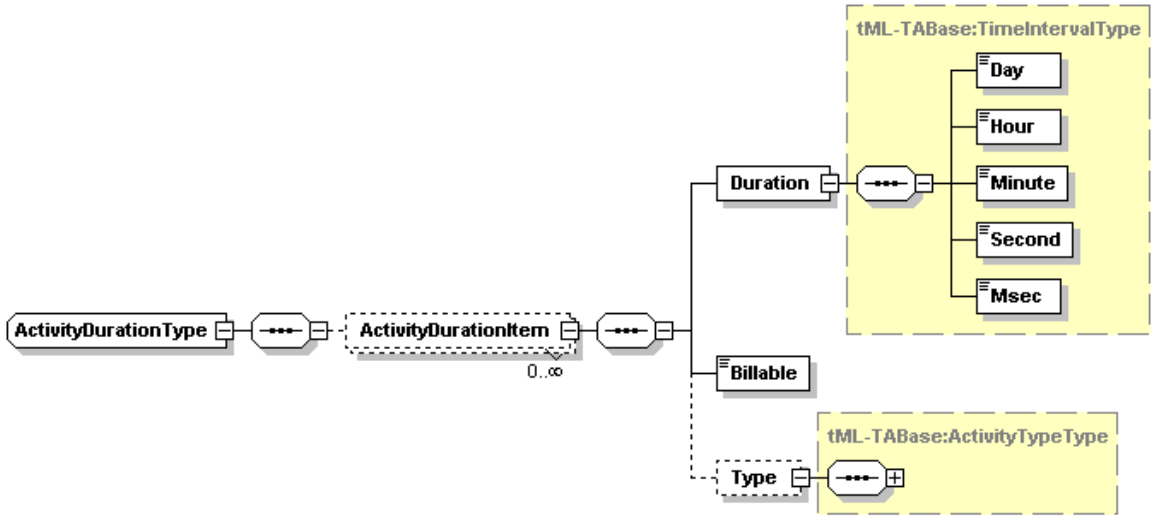
3.8.59 set-of types: Types defined by referencing a single component type; each value in the set-of type is an unordered list of zero, one, or more values of the component type.</td>
<td> 
</td>
<td><pre>
<complexType name="ActivityDurationType">
  <sequence>
    <element name="ActivityDurationItem" minOccurs="0"
     maxOccurs="unbounded">
      <complexType>
        <sequence>
          <element name="Duration"
         type="tML-TABase:TimeIntervalType"/>
          <element name="Billable"
         type="boolean" default="true"/>
          <element name="Type"
         type="tML-TABase:ActivityTypeType"
         minOccurs="0"/>
        </sequence>
      </complexType>
    </element>
  </sequence>
</complexType>
</pre></td>
</tr>
<tr><td colspan="3">Schema Data Structure Diagram</td></tr>
<tr><td colspan="3"> 
</td></tr>
</table>

Table 3.13 – SET OF CHOICE

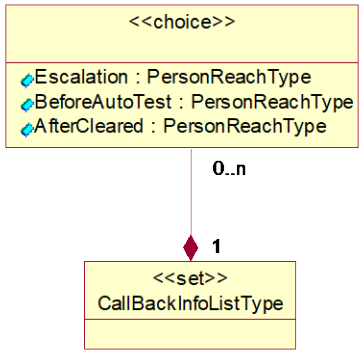
<table>
<tr><th>ASN.1 Definition</th><th>UML Model</th><th>Schema</th></tr>
<tr>
<td>

CallBackInfoList::=SET OF CHOICE{
  escalation
[0]PersonReach,
  beforeAutoTest [1]PersonReach,
  afterCleared [2]PersonReach,
  ...
}

Excerpt from X.680

3.8.59 set-of types: Types defined by referencing a single component type; each value in the set-of type is an unordered list of zero, one, or more values of the component type.

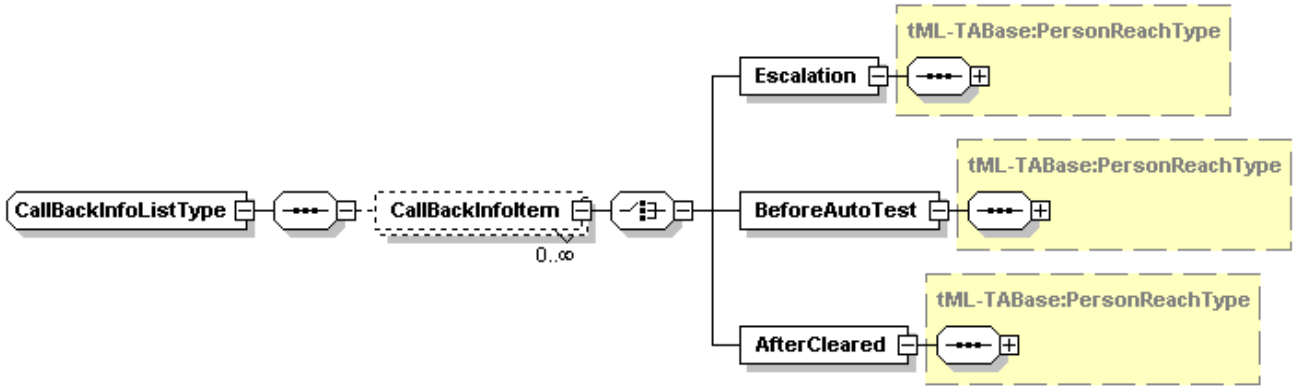
</td>
<td>



</td>
<td>

```
<complexType name="CallBackInfoListType">
    <sequence>
        <element name="CallBackInfoItem"
          minOccurs="0" maxOccurs="unbounded">
            <complexType>
                <choice>
                    <element name="Escalation"
                type="tML-TABase:PersonReachType"/>
                    <element name="BeforeAutoTest"
                type="tML-TABase:PersonReachType"/>
                    <element name="AfterCleared"
                type="tML-TABase:PersonReachType"/>
                </choice>
            </complexType>
        </element>
    </sequence>
</complexType>
```

</td>
</tr>
<tr><th colspan="3">Schema Data Structure Diagram</th></tr>
<tr>
<td colspan="3">



</td>
</tr>
</table>

Table 3.14 – SET OF with SIZE

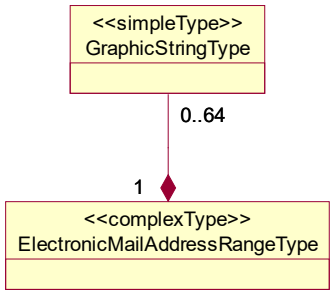
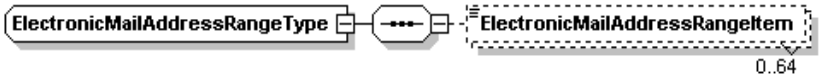
ASN.1 Definition	UML Model	Schema
ElectronicMailAddressRange::= SET SIZE (0..64) OF GraphicString(SIZE(0..64))		<complexType name="ElectronicMailAddressRangeType"> <sequence> <element name="ElectronicMailAddressRangeItem" minOccurs="0" maxOccurs="64"> <simpleType> <restriction base="tML-TABase:GraphicStringType"> <minLength value="0"/> <maxLength value="64"/> </restriction> </simpleType> </element> </sequence> </complexType>
Schema Data Structure Diagram		
		

Table 3.15 – OCTET STRING

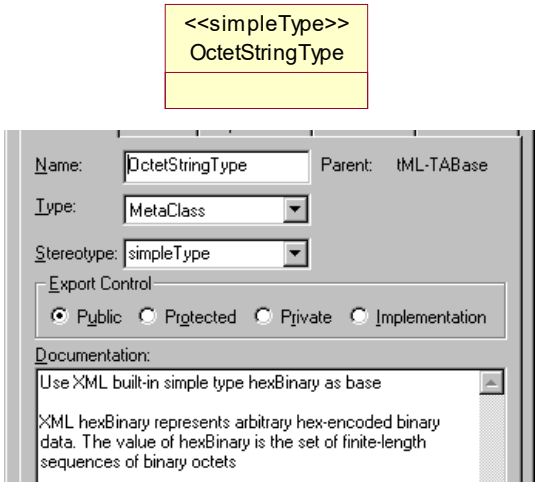
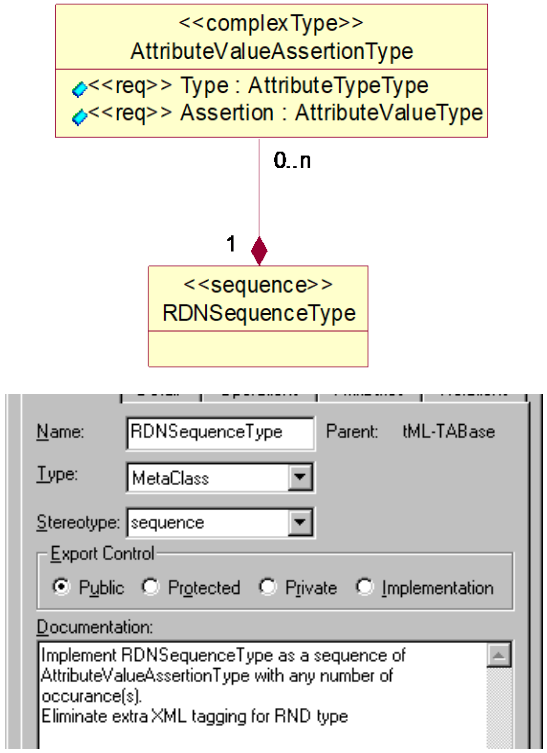
Excerpt from X.680	UML Model	Schema
3.8.48 octetstring type: A simple type whose distinguished values are an ordered sequence of zero, one, or more octets, each octet being an ordered sequence of eight bits. **C.2.6 Octet string** **C.2.6.1** Use an octet string type to model binary data whose format and length are unspecified, or specified elsewhere, and whose length in bits is a multiple of eight. EXAMPLE **G4FacsimileImage ::= OCTET STRING** -- a sequence of octets conforming to -- Recommendations T.5 and T.6. **image G4FacsimilePage ::= '3FE2EBAD471005'H**		<simpleType name="OctetStringType"> <restriction base="hexBinary"/> </simpleType>

Table 3.16 – SEQUENCE OF

<table>
<tr><th>ASN.1 Definition (X.711)</th><th>UML Model</th><th>Schema</th></tr>
<tr>
<td>AttributeValueAssertion ::=
SEQUENCE {
type AttributeType,
assertion AttributeValue
}

RelativeDistinguishedName ::=
SET OF AttributeValueAssertion
RDNSequence ::= SEQUENCE OF
RelativeDistinguishedName

DistinguishedName::= RDNSequence

Excerpt from X.680
3.8.57 sequence-of types: Types defined by referencing a single component type; each value in the sequence-of type is an ordered list of zero, one, or more values of the component type.</td>
<td> 
</td>
<td><pre>
<complexType name="AttributeValueAssertionType">
  <sequence>
    <element name="Type"
      type="tML-TABase:AttributeTypeType"/>
    <element name="Assertion"
      type="tML-TABase:AttributeValueType"/>
  </sequence>
</complexType>

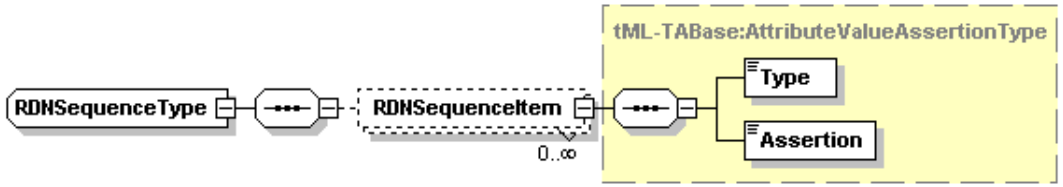
<complexType name="RDNSequenceType">
  <sequence>
    <element name="RDNSequenceItem"
      type="tML-TABase:AttributeValueAssertionType"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
</pre></td>
</tr>
<tr><th colspan="3">Schema Data Structure Diagram</th></tr>
<tr><td colspan="3"> 
</td></tr>
</table>

Table 3.17 – Restricted Inheritance Based on CMIP Specification (see Annexes A through H)

ASN.1 Definition (X.711)

```
ObjectInstance::= CHOICE {
   distinguishedName
      [2] IMPLICIT DistinguishedName,
   nonSpecificForm
      [3] IMPLICIT OCTET STRING,
   localDistinguishedName
      [4] IMPLICIT RDNSequence
}
```

ASN.1 Definition (Annexes A through H)

ManagedObjectInstance::=
ObjectInstance

Excerpt from Annexes A through H

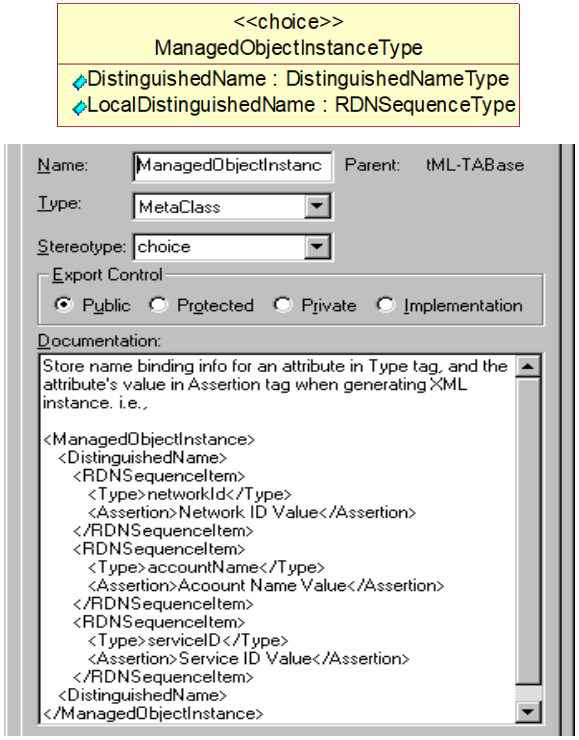
7.5 Supporting Productions

….

AttributeId, ObjectClass, ObjectInstance FROM CMIP-1 {joint-iso-ccitt(2) ms(9) cmip(1) modules(0) protocol(3)};

– Only the Distinguished Name and
– Local Distinguished Name forms
– are supported.

UML Model



Schema

```
<complexType name="ObjectInstanceType">
    <choice>
        <element name="DistinguishedName"
          type="tML-TABase:DistinguishedNameType"/>
        <element name="NonSpecificForm"
          type="tML-TABase:OctetStringType"/>
        <element name="LocalDistinguishedName"
          type="tML-TABase:RDNSequenceType"/>
    </choice>
</complexType>


<complexType name="ManagedObjectInstanceType">
    <complexContent>
        <restriction base="tML-TABase:ObjectInstanceType">
            <choice>
                <element name="DistinguishedName"
              type="tML-TABase:DistinguishedNameType"/>
                <element name="LocalDistinguishedName"
              type="tML-TABase:RDNSequenceType"/>
            </choice>
        </restriction>
    </complexContent>
</complexType>
```

Schema Data Structure Diagram

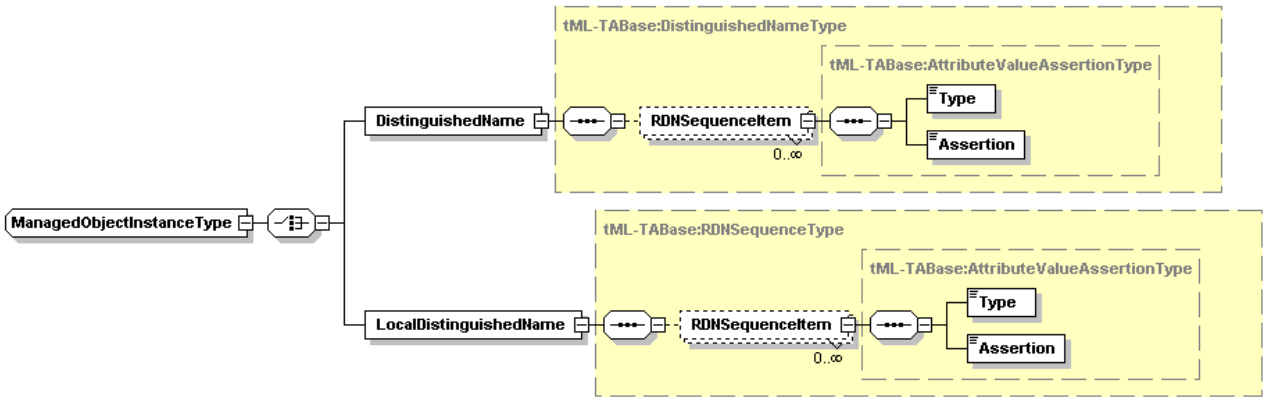


Table 3.18 – Simple "typedef"

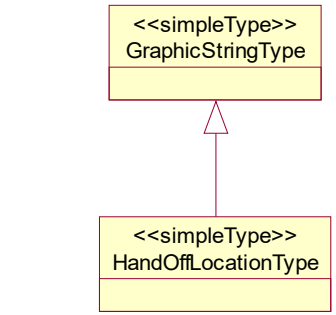
<table>
<tr><th>ASN.1 Definition</th><th>UML Model</th><th>Schema</th></tr>
<tr>
<td>
HandOffLocation::=
GraphicString(SIZE(0..64))

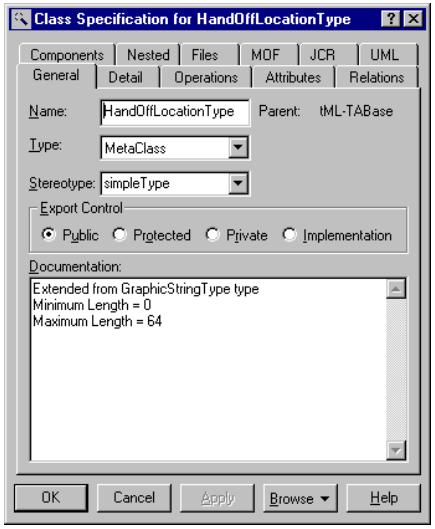
Excerpt from Annexes A through H

5.1 Use of GraphicString syntax
Within North America, the character set for GraphicString should be the same as the set of characters defined for ASCII printableString.

Excerpt from X.680

Table 5 – PrintableString

<table>
<tr><th>Name</th><th>Graphic</th></tr>
<tr><td>Capital letters</td><td>A, B, ... Z</td></tr>
<tr><td>Small letters</td><td>a, b, ... z</td></tr>
<tr><td>Digits</td><td>0, 1, ... 9</td></tr>
<tr><td>Space</td><td>(space)</td></tr>
<tr><td>Apostrophe</td><td>'</td></tr>
<tr><td>Left Parenthesis</td><td>(</td></tr>
<tr><td>Right Parenthesis</td><td>)</td></tr>
<tr><td>Plus sign</td><td>+</td></tr>
<tr><td>Comma</td><td>,</td></tr>
<tr><td>Hyphen</td><td>-</td></tr>
<tr><td>Full stop</td><td>.</td></tr>
<tr><td>Solidus</td><td>/</td></tr>
<tr><td>Colon</td><td>:</td></tr>
<tr><td>Equal sign</td><td>=</td></tr>
<tr><td>Question mark</td><td>?</td></tr>
</table>
</td>
<td>




</td>
<td>
<pre>
<simpleType name="HandOffLocationType">
    <restriction
       base="tML-TABase:GraphicStringType">
        <minLength value="0"/>
        <maxLength value="64"/>
    </restriction>
</simpleType>
</pre>

<pre>
<simpleType name="GraphicStringType">
    <restriction base="string">
        <pattern
      value="[A-Z|a-z|0-9|\s|'|\(|\)|\+|,|\-
|\.|/|:|=|\?|@|;|!|”|#|$|%|*|`|\[|\]|\\|^|_|{|\|}|~|&|<|>|"]*"/>
    </restriction>
</simpleType>
</pre>
</td>
</tr>
</table>

Table 3.19 – Complex "typedef": Case 1

ASN.1 Definition

AdditionalTroubleInfoList::=
SET OF GraphicString

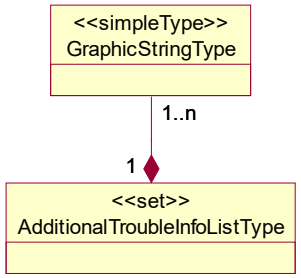
GDMO Definition

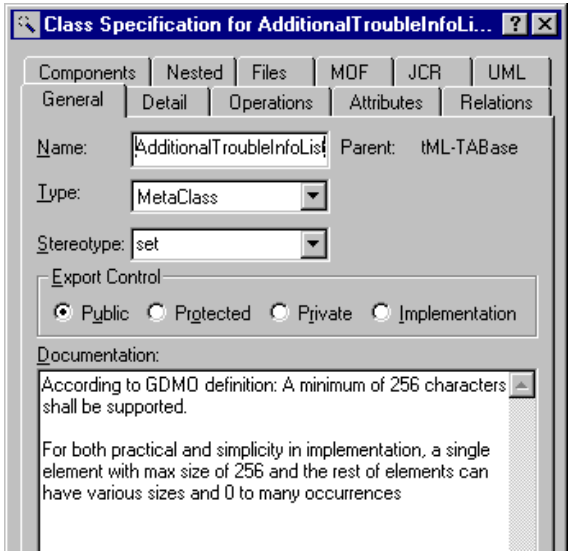
7.2.8 Additional Trouble Information List

The Additional Trouble Information List attribute further describes the selected Trouble Type. A minimum of 256 octets shall be supported, regardless of the number of values in the set. The manager can only add information, but not remove it. It is possible that the oldest information may be lost if an implementation has restrictions on the maximum size.

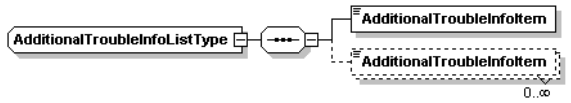
additionalTroubleInfoList
ATTRIBUTE
WITH ATTRIBUTE SYNTAX GNMTA.AdditionalTroubleInfoList;
MATCHES FOR
EQUALITY,
SET-INTERSECTION,
SET-COMPARISON;
REGISTERED AS {trAttribute 8};

UML Model





Schema Data Structure Diagram



Schema

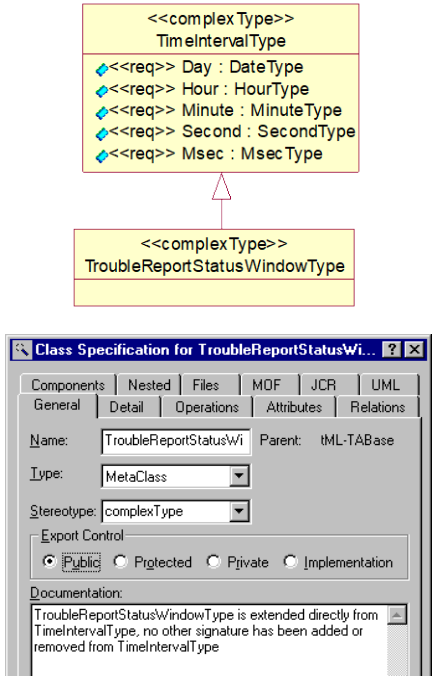
```
<complexType name="AdditionalTroubleInfoListType">
    <sequence>
        <element name="AdditionalTroubleInfoItem" minOccurs="1"
maxOccurs="unbounded">
            <simpleType>
                <restriction base="tML-TABase:GraphicStringType">
                    <minLength value="0"/>
                    <maxLength value="400"/>
                </restriction>
            </simpleType>
        </element>

    </sequence>
</complexType>
```

Table 3.20 – Complex "typedef": Case 2

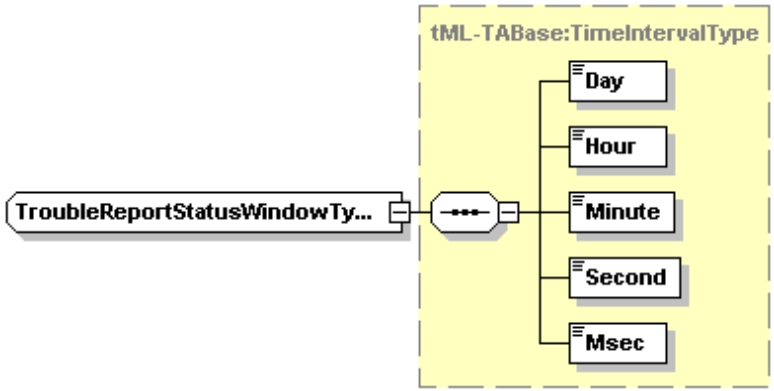
ASN.1 Definition

```
TimeInterval ::= SEQUENCE {
      day         [0] INTEGER (0..31) DEFAULT 0,
       hour       [1] INTEGER (0..23) DEFAULT 0,
       minute     [2] INTEGER (0..59) DEFAULT 0,
       second     [3] INTEGER (0..59) DEFAULT 0,
      msec        [4] INTEGER (0..999) DEFAULT 0
      }

TroubleReportStatusWindow::=TimeInterval
```

UML Model



Schema Data Structure Diagram



Schema

```
<simpleType name="DayType">
    <restriction base="integer">
        <minInclusive value="0"/><maxInclusive value="31"/>
    </restriction>
</simpleType>

<simpleType name="HourType">
    <restriction base="integer">
        <minInclusive value="0"/><maxInclusive value="23"/>
    </restriction>
</simpleType>

<simpleType name="MinuteType">
    <restriction base="integer">
        <minInclusive value="0"/><maxInclusive value="59"/>
    </restriction>
</simpleType>

…

<complexType name="TimeIntervalType">
    <sequence>
        <element name="Day" type="tML-TABase:DayType"
          default="0"/>
        <element name="Hour" type="tML-TABase:HourType"
          default="0"/>
        <element name="Minute" type="tML-TABase:MinuteType"
          default="0"/>
        <element name="Second" type="tML-TABase:SecondType"
          default="0"/>
        <element name="Msec" type="tML-TABase:MsecType"
          default="0"/>
    </sequence>
</complexType>


<complexType name="TroubleReportStatusWindowType">
    <complexContent>
        <extension base="tML-TABase:TimeIntervalType"/>
    </complexContent>
</complexType>
```

3.2 Trouble Administration Services Mapping Rules

The tML TA interface is developed based on the analysis on trouble administration functions specified in clause I.1.2 of Annex I, and trouble administration service definitions in clause 1.4 of the same Annex. The following table lists comparison between the tML TA interface and trouble administration functions and services specified in Tables 1 and 4 of Annex I.

Table 3.21 – Comparison between Trouble Administration Functions/Services and the tML TA Interface

Trouble Administration Functional Unit	CMISE Service(s)	XML Interface(s)
Kernel	• PT-CREATE on Telecommunications Trouble Report object • PT-GET on Telecommunications Trouble Report object	• RequestTroubleReportCreationRequest/Response • RetrieveTroubleReportStatusRequest/Response
Request Trouble Report Format	• PT-GET on Trouble Report Format Definition object	• RetrieveTroubleReportFormatRequest/Response • RetrieveTroubleReportFormatByServiceIdRequest/Response
Trouble History Event Notification	• Trouble History Event Notification	• TroubleHistoryEventNotification
Review Trouble History Record	• PT-GET on Trouble History Record object	• RetrieveTroubleHistoryByServiceIdRequest/Response
Add Trouble Information	• PT-SET on Additional Trouble Information List attribute in Telecommunications Trouble Report object	• AddTroubleInfoRequest/Response
Trouble Report Status/Commitment Time Update Notification	• Trouble Report Status/Commitment Time Update Notification	• TroubleReportStatusOrCommitmentTimeUpdateNotification
Verify Trouble Repair Completion	• PT-SET on Close Out Verification attribute in Telecommunications Trouble Report object	• VerifyRepairCompletionRequest/Response
Modify Trouble Administration Information	• PT-SET on Telecommunications Trouble Report object	• ModifyAttributesRequest/Response
Trouble Administration Configuration Event Notification	• Attribute Value Change Notification • Object Creation/Deletion Notification	• AttributeValueChangeNotification • AVCTroubleReportFormatDefinitionNotification • TroubleReportCreatedNotification • EnrolTroubleReportFormatDefinitionNotification • DeleteTroubleReportNotification • DeenrolTroubleReportFormatDefinitionNotification
Trouble Report Progress Notification	• Trouble Report Progress Notification	• TroubleReportProgressNotification
Cancel Trouble Report	• PT-SET on Cancel Requested by Manager attribute in Telecommunications Trouble Report object	• CancelTroubleReportRequest/Response
--	--	• RetrieveTroubleReportIdByServiceIdRequest/Response[10]
--	--	• RetrieveAttributesByTroubleReportIdRequest/Response[11]
--	--	• ConfirmNotificationReceiptRequest[12]

[10] The service shown to have no equivalence in CMIP. However, a generic get service along with the filter field satisfies this service.

[11] The service shown to have no equivalence in CMIP. However, a generic get service along with the filter field satisfies this service.

[12] ConfirmNotificationReceiptRequest is the service customer's acknowledgement to a service provider's Notification. Therefore, it has no corresponding Response.

The remainder of this clause will provide an example (TR Creation) that explains how to generate a specific XML interface (a specific Request/Response/Notification for TR creation) for services provided by trouble administration functions/services.

Table 3.22 – Telecommunications Trouble Report Creation

<table>
<tr><th>Excerpt from Annexes A through H</th><th>UML Model (Request)</th><th>Schema (Request)</th></tr>
<tr>
<td>

7.1.5 Telecommunications Trouble Report

… As part of create, the manager is required to supply the following attributes:

- Managed Object Instance;
- Trouble Type;
- Additional Trouble Information List;

plus any manager-supplied attributes in conditional packages identified as “must be present” attributes.

…

-- The following attributes can only supplied by the manager: managedObjectInstance

…

Excerpt from Annex I

9. Trouble administration service definitions

…

9.1 Kernel functional unit

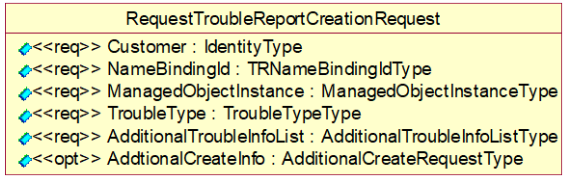
…

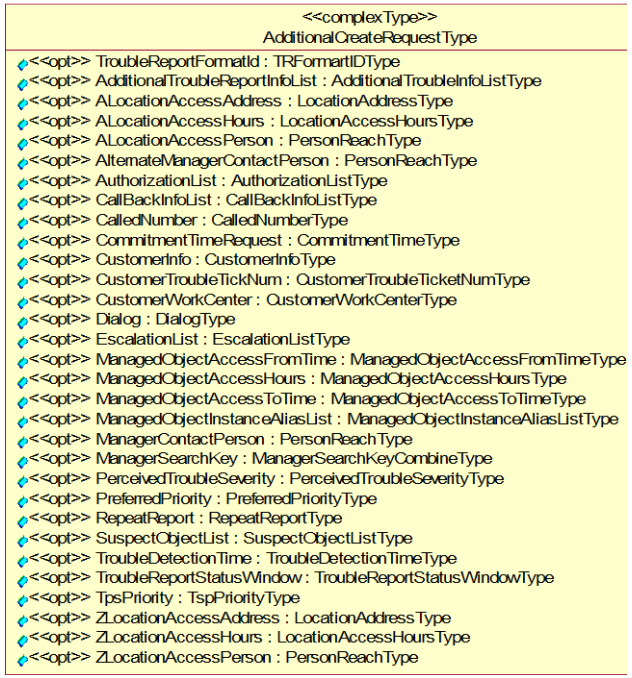
9.1.1 Enter Trouble Report

The PT-CREATE service, as described in ISO/IEC 10164-1, is used to allow a manager to request that a trouble report be created by the agent with the appropriate information…If the manager chooses not to use the reference object option, the manager shall supply the following attributes as part of the create operation:

- Managed Object Instance;
- …

</td>
<td>





</td>
<td>

```
<element name="RequestTroubleReportCreationRequest"
      type="tML-TA:RequestTroubleReportCreationRequestType"/>

<complexType name="RequestTroubleReportCreationRequestType">
    <sequence>
        <element name="Customer" type="tML-TABase:IdentityType"/>
        <element name="NameBindingId"
         type="tML-TABase:TRNameBindingIdType"/>
        <element name="ManagedObjectInstance"
         type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="TroubleType" type="tML-TABase:TroubleTypeType"/>
        <element name="AdditionalTroubleInfoList"
         type="tML-TABase:AdditionalTroubleInfoListType"/>
        <element name="AdditionalCreateInfo"
         type="tML-TABase:AdditionalCreateRequestType"
         minOccurs="0"/>
    </sequence>
</complexType>

<simpleType name=”NameBindingIdtype”>
  <restriction base=”string”>
    <enumeration value=”1.2.840.10015.0.6.24”/>
    <enumeration value=”1.2.840.10015.0.6.25”/>
    <enumeration value=”1.2.840.10015.0.6.26”/>
    <enumeration value=”1.2.840.10015.0.6.27”/>
  </restriction>
</simpleType>


<complexType name="TRNameBindingIdType">
    <sequence>
        <element name="NameBindingId" type="tML-TABase:NameBindingIdType"/>

    </sequence>
</complexType>
```

</td>
</tr>
</table>

Table 3.22 (con't)

Excerpt from Annexes A through H

…
As part of the instantiation of a Telecommunications Trouble Report, the agent is also required to supply values for the remaining "CHARACTERIZED BY" attributes of the Telecommunications Trouble Report object class plus any agent-supplied attributes in conditional packages identified as "must be present"…

– The following attributes can only be supplied by the agent:
  initiatingMode
  receivedTime
  troubleReportID
– The following attributes can only be supplied by the agent and updated by the agent:
  activityDuration *
  additionalTroubleStatusInfo
  agentContactPerson
  …
  commitmentTime
  …

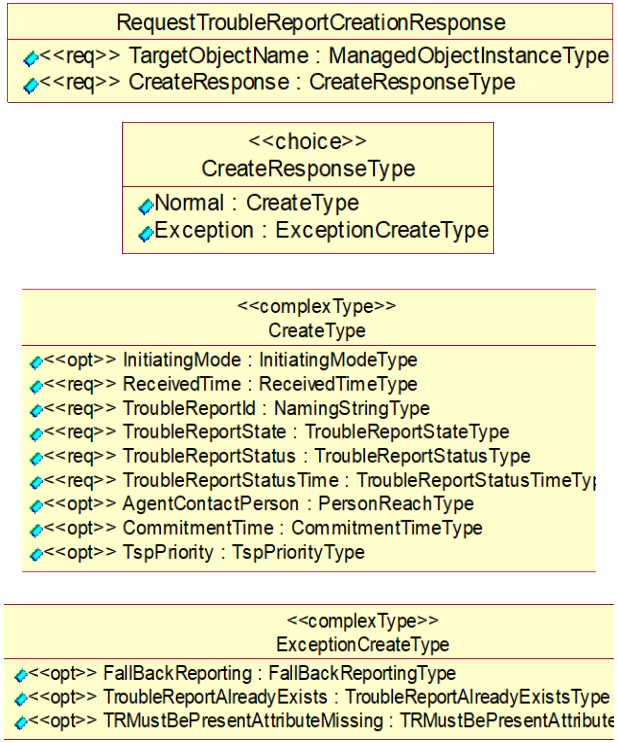
Excerpt from Annex I

…
If the input information is correct, the agent will respond with the name (Trouble Report ID)…

The Enter Trouble Report service supports the following "processing failure" specific errors.

a) *Trouble report already exists for this CNM Service instance*:…
b) *Fallback trouble reporting used*:…

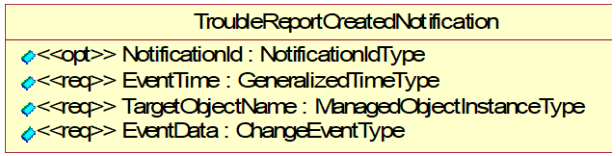
UML Model (Response)



Schema (Response)

```
<element name="RequestTroubleReportCreationResponse"
      type="tML-TA:RequestTroubleReportCreationResponseType"/>

<complexType name="RequestTroubleReportCreationResponseType">
    <sequence>
        <element name="TargetObjectName"
        type="tML-TABase:ManagedObjectInstanceType"/>
   <element name="CreateResponse" type="tML-TABase:CreateResponseType"/>
  </sequence>
</complexType>

<complexType name="CreateResponseType">
    <choice>
        <element name="Normal" type="tML-TABase:CreateType"/>
        <element name="Exception" type="tML-TABase:ExceptionCreateType"/>
    </choice>
</complexType>

<complexType name="CreateType">
    <sequence>
        <element name="InitiatingMode" type="tML-TABase:InitiatingModeType"
        minOccurs="0"/>
        <element name="ReceivedTime" type="tML-TABase:ReceivedTimeType"/>
        <element name="TroubleReportId" type="tML-TABase:NamingStringType"/>
        <element name="TroubleReportState"
        type="tML-TABase:TroubleReportStateType"/>
        <element name="TroubleReportStatus"
        type="tML-TABase:TroubleReportStatusType"/>
        <element name="TroubleReportStatusTime"
        type="tML-TABase:TroubleReportStatusTimeType"/>
        <element name="AgentContactPerson" type="tML-TABase:PersonReachType"
        minOccurs="0"/>
        <element name="CommitmentTime" type="tML-TABase:CommitmentTimeType"
        minOccurs="0"/>
        <element name="TspPriority" type="tML-TABase:TspPriorityType"
        minOccurs="0"/>
    </sequence>
     </complexType>

<complexType name="ExceptionCreateType">
    <complexContent>
        <extension base="tML-TABase:ExceptionGenericType">
            <sequence>
                <element name="FallBackReporting"
            type="tML-TABase:FallBackReportingType" minOccurs="0"/>
                <element name="TroubleReportAlreadyExists"
            type="tML-TABase:TroubleReportAlreadyExistsType"
            minOccurs="0"/>
                <element name="TRMustBePresentAttributeMissing"
            type="tML-TABase:TRMustBePresentAttributeMissingType"
            minOccurs="0"/>
              </sequence>
        </extension>
    </complexContent>
</complexType>
```

Table 3.22 (con't)

Excerpt from Annexes A through H	UML Model (Notification)	Schema (Notification)
troubleReport MANAGED OBJECT CLASS DERIVED FROM "Rec. X.721|ISO/IEC 10165-2 : 1992":top; … trObjectCreationDeletionPkg PACKAGE NOTIFICATIONS "Rec.X.721|ISO/IEC 10165-2:1992": objectCreation, "Rec.X.721|ISO/IEC 10165-2: 1992": objectDeletion; REGISTERED AS {trPackage 84}; PRESENT IF "an instance supports it.", … REGISTERED AS {trMObjectClass 7}; **Excerpt from Annex I** **9.9 Trouble Administration Configuration Event Notification functional unit** … **9.9.2 Object CreationNotification** The Object Creation Notification service is defined in ISO/IEC 10164-1. In this FU, the Object Creation Notification service allows the agent to notify the manager when a Telecommunications Trouble Report or a Trouble Report Format Definition object is created through local administrative procedures.		<element name="TroubleReportCreatedNotification" type="tML-TA:TroubleReportCreatedNotificationType"/> <complexType name="TroubleReportCreatedNotificationType"> <sequence> <element name="NotificationId" type="tML-TABase:NotificationIdType" minOccurs="0"/> <element name="EventTime" type="tML-TABase:GeneralizedTimeType"/> <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/> <element name="EventData" type="tML-TABase:ChangeEventType"/> </sequence> </complexType>

3.3 UML Diagrams for tML TA

3.3.1 Request

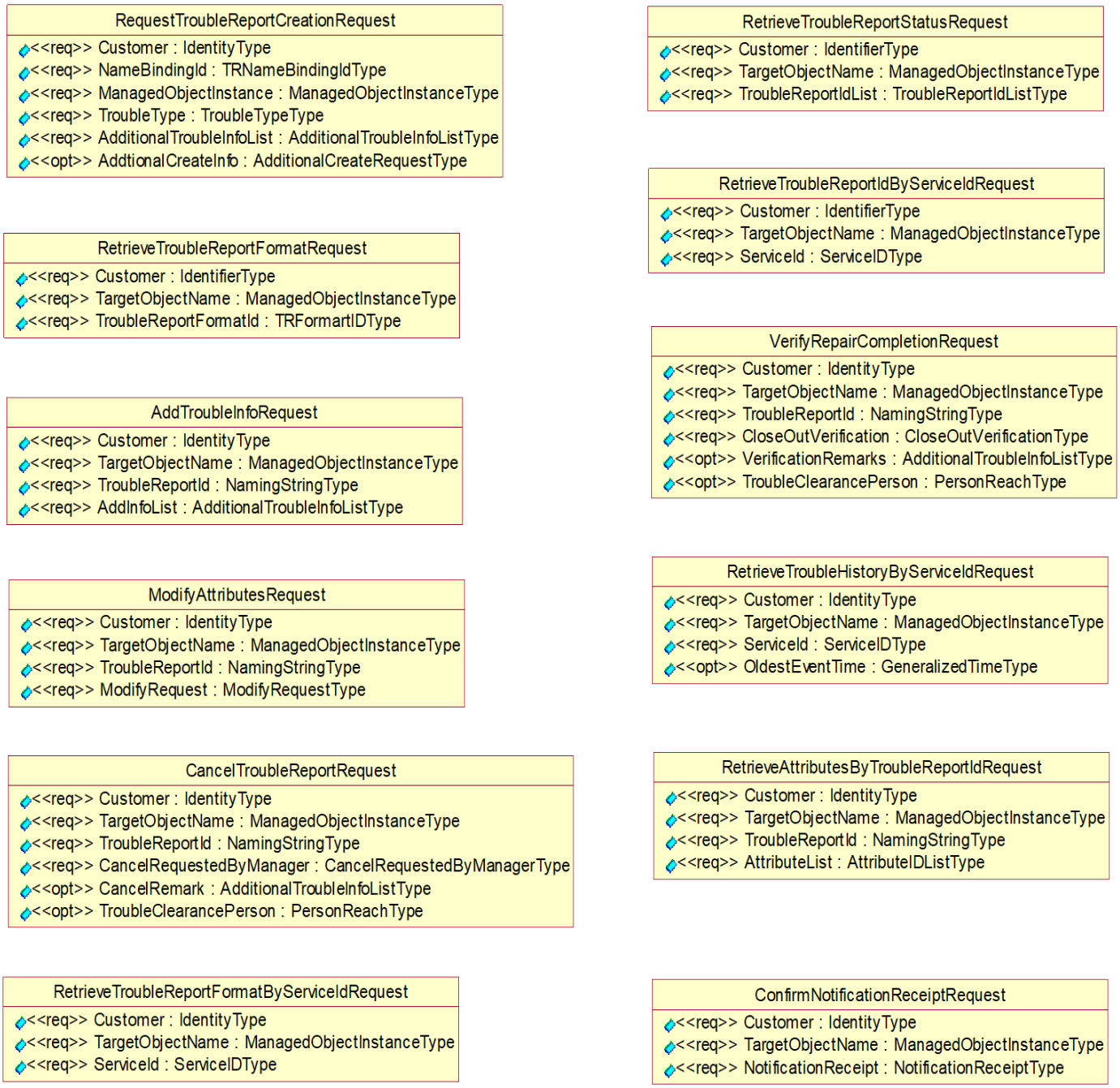


Figure 3.1 – Telecommunications Trouble Report Request PDUs

3.3.2 Response

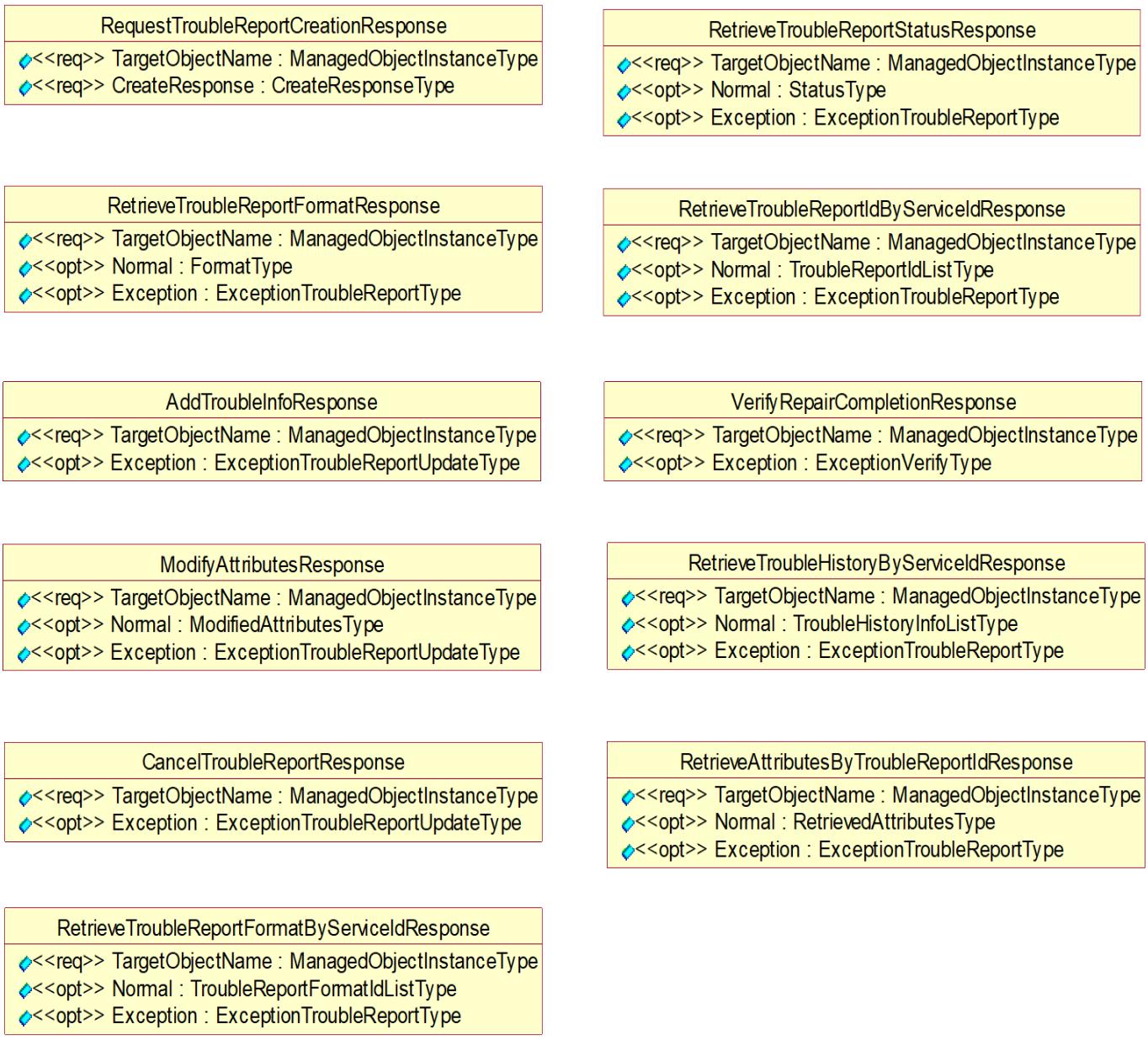


Figure 3.2 – Telecommunications Trouble Report Response PDUs

3.3.3 Notification

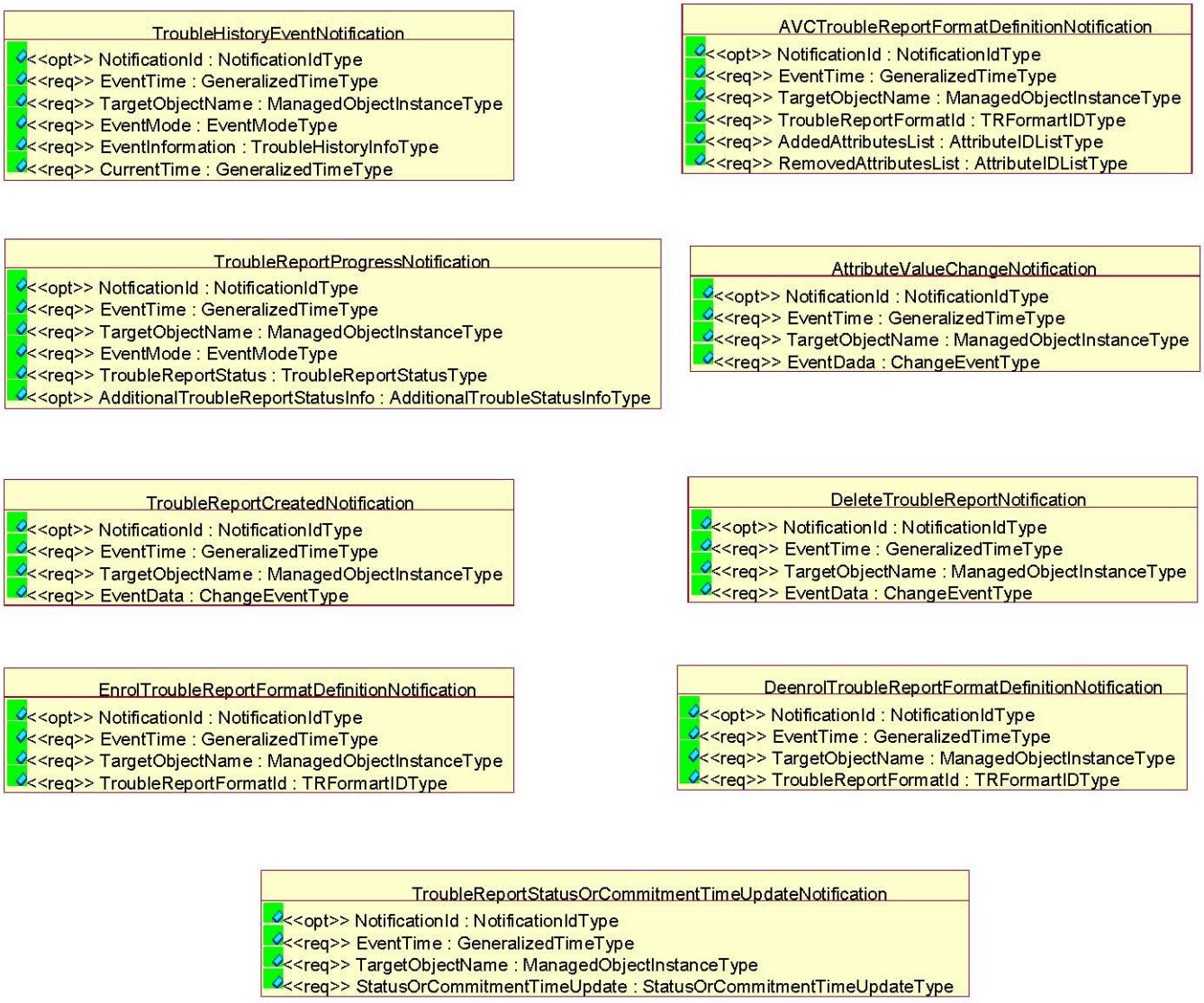


Figure 3.3 – Telecommunications Trouble Report Notification PDUs

3.4 UML Diagrams for tML TABase

3.4.1 Data Types Used in Telecommunications Trouble Report Request

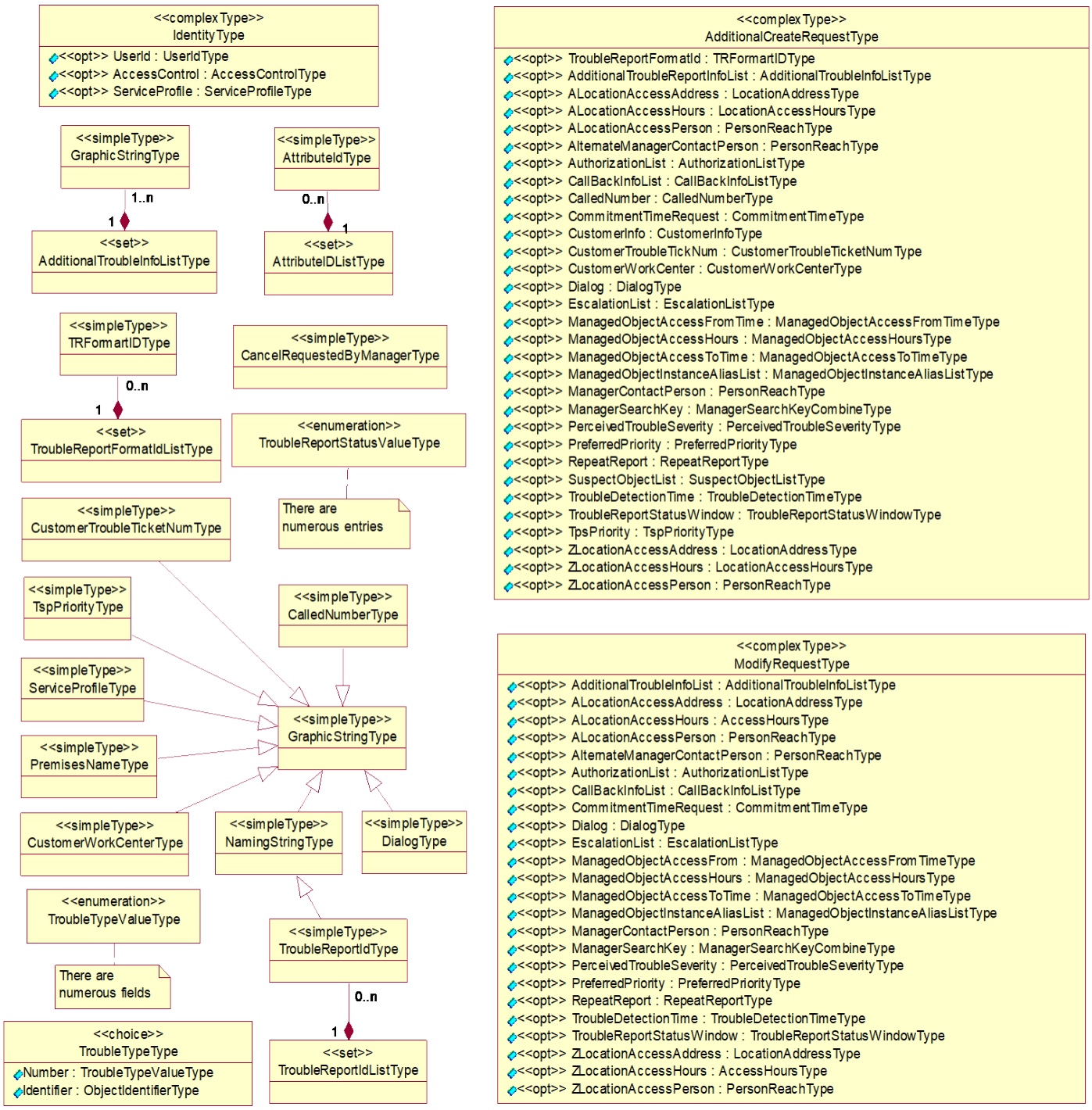


Figure 3.4 – Data Types Used in Telecommunications Trouble Report Request: Part 1

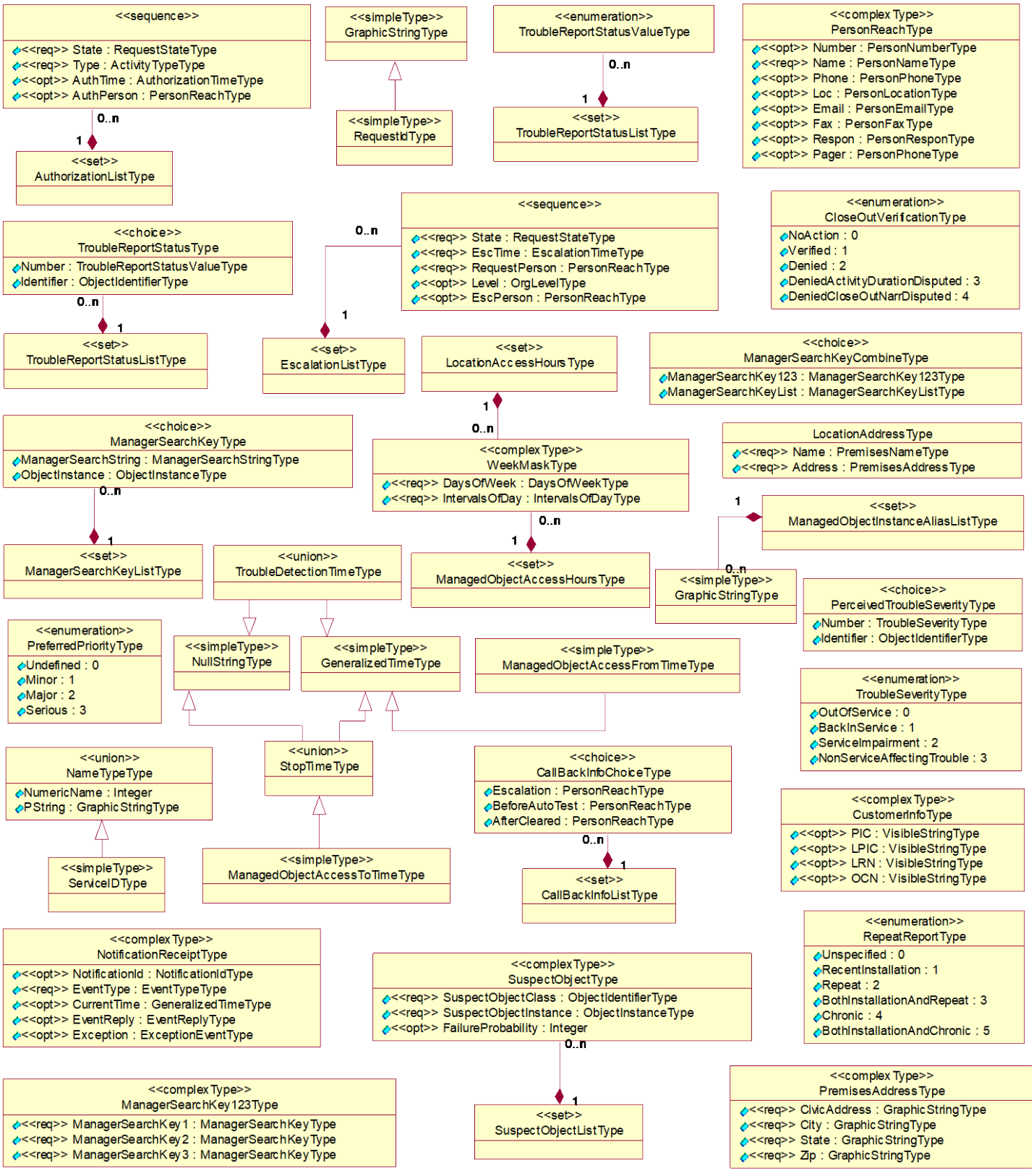


Figure 3.5 – Data Types Used in Telecommunications Trouble Report Request: Part 2

3.4.2 Data Types Used in Telecommunications Trouble Report Response

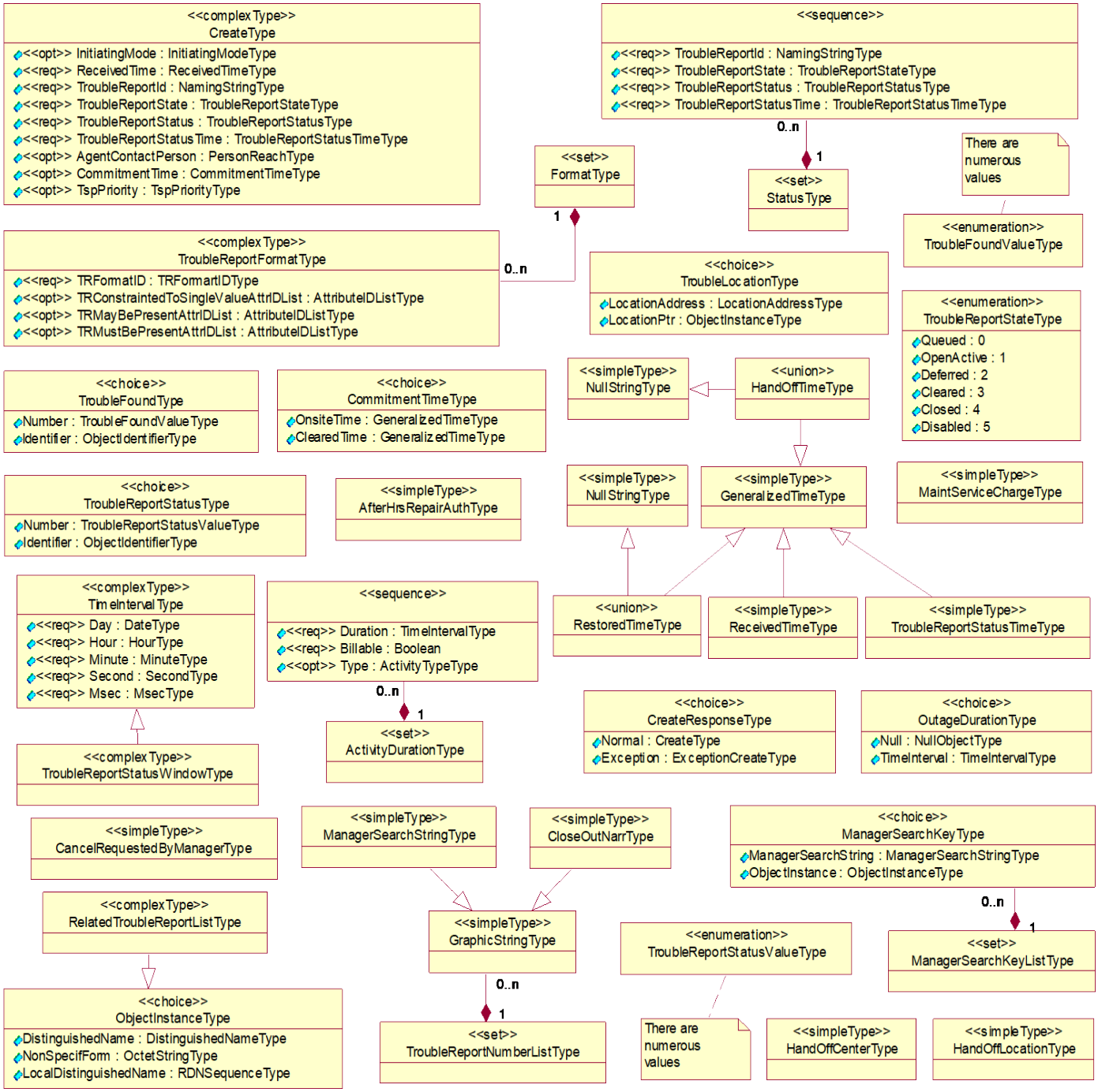


Figure 3.6 – Data Types Used in Telecommunications Trouble Report Response: Part 1

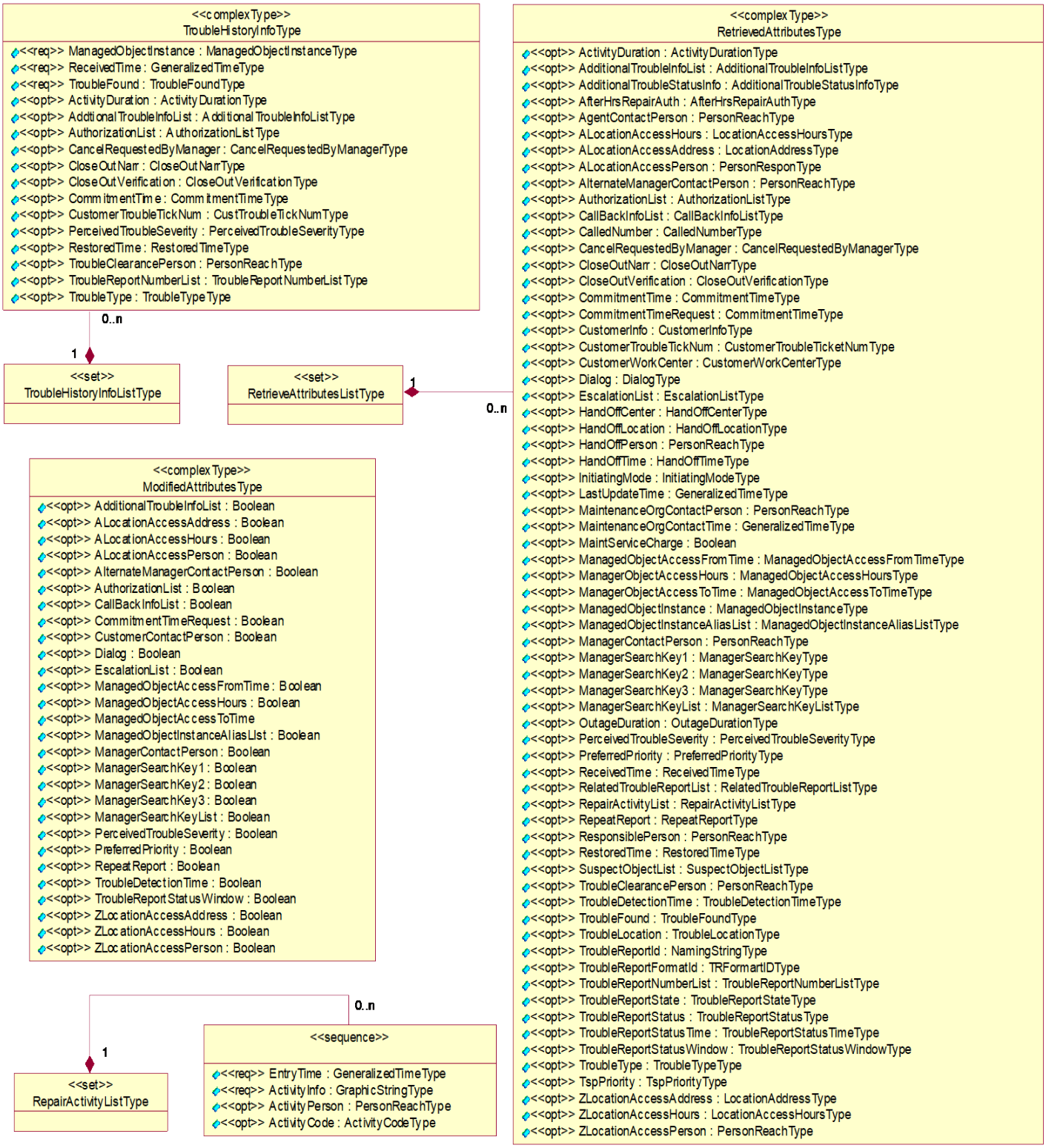


Figure 3.7 – Data Types Used in Telecommunications Trouble Report Response: Part 2

3.4.3 Data Types Used in Telecommunications Trouble Report Notification

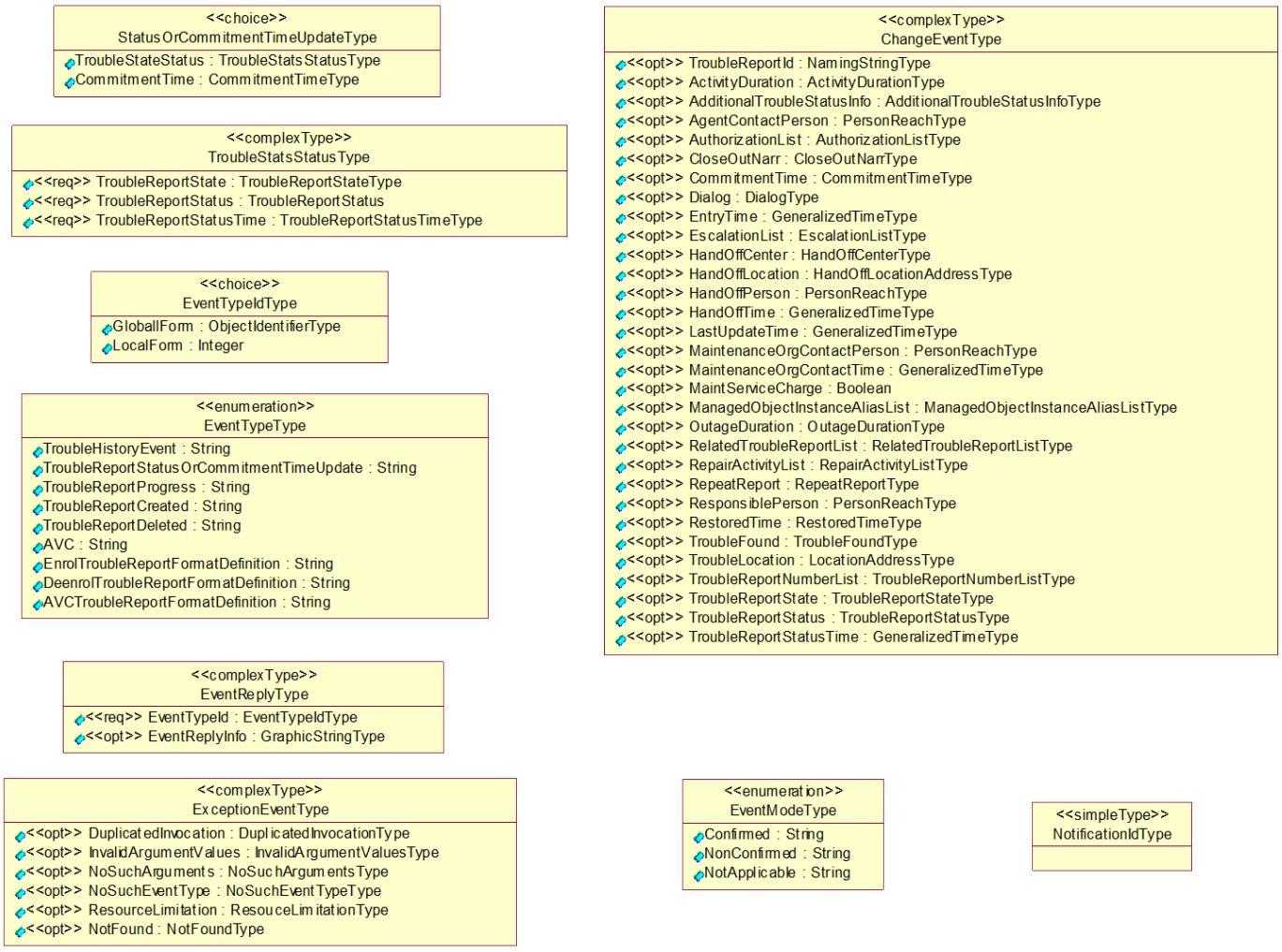


Figure 3.8 – Data Types Used in Telecommunications Trouble Report Notification

3.4.4 Exception Types

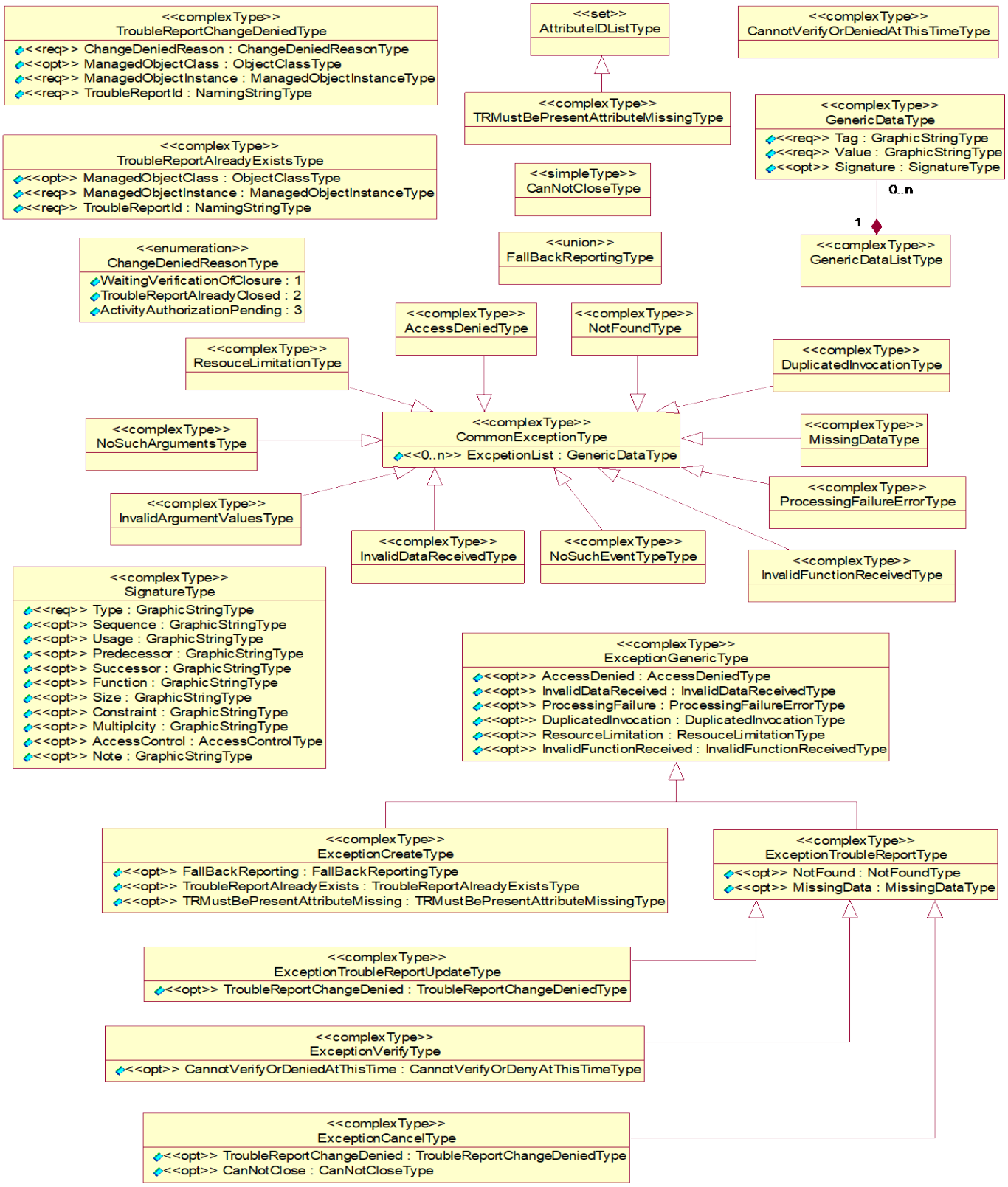


Figure 3.9 – Exception Classes

4 Trouble Administration XML Schemas

4.1 Overview of tML TA Schema Packages

4.1.1 tML TA Schema Packges

The suggested tML TA schema packages dependency is the following:

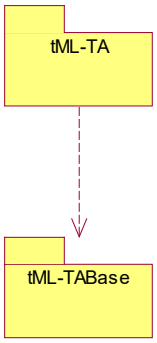


Figure 4.1 – tML TA Schema Package Dependency Diagram

The impact of the dependencies in tML TA Schemas is shown in following clauses.

4.1.1.1 tML-TA Package

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="http://www.ansi.org/tML/TA/tML-TA"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:tML="urn:itu.int/tML/tMLSchemaMetadata"
    xmlns:tML-TABase="http://www.ansi.org/tML/TA/tML-TABase"
    xmlns:tML-TA="http://www.ansi.org/tML/TA/tML-TA"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
    version="1.2 2012/03/20 12:23:09"
    id="tML-TA.xsd"
    xml:lang="en"
>
    <annotation>
        <documentation>
            <tML:SchemaMetadata>
                <tML:OriginalAuthor>Wu Liu</tML:OriginalAuthor>
                <tML:CreationDate>07-05-2002</tML:CreationDate>
                <tML:Description>This schema contains Trouble Administration services
                                        provided by Telecommunications Trouble Report functions
                                        This version (1.2) contains a new ActivityType of AuthorizeToWork
                </tML:Description>
                <tML:Source>ANSI T1.227/T1.228</tML:Source>
                <tML:SchemaHistory/>
            </tML:SchemaMetadata>
        </documentation>
    </annotation>

    <import namespace="http://www.ansi.org/tML/TA/tML-TABase"
              schemaLocation="tML-TABase.xsd"/>

        <!— detailed in clause 4.2 -->
</schema>
```

4.1.1.2 tML-TABase Package

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="http://www.ansi.org/tML/TA/tML-TABase" xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:tML="urn:itu.int/tML/tMLSchemaMetadata"
    xmlns:tML-TABase="http://www.ansi.org/tML/TA/tML-TABase"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
    version="1.1 2004/04/15 14:19:30"
    id="tML-TABase.xsd"
    xml:lang="en"
>

    <annotation>
        <documentation>
            <tML:SchemaMetadata>
                <tML:OriginalAuthor>Wu Liu</tML:OriginalAuthor>
                <tML:CreationDate>07-05-2002</tML:CreationDate>
                <tML:Description>This schema contains Attribute Types defined in T1.227/T1.228
                                 and related documents. It also provides data types used in
                                 Trouble Administration services (tML-TA schema).
                </tML:Description>
                <tML:Source>ANSI T1.227/T1.228</tML:Source>
                <tML:SchemaHistory/>
            </tML:SchemaMetadata>
        </documentation>
    </annotation>

    <!— detailed in clause 4.3 -->
</schema>
```

4.2 Schema for tML-TA

4.2.1 Request Portion[13]

```
    <element name="RequestTroubleReportCreationRequest"
             type="tML-TA:RequestTroubleReportCreationRequestType"/>
    <element name="RetrieveTroubleReportStatusRequest"
             type="tML-TA:RetrieveTroubleReportStatusRequestType"/>
    <element name="RetrieveTroubleReportFormatRequest"
             type="tML-TA:RetrieveTroubleReportFormatRequestType"/>
    <element name="RetrieveTroubleReportIdByServiceIdRequest"
             type="tML-TA:RetrieveTroubleReportIdByServiceIdRequestType"/>
    <element name="AddTroubleInfoRequest" type="tML-TA:AddTroubleInfoRequestType"/>
    <element name="VerifyRepairCompletionRequest"
             type="tML-TA:VerifyRepairCompletionRequestType"/>
    <element name="ModifyAttributesRequest" type="tML-TA:ModifyAttributesRequestType"/>
    <element name="CancelTroubleReportRequest" type="tML-TA:CancelTroubleReportRequestType"/>
```

[13] Message Identification (commonly known as Request Id) is used to correlate a request and a corresponding response for a specific tML message exchanged between a Service Provider and a Service Customer. When this standard was first published as a Trial-Use standard, transporting tML message under SOAP over HTTP(S) by certain Web Services imposed limitations on how the tML Header content could be constructed. Those limitations have led some Service Provider(s) and Service Customer(s) to move the Message Identification field from the tML Header into the tML message itself (as in a tML TA request/response/notification PDU where the identification is the first field in the message). Even with the Web Services software vendors progressing their products so that supporting a full tML Header in a SOAP message has become possible, those Service Provider(s) or Service Customer(s) who have already implemented the tML TA interfaces using the Message Identification (Request Id) as the first field in the tML TA request/response/notification PDUs may continue do so, and this practice would not be considered a violation of this standard.

```
<element name="RetrieveTroubleHistoryByServiceIdRequest"
            type="tML-TA:RetrieveTroubleHistoryByServiceIdRequestType"/>
<element name="RetrieveAttributesByTroubleReportIdRequest"
            type="tML-TA:RetrieveAttributesByTroubleReportIdRequestType"/>
<element name="RetrieveTroubleReportFormatByServiceIdRequest"
            type="tML-TA:RetrieveTroubleReportFormatByServiceIdRequestType"/>
<element name="ConfirmNotificationReceiptRequest"
            type="tML-TA:ConfirmNotificationReceiptRequestType"/>

<complexType name="RequestTroubleReportCreationRequestType">
    <sequence>
        <element name="Customer" type="tML-TABase:IdentityType"/>
        <element name="NameBindingId" type="tML-TABase:TRNameBindingIdType"/>
        <element name="ManagedObjectInstance" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="TroubleType" type="tML-TABase:TroubleTypeType"/>
        <element name="AdditionalTroubleInfoList"
                    type="tML-TABase:AdditionalTroubleInfoListType"/>
        <element name="AdditionalCreateInfo" type="tML-TABase:AdditionalCreateRequestType"
                    minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="RetrieveTroubleReportStatusRequestType">
    <sequence>
        <element name="Customer" type="tML-TABase:IdentityType"/>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="TroubleReportIdList" type="tML-TABase:TroubleReportIdListType"/>
    </sequence>
</complexType>
<complexType name="RetrieveTroubleReportFormatRequestType">
    <sequence>
        <element name="Customer" type="tML-TABase:IdentityType"/>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="TroubleReportFormatId" type="tML-TABase:TRFormatIDType"/>
    </sequence>
</complexType>
<complexType name="RetrieveTroubleReportIdByServiceIdRequestType">
    <sequence>
        <element name="Customer" type="tML-TABase:IdentityType"/>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="ServiceId" type="tML-TABase:ServiceIDType"/>
    </sequence>
</complexType>
<complexType name="AddTroubleInfoRequestType">
    <sequence>
        <element name="Customer" type="tML-TABase:IdentityType"/>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="TroubleReportId" type="tML-TABase:NamingStringType"/>
        <element name="AddInfoList" type="tML-TABase:AdditionalTroubleInfoListType"/>
    </sequence>
</complexType>
<complexType name="VerifyRepairCompletionRequestType">
    <sequence>
        <element name="Customer" type="tML-TABase:IdentityType"/>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="TroubleReportId" type="tML-TABase:NamingStringType"/>
        <element name="CloseOutVerification" type="tML-TABase:CloseOutVerificationType"/>
        <element name="VerificationRemarks" type="tML-TABase:AdditionalTroubleInfoListType"
                    minOccurs="0"/>
        <element name="TroubleClearancePerson" type="tML-TABase:PersonReachType"
                    minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="ModifyAttributesRequestType">
    <sequence>
        <element name="Customer" type="tML-TABase:IdentityType"/>
```

```
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="TroubleReportId" type="tML-TABase:NamingStringType"/>
        <element name="ModifyRequest" type="tML-TABase:ModifyRequestType"/>
    </sequence>
</complexType>
<complexType name="CancelTroubleReportRequestType">
    <sequence>
        <element name="Customer" type="tML-TABase:IdentityType"/>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="TroubleReportId" type="tML-TABase:NamingStringType"/>
        <element name="CancelRequestedByManager"
                 type="tML-TABase:CancelRequestedByManagerType"/>
        <element name="CancelRemark" type="tML-TABase:AdditionalTroubleInfoListType"
                 minOccurs="0"/>
        <element name="TroubleClearancePerson" type="tML-TABase:PersonReachType"
                 minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="RetrieveTroubleHistoryByServiceIdRequestType">
    <sequence>
        <element name="Customer" type="tML-TABase:IdentityType"/>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="ServiceId" type="tML-TABase:ServiceIDType"/>
        <element name="OldestEventTime" type="tML-TABase:GeneralizedTimeType" minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="RetrieveAttributesByTroubleReportIdRequestType">
    <sequence>
        <element name="Customer" type="tML-TABase:IdentityType"/>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="TroubleReportId" type="tML-TABase:NamingStringType"/>
        <element name="RetrieveAttributes" type="tML-TABase:AttributeIDListType"/>
    </sequence>
</complexType>
<complexType name="RetrieveTroubleReportFormatByServiceIdRequestType">
    <sequence>
        <element name="Customer" type="tML-TABase:IdentityType"/>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="ServiceId" type="tML-TABase:ServiceIDType"/>
    </sequence>
</complexType>
<complexType name="ConfirmNotificationReceiptRequestType">
    <sequence>
        <element name="Customer" type="tML-TABase:IdentityType"/>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="NotificationReceipt" type="tML-TABase:NotificationReceiptType"/>
    </sequence>
</complexType>
```

4.2.2 Response Portion

```
<element name="RequestTroubleReportCreationResponse"
         type="tML-TA:RequestTroubleReportCreationResponseType"/>
<element name="RetrieveTroubleReportStatusResponse"
         type="tML-TA:RetrieveTroubleReportStatusResponseType"/>
<element name="RetrieveTroubleReportFormatResponse"
         type="tML-TA:RetrieveTroubleReportFormatResponseType"/>
<element name="RetrieveTroubleReportIdByServiceResponse"
         type="tML-TA:RetrieveTroubleReportIdByServiceResponseType"/>
<element name="AddTroubleInfoResponse" type="tML-TA:AddTroubleInfoResponseType"/>
<element name="VerifyRepairCompletionResponse"
         type="tML-TA:VerifyRepairCompletionResponseType"/>
<element name="ModifyAttributesResponse" type="tML-TA:ModifyAttributesResponseType"/>
<element name="CancelTroubleReportResponse" type="tML-TA:CancelTroubleReportResponseType"/>
```

```
<element name="RetrieveTroubleHistoryByServiceIdResponse"
         type="tML-TA:RetrieveTroubleHistoryByServiceIdResponseType"/>
<element name="RetrieveAttributesByTroubleReportIdResponse"
         type="tML-TA:RetrieveAttributesByTroubleReportIdResponseType"/>
<element name="RetrieveTroubleReportFormatByServiceIdResponse"
         type="tML-TA:RetrieveTroubleReportFormatByServiceIdResponseType"/>


<complexType name="RequestTroubleReportCreationResponseType">
    <sequence>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="CreateResponse" type="tML-TABase:CreateResponseType"/>
    </sequence>
</complexType>
<complexType name="RetrieveTroubleReportStatusResponseType">
    <sequence>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="Normal" type="tML-TABase:StatusType" minOccurs="0"/>
        <element name="Exception" type="tML-TABase:ExceptionTroubleReportType" minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="RetrieveTroubleReportFormatResponseType">
    <sequence>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="Normal" type="tML-TABase:FormatType" minOccurs="0"/>
        <element name="Exception" type="tML-TABase:ExceptionTroubleReportType" minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="RetrieveTroubleReportIdByServiceResponseType">
    <sequence>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="Normal" type="tML-TABase:TroubleReportIdListType" minOccurs="0"/>
        <element name="Exception" type="tML-TABase:ExceptionTroubleReportType" minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="AddTroubleInfoResponseType">
    <sequence>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="Exception" type="tML-TABase:ExceptionTroubleReportUpdateType"
                 minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="VerifyRepairCompletionResponseType">
    <sequence>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="Exception" type="tML-TABase:ExceptionVerifyType" minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="ModifyAttributesResponseType">
    <sequence>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="Normal" type="tML-TABase:ModifiedAttributesType" minOccurs="0"/>
        <element name="Exception" type="tML-TABase:ExceptionTroubleReportUpdateType"
                 minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="CancelTroubleReportResponseType">
    <sequence>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="Exception" type="tML-TABase:ExceptionTroubleReportUpdateType"
                 minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="RetrieveTroubleHistoryByServiceIdResponseType">
    <sequence>
```

```
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="Normal" type="tML-TABase:TroubleHistoryInfoListType" minOccurs="0"/>
        <element name="Exception" type="tML-TABase:ExceptionTroubleReportType" minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="RetrieveAttributesByTroubleReportIdResponseType">
    <sequence>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="Normal" type="tML-TABase:RetrievedAttributesType" minOccurs="0"/>
        <element name="Exception" type="tML-TABase:ExceptionTroubleReportType" minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="RetrieveTroubleReportFormatByServiceIdResponseType">
    <sequence>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="Normal" type="tML-TABase:TroubleReportFormatIdListType" minOccurs="0"/>
        <element name="Exception" type="tML-TABase:ExceptionTroubleReportType" minOccurs="0"/>
    </sequence>
</complexType>
```

4.2.3 Notification Portion

```
<element name="TroubleHistoryEventNotification"
            type="tML-TA:TroubleHistoryEventNotificationType"/>
<element name="TroubleReportStatusOrCommitmentTimeUpdateNotification"
            type="tML-TA:TroubleReportStatusOrCommitmentTimeUpdateNotificationType"/>
<element name="TroubleReportProgressNotification"
            type="tML-TA:TroubleReportProgressNotificationType"/>
<element name="TroubleReportCreatedNotification"
            type="tML-TA:TroubleReportCreatedNotificationType"/>
<element name="AttributeValueChangeNotification"
            type="tML-TA:AttributeValueChangeNotificationType"/>
<element name="DeleteTroubleReportNotification"
            type="tML-TA:DeleteTroubleReportNotificationType"/>
<element name="EnrolTroubleReportFormatNotification"
            type="tML-TA:EnrolTroubleReportFormatNotificationType"/>
<element name="DeenrolTroubleReportFormatNotification"
            type="tML-TA:DeenrolTroubleReportFormatNotificationType"/>
<element name="AVCTroubleReportFormatDefinitionNotification"
            type="tML-TA:AVCTroubleReportFormatDefinitionNotificationType"/>

<complexType name="TroubleHistoryEventNotificationType">
    <sequence>
        <element name="NotificationId" type="tML-TABase:NotificationIdType" minOccurs="0"/>
        <element name="EventTime" type="tML-TABase:GeneralizedTimeType"/>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="EventMode" type="tML-TABase:EventModeType"/>
        <element name="EventInformation" type="tML-TABase:TroubleHistoryInfoType"/>
        <element name="CurrentTime" type="tML-TABase:GeneralizedTimeType"/>
    </sequence>
</complexType>
<complexType name="TroubleReportStatusOrCommitmentTimeUpdateNotificationType">
    <sequence>
        <element name="NotificationId" type="tML-TABase:NotificationIdType" minOccurs="0"/>
        <element name="EventTime" type="tML-TABase:GeneralizedTimeType"/>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="StatusOrCommitmentTimeUpdate"
                    type="tML-TABase:StatusOrCommitmentTimeUpdateType"/>
    </sequence>
</complexType>
<complexType name="TroubleReportProgressNotificationType">
    <sequence>
        <element name="NotificationId" type="tML-TABase:NotificationIdType" minOccurs="0"/>
        <element name="EventTime" type="tML-TABase:GeneralizedTimeType"/>
```

```
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="EventMode" type="tML-TABase:EventModeType"/>
        <element name="TroubleReportStatus" type="tML-TABase:TroubleReportStatusType"/>
        <element name="AdditionalTroubleStatusInfo"
                 type="tML-TABase:AdditionalTroubleStatusInfoType" minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="TroubleReportCreatedNotificationType">
    <sequence>
        <element name="NotificationId" type="tML-TABase:NotificationIdType" minOccurs="0"/>
        <element name="EventTime" type="tML-TABase:GeneralizedTimeType"/>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="EventData" type="tML-TABase:ChangeEventType"/>
    </sequence>
</complexType>
<complexType name="AttributeValueChangeNotificationType">
    <sequence>
        <element name="NotificationId" type="tML-TABase:NotificationIdType" minOccurs="0"/>
        <element name="EventTime" type="tML-TABase:GeneralizedTimeType"/>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="EventData" type="tML-TABase:ChangeEventType"/>
    </sequence>
</complexType>
<complexType name="DeleteTroubleReportNotificationType">
    <sequence>
        <element name="NotificationId" type="tML-TABase:NotificationIdType" minOccurs="0"/>
        <element name="EventTime" type="tML-TABase:GeneralizedTimeType"/>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="EventData" type="tML-TABase:ChangeEventType"/>
    </sequence>
</complexType>
<complexType name="EnrolTroubleReportFormatNotificationType">
    <sequence>
        <element name="NotificationId" type="tML-TABase:NotificationIdType" minOccurs="0"/>
        <element name="EventTime" type="tML-TABase:GeneralizedTimeType"/>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="TroubleReportFormatId" type="tML-TABase:TRFormatIDType"/>
    </sequence>
</complexType>
<complexType name="DeenrolTroubleReportFormatNotificationType">
    <sequence>
        <element name="NotificationId" type="tML-TABase:NotificationIdType" minOccurs="0"/>
        <element name="EventTime" type="tML-TABase:GeneralizedTimeType"/>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="TroubleReportFormatId" type="tML-TABase:TRFormatIDType"/>
    </sequence>
</complexType>
<complexType name="AVCTroubleReportFormatDefinitionNotificationType">
    <sequence>
        <element name="NotificationId" type="tML-TABase:NotificationIdType" minOccurs="0"/>
        <element name="EventTime" type="tML-TABase:GeneralizedTimeType"/>
        <element name="TargetObjectName" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="TroubleReportFormatId" type="tML-TABase:TRFormatIDType"/>
        <element name="AddedAttributesList" type="tML-TABase:AttributeIDListType"/>
        <element name="RemovedAttributesList" type="tML-TABase:AttributeIDListType"/>
    </sequence>
</complexType>
```

4.3 Schema for tML-TABase

4.3.1 Attributes Types

```
<simpleType name="NullStringType">
    <restriction base="string">
        <length value="0"/>
    </restriction>
</simpleType>
<complexType name="NullObjectType" final="#all"/>
<simpleType name="GraphicStringType">
    <restriction base="string">
        <pattern value="[A-Z|a-z|0-9|\s|'|\(|\)|\+|,|\-|\.|/|:|=|\?|@|;|!|”|#|$|%|*|`|\[|\]|\\|^|_|{|\||}|~|&amp;|&lt;|&gt;|&quot;]*"/>
    </restriction>
</simpleType>
<simpleType name="PrintableStringType">
    <restriction base="string">
        <pattern value="[A-Z|a-z|0-9|\s|'|\(|\)|\+|,|\-|\.|/|:|=|\?]*"/>
    </restriction>
</simpleType>
<simpleType name="VisibleStringType">
    <restriction base="string">
        <pattern value="[A-Z|a-z|0-9|\s|'|\(|\)|\+|,|\-|\.|/|:|=|\?]*"/>
    </restriction>
</simpleType>
<simpleType name="VisibleString64Type">
    <restriction base="tML-TABase:VisibleStringType">
        <minLength value="0"/>
        <maxLength value="64"/>
    </restriction>
</simpleType>
<simpleType name="SingleBitStringType">
    <restriction base="string">
        <length value="1"/>
        <pattern value="[0-1]"/>
    </restriction>
</simpleType>
<simpleType name="AttributeIdType">
    <restriction base="string"/>
</simpleType>
<simpleType name="GraphicString64Type">
    <restriction base="tML-TABase:GraphicStringType">
        <minLength value="0"/>
        <maxLength value="64"/>
    </restriction>
</simpleType>
<simpleType name="GraphicString128Type">
    <restriction base="tML-TABase:GraphicStringType">
        <minLength value="0"/>
        <maxLength value="128"/>
    </restriction>
</simpleType>
<simpleType name="GraphicString256Type">
    <restriction base="tML-TABase:GraphicStringType">
        <minLength value="0"/>
        <maxLength value="256"/>
    </restriction>
</simpleType>
<simpleType name="ObjectIdentifierType">
    <annotation>
        <documentation>
        When an ObjectIdentifier is being used for creating a company specific extension
            (seen as Identifier tag), use a company's ECC or ACNA as the prefix, followed by an
```

```
            underscore (“_”) then the company own definition obeying W3C XML tag rules to ensure
            uniqueness.
        </documentation>
    </annotation>
    <restriction base="tML-TABase:GraphicStringType"/>
</simpleType>
<simpleType name="GeneralizedTimeType">
    <restriction base="dateTime"/>
</simpleType>
<simpleType name="Time24Type">
    <restriction base="time"/>
</simpleType>
<complexType name="SignatureType">
    <sequence>
        <element name="Type" type="tML-TABase:GraphicString64Type"/>
        <element name="Sequence" type="tML-TABase:GraphicString64Type" minOccurs="0"/>
        <element name="Usage" type="tML-TABase:GraphicString64Type" minOccurs="0"/>
        <element name="Predecessor" type="tML-TABase:GraphicString64Type" minOccurs="0"/>
        <element name="Successor" type="tML-TABase:GraphicString64Type" minOccurs="0"/>
        <element name="Function" type="tML-TABase:GraphicString64Type" minOccurs="0"/>
        <element name="Size" type="tML-TABase:GraphicString64Type" minOccurs="0"/>
        <element name="Constraint" type="tML-TABase:GraphicString256Type" minOccurs="0"/>
        <element name="Multiplicity" type="tML-TABase:GraphicString64Type" minOccurs="0"/>
        <element name="Note" type="tML-TABase:GraphicString256Type" minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="GenericDataType">
    <sequence>
        <element name="Tag" type="tML-TABase:GraphicString64Type"/>
        <element name="Value" type="tML-TABase:GraphicStringType"/>
        <element name="Signature" type="tML-TABase:SignatureType" minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="GenericDataListType">
    <sequence>
        <element name="GenericDataItem" type="tML-TABase:GenericDataType"
                 minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="IdentityType">
    <sequence>
        <element name="UserId" type="tML-TABase:UserIdType" minOccurs="0"/>
        <element name="AccessControl" type="tML-TABase:AccessControlType" minOccurs="0"/>
        <element name="ServiceProfile" type="tML-TABase:ServiceProfileType" minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="CustomerRecordType">
    <all>
        <element name="Name" type="tML-TABase:VisibleStringType"/>
        <element name="MailingAddress" type="tML-TABase:VisibleStringType"/>
        <element name="AccountNumber" type="integer"/>
        <element name="BalanceDue" type="integer" minOccurs="0"/>
    </all>
</complexType>
<complexType name="AccessControl1Type" final="#all">
    <sequence>
        <element name="EntityIdentifier">
            <simpleType>
                <restriction base="tML-TABase:GraphicStringType">
                    <minLength value="1"/>
                    <maxLength value="64"/>
                </restriction>
            </simpleType>
        </element>
        <element name="EncrytedString">
```

```
            <simpleType>
                <restriction base="tML-TABase:OctetStringType">
                    <minLength value="1"/>
                    <maxLength value="64"/>
                </restriction>
            </simpleType>
        </element>
    </sequence>
</complexType>
<complexType name="AccessControl2Type" final="#all">
    <sequence>
        <element name="EntityIdentifier">
            <simpleType>
                <restriction base="tML-TABase:GraphicStringType">
                    <minLength value="1"/>
                    <maxLength value="64"/>
                </restriction>
            </simpleType>
        </element>
        <element name="InitializationVector">
            <simpleType>
                <restriction base="tML-TABase:OctetStringType">
                    <length value="8"/>
                </restriction>
            </simpleType>
        </element>
        <element name="KeyIdentifier" type="integer"/>
        <element name="EncrytedString">
            <simpleType>
                <restriction base="tML-TABase:OctetStringType">
                    <minLength value="8"/>
                    <maxLength value="64"/>
                </restriction>
            </simpleType>
        </element>
    </sequence>
</complexType>
<complexType name="AccessControlType">
    <choice>
        <element name="AccessControl1" type="tML-TABase:AccessControl1Type"/>
        <element name="AccessControl2" type="tML-TABase:AccessControl2Type"/>
    </choice>
</complexType>
<simpleType name="OctetStringType">
    <restriction base="hexBinary"/>
</simpleType>
<simpleType name="NotificationIdType">
    <restriction base="integer"/>
</simpleType>
<simpleType name="AttributeTypeType">
    <restriction base="tML-TABase:ObjectIdentifierType"/>
</simpleType>
<simpleType name="AttributeValueType">
    <restriction base="tML-TABase:PrintableStringType"/>
</simpleType>
<complexType name="AttributeValueAssertionType">
    <sequence>
        <element name="Type" type="tML-TABase:AttributeTypeType"/>
        <element name="Assertion" type="tML-TABase:AttributeValueType"/>
    </sequence>
</complexType>
<complexType name="RelativeDistinguishedNameType">
    <sequence>
        <element name="RelativeDistinguishedNameItem"
                 type="tML-TABase:AttributeValueAssertionType"
```

```
                minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="RDNSequenceType">
    <sequence>
        <element name="RDNSequenceItem" type="tML-TABase:AttributeValueAssertionType"
                minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="DistinguishedNameType">
    <complexContent>
        <extension base="tML-TABase:RDNSequenceType"/>
    </complexContent>
</complexType>
<complexType name="ObjectClassType">
    <choice>
        <element name="GlobalForm" type="tML-TABase:ObjectIdentifierType"/>
        <element name="LocalForm" type="integer"/>
    </choice>
</complexType>
<complexType name="ObjectInstanceType">
    <choice>
        <element name="DistinguishedName" type="tML-TABase:DistinguishedNameType"/>
        <element name="NonSpecificForm" type="tML-TABase:OctetStringType"/>
        <element name="LocalDistinguishedName" type="tML-TABase:RDNSequenceType"/>
    </choice>
</complexType>
<simpleType name="ActivityCodeValueType">
    <restriction base="integer">
        <enumeration value="0"/>
        <enumeration value="1"/>
        <enumeration value="2"/>
        <enumeration value="3"/>
        <enumeration value="4"/>
        <enumeration value="5"/>
        <enumeration value="6"/>
        <enumeration value="7"/>
        <enumeration value="8"/>
        <enumeration value="9"/>
        <enumeration value="10"/>
        <enumeration value="11"/>
        <enumeration value="12"/>
    </restriction>
</simpleType>
<complexType name="ActivityCodeType" final="#all">
    <choice>
        <element name="Number" type="tML-TABase:ActivityCodeValueType"/>
        <element name="Identifier" type="tML-TABase:ObjectIdentifierType"/>
    </choice>
</complexType>
<complexType name="ActivityDurationType">
    <sequence>
        <element name="ActivityDurationItem" minOccurs="0" maxOccurs="unbounded">
            <complexType>
                <sequence>
                    <element name="Duration" type="tML-TABase:TimeIntervalType"/>
                    <element name="Billable" type="boolean" default="true"/>
                    <element name="Type" type="tML-TABase:ActivityTypeType" minOccurs="0"/>
                </sequence>
            </complexType>
        </element>
    </sequence>
</complexType>
<simpleType name="ActivityInfoType">
    <restriction base="tML-TABase:GraphicString256Type"/>
```

```
</simpleType>
<complexType name="ActivityPersonType">
    <complexContent>
        <extension base="tML-TABase:PersonReachType"/>
    </complexContent>
</complexType>
<complexType name="ActivityTypeType">
    <sequence>
        <element name="AfterHoursRepair" type="boolean"/>
        <element name="Standby" type="boolean"/>
        <element name="AfterHoursStandby" type="boolean"/>
        <element name="Test" type="boolean"/>
        <element name="ManagerInitiatedTest" type="boolean"/>
        <element name="Dispatch" type="boolean"/>
        <element name="NoAccess" type="boolean"/>
        <element name="DelayedMaintenance" type="boolean"/>
        <element name="Release" type="boolean"/>
        <element name="DeregulatedWork" type="boolean"/>
        <element name="AuthorizeToWork" type="boolean"/>
    </sequence>
</complexType>
<simpleType name="AdditionalTextType">
    <restriction base="tML-TABase:GraphicStringType">
        <minLength value="0"/>
        <maxLength value="256"/>
    </restriction>
</simpleType>
<complexType name="AdditionalTroubleInfoListType">
    <sequence>
        <element name="AdditionalTroubleInfoItem" minOccurs=”1” maxOccurs=”unbounded”>
            <simpleType>
                <restriction base="tML-TABase:GraphicStringType">
                    <minLength value="0"/>
                    <maxLength value="400"/>
                </restriction>
            </simpleType>
        </element>

    </sequence>
</complexType>
<complexType name="AdditionalTroubleStatusInfoType">
    <sequence>
        <element name="AdditionalTroubleStatusInfoItem" minOccurs=”0” maxOccurs=”unbounded”>
            <simpleType>
                <restriction base="tML-TABase:GraphicStringType">
                    <minLength value="0"/>
                    <maxLength value="400"/>
                </restriction>
            </simpleType>
        </element>
    </sequence>

</complexType>
<simpleType name="AfterHrsRepairAuthType">
    <restriction base="boolean"/>
</simpleType>
<complexType name="AgentContactPersonType">
    <complexContent>
        <extension base="tML-TABase:PersonReachType"/>
    </complexContent>
</complexType>
<complexType name="AlarmRecordPtrListType">
    <sequence>
        <element name="AlarmRecordPtrItem" type="tML-TABase:ObjectInstanceType"
                 minOccurs="0" maxOccurs="unbounded"/>
```

```
    </sequence>
</complexType>
<complexType name="AlternateManagerContactPersonType">
    <complexContent>
        <extension base="tML-TABase:PersonReachType"/>
    </complexContent>
</complexType>
<complexType name="ApplicableManagedObjectClassListType">
    <sequence>
        <element name="ApplicableManagedObjectClassItem" type="tML-TABase:ObjectClassType"
                 minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="ApplicableManagedObjectInstanceType">
    <sequence>
        <element name="ApplicableManagedObjectInstanceItem"
                 type="tML-TABase:ObjectInstanceType" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="AttributeIDListType">
    <sequence>
        <element name="AttributeIDItem" type="tML-TABase:AttributeIdType"
                 minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="AuthorizationListType">
    <sequence>
        <element name="AuthorizationItem" minOccurs="0" maxOccurs="unbounded">
            <complexType>
                <sequence>
                    <element name="State" type="tML-TABase:RequestStateType"/>
                    <element name="Type" type="tML-TABase:ActivityTypeType"/>
                    <element name="AuthTime" type="tML-TABase:AuthorizationTimeType"
                             minOccurs="0"/>
                    <element name="AuthPerson" type="tML-TABase:PersonReachType" minOccurs="0"/>
                </sequence>
            </complexType>
        </element>
    </sequence>
</complexType>
<simpleType name="AuthorizationTimeType">
    <restriction base="tML-TABase:GraphicStringType"/>
</simpleType>
<complexType name="CallBackInfoListType">
    <sequence>
        <element name="CallBackInfoItem" minOccurs="0" maxOccurs="unbounded">
            <complexType>
                <choice>
                    <element name="Escalation" type="tML-TABase:PersonReachType"/>
                    <element name="BeforeAutoTest" type="tML-TABase:PersonReachType"/>
                    <element name="AfterCleared" type="tML-TABase:PersonReachType"/>
                </choice>
            </complexType>
        </element>
    </sequence>
</complexType>
<simpleType name="CalledNumberType">
    <restriction base="tML-TABase:GraphicString64Type"/>
</simpleType>
<simpleType name="CancelRequestedByManagerType">
    <restriction base="boolean"/>
</simpleType>
<simpleType name="ChangeDeniedReasonType" final="#all">
    <restriction base="integer">
        <enumeration value="1"/>
```

```
            <enumeration value="2"/>
            <enumeration value="3"/>
        </restriction>
    </simpleType>
    <simpleType name="CloseOutNarrType">
        <restriction base="tML-TABase:GraphicString256Type"/>
    </simpleType>
    <simpleType name="CloseOutVerificationType">
        <restriction base="integer">
            <enumeration value="0"/>
            <enumeration value="1"/>
            <enumeration value="2"/>
            <enumeration value="3"/>
            <enumeration value="4"/>
        </restriction>
    </simpleType>
    <complexType name="CommitmentTimeType">
        <choice>
            <element name="OnsiteTime" type="tML-TABase:GeneralizedTimeType"/>
            <element name="ClearedTime" type="tML-TABase:GeneralizedTimeType"/>
        </choice>
    </complexType>
    <simpleType name="CustomerWorkCenterType">
        <restriction base="tML-TABase:GraphicString64Type"/>
    </simpleType>
    <simpleType name="CustTroubleTickNumType">
        <restriction base="tML-TABase:GraphicString64Type"/>
    </simpleType>
    <complexType name="CustomerInfoType">
        <sequence>
            <element name="PIC" type="tML-TABase:VisibleString64Type" minOccurs="0"/>
            <element name="LPIC" type="tML-TABase:VisibleString64Type" minOccurs="0"/>
            <element name="LRN" type="tML-TABase:VisibleString64Type" minOccurs="0"/>
            <element name="OCN" type="tML-TABase:VisibleString64Type" minOccurs="0"/>
        </sequence>
    </complexType>
    <simpleType name="DialogType">
        <restriction base="tML-TABase:GraphicStringType">
            <minLength value="0"/>
            <maxLength value="640"/>
        </restriction>
    </simpleType>
    <simpleType name="EntryTimeType">
        <restriction base="tML-TABase:GeneralizedTimeType"/>
    </simpleType>
    <complexType name="EscalationListType">
        <sequence>
            <element name="EscalationItem" minOccurs="0" maxOccurs="unbounded">
                <complexType>
                    <sequence>
                        <element name="State" type="tML-TABase:RequestStateType"/>
                        <element name="EscTime" type="tML-TABase:EscalationTimeType"/>
                        <element name="RequestPerson" type="tML-TABase:PersonReachType"/>
                        <element name="Level" type="tML-TABase:OrgLevelType" minOccurs="0"/>
                        <element name="EscPerson" type="tML-TABase:PersonReachType" minOccurs="0"/>
                    </sequence>
                </complexType>
            </element>
        </sequence>
    </complexType>
    <simpleType name="EscalationTimeType">
        <restriction base="tML-TABase:GeneralizedTimeType"/>
    </simpleType>
    <simpleType name="EstimatedTimeOfRestoralType">
        <restriction base="tML-TABase:GeneralizedTimeType"/>
```

```
</simpleType>
<simpleType name="EventModeType">
    <restriction base="string">
        <enumeration value="Confirmed"/>
        <enumeration value="NonConfirmed"/>
        <enumeration value="NotApplicable"/>
    </restriction>
</simpleType>
<complexType name="EventReplyType">
    <sequence>
        <element name="EventTypeId" type="tML-TABase:EventTypeIdType"/>
        <element name="EventReplyInfo" type="tML-TABase:GraphicStringType"/>
    </sequence>
</complexType>
<complexType name="EventTypeIdType">
    <choice>
        <element name="GlobalForm" type="tML-TABase:ObjectIdentifierType"/>
        <element name="LocalForm" type="integer"/>
    </choice>
</complexType>
<simpleType name="EventTypeType">
    <restriction base="string">
        <enumeration value="TroubleHistoryEvent"/>
        <enumeration value="TroubleReportStatusOrCommitmentTimeUpdate"/>
        <enumeration value="TroubleReportProgress"/>
        <enumeration value="TroubleReportCreated"/>
        <enumeration value="TroubleReportDeleted"/>
        <enumeration value="AVC"/>
        <enumeration value="EnrolTroubleReportFormatDefinition"/>
        <enumeration value="DeenrolTroubleReportFormatDefinition"/>
        <enumeration value="AVCTroubleReportFormatDefinition"/>
    </restriction>
</simpleType>
<simpleType name="HandOffCenterType">
    <restriction base="tML-TABase:GraphicString64Type"/>
</simpleType>
<simpleType name="HandOffLocationType">
    <restriction base="tML-TABase:GraphicStringType">
        <minLength value="0"/>
        <maxLength value="64"/>
    </restriction>
</simpleType>
<complexType name="HandOffPersonType">
    <complexContent>
        <extension base="tML-TABase:PersonReachType"/>
    </complexContent>
</complexType>
<simpleType name="HandOffTimeType">
    <union memberTypes="tML-TABase:NullStringType tML-TABase:GeneralizedTimeType"/>
</simpleType>
<simpleType name="InitiatingModeType">
    <restriction base="integer">
        <enumeration value="0"/>
        <enumeration value="1"/>
        <enumeration value="2"/>
        <enumeration value="4"/>
        <enumeration value="5"/>
        <enumeration value="6"/>
        <enumeration value="7"/>
    </restriction>
</simpleType>
<simpleType name="LastUpdateTimeType">
    <restriction base="tML-TABase:GeneralizedTimeType"/>
</simpleType>
<complexType name="LocationAddressType">
```

```
    <sequence>
        <element name="PremisesName" type="tML-TABase:PremisesNameType"/>
        <element name="PremisesAddress" type="tML-TABase:PremisesAddressType"/>
    </sequence>
</complexType>
<complexType name="LocationAccessHoursType">
    <sequence>
        <element name="LocationAccessHoursItem" type="tML-TABase:WeekMaskType"
                 minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="MaintenanceOrgContactPersonType">
    <complexContent>
        <extension base="tML-TABase:PersonReachType"/>
    </complexContent>
</complexType>
<complexType name="MaintenanceOrgContactTimeType">
    <choice>
        <element name="Null" type="tML-TABase:NullObjectType"/>
        <element name="MaintenanceOrgContactTime" type="tML-TABase:GeneralizedTimeType"/>
    </choice>
</complexType>
<simpleType name="MaintServiceChargeType">
    <restriction base="boolean"/>
</simpleType>
<simpleType name="ManagedObjectAccessFromTimeType">
    <restriction base="tML-TABase:GeneralizedTimeType"/>
</simpleType>
<complexType name="ManagedObjectAccessHoursType">
    <sequence>
        <element name="ManagedObjectAccessHoursItem" type="tML-TABase:WeekMaskType"
                 minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="ManagedObjectAccessToTimeType">
    <complexContent>
        <extension base="tML-TABase:StopTimeType"/>
    </complexContent>
</complexType>
<complexType name="ManagedObjectInstanceType">
    <complexContent>
        <restriction base="tML-TABase:ObjectInstanceType">
            <choice>
                <element name="DistinguishedName" type="tML-TABase:DistinguishedNameType"/>
                <element name="LocalDistinguishedName" type="tML-TABase:RDNSequenceType"/>
            </choice>
        </restriction>
    </complexContent>
</complexType>
<complexType name="ManagedObjectInstanceAliasListType">
    <sequence>
        <element name="ManagedObjectInstanceAliasItem" type="tML-TABase:GraphicString256Type"
                 minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="ManagerContactPersonType">
    <complexContent>
        <extension base="tML-TABase:PersonReachType"/>
    </complexContent>
</complexType>
<simpleType name="ManagerSearchStringType">
    <restriction base="tML-TABase:GraphicString64Type"/>
</simpleType>
<complexType name="ManagerSearchKeyType">
    <choice>
```

```
        <element name="ManagerSearchString" type="tML-TABase:ManagerSearchStringType"/>
        <element name="ObjectInstance" type="tML-TABase:ObjectInstanceType"/>
    </choice>
</complexType>
<complexType name="ManagerSearchKey123Type">
    <sequence>
        <element name="ManagerSearchKey1" type="tML-TABase:ManagerSearchKeyType"/>
        <element name="ManagerSearchKey2" type="tML-TABase:ManagerSearchKeyType"/>
        <element name="ManagerSearchKey3" type="tML-TABase:ManagerSearchKeyType"/>
    </sequence>
</complexType>
<complexType name="ManagerSearchKeyListType">
    <sequence>
        <element name="ManagerSearchKeyItem" type="tML-TABase:ManagerSearchKeyType"
                 minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="ManagerSearchKeyCombineType">
    <choice>
        <element name="ManagerSearchKey123" type="tML-TABase:ManagerSearchKey123Type"/>
        <element name="ManagerSearchKeyList" type="tML-TABase:ManagerSearchKeyListType"/>
    </choice>
</complexType>
<simpleType name="NamingStringType">
    <restriction base="tML-TABase:GraphicStringType">
        <minLength value="0"/>
        <maxLength value="32"/>
    </restriction>
</simpleType>
<simpleType name="NameTypeType">
    <union memberTypes="integer tML-TABase:GraphicStringType"/>
</simpleType>
<complexType name="NotificationReceiptType">
    <sequence>
        <element name="NotificationId" type="tML-TABase:NotificationIdType" minOccurs="0"/>
        <element name="EventType" type="tML-TABase:EventTypeType"/>
        <element name="CurrentTime" type="tML-TABase:GeneralizedTimeType" minOccurs="0"/>
        <element name="EventReply" type="tML-TABase:EventReplyType" minOccurs="0"/>
        <element name="Exception" type="tML-TABase:ExceptionEventType" minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="OutageDurationType">
    <choice>
        <element name="Null" type="tML-TABase:NullObjectType"/>
        <element name="TimeInterval" type="tML-TABase:TimeIntervalType"/>
    </choice>
</complexType>
<simpleType name="OrgLevelType">
    <restriction base="integer">
        <enumeration value="0"/>
        <enumeration value="1"/>
        <enumeration value="2"/>
        <enumeration value="3"/>
        <enumeration value="4"/>
        <enumeration value="5"/>
        <enumeration value="6"/>
    </restriction>
</simpleType>
<complexType name="PerceivedTroubleSeverityType">
    <choice>
        <element name="Number">
            <simpleType>
                <restriction base="integer">
                    <enumeration value="0"/>
                    <enumeration value="1"/>
```

```
                    <enumeration value="2"/>
                    <enumeration value="3"/>
                </restriction>
            </simpleType>
        </element>
        <element name="identifier" type="tML-TABase:ObjectIdentifierType"/>
    </choice>
</complexType>
<simpleType name="PersonEmailType">
    <restriction base="tML-TABase:GraphicString64Type"/>
</simpleType>
<simpleType name="PersonFaxType">
    <restriction base="tML-TABase:GraphicString64Type"/>
</simpleType>
<complexType name="PersonLocationType">
    <complexContent>
        <extension base="tML-TABase:PremisesAddressType"/>
    </complexContent>
</complexType>
<simpleType name="PersonNameType">
    <restriction base="tML-TABase:GraphicString64Type"/>
</simpleType>
<simpleType name="PersonNumberType">
    <restriction base="tML-TABase:GraphicString64Type"/>
</simpleType>
<simpleType name="PersonPhoneType">
    <restriction base="tML-TABase:GraphicString64Type"/>
</simpleType>
<simpleType name="PersonResponType">
    <restriction base="tML-TABase:GraphicString64Type"/>
</simpleType>
<complexType name="PersonReachType">
    <sequence>
        <element name="Number" minOccurs="0" default="" type="tML-TABase:PersonNumberType"/>
        <element name="Name" default="" type="tML-TABase:PersonNameType"/>
        <element name="Phone" minOccurs="0" type="tML-TABase:PersonPhoneType"/>
        <element name="Loc" minOccurs="0" type="tML-TABase:PersonLocationType"/>
        <element name="Email" minOccurs="0" type="tML-TABase:PersonEmailType"/>
        <element name="Fax" minOccurs="0" type="tML-TABase:PersonFaxType"/>
        <element name="Respon" minOccurs="0" type="tML-TABase:PersonResponType"/>
        <element name="Pager" minOccurs="0" type="tML-TABase:PersonPhoneType"/>
    </sequence>
</complexType>
<simpleType name="PreferredPriorityType">
    <restriction base="integer">
        <enumeration value="0"/>
        <enumeration value="1"/>
        <enumeration value="2"/>
        <enumeration value="3"/>
        <enumeration value="4"/>
    </restriction>
</simpleType>
<simpleType name="PremisesNameType">
    <restriction base="tML-TABase:GraphicString64Type"/>
</simpleType>
<complexType name="PremisesAddressType">
    <sequence>
        <element name="CivicAddress" type="tML-TABase:GraphicString64Type"/>
        <element name="City" type="tML-TABase:GraphicString64Type"/>
        <element name="State" type="tML-TABase:GraphicString64Type"/>
        <element name="Zip" type="tML-TABase:GraphicString64Type"/>
    </sequence>
</complexType>
<simpleType name="ReceivedTimeType">
    <restriction base="tML-TABase:GeneralizedTimeType"/>
```

```
    </simpleType>
    <complexType name="RelatedTroubleReportListType">
        <sequence>
            <element name="RelatedTroubleReporItem" minOccurs="0" maxOccurs="unbounded">
                <complexType>
                    <complexContent>
                        <restriction base="tML-TABase:ObjectInstanceType">
                            <choice>
                                <element name="DistinguishedName"
                                        type="tML-TABase:DistinguishedNameType"/>
                                <element name="LocalDistinguishedName"
                                        type="tML-TABase:RDNSequenceType"/>
                            </choice>
                        </restriction>
                    </complexContent>
                </complexType>
            </element>
        </sequence>
    </complexType>
    <simpleType name="RepairActivityIDType">
        <restriction base="integer"/>
    </simpleType>
    <complexType name="RepairActivityListType">
        <sequence>
            <element name="RepairActivityItem" minOccurs="0" maxOccurs="unbounded">
                <complexType>
                    <sequence>
                        <element name="EntryTime" type="tML-TABase:GeneralizedTimeType"/>
                        <element name="ActivityInfo" type="tML-TABase:GraphicStringType"/>
                        <element name="ActivityPerson" type="tML-TABase:PersonReachType"
                                minOccurs="0"/>
                        <element name="ActivityCode" type="tML-TABase:ActivityCodeType" minOccurs="0"/>
                    </sequence>
                </complexType>
            </element>
        </sequence>
    </complexType>
    <simpleType name="RepeatReportType">
        <restriction base="integer">
            <enumeration value="0"/>
            <enumeration value="1"/>
            <enumeration value="2"/>
            <enumeration value="3"/>
            <enumeration value="4"/>
            <enumeration value="5"/>
        </restriction>
    </simpleType>
    <simpleType name="RequestStateType">
        <restriction base="integer">
            <enumeration value="1"/>
            <enumeration value="2"/>
            <enumeration value="3"/>
        </restriction>
    </simpleType>
    <simpleType name="RestoredTimeType">
        <union memberTypes="tML-TABase:NullStringType tML-TABase:GeneralizedTimeType"/>
    </simpleType>
    <simpleType name="ServiceAliasType">
        <restriction base="tML-TABase:Prin
StringType">
            <maxLength value="64"/>
        </restriction>
    </simpleType>
    <complexType name="ServiceAliasListType">
        <sequence>
```

```
        <element name="ServiceAliasItem" minOccurs="0" maxOccurs="unbounded">
            <simpleType>
                <restriction base="tML-TABase:GraphicStringType">
                    <minLength value="0"/>
                    <maxLength value="64"/>
                </restriction>
            </simpleType>
        </element>
    </sequence>
</complexType>
<simpleType name="ServiceIDType">
    <restriction base="tML-TABase:NameTypeType"/>
</simpleType>
<complexType name="ServiceLocationListType">
    <sequence>
        <element name="ServiceLocationItem" minOccurs="0" maxOccurs="unbounded">
            <complexType>
                <sequence>
                    <element name="PremisesName" type="tML-TABase:PremisesNameType"/>
                    <element name="PremisesAddress" type="tML-TABase:PremisesAddressType"/>
                </sequence>
            </complexType>
        </element>
    </sequence>
</complexType>
<simpleType name="ServiceProfileType">
    <restriction base="tML-TABase:GraphicStringType"/>
</simpleType>
<complexType name="ServiceTypeType">
    <choice>
        <element name="Number" type="integer"/>
        <element name="PString" type="tML-TABase:PrintableStringType"/>
        <element name="Identifier" type="tML-TABase:ObjectIdentifierType"/>
    </choice>
</complexType>
<simpleType name="StatusOfSystemType">
    <restriction base="integer">
        <enumeration value="0"/>
        <enumeration value="1"/>
        <enumeration value="2"/>
        <enumeration value="3"/>
    </restriction>
</simpleType>
<complexType name="StopTimeType">
    <choice>
        <element name="Specific" type="tML-TABase:GeneralizedTimeType"/>
        <element name="Continual" type="tML-TABase:NullStringType"/>
    </choice>
</complexType>
<complexType name="SuspectObjectType">
    <sequence>
        <element name="SuspectObjectClass" type="tML-TABase:ObjectIdentifierType"/>
        <element name="SuspectObjectInstance" type="tML-TABase:ObjectInstanceType"/>
        <element name="FailureProbability" minOccurs="0">
            <simpleType>
                <restriction base="integer">
                    <minInclusive value="1"/>
                    <maxInclusive value="100"/>
                </restriction>
            </simpleType>
        </element>
    </sequence>
</complexType>
<complexType name="SuspectObjectListType">
    <sequence>
```

```
        <element name="SuspectObjectItem" type="tML-TABase:SuspectObjectType"
                 minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<simpleType name="DayType">
    <restriction base="integer">
        <minInclusive value="0"/>
        <maxInclusive value="31"/>
    </restriction>
</simpleType>
<simpleType name="HourType">
    <restriction base="integer">
        <minInclusive value="0"/>
        <maxInclusive value="23"/>
    </restriction>
</simpleType>
<simpleType name="MinuteType">
    <restriction base="integer">
        <minInclusive value="0"/>
        <maxInclusive value="59"/>
    </restriction>
</simpleType>
<simpleType name="SecondType">
    <restriction base="integer">
        <minInclusive value="0"/>
        <maxInclusive value="59"/>
    </restriction>
</simpleType>
<simpleType name="MsecType">
    <restriction base="integer">
        <minInclusive value="0"/>
        <maxInclusive value="999"/>
    </restriction>
</simpleType>
<complexType name="TimeIntervalType">
    <sequence>
        <element name="Day" type="tML-TABase:DayType" default="0"/>
        <element name="Hour" type="tML-TABase:HourType" default="0"/>
        <element name="Minute" type="tML-TABase:MinuteType" default="0"/>
        <element name="Second" type="tML-TABase:SecondType" default="0"/>
        <element name="Msec" type="tML-TABase:MsecType" default="0"/>
    </sequence>
</complexType>
<simpleType name="TRFormatIDType">
    <restriction base="integer"/>
</simpleType>
<complexType name="TroubleReportFormatIdListType">
    <sequence>
        <element name="TroubleReportFormatIdItem" type="tML-TABase:TRFormatIDType"
                 minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<simpleType name="TroubleDetectionTimeType">
    <union memberTypes="tML-TABase:NullStringType tML-TABase:GeneralizedTimeType"/>
</simpleType>
<simpleType name="TroubleFoundValueType">
    <restriction base="integer">
        <enumeration value="0"/>
        <enumeration value="1"/>
        <enumeration value="2"/>
        <enumeration value="3"/>
        <enumeration value="4"/>
        <enumeration value="5"/>
        <enumeration value="6"/>
        <enumeration value="7"/>
```

```
            <enumeration value="8"/>
            <enumeration value="9"/>
            <enumeration value="10"/>
            <enumeration value="11"/>
            <enumeration value="12"/>
            <enumeration value="13"/>
            <enumeration value="14"/>
            <enumeration value="15"/>
            <enumeration value="16"/>
            <enumeration value="17"/>
            <enumeration value="18"/>
            <enumeration value="19"/>
            <enumeration value="20"/>
            <enumeration value="21"/>
            <enumeration value="22"/>
            <enumeration value="23"/>
            <enumeration value="24"/>
            <enumeration value="25"/>
            <enumeration value="26"/>
            <enumeration value="27"/>
            <enumeration value="28"/>
            <enumeration value="29"/>
            <enumeration value="30"/>
            <enumeration value="31"/>
            <enumeration value="32"/>
            <enumeration value="33"/>
            <enumeration value="34"/>
            <enumeration value="35"/>
            <enumeration value="36"/>
            <enumeration value="37"/>
            <enumeration value="38"/>
            <enumeration value="39"/>
            <enumeration value="40"/>
            <enumeration value="41"/>
            <enumeration value="42"/>
            <enumeration value="43"/>
            <enumeration value="44"/>
            <enumeration value="45"/>
            <enumeration value="46"/>
            <enumeration value="47"/>
            <enumeration value="48"/>
            <enumeration value="49"/>
            <enumeration value="50"/>
            <enumeration value="51"/>
            <enumeration value="52"/>
            <enumeration value="53"/>
            <enumeration value="54"/>
            <enumeration value="55"/>
            <enumeration value="56"/>
            <enumeration value="57"/>
            <enumeration value="58"/>
            <enumeration value="59"/>
            <enumeration value="60"/>
        </restriction>
    </simpleType>
    <complexType name="TroubleFoundType">
        <choice>
            <element name="Number" type="tML-TABase:TroubleFoundValueType"/>
            <element name="Identifier" type="tML-TABase:ObjectIdentifierType"/>
        </choice>
    </complexType>
    <complexType name="TroubleHistoryInfoType">
        <sequence>
            <element name="ManagedObjectInstance" type="tML-TABase:ManagedObjectInstanceType"/>
            <element name="ReceivedTime" type="tML-TABase:ReceivedTimeType"/>
```

```
        <element name="TroubleFound" type="tML-TABase:TroubleFoundType"/>
        <element name="ActivityDuration" type="tML-TABase:ActivityDurationType" minOccurs="0"/>
        <element name="AdditionalTroubleInfoList"
                 type="tML-TABase:AdditionalTroubleInfoListType" minOccurs="0"/>
        <element name="AuthorizationList" type="tML-TABase:AuthorizationListType"
                 minOccurs="0"/>
        <element name="CancelRequestedByManager" type="tML-TABase:CancelRequestedByManagerType"
                 minOccurs="0"/>
        <element name="CloseOutNarr" type="tML-TABase:CloseOutNarrType" minOccurs="0"/>
        <element name="CloseOutVerification" type="tML-TABase:CloseOutVerificationType"
                 minOccurs="0"/>
        <element name="CommitmentTime" type="tML-TABase:CommitmentTimeType" minOccurs="0"/>
        <element name="CustomerTroubleTickNum" type="tML-TABase:CustTroubleTickNumType"
                 minOccurs="0"/>
        <element name="PerceivedTroubleSeverity" type="tML-TABase:PerceivedTroubleSeverityType"
                 minOccurs="0"/>
        <element name="RestoredTime" type="tML-TABase:RestoredTimeType" minOccurs="0"/>
        <element name="TroubleClearancePerson" type="tML-TABase:PersonReachType"
                 minOccurs="0"/>
        <element name="TroubleReportNumberList" type="tML-TABase:TroubleReportNumberListType"
                 minOccurs="0"/>
        <element name="TroubleType" type="tML-TABase:TroubleTypeType" minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="TroubleHistoryInfoListType">
    <sequence>
        <element name="TroubleHistoryInfoItem" type="tML-TABase:TroubleHistoryInfoType"
                 minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="TroubleLocationType">
    <choice>
        <element name="LocationAddress" type="tML-TABase:LocationAddressType"/>
        <element name="LocationPtr" type="tML-TABase:ObjectInstanceType"/>
    </choice>
</complexType>
<complexType name="TroubleReportIdListType">
    <sequence>
        <element name="TroubleReportIdItem" type="tML-TABase:NamingStringType"
                 minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="TroubleReportNumberListType">
    <sequence>
        <element name="TroubleReportNumberItem" type="tML-TABase:GraphicString64Type"
                 minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<simpleType name="TroubleReportStateType">
    <restriction base="integer">
        <enumeration value="0"/>
        <enumeration value="1"/>
        <enumeration value="2"/>
        <enumeration value="3"/>
        <enumeration value="4"/>
        <enumeration value="5"/>
    </restriction>
</simpleType>
<simpleType name="TroubleReportStatusValueType">
    <restriction base="integer">
        <enumeration value="1"/>
        <enumeration value="2"/>
        <enumeration value="3"/>
        <enumeration value="4"/>
        <enumeration value="5"/>
```

```
            <enumeration value="6"/>
            <enumeration value="7"/>
            <enumeration value="8"/>
            <enumeration value="9"/>
            <enumeration value="10"/>
            <enumeration value="11"/>
            <enumeration value="12"/>
            <enumeration value="13"/>
            <enumeration value="14"/>
            <enumeration value="15"/>
            <enumeration value="16"/>
            <enumeration value="17"/>
            <enumeration value="18"/>
            <enumeration value="19"/>
            <enumeration value="20"/>
            <enumeration value="21"/>
            <enumeration value="22"/>
            <enumeration value="23"/>
            <enumeration value="24"/>
            <enumeration value="25"/>
            <enumeration value="26"/>
            <enumeration value="27"/>
            <enumeration value="28"/>
            <enumeration value="29"/>
            <enumeration value="30"/>
            <enumeration value="31"/>
            <enumeration value="32"/>
            <enumeration value="33"/>
            <enumeration value="34"/>
            <enumeration value="35"/>
        </restriction>
    </simpleType>
    <complexType name="TroubleReportStatusType">
        <choice>
            <element name="Number" type="tML-TABase:TroubleReportStatusValueType"/>
            <element name="Identifier" type="tML-TABase:ObjectIdentifierType"/>
        </choice>
    </complexType>
    <complexType name="TroubleReportStatusListType">
        <sequence>
            <element name="TroubleReportStatusItem" type="tML-TABase:TroubleReportStatusType"
                     minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <simpleType name="TroubleReportStatusTimeType">
        <restriction base="tML-TABase:GeneralizedTimeType"/>
    </simpleType>
    <complexType name="TroubleReportStatusWindowType">
        <complexContent>
            <extension base="tML-TABase:TimeIntervalType"/>
        </complexContent>
    </complexType>
    <simpleType name="TroubleTypeValueType">
        <restriction base="integer">
      <!- Integer values are defined in ATIS-Standard-.-->

        </restriction>
    </simpleType>
    <complexType name="TroubleTypeType">
        <choice>
            <element name="Number" type="tML-TABase:TroubleTypeValueType"/>
            <element name="Identifier" type="tML-TABase:ObjectIdentifierType"/>
        </choice>
    </complexType>
    <simpleType name="TspPriorityType">
```

```
        <restriction base="tML-TABase:GraphicStringType">
            <length value="2"/>
            <pattern value="[E0-5][0-5]"/>
        </restriction>
    </simpleType>
    <complexType name="DaysOfWeekType">
        <sequence>
            <element name="Sunday" type="boolean" default="true"/>
            <element name="Monday" type="boolean" default="true"/>
            <element name="Tuesday" type="boolean" default="true"/>
            <element name="Wednesday" type="boolean" default="true"/>
            <element name="Thursday" type="boolean" default="true"/>
            <element name="Friday" type="boolean" default="true"/>
            <element name="Saturday" type="boolean" default="true"/>
        </sequence>
    </complexType>
    <complexType name="IntervalsOfDayType">
        <sequence>
            <element name="IntervalsOfDayItem" minOccurs="0" maxOccurs="unbounded">
                <complexType>
                    <sequence>
                        <element name="IntervalStart" type="tML-TABase:Time24Type" default="00:00:00"/>
                        <element name="IntervalEnd" type="tML-TABase:Time24Type" default="23:59:00"/>
                    </sequence>
                </complexType>
            </element>
        </sequence>
    </complexType>
    <complexType name="WeekMaskType">
        <sequence>
            <element name="DaysOfWeek" type="tML-TABase:DaysOfWeekType"/>
            <element name="IntervalsOfDay" type="tML-TABase:IntervalsOfDayType"/>
        </sequence>
    </complexType>
    <simpleType name="UserIdType">
        <restriction base="tML-TABase:GraphicString64Type"/>
    </simpleType>
```

4.3.2 Parameter Types Used in Request and Response

```
  <simpleType name=”NameBindingIdtype”>
    <restriction base=”string”>
      <enumeration value=”1.2.840.10015.0.6.24”/>
      <enumeration value=”1.2.840.10015.0.6.25”/>
      <enumeration value=”1.2.840.10015.0.6.26”/>
      <enumeration value=”1.2.840.10015.0.6.27”/>
    </restriction>
  </simpleType>

    <complexType name="TRNameBindingIdType">
        <sequence>
            <element name="NameBindingId" type="tML-TABase:NameBindingIdType"/>

        </sequence>
    </complexType>
    <complexType name="AdditionalCreateRequestType">
        <sequence>
            <element name="TroubleReportFormatId" type="tML-TABase:TRFormatIDType" minOccurs="0"/>
            <element name="AdditionalTroubleInfoList"
                     type="tML-TABase:AdditionalTroubleInfoListType" minOccurs="0"/>
            <element name="ALocationAccessAddress" type="tML-TABase:LocationAddressType"
                     minOccurs="0"/>
            <element name="ALocationAccessHours" type="tML-TABase:LocationAccessHoursType"
                     minOccurs="0"/>
```

```
        <element name="ALocationAccessPerson" type="tML-TABase:PersonReachType" minOccurs="0"/>
        <element name="AlternateManagerContactPerson" type="tML-TABase:PersonReachType"
                 minOccurs="0"/>
        <element name="AuthorizationList" type="tML-TABase:AuthorizationListType"
                 minOccurs="0"/>
        <element name="CallBackInfoList" type="tML-TABase:CallBackInfoListType" minOccurs="0"/>
        <element name="CalledNumber" type="tML-TABase:CalledNumberType" minOccurs="0"/>
        <element name="CommitmentTimeRequest" type="tML-TABase:CommitmentTimeType"
                 minOccurs="0"/>
        <element name="CustomerInfo" type="tML-TABase:CustomerInfoType" minOccurs="0"/>
        <element name="CustomerTroubleTickNum" type="tML-TABase:CustTroubleTickNumType"
                 minOccurs="0"/>
        <element name="CustomerWorkCenter" type="tML-TABase:CustomerWorkCenterType"
                 minOccurs="0"/>
        <element name="Dialog" type="tML-TABase:DialogType" minOccurs="0"/>
        <element name="EscalationList" type="tML-TABase:EscalationListType" minOccurs="0"/>
        <element name="ManagedObjectAccessFromTime"
                 type="tML-TABase:ManagedObjectAccessFromTimeType" minOccurs="0"/>
        <element name="ManagedObjectAccessHours" type="tML-TABase:ManagedObjectAccessHoursType"
                 minOccurs="0"/>
        <element name="ManagedObjectAccessToTime"
                 type="tML-TABase:ManagedObjectAccessToTimeType" minOccurs="0"/>
        <element name="ManagedObjectInstanceAliasList"
                 type="tML-TABase:ManagedObjectInstanceAliasListType" minOccurs="0"/>
        <element name="ManagerContactPerson" type="tML-TABase:PersonReachType" minOccurs="0"/>
        <element name="ManagerSearchKey" type="tML-TABase:ManagerSearchKeyCombineType"
                 minOccurs="0"/>
        <element name="PerceivedTroubleSeverity" type="tML-TABase:PerceivedTroubleSeverityType"
                 minOccurs="0"/>
        <element name="PreferredPriority" type="tML-TABase:PreferredPriorityType"
                 minOccurs="0"/>
        <element name="RepeatReport" type="tML-TABase:RepeatReportType" minOccurs="0"/>
        <element name="SuspectObjectList" type="tML-TABase:SuspectObjectListType"
                 minOccurs="0"/>
        <element name="TroubleDetectionTime" type="tML-TABase:TroubleDetectionTimeType"
                 minOccurs="0"/>
        <element name="TroubleReportStatusWindow"
                 type="tML-TABase:TroubleReportStatusWindowType" minOccurs="0"/>
        <element name="TspPriority" type="tML-TABase:TspPriorityType" minOccurs="0"/>
        <element name="ZLocationAccessAddress" type="tML-TABase:LocationAddressType"
                 minOccurs="0"/>
        <element name="ZLocationAccessHours" type="tML-TABase:LocationAccessHoursType"
                 minOccurs="0"/>
        <element name="ZLocationAccessPerson" type="tML-TABase:PersonReachType" minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="ModifyRequestType">
    <sequence>
        <element name="AdditionalTroubleInfoList"
                 type="tML-TABase:AdditionalTroubleInfoListType" minOccurs="0"/>
        <element name="ALocationAccessAddress" type="tML-TABase:LocationAddressType"
                 minOccurs="0"/>
        <element name="ALocationAccessHours" type="tML-TABase:LocationAccessHoursType"
                 minOccurs="0"/>
        <element name="ALocationAccessPerson" type="tML-TABase:PersonReachType" minOccurs="0"/>
        <element name="AlternateManagerContactPerson" type="tML-TABase:PersonReachType"
                 minOccurs="0"/>
        <element name="AuthorizationList" type="tML-TABase:AuthorizationListType"
                 minOccurs="0"/>
        <element name="CallBackInfoList" type="tML-TABase:CallBackInfoListType" minOccurs="0"/>
        <element name="CommitmentTimeRequest" type="tML-TABase:CommitmentTimeType"
                 minOccurs="0"/>
        <element name="Dialog" type="tML-TABase:DialogType" minOccurs="0"/>
        <element name="EscalationList" type="tML-TABase:EscalationListType" minOccurs="0"/>
        <element name="ManagedObjectAccessFromTime"
```

```
                type="tML-TABase:ManagedObjectAccessFromTimeType" minOccurs="0"/>
            <element name="ManagedObjectAccessHours" type="tML-TABase:ManagedObjectAccessHoursType"
                minOccurs="0"/>
            <element name="ManagedObjectAccessToTime"
                type="tML-TABase:ManagedObjectAccessToTimeType" minOccurs="0"/>
            <element name="ManagedObjectInstanceAliasList"
                type="tML-TABase:ManagedObjectInstanceAliasListType" minOccurs="0"/>
            <element name="ManagerContactPerson" type="tML-TABase:PersonReachType" minOccurs="0"/>
            <element name="ManagerSearchKey" type="tML-TABase:ManagerSearchKeyCombineType"
                minOccurs="0"/>
            <element name="PerceivedTroubleSeverity" type="tML-TABase:PerceivedTroubleSeverityType"
                minOccurs="0"/>
            <element name="PreferredPriority" type="tML-TABase:PreferredPriorityType"
                minOccurs="0"/>
            <element name="RepeatReport" type="tML-TABase:RepeatReportType" minOccurs="0"/>
            <element name="TroubleDetectionTime" type="tML-TABase:TroubleDetectionTimeType"
                minOccurs="0"/>
            <element name="TroubleReportStatusWindow"
                type="tML-TABase:TroubleReportStatusWindowType" minOccurs="0"/>
            <element name="ZLocationAccessAddress" type="tML-TABase:LocationAddressType"
                minOccurs="0"/>
            <element name="ZLocationAccessHours" type="tML-TABase:LocationAccessHoursType"
                minOccurs="0"/>
            <element name="ZLocationAccessPerson" type="tML-TABase:PersonReachType" minOccurs="0"/>
        </sequence>
    </complexType>
    <complexType name="CreateType">
        <sequence>
            <element name="InitiatingMode" type="tML-TABase:InitiatingModeType" minOccurs="0"/>
            <element name="ReceivedTime" type="tML-TABase:ReceivedTimeType"/>
            <element name="TroubleReportId" type="tML-TABase:NamingStringType"/>
            <element name="TroubleReportState" type="tML-TABase:TroubleReportStateType"/>
            <element name="TroubleReportStatus" type="tML-TABase:TroubleReportStatusType"/>
            <element name="TroubleReportStatusTime" type="tML-TABase:TroubleReportStatusTimeType"/>
            <element name="AgentContactPerson" type="tML-TABase:PersonReachType" minOccurs="0"/>
            <element name="CommitmentTime" type="tML-TABase:CommitmentTimeType" minOccurs="0"/>
            <element name="TspPriority" type="tML-TABase:TspPriorityType" minOccurs="0"/>
        </sequence>
    </complexType>
    <complexType name="CreateResponseType">
        <choice>
            <element name="Normal" type="tML-TABase:CreateType"/>
            <element name="Exception" type="tML-TABase:ExceptionCreateType"/>
        </choice>
    </complexType>
    <complexType name="StatusType">
        <sequence>
            <element name="StatusItem" minOccurs="0" maxOccurs="unbounded">
                <complexType>
                    <sequence>
                        <element name="TroubleReportId" type="tML-TABase:NamingStringType"/>
                        <element name="TroubleReportState" type="tML-TABase:TroubleReportStateType"/>
                        <element name="TroubleReportStatus" type="tML-TABase:TroubleReportStatusType"/>
                        <element name="TroubleReportStatusTime"
                                type="tML-TABase:TroubleReportStatusTimeType"/>
                    </sequence>
                </complexType>
            </element>
        </sequence>
    </complexType>
    <complexType name="TroubleStateStatusType">
        <sequence>
            <element name="TroubleReportState" type="tML-TABase:TroubleReportStateType"/>
            <element name="TroubleReportStatus" type="tML-TABase:TroubleReportStatusType"/>
            <element name="TroubleReportStatusTime"
```

```
                type="tML-TABase:TroubleReportStatusTimeType"/>
    </sequence>
</complexType>
<complexType name="TroubleReportFormatType">
    <sequence>
        <element name="TRFormatID" type="tML-TABase:TRFormatIDType"/>
        <element name="TRConstraintedToSingleValueAttrIDList"
                type="tML-TABase:AttributeIDListType" minOccurs="0"/>
        <element name="TRMayBePresentAttrIDList" type="tML-TABase:AttributeIDListType"
                minOccurs="0"/>
        <element name="TRMustBePresentAttrIDList" type="tML-TABase:AttributeIDListType"
                minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="FormatType">
    <sequence>
        <element name="FormatItem" type="tML-TABase:TroubleReportFormatType"
                minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="RetrievedAttributesType">
    <sequence>
        <element name="ActivityDurationList" type="tML-TABase:ActivityDurationType"
                minOccurs="0"/>
        <element name="AdditionalTroubleInfoList"
                type="tML-TABase:AdditionalTroubleInfoListType" minOccurs="0"/>
        <element name="AdditionalTroubleStatusInfo"
                type="tML-TABase:AdditionalTroubleStatusInfoType" minOccurs="0"/>
        <element name="AgentContactPerson" type="tML-TABase:PersonReachType" minOccurs="0"/>
        <element name="AfterHrsRepairAuth" type="tML-TABase:AfterHrsRepairAuthType"
                minOccurs="0"/>
        <element name="ALocationAccessHours" type="tML-TABase:LocationAccessHoursType"
                minOccurs="0"/>
        <element name="ALocationAccessAddress" type="tML-TABase:LocationAddressType"
                minOccurs="0"/>
        <element name="ALocationAccessPerson" type="tML-TABase:PersonReachType" minOccurs="0"/>
        <element name="AlternateCustomerContactPerson" type="tML-TABase:PersonReachType"
                minOccurs="0"/>
        <element name="AuthorizationList" type="tML-TABase:AuthorizationListType"
                minOccurs="0"/>
        <element name="CallBackInfoList" type="tML-TABase:CallBackInfoListType" minOccurs="0"/>
        <element name="CalledNumber" type="tML-TABase:CalledNumberType" minOccurs="0"/>
        <element name="CancelRequestedByManager" type="tML-TABase:CancelRequestedByManagerType"
                minOccurs="0"/>
        <element name="CloseOutNarr" type="tML-TABase:CloseOutNarrType" minOccurs="0"/>
        <element name="CloseOutVerification" type="tML-TABase:CloseOutVerificationType"
                minOccurs="0"/>
        <element name="CommitmentTime" type="tML-TABase:CommitmentTimeType" minOccurs="0"/>
        <element name="CommitmentTimeRequest" type="tML-TABase:CommitmentTimeType"
                minOccurs="0"/>
        <element name="CustomerContactPerson" type="tML-TABase:PersonReachType" minOccurs="0"/>
        <element name="CustomerInfo" type="tML-TABase:CustomerInfoType" minOccurs="0"/>
        <element name="CustomerTroubleTickNum" type="tML-TABase:CustTroubleTickNumType"
                minOccurs="0"/>
        <element name="CustomerWorkCenter" type="tML-TABase:CustomerWorkCenterType"
                minOccurs="0"/>
        <element name="Dialog" type="tML-TABase:DialogType" minOccurs="0"/>
        <element name="EscalationList" type="tML-TABase:EscalationListType" minOccurs="0"/>
        <element name="HandOffCenter" type="tML-TABase:HandOffCenterType" minOccurs="0"/>
        <element name="HandOffLocation" type="tML-TABase:HandOffLocationType" minOccurs="0"/>
        <element name="HandOffPerson" type="tML-TABase:PersonReachType" minOccurs="0"/>
        <element name="HandOffTime" type="tML-TABase:HandOffTimeType" minOccurs="0"/>
        <element name="InitiatingMode" type="tML-TABase:InitiatingModeType" minOccurs="0"/>
        <element name="LastUpdateTime" type="tML-TABase:GeneralizedTimeType" minOccurs="0"/>
        <element name="MaintenanceOrgContactPerson" type="tML-TABase:PersonReachType"
```

```
                minOccurs="0"/>
    <element name="MaintenanceOrgContactTime"
                type="tML-TABase:MaintenanceOrgContactTimeType" minOccurs="0"/>
    <element name="MaintServiceCharge"
                type="tML-TABase:MaintServiceChargeType" minOccurs="0"/>
    <element name="ManagedObjectAccessFromTime"
                type="tML-TABase:ManagedObjectAccessFromTimeType" minOccurs="0"/>
    <element name="ManagedObjectAccessHours" type="tML-TABase:ManagedObjectAccessHoursType"
                minOccurs="0"/>
    <element name="ManagedObjectAccessToTime"
                type="tML-TABase:ManagedObjectAccessToTimeType" minOccurs="0"/>
    <element name="ManagedObjectInstance" type="tML-TABase:ManagedObjectInstanceType"
                minOccurs="0"/>
    <element name="ManagedObjectInstanceAliasList"
                type="tML-TABase:ManagedObjectInstanceAliasListType" minOccurs="0"/>
    <element name="ManagerContactPerson" type="tML-TABase:PersonReachType" minOccurs="0"/>
    <element name="ManagerSearchKey1" type="tML-TABase:ManagerSearchKeyType"
                minOccurs="0"/>
    <element name="ManagerSearchKey2" type="tML-TABase:ManagerSearchKeyType"
                minOccurs="0"/>
    <element name="ManagerSearchKey3" type="tML-TABase:ManagerSearchKeyType"
                minOccurs="0"/>
    <element name="ManagerSearchKeyList" type="tML-TABase:ManagerSearchKeyListType"
                minOccurs="0"/>
    <element name="OutageDuration" type="tML-TABase:OutageDurationType" minOccurs="0"/>
    <element name="PerceivedTroubleSeverity" type="tML-TABase:PerceivedTroubleSeverityType"
                minOccurs="0"/>
    <element name="PreferredPriority" type="tML-TABase:PreferredPriorityType"
                minOccurs="0"/>
    <element name="ReceivedTime" type="tML-TABase:GeneralizedTimeType" minOccurs="0"/>
    <element name="RelatedTroubleReportList" type="tML-TABase:RelatedTroubleReportListType"
                minOccurs="0"/>
    <element name="RepairActivityList" type="tML-TABase:RepairActivityListType"
                minOccurs="0"/>
    <element name="RepeatReport" type="tML-TABase:RepeatReportType" minOccurs="0"/>
    <element name="ResponsiblePerson" type="tML-TABase:PersonReachType" minOccurs="0"/>
    <element name="RestoredTime" type="tML-TABase:RestoredTimeType" minOccurs="0"/>
    <element name="SuspectObjectList" type="tML-TABase:SuspectObjectListType"
                minOccurs="0"/>
    <element name="TroubleClearancePerson" type="tML-TABase:PersonReachType"
                minOccurs="0"/>
    <element name="TroubleDetectionTime" type="tML-TABase:TroubleDetectionTimeType"
                minOccurs="0"/>
    <element name="TroubleFound" type="tML-TABase:TroubleFoundType" minOccurs="0"/>
    <element name="TroubleLocation" type="tML-TABase:LocationAddressType" minOccurs="0"/>
    <element name="TroubleReportId" type="tML-TABase:NamingStringType" minOccurs="0"/>
    <element name="TroubleReportFormatId" type="tML-TABase:TRFormatIDType" minOccurs="0"/>
    <element name="TroubleReportNumberList" type="tML-TABase:TroubleReportNumberListType"
                minOccurs="0"/>
    <element name="TroubleReportState" type="tML-TABase:TroubleReportStateType"
                minOccurs="0"/>
    <element name="TroubleReportStatus" type="tML-TABase:TroubleReportStatusType"
                minOccurs="0"/>
    <element name="TroubleReportStatusTime" type="tML-TABase:TroubleReportStatusTimeType"
                minOccurs="0"/>
    <element name="TroubleReportStatusWindow"
                type="tML-TABase:TroubleReportStatusWindowType" minOccurs="0"/>
    <element name="TroubleType" type="tML-TABase:TroubleTypeType" minOccurs="0"/>
    <element name="TspPriority" type="tML-TABase:TspPriorityType" minOccurs="0"/>
    <element name="ZLocationAccessAddress" type="tML-TABase:LocationAddressType"
                minOccurs="0"/>
    <element name="ZLocationAccessHours" type="tML-TABase:LocationAccessHoursType"
                minOccurs="0"/>
    <element name="ZLocationAccessPerson" type="tML-TABase:PersonReachType" minOccurs="0"/>
</sequence>
```

```
</complexType>
<complexType name="RetrieveAttributesListType">
    <sequence>
        <element name="RetrieveAttributesItem" type="tML-TABase:RetrievedAttributesType"
                 minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="ModifiedAttributesType">
    <sequence>
        <element name="AdditionalTroubleInfoList" type="boolean" minOccurs="0"/>
        <element name="ALocationAccessAddress" type="boolean" minOccurs="0"/>
        <element name="ALocationAccessHours" type="boolean" minOccurs="0"/>
        <element name="ALocationAccessPerson" type="boolean" minOccurs="0"/>
        <element name="AlternateCustomerContactPerson" type="boolean" minOccurs="0"/>
        <element name="AuthorizationList" type="boolean" minOccurs="0"/>
        <element name="CallBackInfoList" type="boolean" minOccurs="0"/>
        <element name="CommitmentTimeRequest" type="boolean" minOccurs="0"/>
        <element name="CustomerContactPerson" type="boolean" minOccurs="0"/>
        <element name="Dialog" type="boolean" minOccurs="0"/>
        <element name="EscalationList" type="boolean" minOccurs="0"/>
        <element name="ManagedObjectAccessFromTime" type="boolean" minOccurs="0"/>
        <element name="ManagedObjectAccessHours" type="boolean" minOccurs="0"/>
        <element name="ManagedObjectAccessToTime" type="boolean" minOccurs="0"/>
        <element name="PerceivedTroubleSeverity" type="boolean" minOccurs="0"/>
        <element name="PreferredPriority" type="boolean" minOccurs="0"/>
        <element name="RepeatReport" type="boolean" minOccurs="0"/>
        <element name="TroubleDetectionTime" type="boolean" minOccurs="0"/>
        <element name="TroubleReportStatusWindow" type="boolean" minOccurs="0"/>
        <element name="ZLocationAccessAddress" type="boolean" minOccurs="0"/>
        <element name="ZLocationAccessHours" type="boolean" minOccurs="0"/>
        <element name="ZLocationAccessPerson" type="boolean" minOccurs="0"/>
    </sequence>
</complexType>
```

4.3.3 Parameter Types Used in Notification

```
<complexType name="ChangeEventType">
    <sequence>
        <element name="TroubleReportId" type="tML-TABase:NamingStringType" minOccurs="0"/>
        <element name="AgentContactPerson" type="tML-TABase:PersonReachType" minOccurs="0"/>
        <element name="ActivityDurationList" type="tML-TABase:ActivityDurationType"
                 minOccurs="0"/>
        <element name="AdditionalTroubleStatusInfo"
                 type="tML-TABase:AdditionalTroubleStatusInfoType" minOccurs="0"/>
        <element name="AuthorizationList" type="tML-TABase:AuthorizationListType"
                 minOccurs="0"/>
        <element name="CloseOutNarr" type="tML-TABase:CloseOutNarrType" minOccurs="0"/>
        <element name="CommitmentTime" type="tML-TABase:CommitmentTimeType" minOccurs="0"/>
        <element name="Dialog" type="tML-TABase:DialogType" minOccurs="0"/>
        <element name="EscalationList" type="tML-TABase:EscalationListType" minOccurs="0"/>
        <element name="HandOffCenter" type="tML-TABase:HandOffCenterType" minOccurs="0"/>
        <element name="HandOffLocation" type="tML-TABase:HandOffLocationType" minOccurs="0"/>
        <element name="HandOffPersonName" type="tML-TABase:PersonReachType" minOccurs="0"/>
        <element name="HandOffTime" type="tML-TABase:HandOffTimeType" minOccurs="0"/>
        <element name="LastUpdateTime" type="tML-TABase:GeneralizedTimeType" minOccurs="0"/>
        <element name="ManagedObjectInstanceAliasList"
                 type="tML-TABase:ManagedObjectInstanceAliasListType" minOccurs="0"/>
        <element name="MaintenanceOrgContactPerson" type="tML-TABase:PersonReachType"
                 minOccurs="0"/>
        <element name="MaintenanceOrgContactTime"
                 type="tML-TABase:MaintenanceOrgContactTimeType" minOccurs="0"/>
        <element name="MaintServiceCharge" type="tML-TABase:MaintServiceChargeType"
                 minOccurs="0"/>
        <element name="OutageDuration" type="tML-TABase:OutageDurationType" minOccurs="0"/>
```

```
        <element name="RelatedTroubleReportList" type="tML-TABase:RelatedTroubleReportListType"
                 minOccurs="0"/>
        <element name="RepairActivityList" type="tML-TABase:RepairActivityListType"
                 minOccurs="0"/>
        <element name="RepeatReport" type="tML-TABase:RepeatReportType" minOccurs="0"/>
        <element name="ResponsiblePerson" type="tML-TABase:PersonReachType" minOccurs="0"/>
        <element name="RestoredTime" type="tML-TABase:RestoredTimeType" minOccurs="0"/>
        <element name="TroubleFound" type="tML-TABase:TroubleFoundType" minOccurs="0"/>
        <element name="TroubleLocation" type="tML-TABase:LocationAddressType" minOccurs="0"/>
        <element name="TroubleReportNumberList" type="tML-TABase:TroubleReportNumberListType"
                 minOccurs="0"/>
        <element name="TroubleReportState" type="tML-TABase:TroubleReportStateType"
                 minOccurs="0"/>
        <element name="TroubleReportStatus" type="tML-TABase:TroubleReportStatusType"
                 minOccurs="0"/>
        <element name="TroubleReportStatusTime" type="tML-TABase:TroubleReportStatusTimeType"
                 minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="ChangeEventListType">
    <sequence>
        <element name="ChangeEventItem" type="tML-TABase:ChangeEventType"
                 minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="StatusOrCommitmentTimeUpdateType">
    <choice>
        <element name="TroubleStateStatus" type="tML-TABase:TroubleStateStatusType"/>
        <element name="CommitmentTime" type="tML-TABase:CommitmentTimeType"/>
    </choice>
</complexType>
```

4.3.4 Exception Data Types

```
<simpleType name="CanNotCloseType">
    <restriction base="integer">
        <enumeration value="0"/>
    </restriction>
</simpleType>
<complexType name="CannotVerifyOrDeniedAtThisTimeType" final="#all"/>
<simpleType name="FallBackReportingType">
    <union memberTypes="tML-TABase:NullStringType tML-TABase:GraphicStringType"/>
</simpleType>
<complexType name="TroubleReportChangeDeniedType">
    <sequence>
        <element name="ChangeDeniedReason" type="tML-TABase:ChangeDeniedReasonType"/>
        <element name="ManagedObjectClass" type="tML-TABase:ObjectClassType" minOccurs="0"/>
        <element name="ManagedObjectInstance" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="TroubleReportId" type="tML-TABase:NamingStringType"/>
    </sequence>
</complexType>
<complexType name="TroubleReportAlreadyExistsType">
    <sequence>
        <element name="ManagedObjectClass" type="tML-TABase:ObjectClassType" minOccurs="0"/>
        <element name="ManagedObjectInstance" type="tML-TABase:ManagedObjectInstanceType"/>
        <element name="TroubleReportId" type="tML-TABase:NamingStringType"/>
    </sequence>
</complexType>
<complexType name="TRMustBePresentAttributeMissingType">
    <complexContent>
        <extension base="tML-TABase:AttributeIDListType"/>
    </complexContent>
</complexType>
<complexType name="CommonExceptionType">
```

```
    <sequence>
        <element name="ExceptionList" type="tML-TABase:GenericDataType"
                 minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="DuplicatedInvocationType">
    <complexContent>
        <extension base="tML-TABase:CommonExceptionType"/>
    </complexContent>
</complexType>
<complexType name="InvalidDataReceivedType">
    <complexContent>
        <extension base="tML-TABase:CommonExceptionType"/>
    </complexContent>
</complexType>
<complexType name="AccessDeniedType">
    <complexContent>
        <extension base="tML-TABase:CommonExceptionType"/>
    </complexContent>
</complexType>
<complexType name="ResourceLimitationType">
    <complexContent>
        <extension base="tML-TABase:CommonExceptionType"/>
    </complexContent>
</complexType>
<complexType name="ProcessingFailureErrorType">
    <complexContent>
        <extension base="tML-TABase:CommonExceptionType"/>
    </complexContent>
</complexType>
<complexType name="NotFoundType">
    <complexContent>
        <extension base="tML-TABase:CommonExceptionType"/>
    </complexContent>
</complexType>
<complexType name="MissingDataType">
    <complexContent>
        <extension base="tML-TABase:CommonExceptionType"/>
    </complexContent>
</complexType>
<complexType name="NoSuchArgumentsType">
    <complexContent>
        <extension base="tML-TABase:CommonExceptionType"/>
    </complexContent>
</complexType>
<complexType name="InvalidArgumentValuesType">
    <complexContent>
        <extension base="tML-TABase:CommonExceptionType"/>
    </complexContent>
</complexType>
<complexType name="NoSuchEventTypeType">
    <complexContent>
        <extension base="tML-TABase:CommonExceptionType"/>
    </complexContent>
</complexType>
<complexType name="InvalidFunctionReceivedType">
    <complexContent>
        <extension base="tML-TABase:CommonExceptionType"/>
    </complexContent>
</complexType>
<complexType name="ExceptionGenericType">
    <sequence>
        <element name="AccessDenied" type="tML-TABase:AccessDeniedType" minOccurs="0"/>
        <element name="InvalidDataReceived" type="tML-TABase:InvalidDataReceivedType"
                 minOccurs="0"/>
```

```
        <element name="ProcessingFailure" type="tML-TABase:ProcessingFailureErrorType"
                 minOccurs="0"/>
        <element name="ResourceLimitation" type="tML-TABase:ResourceLimitationType"
                 minOccurs="0"/>
        <element name="DuplicatedInvocation" type="tML-TABase:DuplicatedInvocationType"
                 minOccurs="0"/>
        <element name="InvalidFunctionReceived" type="tML-TABase:InvalidFunctionReceivedType"
                 minOccurs="0"/>
    </sequence>
</complexType>
<complexType name="ExceptionCreateType">
    <complexContent>
        <extension base="tML-TABase:ExceptionGenericType">
            <sequence>
                <element name="FallBackReporting" type="tML-TABase:FallBackReportingType"
                         minOccurs="0"/>
                <element name="TroubleReportAlreadyExists"
                         type="tML-TABase:TroubleReportAlreadyExistsType" minOccurs="0"/>
                <element name="TRMustBePresentAttributeMissing"
                         type="tML-TABase:TRMustBePresentAttributeMissingType" minOccurs="0"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="ExceptionTroubleReportType">
    <complexContent>
        <extension base="tML-TABase:ExceptionGenericType">
            <sequence>
                <element name="NotFound" type="tML-TABase:NotFoundType" minOccurs="0"/>
                <element name="MissingData" type="tML-TABase:MissingDataType" minOccurs="0"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="ExceptionTroubleReportUpdateType">
    <complexContent>
        <extension base="tML-TABase:ExceptionTroubleReportType">
            <sequence>
                <element name="TroubleReportChangeDenied"
                         type="tML-TABase:TroubleReportChangeDeniedType" minOccurs="0"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="ExceptionVerifyType">
    <complexContent>
        <extension base="tML-TABase:ExceptionTroubleReportType">
            <sequence>
                <element name="CannotVerifyOrDeniedAtThisTime"
                         type="tML-TABase:CannotVerifyOrDeniedAtThisTimeType" minOccurs="0"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="ExceptionCancelType">
    <complexContent>
        <extension base="tML-TABase:ExceptionTroubleReportType">
            <sequence>
                <element name="TroubleReportChangeDenied"
                         type="tML-TABase:TroubleReportChangeDeniedType" minOccurs="0"/>
                <element name="CanNotClose" type="tML-TABase:CanNotCloseType" minOccurs="0"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

```
<complexType name="ExceptionEventType">
    <sequence>
        <element name="DuplicatedInvocation" type="tML-TABase:DuplicatedInvocationType"
                 minOccurs="0"/>
        <element name="InvalidArgumentValues" type="tML-TABase:InvalidArgumentValuesType"
                 minOccurs="0"/>
        <element name="NoSuchArguments" type="tML-TABase:NoSuchArgumentsType" minOccurs="0"/>
        <element name="NoSuchEventType" type="tML-TABase:NoSuchEventTypeType" minOccurs="0"/>
        <element name="ResourceLimitation" type="tML-TABase:ResourceLimitationType"
                 minOccurs="0"/>
        <element name="NotFound" type="tML-TABase:NotFoundType" minOccurs="0"/>
    </sequence>
</complexType>
```

Annex A: CMIP (ASN.1) Model & Accompanying Documentation

(informative)

Note – The text in Annexes A through H was formerly contained in ATIS-0300227, which has since been withdrawn, as CMIP is generally no longer used for Trouble Management.

A.1 Overview of the Service Model

A trouble report can be generated for any GNIM-managed object in ATIS-0300240. In addition, this clause provides the rationale on why some administrations may require support for the CNM Service object class for the trouble administration interface.

Customers manage their services, not the components that the provider uses to provide the service. A customer's request to test a service should produce a result in a form useful and meaningful to a customer, while hiding that information about how the service is provided that may be proprietary to the service provider. Similarly, reconfiguration requests or trouble reports should use a syntax and semantics that are friendly to the customer. The service should be managed in terms of how the service is perceived by the customer. Changes in the provider's underlying configuration that do not affect the customer's service should usually be hidden from the customer. Nor should a customer's Network Management systems or personnel need to interact with the details of the provider's system if they can be hidden by a more coherent customer-oriented service view.

Although services are relatively independent of network elements providing the service, they aren't completely independent. Clearly, a customer will know of certain network elements and manage them directly (features of analog Lines, for example). Also, it would be presumptuous of the service provider to assume complete ignorance of their architecture by their sophisticated customers. A model of customer network management needs to preserve this information where appropriate, if only to allow a common discourse between the customers and the service provider's telecommunications personnel who share a common training and language.

The service provider shall map transactions from the customer about services to management activities performed in terms of network elements. For example, a trouble report on a customer's service may map into several trouble reports on distinct network elements. A model of customer network management shall allow the definition of this mapping from the service oriented customer view to the network oriented internal view.

A service encapsulates common features that are not part of the network element world. These include such qualities as billing information and tariffs, customer contact information, Centrex Group relationships, customer-oriented naming and ownership. Additionally, services that are provided by concrete network elements are abstracted from the network elements themselves. Identical services can be provided by different types of elements and different services are provided over the same elements (often simultaneously). Regardless of how a service is provided to the customer, its definition should present a coherent view of the product purchased from the provider.

A service may be made up of discrete elements that themselves are services, for example, a Centrex Group or a Private Virtual Network. Features may be shared among the discrete components. These include billing, tariff information, common operational features, common reports, common engineering criteria, common alarm formats, and common trouble reporting. To avoid redundancy, this information needs to be stored in a common object whose scope subtends the separate components of the service.

Customer Network Management, in which a customer interacts with the provider of a service to manage that service may need a partitioning of the "world" far different from that required to manage a network by the "network's" owner. While the owner or operator of the network needs to view the components of the system and manage them as a complete entity providing many services to many customers, the user uses and manages the services provided by the network as services, not as a series of discrete components.

Services may be abstract concepts. The same network elements may define several different services for one or more customers. Or identical services could be provided by different sets of network elements at different times or for different customers. In either of these cases, the definition of a service shall describe the relationship between the service offered the customer and the actual network elements that provide the service. The description of the service also will map the behavior of the service and the attributes that describe how it may be managed to the actual elements of the service provider.

A service provided to a customer may itself contain other services. For example, a “Basic Business Group” (Centrex Group) is a service composed of several services (voice lines, for example) provided to a customer as a coherent package and managed as a single entity. The individual components of the business group may themselves be services and be separately manageable. The Basic Business Group contains these component services and would be responsible for the definition and management of qualities that are common to all members of the Centrex Group.

Features and attributes of services to a customer may be common across many instances of the service. These common features or qualities may carry down to the specific instances of the services. A Service Profile object could model these shared features.

A.1.1 Object Class & Attribute Type

This clause contains the definitions of the object classes and attribute types that form the basis for the extensions to the GNIM needed for interfaces between OSs across jurisdictional boundaries (see ATIS-0300240.1998(R2007)). The various object classes and attribute types are described in A.2–A.6, as follows:

Subclause	Object class/ Attribute type
A.2	Object classes
	Attribute types using the template notation described in ISO/IEC 10165-4.
	Specific error messages identified for the interface.
	Type definitions of the various attributes introduced in A.3 through the Abstract Syntax Notation-One (ASN.1) language.

The object classes and attribute types defined here should be used wherever possible. However, where the need is justified, additional object classes or attribute types may be required.

At the time of writing, no stand-alone Service Model exists within the standards, either regionally or internationally. Therefore, a preliminary view of three service objects (i.e., CNM Service, Account, and Service Profile) is provided to support customer trouble administration interfaces.

The subject of security is for further study, and there is a possibility that some objects must be defined for that purpose, probably using authentication, as detailed in ISO/IEC 8649. For notifications regarding security violations, Network Mangement Forum’s Forum Security Notifier object may be used.

Access permissions are necessary to provide mutual privacy among accounts. Access to an Account object shall be granted on the basis of user permissions specific to that object. Granting access to an Account object may grant access to objects subordinate to that object, or, permission may have to be specifically granted individually for each subordinate and superior object.

A.2 Object class definitions

The object classes currently defined in this standard apply only to the trouble report administration aspect of the Fault Management functional area. As message specifications for other functional areas are developed, the applicability of an object may expand beyond Fault Management.

For the convenience of the reader, Annex G gives a pictorial representation of the pointer attribute relationships between objects in this document. Annex G does not represent normative information.

The remainder of this clause contains the definition of the object classes specified in this standard. Each object definition consists of a text definition followed by a template specification, as specified in ISO/IEC 10165-4. Name bindings are supplied in Annex B.

A.2.1 Account

This object class contains information that describes a customer account that interacts with the carrier. Naming an account by another account allows a flexible hierarchical organization of the managed objects.

```
account MANAGED OBJECT CLASS
   DERIVED FROM “Rec. X.721|ISO/IEC 10165-2 :1992”:top;
   CHARACTERIZED BY accountPkg PACKAGE
      ATTRIBUTES
         accountName GET,
         additionalText GET-REPLACE;;;
   CONDITIONAL PACKAGES
      aAccountContactAttributeListPkg PACKAGE
      ATTRIBUTES
         accountContactList GET-REPLACE ADD-REMOVE;
   REGISTERED AS {trPackage 1};
         PRESENT IF “an instance supports it and accountContactObjectListPkg is not present.”,
   aAccountContactObjectListPkg PACKAGE
    BEHAVIOUR accountContactObjectListBehaviour BEHAVIOUR
          DEFINED AS  “The Account Contact Object List package contains one attribute
                       which points to instances of the Contact object that
                       represent individuals in the manager’s organization
                       who can be contacted about an account.";;
    ATTRIBUTES
     contactObjectPtrList GET-REPLACE ADD-REMOVE;
   REGISTERED AS {trPackage 2};
     PRESENT IF “an instance supports it and accountContactAttributeListPkg is not present.”;
REGISTERED AS {trMObjectClass 1};
```

A.2.2 CNM Service

The Customer Network Management (CNM) Service object class is a specialization of the Service object class in the GNIM. These refinements are necessary to support the service modeling concept for CNM OS–OS interfaces across jurisdictional boundaries.

The CNM Service object class represents the specific functionality that a provider supplies to customers. The Service ID attribute identifies the service independent of the Service Aliases (network-specific identifiers such as phone numbers or circuit IDs). The CNM Service object decouples the relationship between services offered to the customer and the specific network components that provide the services.

The cnmService object class has been updated, and a new object class, cnmServiceR0, is created here. The GDMO template is provided below. The original cnmService object class is still being preserved, and its template is provided in Annex H of this standard.

```
cnmServiceR0 MANAGED OBJECT CLASS
    DERIVED FROM "T1.240-1998":service;
    CHARACTERIZED BY cnmServiceR0Pkg PACKAGE
    ATTRIBUTES
        -- 2 mandatory attributes inherited from service:
        -- serviceID GET,
        -- serviceType GET-REPLACE,
        -- some implementations may not support a REPLACE on serviceType
        serviceLocationList GET,
        serviceDescription GET;;;
        -- 1 notification inherited from service:
        -- qualityOfServiceAlarm
        -- may need to be modified in this class
CONDITIONAL PACKAGES
            -- 7 optional attributes inherited from service:
            -- administrativeState GET-REPLACE,
            -- alarmState GET,
            -- currentProblemList GET,
            -- operationalState GET,
            -- supportedServiceNameList
            -- GET-REPLACE-ADD-REMOVE,
            -- supportedByObjectList
            -- GET-REPLACE-ADD-REMOVE,
            -- usageState GET,
    csServiceAliasPkg PRESENT IF "an instance supports it.",
    csServiceProfileObjectPtrPkg PRESENT IF "an instance supports it.",
    csTroubleReportFormatObjectPtrPkg PRESENT IF "an instance supports it.";
REGISTERED AS {trMObjectClass 11};
```

A.2.2.1 Packages for CNM Service Object Classes

This subclause defines the conditional packages that are used by the cnmServiceR0 (defined in 7.1.2, CNM Service) and cnmService (defined in Annex I) object classes. These packages were originally defined in-line in cnmService and now are defined separately.

csServiceAliasPkg PACKAGE

ATTRIBUTES

serviceAliasList GET;

-- Administrations may specify structure

-- of serviceAliasList GraphicString

REGISTERED AS {trPackage 3};

csServiceProfileObjectPtrPkg PACKAGE

ATTRIBUTES

serviceProfileObjectPtr GET;

REGISTERED AS {trPackage 4};

csTroubleReportFormatObjectPtrPkg PACKAGE

ATTRIBUTES

troubleReportFormatObjectPtr GET SET-BY-CREATE;

REGISTERED AS {trPackage 5};

A.2.3 Contact

This standard makes use of the Contact object defined in ITU-T Recommendation X.790. For the convenience of the reader, the details of the Contact object from this document are in Annex B.

A.2.4 Forum Security Notifier

This standard makes use of the Network Management Forum's Forum Security Notifier object defined in the "Forum Library - Volume 4: OMNIPoint 1 Definitions, Issue 1.0, August 1992". For the convenience of the reader, the details of the Forum's Forum Security Notifier object from this document are in Annex B. Although the object is considered stable in this document, the official Forum Security Notifier object, including its registration, is available from the NM Forum. OMNI*Point* 1 was issued in October 1992.

A.2.5 Telecommunications Trouble Report

The Telecommunications Trouble Report is inherited from the Trouble Report. The Telecommunications Trouble Report object represents reported troubles on telecommunications services or resources. Instances of this class describe the nature of the problem as well as ongoing status.

Local administrations may put restrictions on the number of open Telecommunications Trouble Reports per managed object through business agreements.

The Trouble Report Administration model allows multiple trouble report formats as defined by instances of the Trouble Report Format Definition object. Each trouble report format is a predefined combination of Telecommunications Trouble Report attributes. The trouble report format applicable to a particular CNM Service or GNIM-managed object instance can be dynamically specified by the service provider through the Trouble Report Format Definition object. When the trouble report format is explicitly defined through the Trouble Report Format Definition object for a CNM Service or GNIM-managed object, a Telecommunications Trouble Report instance for that CNM Service or GNIM-managed object shall consist of:

- The mandatory attributes of the Telecommunications Trouble Report object class as defined in the Telecommunications Trouble Report object class specification;
- The attributes in conditional packages of the Telecommunications Trouble Report that “must be present” as defined by the corresponding Trouble Report Format Definition object;
- Optionally the attributes in conditional packages of the Telecommunications Trouble Report that “may be present” as defined by the corresponding Trouble Report Format Definition object.

The appropriate instance of the Trouble Report Format Definition object is identified either:

a) By a pointer attribute (troubleReportFormatObjectPtr) in the CNM Service object (when the format must be defined on an object instance basis) or,

b) By inclusion of the managed object class in an applicableManagedObjectClassList attribute in the Trouble Report Format Definition object (when the format is the same for an entire object class) or,

c) By inclusion of an instance of a GNIM object class that represents a telecommunications resource in an applicableManagedObjectInstanceList attribute in the Trouble Report Format Definition object (when the format is specific to the object instance).

The manager is allowed to create Telecommunications Trouble Reports. As part of create, the manager is required to supply the following attributes:

- Managed Object Instance;
- Trouble Type;
- Additional Trouble Information List;

plus any manager-supplied attributes in conditional packages identified as “must be present”. The manager also has the option to include manager-supplied attributes in conditional packages identified as “may be present” attributes.

As part of the instantiation of a Telecommunications Trouble Report, the agent is also required to supply values for the remaining “CHARACTERIZED BY” attributes of the Telecommunications Trouble Report object class plus any agent-supplied attributes in conditional packages identified as “must be present”. The agent also has the option to include agent-supplied attributes in conditional packages identified as “may be present” attributes.

For each of the Packages of the Telecommunications Trouble Report, the following specifies whether the attribute can be supplied by the manager or agent and whether it can be later updated by the manager or agent.

- The following attributes can only be supplied by the manager:

  managedObjectInstance

  suspectObjectList

  troubleType

  calledNumber

  customerWorkCenter

  custTroubleTickNum

  troubleReportFormatObjectPtr

  tspPriority

- The following attributes can only be supplied by the manager and updated by the manager:

  aLocationAccessAddress

zLocationAccessAddress
aLocationAccessHours
zLocationAccessHours
aLocationAccessPerson
zLocationAccessPerson
additionalTroubleInfoList
alternateManagerContactPerson
alternateManagerContactObjectPtr
callBackInfoList
commitmentTimeRequest
managedObjectAccessHours
managedObjectAccessFromTime
managedObjectAccessToTime
managerContactPerson
managerContactObjectPtr
managerSearchKey1
managerSearchKey2
managerSearchKey3
managerSearchKeyList
perceivedTroubleSeverity
preferredPriority
troubleReportStatusWindow

- The following attributes can only be supplied by the agent:

  initiatingMode
  receivedTime
  troubleReportID

- The following attributes can only be supplied by the agent and updated by the agent:

  activityDuration *
  additionalTroubleStatusInfo
  agentContactPerson
  agentContactObjectPtr
  alarmRecordPtrList
  commitmentTime
  lastUpdateTime
  relatedTroubleReportList
  responsiblePersonName *
  responsiblePersonPtr *
  troubleLocation *

troubleReportNumberList

troubleReportState *

troubleReportStatus *

troubleReportStatusTime

* These attributes are required to be updatable by the manager in the service-provider-to-service-provider interface.

- The following attributes can be set to default by the agent and only updated by the agent:

  closeOutNarr *

  handOffCenter

  handOffLocation

  handOffPersonName *

  handOffPersonPtr *

  handOffTime

  maintenanceOrgContactName *

  maintenanceOrgContactPtr *

  maintenanceOrgContactTime *

  maintServiceCharge

  outageDuration

  repairActivityList *

  restoredTime *

  troubleFound *

* These attributes are required to be updatable by the manager in the service-provider-to-service-provider interface.

- The following attributes can be set to default by the agent and only updated by the manager:

  afterHrsRepairAuth

  cancelRequestedByManager

  closeOutVerification

  troubleClearancePerson

  troubleDetectionTime

- The following attribute can be supplied by the manager or set to default by the agent and updated by the manager or agent.

  authorizationList

- The following attributes can only be supplied by the manager and updated by the agent:

  managedObjectInstanceAliasList

- The following attributes can only be supplied by the manager and updated by the manager or agent:

  dialog

  escalationList

  repeatReport

The manager can create Telecommunications Trouble Report instances in the agent system. Telecommunications Trouble Reports may also be created locally by the agent on behalf of the manager. The Initiating Mode attribute may be used to indicate the source of the trouble report – manager or agent.

When instantiating a Telecommunications Trouble Report object, the manager may use the packages attribute to identify attributes defined for the object as part of the conditional packages, but for which the manager has no initial value to supply and for which there is no default value specified in the package definition.

Manager deletion of Telecommunications Trouble Reports is not supported on interjurisdictional interfaces. A manager may request that a Telecommunications Trouble Report be canceled, which may or may not result in the Telecommunications Trouble Report being closed-out immediately. Closed-out Telecommunications Trouble Reports are deleted locally by the agent according to some storage period criteria (e.g., 3, 12, or 18 months).

The Telecommunications Trouble Report object generates the object creation and object deletion notifications whenever the agent creates the Telecommunications Trouble Report object or deletes the Telecommunications Trouble Report object through local administrative procedures.

An Attribute Value Change Notification is emitted when there is a change in the value of a Telecommunications Trouble Report attribute. In some implementations, only changes in the Trouble Report Status or Commitment Time attributes are emitted.

The Telecommunications Trouble Report object generates a Trouble History Event Notification with Trouble History information whenever the Trouble Report Status attribute value changes to a closed-out value.

> NOTE – This notification is in addition to the attribute value change notification for the Trouble Report Status attribute. This notification is offered to a Log where the discriminator attribute of the log decides whether the notification will be logged. In some implementations the attributes that allow selective logging will be absent or not under the control of the interface. The concept of logs is introduced in ISO/IEC 10164-6.

```
telecommunicationsTroubleReport MANAGED OBJECT CLASS
    DERIVED FROM troubleReport;
    CONDITIONAL PACKAGES
        trAfterHrsRepairAuthPkg PACKAGE
            ATTRIBUTES
                afterHrsRepairAuth
                    DEFAULT VALUE GNMTA.afterHrsRepairAuthAfterHrsRepairAuthDefault
                    GET-REPLACE;
        REGISTERED AS {trPackage 6};
            PRESENT IF “an instance supports it and trAuthorizationListPkg is not present.”,
        trAlarmRecordPtrListPkg PACKAGE
            ATTRIBUTES
                alarmRecordPtrList GET;
                -- Necessary condition: Trouble Report
                -- shall have been generated as a result
                -- of an alarm.
                -- Not a sufficient condition: some
                -- instances may choose not to support
                -- even if the trouble report was
                -- generated as a result of an alarm
                -- received or generated in the agent.
```

REGISTERED AS {trPackage 7};

PRESENT IF “an instance supports it.”,

trAlternateManagerContactPersonAttributePkg PACKAGE

ATTRIBUTES

alternateManagerContactPerson GET-REPLACE;

REGISTERED AS {trPackage 8};

PRESENT IF “an instance supports it and trAlternateManagerContactPersonObjectPkg is not present.”,

trAlternateManagerContactPersonObjectPkg PACKAGE

BEHAVIOUR alternateManagerContactPersonObjectBehaviour BEHAVIOUR

DEFINED AS “The Alternate Manager Contact Person Object package contains one attribute which points to an instance of the Contact object which represents the alternative person to the manager contact who can be contacted regarding the reported trouble.”;;

ATTRIBUTES

alternateManagerContactObjectPtr GET-REPLACE;

REGISTERED AS {trPackage 9};

PRESENT IF “an instance supports it and trAlternateManagerContactPersonAttributePkg is not present.”,

trAuthorizationListPkg PACKAGE

ATTRIBUTES

authorizationList

DEFAULT VALUE GNMTA.authorizationAuthorizationDefault

GET-REPLACE ADD-REMOVE troubleReportChangeDenied;

REGISTERED AS {trPackage 10};

PRESENT IF “an instance supports it and trAfterHrsRepairAuthPkg is not present.”,

trCallBackInfoListPkg PACKAGE

ATTRIBUTES

callBackInfoList GET-REPLACE ADD-REMOVE;

REGISTERED AS {trPackage 11};

PRESENT IF “an instance supports it.”,

trCalledNumberPkg PACKAGE

ATTRIBUTES

calledNumber GET SET-BY-CREATE;

REGISTERED AS {trPackage 12};

PRESENT IF “an instance supports it.”,

```
trCancelRequestedByManagerPkg PACKAGE
   ATTRIBUTES
      cancelRequestedByManager
         INITIAL VALUE GNMTA.troubleReportCancelRequestedByManagerInitial
         GET-REPLACE troubleReportChangeDenied canNotClose;
REGISTERED AS {trPackage 13};
   PRESENT IF “an instance supports it.”,
trCloseOutVerificationPkg PACKAGE
   ATTRIBUTES
      closeOutVerification
            INITIAL VALUE GNMTA.closeOutVerificationCloseOutVerificationInitial
            GET-REPLACE cannotVerifyOrDenyAtThisTime;
REGISTERED AS {trPackage 14};
   PRESENT IF “an instance supports it.”,
trCommitmentTimePkg PACKAGE
   ATTRIBUTES
      commitmentTime GET;
      -- some implementations may not
      -- support a GET
REGISTERED AS {trPackage 15};
   PRESENT IF “an instance supports it.”,
trCommitmentTimeRequestPkg PACKAGE
   ATTRIBUTES
      commitmentTimeRequest GET-REPLACE troubleReportChangeDenied;
REGISTERED AS {trPackage 16};
   PRESENT IF “an instance supports it.”,
trCustomerWorkCenterPkg PACKAGE
   ATTRIBUTES
      customerWorkCenter GET SET-BY-CREATE;
REGISTERED AS {trPackage 17};
   PRESENT IF “an instance supports it.”,
trCustTroubleTickNumPkg PACKAGE
   ATTRIBUTES
      custTroubleTickNum GET SET-BY-CREATE;
REGISTERED AS {trPackage 18};
      PRESENT IF “an instance supports it.”,
trDialogPkg PACKAGE
   ATTRIBUTES
```

```
        dialog GET-REPLACE;
REGISTERED AS {trPackage 19};
    PRESENT IF “an instance supports it.”,
trEscalationListPkg PACKAGE
    ATTRIBUTES
        escalationList GET ADD troubleReportChangeDenied;
REGISTERED AS {trPackage 20};
        PRESENT IF “an instance supports it.”,
trHandOffCenterPkg PACKAGE
    ATTRIBUTES
        handOffCenter
                INITIAL VALUE GNMTA.handOffCenterHandOffCenterInitial
                GET;
REGISTERED AS {trPackage 21};
        PRESENT IF “an instance supports it.”,
trHandOffLocationPkg PACKAGE
    ATTRIBUTES
        handOffLocation
                INITIAL VALUE GNMTA.handOffLocationHandOffLocationInitial
                GET;
REGISTERED AS {trPackage 22};
        PRESENT IF “an instance supports it.”,
trHandOffPersonNamePkg PACKAGE
 BEHAVIOUR handOffPersonNameBehaviour BEHAVIOUR
        DEFINED AS  “Modifications to
                        handOffPersonName are required only in the
                        service provider to service provider interface.
                        The CMIS error ‘access denied’ may be issued
                        in response to attempts to modify this attribute
                        on any other interface.”;;
    ATTRIBUTES
        handOffPersonName
                INITIAL VALUE GNMTA.handOffPersonNameHandOffPersonNameInitial
                GET-REPLACE;
REGISTERED AS {trPackage 23};
        PRESENT IF “an instance supports it and trHandOffPersonPtrPkg is not present.”,
trHandOffPersonPtrPkg PACKAGE
 BEHAVIOUR handOffPersonPtrBehaviour BEHAVIOUR
```

```
        DEFINED AS “Modifications to
                handOffPersonPtr are required only in the
                service provider to service provider interface.
                The CMIS error ‘access denied’ may be issued
                in response to attempts to modify this attribute
                on any other interface.”;;
    ATTRIBUTES
        handOffPersonPtr GET-REPLACE;
REGISTERED AS {trPackage 24};
        PRESENT IF “an instance supports it and trHandOffPersonNamePkg is not present.”,
trHandOffTimePkg PACKAGE
    ATTRIBUTES
        handOffTime
            INITIAL VALUE GNMTA.handOffTimeHandOffTimeInitial
            GET;
REGISTERED AS {trPackage 25};
        PRESENT IF “an instance supports it.”,
trInitiatingModePkg PACKAGE
    ATTRIBUTES
        initiatingMode GET;
REGISTERED AS {trPackage 26};
        PRESENT IF “an instance supports it.”,
trLastUpdateTimePkg PACKAGE
    ATTRIBUTES
        lastUpdateTime GET;
REGISTERED AS {trPackage 27};
        PRESENT IF “an instance supports it.”,
trALocationPkg PACKAGE
    ATTRIBUTES
        aLocationAccessAddress GET-REPLACE;
REGISTERED AS {trPackage 28};
        PRESENT IF “an instance supports it.”,
trZLocationpkg PACKAGE
    ATTRIBUTES
        zLocationAccessAddress GET-REPLACE;
REGISTERED AS {trPackage 29};
        PRESENT IF “an instance supports it.”,
trALocationAccessHoursPkg PACKAGE
```

```
    ATTRIBUTES
        aLocationAccessHours GET-REPLACE ADD-REMOVE;
REGISTERED AS {trPackage 30};
        PRESENT IF “an instance supports it.”,
trZLocationAccessHoursPkg PACKAGE
    ATTRIBUTES
        zLocationAccessHours GET-REPLACE ADD-REMOVE;
REGISTERED AS {trPackage 31};
        PRESENT IF “an instance supports it.”,
trALocationAccessPersonPkg PACKAGE
    ATTRIBUTES
        aLocationAccessPerson GET-REPLACE;
REGISTERED AS {trPackage 32};
        PRESENT IF “an instance supports it.”,
trZLocationAccessPersonPkg PACKAGE
    ATTRIBUTES
        zLocationAccessPerson GET-REPLACE;
REGISTERED AS {trPackage 33};
        PRESENT IF “an instance supports it.”,
trMaintenanceOrgContactNamePkg PACKAGE
     BEHAVIOUR maintenanceOrgContactNameBehaviour BEHAVIOUR
        DEFINED AS “Modifications to
                    maintenanceOrgContactName are required only in the
                    service provider to service provider interface.
                    The CMIS error ‘access denied’ may be issued
                    in response to attempts to modify this attribute
                    on any other interface.”;;
    ATTRIBUTES
        maintenanceOrgContactName
            INITIAL VALUE GNMTA.maintenanceOrgContactNameMaintenanceOrgContactNameInitial
            GET-REPLACE;
REGISTERED AS {trPackage 34};
        PRESENT IF “an instance supports it and
                trMaintenanceOrgContactPtrPkg is not present.”,
trMaintenanceOrgContactPtrPkg PACKAGE
 BEHAVIOUR maintenanceOrgContactPtrBehaviour BEHAVIOUR
        DEFINED AS “Modifications to
                    maintenanceOrgContactPtr are required only in the
```

service provider to service provider interface.
The CMIS error ‘access denied’ may be issued
in response to attempts to modify this attribute
on any other interface.”;;

ATTRIBUTES
maintenanceOrgContactPtr GET-REPLACE;
REGISTERED AS {trPackage 35};

PRESENT IF “an instance supports it and
trMaintenanceOrgContactNamePkg is not present.”,
trMaintenanceOrgContactTimePkg PACKAGE
BEHAVIOUR maintenanceOrgContactTimeBehaviour BEHAVIOUR

DEFINED AS “Modifications to
maintenanceOrgContactTime are required only in the
service provider to service provider interface.
The CMIS error ‘access denied’ may be issued
in response to attempts to modify this attribute
on any other interface.”;;

ATTRIBUTES
maintenanceOrgContactTime
INITIAL VALUE GNMTA.maintenanceOrgContactTimeMaintenanceOrgContactTimeInitial
GET-REPLACE;
REGISTERED AS {trPackage 36};

PRESENT IF “an instance supports it.”,
trMaintServiceChargePkg PACKAGE
ATTRIBUTES
maintServiceCharge
INITIAL VALUE GNMTA.maintServiceChargeMaintServiceChargeInitial
GET;
REGISTERED AS {trPackage 37};

PRESENT IF “an instance supports it.”,
trManagedObjectAccessHoursPkg PACKAGE
ATTRIBUTES
managedObjectAccessHours GET-REPLACE ADD-REMOVE;
REGISTERED AS {trPackage 38};

PRESENT IF “an instance supports it.”,
trManagedObjectAccessFromTimePkg PACKAGE
ATTRIBUTES
managedObjectAccessFromTime GET-REPLACE;

```
REGISTERED AS {trPackage 39};
    PRESENT IF “an instance supports it.”,
trManagedObjectAccessToTimePkg PACKAGE
  ATTRIBUTES
    managedObjectAccessToTime GET-REPLACE;
REGISTERED AS {trPackage 40};
    PRESENT IF “an instance supports it.”,
trManagerContactPersonAttributePkg PACKAGE
  ATTRIBUTES
    managerContactPerson GET-REPLACE;
REGISTERED AS {trPackage 41};
    PRESENT IF “an instance supports it and trManagerContactPersonObjectPkg is not present.”,
trManagerContactPersonObjectPkg PACKAGE
 BEHAVIOUR managerContactPersonObjectBehaviour BEHAVIOUR
    DEFINED AS “The Manager Contact Person Object package contains one
                attribute which points to an instance of the Contact object
                that identifies an individual in the manager’s organization
                who can be contacted regarding the reported trouble.”;;
  ATTRIBUTES
    managerContactObjectPtr GET-REPLACE;
REGISTERED AS {trPackage 42};
    PRESENT IF “an instance supports it and trManagerContactPersonAttributePkg is not present.”,
trManagerSearchKeyPkg PACKAGE
  ATTRIBUTES
    managerSearchKey1 GET-REPLACE,
    managerSearchKey2 GET-REPLACE,
    managerSearchKey3 GET-REPLACE;
REGISTERED AS {trPackage 43};
    PRESENT IF “an instance supports it and trManagerSearchKeyListPkg is not present.”,
trManagerSearchKeyListPkg PACKAGE
  ATTRIBUTES
    managerSearchKeyList GET-REPLACE ADD-REMOVE;
REGISTERED AS {trPackage 44};
    PRESENT IF “an instance supports it and trManagerSearchKeyPkg is not present.”,
trOutageDurationPkg PACKAGE
  ATTRIBUTES
    outageDuration
        INITIAL VALUE GNMTA.outageDurationOutageDurationInitial
```

```
            GET;
REGISTERED AS {trPackage 45};
      PRESENT IF “an instance supports it.”,
trPerceivedTroubleSeverityPkg PACKAGE
   ATTRIBUTES
      perceivedTroubleSeverity GET-REPLACE;
      -- some implementations may not
      -- support GET-REPLACE
REGISTERED AS {trPackage 46};
      PRESENT IF “an instance supports it.”,
trPreferredPriorityPkg PACKAGE
   ATTRIBUTES
      preferredPriority GET-REPLACE;
REGISTERED AS {trPackage 47};
      PRESENT IF “an instance supports it.”,
trRepeatReportPkg PACKAGE
   ATTRIBUTES
      repeatReport GET-REPLACE;
REGISTERED AS {trPackage 48};
      PRESENT IF “an instance supports it.”,
trResponsiblePersonNamePkg PACKAGE
    BEHAVIOUR responsiblePersonNameBehaviour BEHAVIOUR
      DEFINED AS “Modifications to
                  responsiblePersonName are required only in the
                  service provider to service provider interface.
                  The CMIS error ‘access denied’ may be issued
                  in response to attempts to modify this attribute
                  on any other interface.”;;
   ATTRIBUTES
      responsiblePersonName GET-REPLACE;
REGISTERED AS {trPackage 49};
      PRESENT IF “an instance supports it and trResponsiblePersonPtrPkg
            is not present.”,
trResponsiblePersonPtrPkg PACKAGE
 BEHAVIOUR responsiblePersonPtrBehaviour BEHAVIOUR
      DEFINED AS “Modifications to
                  responsiblePersonPtr are required only in the
                  service provider to service provider interface.
```

```
                The CMIS error ‘access denied’ may be issued
                in response to attempts to modify this attribute
                on any other interface.”;;
    ATTRIBUTES
        responsiblePersonPtr GET-REPLACE;
REGISTERED AS {trPackage 50};
        PRESENT IF “an instance supports it and trResponsiblePersonNamePkg
                is not present.”,
trSuspectObjectListPkg PACKAGE
    ATTRIBUTES
        suspectObjectList GET SET-BY-CREATE;
REGISTERED AS {trPackage 51};
        PRESENT IF “an instance supports it.”,
trTroubleDetectionTimePkg PACKAGE
    ATTRIBUTES
        troubleDetectionTime
            DEFAULT VALUE GNMTA.troubleDetectionTimeTroubleDetectionTimeDefault
            GET-REPLACE;
REGISTERED AS {trPackage 52};
        PRESENT IF “an instance supports it.”,
trTroubleLocationPkg PACKAGE
 BEHAVIOUR troubleLocationBehaviour BEHAVIOUR
        DEFINED AS “Modifications to
                troubleLocation are required only in the
                service provider to service provider interface.
                The CMIS error ‘access denied’ may be issued
                in response to attempts to modify this attribute
                on any other interface.”;;
    ATTRIBUTES
        troubleLocation GET-REPLACE;
REGISTERED AS {trPackage 53};
        PRESENT IF “an instance supports it.”,
trTroubleReportStatusWindowPkg PACKAGE
    ATTRIBUTES
        additionalTroubleStatusInfo GET,
        troubleReportStatusWindow GET-REPLACE;
    NOTIFICATIONS
        “T1.228 : 1995”:troubleReportProgressNotification;
```

REGISTERED AS {trPackage 54};

PRESENT IF “an instance supports it.”,

trTspPriorityPkg PACKAGE

ATTRIBUTES

tspPriority GET SET-BY-CREATE;

REGISTERED AS {trPackage 55};

PRESENT IF “an instance supports it.”;

REGISTERED AS {trMObjectClass 3};

A.2.6 Telecommunications Trouble Report – R1 version

telecommunicationsTroubleReportR1 is a new object class derived from the telecommunications-TroubleReport object class. The telecommunicationsTroubleReportR1 object class is used when there is a need of using the customerInfo attribute with its data sequence values to further detail customer service information. For more detailed information on the customerInfo attribute, see 7.2.102 and the related ASN.1 supporting productions in 7.5.

telecommunicationsTroubleReportR1 MANAGED OBJECT CLASS

DERIVED FROM telecommunicationsTroubleReport;

CHARACTERIZED BY telecommunicationsTroubleReportR1Pkg PACKAGE

BEHAVIOUR

tTRr1Behaviour BEHAVIOUR

DEFINED AS

"This object class is used instead of telecommunicationsTroubleReport when there is a need for the customerInfo attribute.";;

CONDITIONAL PACKAGES

customerInfoPkg PACKAGE

BEHAVIOUR customerInfoBehaviour BEHAVIOUR

DEFINED AS

"The customerInfo attribute identifies data values necessary to further detail customer service information associated with the Managed Object Instance. The Manager will provide this information to the Agent to assist in the trouble resolution process.";

ATTRIBUTES

customerInfo GET;

REGISTERED AS {trPackage 97};

PRESENT IF "an instance supports it.";

REGISTERED AS {trMObjectClass 12};

A.2.7 Repair Activity

The Repair Activity object class will contain parameters and text describing the specific repair functions performed, who performed them, and when they were performed. For each repair activity performed in conjunction with resolving a problem related to a Trouble Report, a Repair Activity object is created.

A Trouble Report is the main point of the coordination of problem-solving activity. The Repair Activity object will provide a user with information regarding the activities carried out so far to resolve the problem. The manager will only be allowed to get its associated attributes and display them.

When a Trouble Report is deleted locally by the agent, the associated Repair Activity object(s) will also be deleted.

Repair Activity information may be alternatively stored in the Repair Activity List attribute in the Trouble Report. Both methods may not be used simultaneously.

```
repairActivity MANAGED OBJECT CLASS
   DERIVED FROM “Rec. X.721|ISO/IEC 10165-2 : 1992”:top;
   CHARACTERIZED BY repairActivityPkg PACKAGE
         ATTRIBUTES
            repairActivityID GET,
            entryTime GET,
            activityInfo DEFAULT VALUE GNMTA.activityInfoActivityInfoDefault GET;;;
   CONDITIONAL PACKAGES
      raActivityPersonPkg PACKAGE
         ATTRIBUTES
            activityPerson GET;
      REGISTERED AS {trPackage 56};
            PRESENT IF “an instance supports it.”,
      raActivityCodePkg PACKAGE
         ATTRIBUTES
            activityCode GET;
      REGISTERED AS {trPackage 57};
            PRESENT IF “an instance supports it.”,
      raObjectCreationPkg PACKAGE
         NOTIFICATIONS
            “Rec. X.721|ISO/IEC 10165-2 : 1992”:objectCreation;
      REGISTERED AS {trPackage 58};
            PRESENT IF “an instance supports it.”;
REGISTERED AS {trMObjectClass 4};
```

A.2.8 Service Profile

The Service Profile object class contains information describing service characteristics that are common to a set of CNM Services. The use of this object is for further study. Specialization of this object is expected as work progresses on additional functional areas of the CNM interface.

```
serviceProfile MANAGED OBJECT CLASS
   DERIVED FROM “Rec. X.721|ISO/IEC 10165-2 : 1992”:top;
   CHARACTERIZED BY serviceProfilePkg PACKAGE
      ATTRIBUTES
```

```
        serviceProfileID GET,
        serviceProfileDescription GET;;;
REGISTERED AS {trMObjectClass 5};
```

A.2.9 Trouble History Record

The Trouble History Record object class is a refinement of the Log Record object class in ISO/IEC 10165-2 and is used to log the Trouble History Event notifications from the Trouble Report object and its subclasses. The Trouble History Record object is a repository for selected information from a Trouble Report object and its subclasses. Instantiated subclasses of the Trouble Report object generate a Trouble History Event notification with Trouble History information whenever the Trouble Report Status attribute value changes to a final closed-out value. The attributes in the Trouble History Event notification (and therefore the Trouble History object) may be a subset of the attributes present in the Trouble Report object.

Trouble History Records are contained in a Log object. If the Log is deleted, all its contained Trouble History Records shall also be automatically deleted by the managed system. The service provider may also remove selected Trouble History Records locally based on some criteria, e.g., reaching a certain age limit or the number of records stored reaching a certain threshold value.

```
troubleHistoryRecord MANAGED OBJECT CLASS
  DERIVED FROM “Rec. X.721|ISO/IEC 10165-2 : 1992”:eventLogRecord;
  CHARACTERIZED BY troubleHistoryRecordPkg PACKAGE
    ATTRIBUTES
      “Rec. X.721|ISO/IEC 10165-2 : 1992”:eventTime GET,
      -- indicates close-out time, optional attribute in eventLogRecord
      managedObjectInstance GET,
      -- Copied from the corresponding trouble report object.
      -- Refers to instance of CNM Service
      -- or GNIM object representing a
      -- telecommunications resource.
      receivedTime GET,
      -- indicates trouble report creation time
      troubleFound GET;;;
CONDITIONAL PACKAGES
  thrActivityDurationPkg PACKAGE
    ATTRIBUTES
      activityDuration GET;
  REGISTERED AS {trPackage 59};
    PRESENT IF “an instance supports it.”,
  thrAdditionalTroubleInfoListPkg PACKAGE
    ATTRIBUTES
      additionalTroubleInfoList GET ADD
      --SET-BY-CREATE--;
  REGISTERED AS {trPackage 60};
```

PRESENT IF “an instance supports it.”,

thrAuthorizationPkg PACKAGE

ATTRIBUTES
authorizationList GET;

REGISTERED AS {trPackage 61};

PRESENT IF “an instance supports it.”,

thrCancelRequestedByManagerPkg PACKAGE

ATTRIBUTES
cancelRequestedByManager GET;

REGISTERED AS {trPackage 62};

PRESENT IF “an instance supports it.”,

thrCloseOutNarrPkg PACKAGE

ATTRIBUTES
closeOutNarr GET;

REGISTERED AS {trPackage 63};

PRESENT IF “an instance supports it.”,

thrCloseOutVerificationPkg PACKAGE

ATTRIBUTES
closeOutVerification GET;

REGISTERED AS {trPackage 64};

PRESENT IF “an instance supports it.”,

thrCommitmentTimePkg PACKAGE

ATTRIBUTES
commitmentTime GET;

REGISTERED AS {trPackage 65};

PRESENT IF “an instance supports it.”,

thrCustTroubleTickNumPkg PACKAGE

ATTRIBUTES
custTroubleTickNum GET;

REGISTERED AS {trPackage 66};

PRESENT IF “an instance supports it.”,

thrPerceivedTroubleSeverityPkg PACKAGE

ATTRIBUTES
perceivedTroubleSeverity GET;

REGISTERED AS {trPackage 67};

PRESENT IF “an instance supports it.”,

thrRestoredTimePkg PACKAGE

ATTRIBUTES
restoredTime GET;

REGISTERED AS {trPackage 68};

PRESENT IF “an instance supports it.”,

thrTroubleClearancePersonPkg PACKAGE

```
    ATTRIBUTES
      troubleClearancePerson GET;
  REGISTERED AS {trPackage 69};

    PRESENT IF “an instance supports it.”,

  thrTroubleReportNumberListPkg PACKAGE

    ATTRIBUTES
      troubleReportNumberList GET;
        REGISTERED AS {trPackage 70};

    PRESENT IF “an instance supports it.”,

  thrTroubleTypePkg PACKAGE

    ATTRIBUTES
      troubleType GET SET-BY-CREATE;
  REGISTERED AS {trPackage 71};

    PRESENT IF “an instance supports it.”;

REGISTERED AS {trMObjectClass 6};
```

A.2.10 Trouble Report

The Trouble Report object is the superclass of the Telecommunications Trouble Report object. The Trouble Report object is not instantiated. Instances of the Telecommunications Trouble Report subclass represent customer-reported troubles on telecommunications services or resources. Trouble reports describe the nature of the problem as well as ongoing status.

Local administrations may put restrictions on the number of open trouble reports per managed object through business agreements.

The Trouble Report Administration model allows multiple Trouble Report formats as defined by instances of the Trouble Report Format Definition object. Each Trouble Report format is a predefined combination of trouble report attributes. The Trouble Report format applicable to a particular CNM-Service-managed or GNIM-managed object instance can be dynamically specified by the service provider through the Trouble Report Format Definition object. When the Trouble Report format is explicitly defined through the Trouble Report Format Definition object for a CNM-Service-managed or GNIM-managed object, a Telecommunications Trouble Report instance for that CNM-Service-managed or GNIM-managed object shall consist of:

- the mandatory attributes of the Telecommunications Trouble Report, as defined in the Telecommunications Trouble Report object specification;
- the attributes in conditional packages of the Telecommunications Trouble Report that “must be present” as defined by the corresponding Trouble Report Format Definition object;
- optionally the attributes in conditional packages of the Telecommunications Trouble Report that “may be present,” as defined by the corresponding Trouble Report Format Definition object.

The appropriate instance of the Trouble Report Format Definition object is identified either:

a) by a pointer attribute (troubleReportFormatObjectPtr) in the CNM Service object (when the format must be defined on an object instance basis);
b) by inclusion of the managed object class in an applicableManagedObjectClassList attribute in the Trouble Report Format Definition object (when the format is the same for an entire object class);
c) by inclusion of an instance of a GNIM object class that represents a telecommunications resource in an applicableManagedObjectInstanceList attribute in the Trouble Report Format Definition object (when the format is specific to the object instance).

The manager is allowed to create Telecommunications Trouble Reports (subclass of Trouble Report). As part of the create, the manager is required to supply the following attributes:

- Managed Object Instance;
- Trouble Type;
- Additional Trouble Information List;

plus any manager-supplied attributes in conditional packages identified as “must be present” attributes. The manager also has the option to include manager-supplied attributes in conditional packages identified as “may be present” attributes.

As part of the instantiation of a Telecommunications Trouble Report object, the agent is also required to supply values for the remaining characterized by attributes of the Telecommunications Trouble Report object class plus any agent-supplied attributes in conditional packages identified as “must be present” attributes. The agent also has the option to include agent-supplied attributes in conditional packages identified as “may be present” attributes.

A manager may request that a trouble report be canceled, which may or may not result in the trouble report being closed-out immediately. Closed-out trouble reports are deleted locally by the agent according to some storage period criteria (e.g., 3, 12, or 18 months).

The Telecommunications Trouble Report object generates the object creation and object deletion notifications whenever the agent creates a Telecommunications Trouble Report object or deletes a Telecommunications Trouble Report object through local administrative procedures. These notifcations are inherited from the Trouble Report object.

An Attribute Value Change Notification is emitted when there is a change in the value of a Telecommunications Trouble Report attribute. This notification is inherited from the Trouble Report object. In some implementations, only changes in the Trouble Report Status or Commitment Time attributes are emitted.

The Telecommunications Trouble Report object generates a Trouble History Event Notification with Trouble History information whenever the Trouble Report Status attribute value changes to a closed-out value. This notification is inherited from the Trouble Report object.

> NOTE – This notification is in addition to the attribute value change notification for the Trouble Report Status attribute. This notification is offered to a Log where the discriminator attribute of the log decides whether the notification will be logged. In some implementations, the attributes that allow selective logging will be absent or not under the control of the interface. The concept of logs is introduced in ISO/IEC 10164-6.

```
troubleReport MANAGED OBJECT CLASS

  DERIVED FROM “Rec. X.721|ISO/IEC 10165-2 : 1992”:top;

  CHARACTERIZED BY troubleReportPkg PACKAGE

    BEHAVIOUR troubleReportBehaviour BEHAVIOUR
      DEFINED AS “Modifications to troubleFound, troubleReportState,
                  and troubleReportStatus are required only in the
                  service provider to service provider interface.
                  The CMIS error ‘access denied’ may be issued
                  in response to attempts to modify these attributes
                  on any other interface.”;;

  ATTRIBUTES

    additionalTroubleInfoList GET ADD,

      -- some implementations may not support a GET

    managedObjectInstance GET SET-BY-CREATE,

                  receivedTime GET,

    troubleFound INITIAL VALUE GNMTA.troubleFoundTroubleFoundInitial

      GET-REPLACE,

    troubleReportID GET,
```

```
    troubleReportState GET-REPLACE,
    troubleReportStatus GET-REPLACE,
    troubleReportStatusTime GET,
    troubleType GET SET-BY-CREATE;;;
CONDITIONAL PACKAGES
  trActivityDurationPkg PACKAGE
    BEHAVIOUR activityDurationBehaviour BEHAVIOUR
      DEFINED AS “Modifications to
                activityDuration are required only in the
                service provider to service provider interface.
                The CMIS error ‘access denied’ may be issued
                in response to attempts to modify this attribute
                on any other interface.”;;
      ATTRIBUTES
        activityDuration
        INITIAL VALUE GNMTA.activityDurationactivityDurationInitial
        GET ADD;
  REGISTERED AS {trPackage 72};
      PRESENT IF “an instance supports it.”,
  trAgentContactPersonAttributePkg PACKAGE
    ATTRIBUTES
      agentContactPerson GET;
  REGISTERED AS {trPackage 73};
      PRESENT IF “an instance supports it and trAgentContactPersonObjectPkg is not present.”,
  trAgentContactPersonObjectPkg PACKAGE
    BEHAVIOUR agentContactPersonObjectBehaviour BEHAVIOUR
      DEFINED AS “The Agent Contact Person Object package points to an
                instance of the Contact object that identifies an
                individual in the agent’s organization who can be contacted
                regarding the reported trouble.”;;
      ATTRIBUTES
        agentContactObjectPtr GET;
  REGISTERED AS {trPackage 74};
      PRESENT IF “an instance supports it and trAgentContactPersonAttributePkg is not present.”,
  trCloseOutNarrPkg PACKAGE
    BEHAVIOUR closeOutNarrBehaviour BEHAVIOUR
      DEFINED AS “Modifications to
                closeOutNarr are required only in the
                service provider to service provider interface.
      The CMIS error ‘access denied’ may be issued
                in response to attempts to modify this attribute
                on any other interface.”;;
      ATTRIBUTES
        closeOutNarr
```

```
        INITIAL VALUE GNMTA.closeOutNarrCloseOutNarrInitial
        GET-REPLACE;
REGISTERED AS {trPackage 75};

    PRESENT IF “an instance supports it.”,
trManagedObjectInstanceAliasListPkg PACKAGE
  ATTRIBUTES
    managedObjectInstanceAliasList GET;
REGISTERED AS {trPackage 76};

    PRESENT IF “an instance supports it.”,
trNumberListPkg PACKAGE
  ATTRIBUTES
    troubleReportNumberList GET;
REGISTERED AS {trPackage 77};

    PRESENT IF “an instance supports it.”,
trRelatedTroubleReportListPkg PACKAGE
  ATTRIBUTES
    relatedTroubleReportList GET;
REGISTERED AS {trPackage 78};

    PRESENT IF “an instance supports it.”,
trRepairActivityListPkg PACKAGE
 BEHAVIOUR repairActivityListBehaviour BEHAVIOUR
   DEFINED AS “Modifications to
              repairActivityList are required only in the
              service provider to service provider interface.
              The CMIS error ‘access denied’ may be issued
              in response to attempts to modify this
              attribute on any other interface.”;;
  ATTRIBUTES
    repairActivityList
      INITIAL VALUE GNMTA.repairActivityListRepairActivityListInitial
      GET ADD;
    -- Support for repairActivityList
    -- determined by policies of
    -- administration performing repair.
    REGISTERED AS {trPackage 79};
   PRESENT IF “an instance supports it and no RepairActivity object is
              contained in an instance of this object class or its subclasses.”,
trRestoredTimePkg PACKAGE
 BEHAVIOUR restoredTimeBehaviour BEHAVIOUR
   DEFINED AS “Modifications to
              restoredTime are required only in the
              service provider to service provider interface.
              The CMIS error ‘access denied’ may be issued
              in response to attempts to modify this attribute
```

```
                on any other interface”;;
    ATTRIBUTES
      restoredTime INITIAL VALUE GNMTA.restoredTimeRestoredTimeInitial
        GET-REPLACE;
  REGISTERED AS {trPackage 80};
    PRESENT IF “an instance supports it.”,
  trTroubleClearancePersonAttributePkg PACKAGE
    ATTRIBUTES
      troubleClearancePerson
        DEFAULT VALUE GNMTA.troubleClearancePersonTroubleClearancePersonDefault
        GET-REPLACE;
  REGISTERED AS {trPackage 81};
    PRESENT IF “an instance supports it.”,
  trTroubleReportFormatObjectPtrPkg PACKAGE
    ATTRIBUTES
      troubleReportFormatObjectPtr GET SET-BY-CREATE;
  REGISTERED AS {trPackage 82};
    PRESENT IF “an instance supports it.”,
  trAttributeValueChangePkg PACKAGE
    NOTIFICATIONS
      “Rec. X.721|ISO/IEC 10165-2 : 1992”:attributeValueChange;
  REGISTERED AS {trPackage 83};
    PRESENT IF “an instance supports it.”,
  trObjectCreationDeletionPkg PACKAGE
    NOTIFICATIONS
      “Rec. X.721|ISO/IEC 10165-2 : 1992”:objectCreation,
      “Rec. X.721|ISO/IEC 10165-2 : 1992”:objectDeletion;
      REGISTERED AS {trPackage 84};
    PRESENT IF “an instance supports it.”,
  trHistoryEventPkg PACKAGE
    NOTIFICATIONS
      “T1.228 : 1995”:troubleHistoryEventNotification;
  REGISTERED AS {trPackage 85};
    PRESENT IF “an instance supports it.”;
REGISTERED AS {trMObjectClass 7};
```

A.2.11 Trouble Report Format Definition

The Trouble Report Format Definition object gives the service provider a flexible scheme that allows definition of Trouble Report formats. It also provides the flexibility to dynamically specify Trouble Report formats for a service/resource object on an object class basis or on an object instance basis. A Trouble Report Format Definition

contains a selected list of attribute identifiers, from the attributes in conditional packages of the Trouble Report object class or its subclasses.

An instance of the Trouble Report Format Definition object defines which Trouble Report conditional package attributes “must be present” or “may be present” in an instance of the Telecommunications Trouble Report. The behavior of the Telecommunications Trouble Report object determines whether the manager or agent supplies and/or updates the attributes associated with these conditional packages.

“Must be present” and “may be present” conditional package attributes are identified by the tRMustBePresentAttrIDList and the tRMayBePresentAttrIDList attributes, respectively. The tRMustBePresentAttrIDList and tRMayBePresentAttrIDList attributes may only contain attribute IDs that have already been defined as attributes in conditional packages of the Trouble Report object class or its subclasses.

The Trouble Report Format Definition object can be used in determining the format for reporting troubles either on an instance of the CNM Service object or on an instance of a GNIM object representing a telecommunications resource. The appropriate instance of the Trouble Report Format Definition object is identified through one of the following ways:

a) By a pointer attribute (Trouble Report format) in the CNM Service object (when the format must be defined on an object instance basis);
b) By inclusion of the managed object class (either CNM Service or GNIM telecommunications resource) in an applicableManagedObjectClassList attribute in the Trouble Report Format Definition object (when the format is the same for an entire object class);
c) By inclusion of a managed object instance of a GNIM telecommunications resource in an applicableManagedObjectInstanceList attribute in the Trouble Report Format Definition object (when the format is specific to the object instance).

NOTE – If all instances of the same object class use the same trouble report format, it is recommended that this be represented using the applicableManagedObjectClassList attribute.

Instances of the Trouble Report Format Definition object class are locally created and updated by the agent system.

Trouble Report format examples for a typical Exchange Carrier are provided in Annex E.

```
troubleReportFormatDefn MANAGED OBJECT CLASS
  DERIVED FROM “Rec. X.721|ISO/IEC 10165-2 : 1992”:top;
  CHARACTERIZED BY troubleReportFormatDefnPkg PACKAGE
      ATTRIBUTES
        tRFormatID GET;;;
  CONDITIONAL PACKAGES
    trfdApplicableManagedObjectClassListPkg PACKAGE
      ATTRIBUTES
        applicableManagedObjectClassList GET;
        -- present if this instance of the format
        -- definition object applies to all objects
        -- of classes in this list
    REGISTERED AS {trPackage 86};
        PRESENT IF “an instance supports it.”,
    trfdApplicableManagedObjectInstanceListPkg PACKAGE
      ATTRIBUTES
        applicableManagedObjectInstanceList GET;
        -- present if this instance of the format
        -- definition object applies to only some
        -- instances of a GNIM object class
    REGISTERED AS {trPackage 87};
        PRESENT IF “an instance supports it.”,
```

trfdTrConstrainedToSingleValueAttrIDListPkg PACKAGE
BEHAVIOUR trfdTrConstrainedToSingleValueAttrIDListBehaviour BEHAVIOUR

DEFINED AS “If a manager attempts to add more than one value to attributes of this list, a CMISE ‘complexity limitation’ error will be generated.”;;

ATTRIBUTES

tRConstrainedToSingleValueAttrIDList GET;
REGISTERED AS {trPackage 88};
PRESENT IF “an instance supports it.”,
trfdTrMayBePresentAttrIDListPkg PACKAGE

ATTRIBUTES

tRMayBePresentAttrIDList GET;
REGISTERED AS {trPackage 89};
PRESENT IF “an instance supports it.”,
-- either tRMayBePresentAttrIDList or tRMustBePresentAttrIDList
-- or both shall be present in an instance of the Trouble Report
-- Format Defn. object
trfdTrMustBePresentAttrIDListPkg PACKAGE

ATTRIBUTES

tRMustBePresentAttrIDList GET;
REGISTERED AS {trPackage 90};
PRESENT IF “an instance supports it.”,
-- either tRMayBePresentAttrIDList or tRMustBePresentAttrIDList
-- or both shall be present in an instance of the Trouble Report
-- Format Defn. object
trfdAttributeValueChangePkg PACKAGE

NOTIFICATIONS

“Rec. X.721|ISO/IEC 10165-2 : 1992”:attributeValueChange;
REGISTERED AS {trPackage 91};
PRESENT IF “an instance supports it.”,
trfdObjectCreationDeletionPkg PACKAGE

NOTIFICATIONS

“Rec. X.721|ISO/IEC 10165-2 : 1992”:objectCreation,
“Rec. X.721|ISO/IEC’10165-2 : 1992”:objectDeletion;
REGISTERED AS {trPackage 92};
PRESENT IF “an instance supports it.”;

REGISTERED AS {trMObjectClass 8};

A.2.12 Agent System Monitor

This object allows the information on status of different subsystems (such as OSs) of the agent to be accessed by the manager. A separate object instance is needed to provide status of each different subsystem. The agent is responsible for locally updating the appropriate attributes in the object instances representing the status of its different subsystems. Whenever the status of one subsystem changes, an attribute value change notification would be emitted, and this would notify the manager of the status change of the subsystem. In addition, the manager may access the attribute values of any of the agentSystemMonitor objects by using m-Get.

agentSystemMonitor MANAGED OBJECT CLASS

DERIVED FROM “Rec. X.721|ISO/IEC 10165-2:1992”: top;

CHARACTERIZED BY agentSystemMonitorPkg PACKAGE

BEHAVIOUR agentSystemMonitorBehaviour BEHAVIOUR

DEFINED AS “This object provides status reports on one subsystem (such as Operations System) of the agent.”;;

ATTRIBUTES

systemMonitorId GET;;;

CONDITIONAL PACKAGES

“Rec. M.3100:1992”: attributeValueChangeNotificationPackage

PRESENT IF “an instance supports it”,

“Rec. M.3100:1992”:createDeleteNotificationsPackage

PRESENT IF “an instance supports it”,

asmStatusOfSystemPkg PACKAGE

ATTRIBUTES

statusOfSystem GET;

REGISTERED AS {trPackage 93};

PRESENT IF “an instance supports it”,

asmEstimatedTimeOfRestoralPkg PACKAGE

ATTRIBUTES

estimatedTimeOfRestoral GET;

REGISTERED AS {trPackage 94};

PRESENT IF “an instance supports it”;

REGISTERED AS {trMObjectClass 9};

A.2.13Manager System Monitor

This object allows the information on status of different subsystems (such as OSs) of the manager to be accessed by the agent. A separate object instance is needed to provide status of each different subsystem. The manager is responsible for requesting the agent to update the attributes of the different object instances representing the status of its different subsystems. The agent can locally access the attributes in these object instances in order to acquire the status information of different susbsytems of the manager.

managerSystemMonitor MANAGED OBJECT CLASS

DERIVED FROM “Rec. X.721|ISO/IEC 10165-2:1992”: top;

CHARACTERIZED BY managerSystemMonitorPkg PACKAGE

BEHAVIOUR managerSystemMonitorBehaviour BEHAVIOUR

DEFINED AS “This object provides status reports on one susbsystem (such as Operations System) of the agent.”;;

ATTRIBUTES

systemMonitorId GET SET-BY-CREATE;;;

CONDITIONAL PACKAGES

“Rec. M.3100:1992”: attributeValueChangeNotificationPackage

PRESENT IF “an instance supports it”,

“Rec. M.3100:1992”:createDeleteNotificationsPackage

PRESENT IF “an instance supports it”,

msmStatusOfSystemPkg PACKAGE

ATTRIBUTES

statusOfSystem GET-REPLACE;

REGISTERED AS {trPackage 95};

PRESENT IF “an instance supports it”,

msmEstimatedTimeOfRestoralPkg PACKAGE

ATTRIBUTES

estimatedTimeOfRestoral GET REPLACE;

REGISTERED AS {trPackage 96};

PRESENT IF “an instance supports it”;

REGISTERED AS {trMObjectClass 10};

A.3 Attribute Type Definitions

This subclause contains the definitions of the attribute types specified in this standard. Each attribute type definition consists of a text definition followed by a template specification, as specified in ISO/IEC 10165-4.

The ability to read and write attributes belonging to object classes defined in this standard can generally be accomplished using the PT-GET and PT-SET services of ISO/IEC 10164-1. However, the ability to read or write any given attribute is constrained by the object class definition in which the attribute appears and also by the Functional Units negotiated at the start of the association.

Some attributes (e.g., Managed Object Instance) are defined as pointers to other objects belonging to an object class specified in this standard (e.g., CNM Service). These pointers may also reference objects belonging to subclasses of the specified object classes (e.g., subclasses of CNM Service). When these subclasses are not known to the peer systems, objects of these subclasses are to be treated as members of an object class specified in this standard (e.g., CNM Service if the pointer refers to a subclass of CNM Service, etc.).

For the convenience of the reader, annex F gives a pictorial representation of the pointer attribute relationships between objects in this document. Annex F does not represent normative information.

Some administrations restrict the use of matching criteria for certain attributes.

A.3.1 Account Contact List

The Account Contact List attribute specifies the individuals in the manager’s organization, who can be contacted regarding the account.

```
accountContactList ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.AccountContactList;
   MATCHES FOR
      EQUALITY,
      SET-INTERSECTION,
      SET-COMPARISON;
REGISTERED AS {trAttribute 1};
```

A.3.2 Account Name

The Account Name attribute is the name given to an account by the customer, where an account is a customer or agency entity that may be billed by the service provider or that may take responsibility for performing network management services for the customer. The Account Name is the RDN of the Account object.

```
accountName ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.AccountName;
   MATCHES FOR
      EQUALITY,
      SUBSTRINGS;
REGISTERED AS {trAttribute 2};
```

A.3.3 Activity Code

The Activity Code attribute identifies a general repair activity category.

```
activityCode ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.ActivityCode;
   MATCHES FOR
      EQUALITY;
REGISTERED AS {trAttribute 3};
```

A.3.4 Activity Duration

The Activity Duration attribute indicates time spent on billable and nonbillable activities. It is possible to indicate the total billable or nonbillable time spent on a group of activities as indicated by the bits with a value of “1” in the bit string.

```
activityDuration ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.ActivityDuration;
REGISTERED AS {trAttribute 4};
```

A.3.5 Activity Information

The Activity Information attribute will contain 256 bytes of text that will indicate what repair activity is being carried out to repair the problem.

```
activityInfo ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.ActivityInfo;
   MATCHES FOR
      EQUALITY,
      SUBSTRINGS;
REGISTERED AS {trAttribute 5};
```

A.3.6 Activity Person

The Activity Person attribute contains information about the operator or supervisor who created the repair activity request. Matching for equality means checking personNumber only in the sequence.

activityPerson ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.ActivityPerson;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 6};

A.3.7 Additional Text

The Additional Text attribute contains additional pertinent enterprise information that describes the Account. This enterprise information pertains to the way the customer and the service provider interact when conducting business.

additionalText ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.AdditionalText;

MATCHES FOR

EQUALITY,

SUBSTRINGS;

REGISTERED AS {trAttribute 7};

A.3.8 Additional Trouble Information List

The Additional Trouble Information List attribute further describes the selected Trouble Type. A minimum of 256 octets shall be supported, regardless of the number of values in the set. The manager can only add information, but not remove it. It is possible that the oldest information may be lost if an implementation has restrictions on the maximum size.

additionalTroubleInfoList ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.AdditionalTroubleInfoList;

MATCHES FOR

EQUALITY,

SET-INTERSECTION,

SET-COMPARISON;

REGISTERED AS {trAttribute 8};

A.3.9 Additional Trouble Status Information

The Additional Trouble Status Information attribute further describes the value of the Trouble Report Status attribute. Information shall only be added and not removed.

additionalTroubleStatusInfo ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.AdditionalTroubleStatusInfo;

MATCHES FOR

EQUALITY,

SET-INTERSECTION,

SET-COMPARISON;

REGISTERED AS {trAttribute 9};

A.3.10 Administrative State

This attribute is imported from ISO/IEC 10164-2.

A.3.11 After Hours Repair Authorization

The After Hours Repair Authorization attribute indicates whether the customer has given the OK to repair the service outside normal business hours (e.g., 9 am–5 pm, Monday through Friday).

afterHrsRepairAuth ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.AfterHrsRepairAuth;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 10};

A.3.12 Agent Contact Person

The Agent Contact Person attribute identifies an individual in the agent's organization who can be contacted regarding the reported trouble. Matching for equality means checking personNumber only in the sequence.

agentContactPerson ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.PersonReach;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 11};

A.3.13 Agent Contact Object Pointer

The Agent Contact Object Pointer attribute points to a Contact object that identifies an individual in the agent's organization who can be contacted regarding the reported trouble.

agentContactObjectPtr ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.AgentContactObjectPtr;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 12};

A.3.14 Alarm Record Pointer List

The Alarm Record Pointer List attribute points to instance(s) of the Alarm Record available in the agent system. A necessary condition for this attribute to be present is that the Trouble Report shall have been generated as a result of an alarm. However, this is not a sufficient condition since some administrations may choose not to support this attribute even if the Trouble Report was generated as a result of an alarm received or generated in the agent.

alarmRecordPtrList ATTRIBUTE

```
    WITH ATTRIBUTE SYNTAX GNMTA.AlarmRecordPtrList;
    MATCHES FOR
        EQUALITY,
        SET-INTERSECTION,
        SET-COMPARISON;
    REGISTERED AS {trAttribute 13};
```

A.3.15 Alarm State

This attribute is imported from the GNIM [see ATIS-0300240.1998(R2007)].

A.3.16 Alternate Manager Contact Person

The Alternate Manager Contact Person attribute identifies an alternative individual to the manager contact in the manager's organization who can be contacted regarding the reported trouble. Matching for equality means checking personNumber only in the sequence.

```
alternateManagerContactPerson ATTRIBUTE
    WITH ATTRIBUTE SYNTAX GNMTA.PersonReach;
    MATCHES FOR
        EQUALITY;
REGISTERED AS {trAttribute 14};
```

A.3.17 Alternate Manager Contact Object Pointer

The Alternate Manager Contact Object Pointer attribute points to a Contact object that identifies an alternative individual to the manager contact in the manager's organization who can be contacted regarding the reported trouble.

```
alternateManagerContactObjectPtr ATTRIBUTE
    WITH ATTRIBUTE SYNTAX GNMTA.AlternateManagerContactObjectPtr;
    MATCHES FOR
        EQUALITY;
REGISTERED AS {trAttribute 15};
```

A.3.18 Applicable Managed Object Class List

The Applicable Managed Object Class List attribute identifies the classes of managed objects to which a particular Trouble Report Format Definition applies.

```
applicableManagedObjectClassList ATTRIBUTE
    WITH ATTRIBUTE SYNTAX GNMTA.ApplicableManagedObjectClassList;
    MATCHES FOR
        EQUALITY,
        SET-INTERSECTION,
        SET-COMPARISON;
REGISTERED AS {trAttribute 16};
```

A.3.19 Applicable Managed Object Instance List

The Applicable Managed Object Instance List attribute identifies the instances of managed objects to which a particular Trouble Report Format Definition applies.

applicableManagedObjectInstanceList ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.ApplicableManagedObjectInstanceList;

MATCHES FOR

EQUALITY,

SET-INTERSECTION,

SET-COMPARISON;

REGISTERED AS {trAttribute 17};

A.3.20 Authorization List

A.3.20.1 General

The Authorization List attribute identifies whether authorization is requested by the agent and granted by the manager. It also specifies the type of activities that are authorized, and optionally the authorizing person, and the time of authorization.

A.3.20.2 Mechanism for Requesting & Granting Authorization

When the troubleReport objectClass is entered, the authorizationList attribute may have been created as an empty SEQUENCE (a sequence of zero length) or may have been created with an initial set of authorizations provided by the manager or with a set of default authorizations provided by the agent (under an assumed prior agreement with the manager). In the former case, no activities are authorized by the manager. In the latter case, the authorizationList attribute would include a requestedState with a value of “provided(2)” (since authorization is always provided by the manager and may be requested by the agent). It would also include an activityType where the bit string would include a “1” for each bit representing an activity that is authorized. It may also include an authTime and an authPerson indicating who in the manager’s organization has authorized the specific activities.

Subsequently, after the troubleReport object instance has been created the agent may request authorization by changing the value of the authorizationList attribute. The sequence shall include a state= “requested(1)” and an activityType plus a bit string indicating which items are being requested (all items for which authorization is required shall be indicated, even if previously authorized). The agent would not include an authTime or a value for the authPerson in the sequence. This change (request for authorization) would be indicated to the manager via an attribute ValueChange notification.

The manager then responds to this request for authorization by adding a new sequence to this attribute. This sequence shall include a state=“provided(2)”, an activityType indicating the total set of authorized activities, an authTime and optionally the person providing the authorization.

If the request is denied, the sequence may include a state = “denied (3)” and the additionalTroubleInfoList attribute may be used to show the reason for the denial. If authorization is simultaneously requested for multiple activities, it is not required that the authorization be provided or denied simultaneously for all activities in the same response The response "denied (3)" can only be sent by the manager for activities that have not been previously authorized.

In addition, the activityType also contains an integer value for deregulatedWork. The deregulatedWork activity is for work performed, by the service provider, beyond the demarcation point of the regulated service. Deregulated work shall be authorized by the manager before the work is performed by the agent. This authorization allows the agent to bill the manager for any work performed, on behalf of the manager, within the rules and regulations of deregulated activities. The authorizations for deregulated work is currently performed outside of the EB interface

because there is no support in the current standard. (It is envisioned that this authorization will become more common once the agents and managers utilize the EB gateways for reporting troubles on resold services.)

The activityType is provided by the manager on the create and updateable by the manager or the agent.

authorizationList ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.AuthorizationList;

REGISTERED AS {trAttribute 18};

A.3.21 Call Back Information List

The Call Back Information List attribute identifies the call back types requested by the manager and the person to be contacted for each call back type. Call back requests of multiple types can be present in the trouble report. The call back types identified are as follows:

- escalation, indicating customer requested a call back from the higher officials of the service provider;
- before_auto_test, indicating customer requested a call back before performing any automatic tests on the service;
- after_cleared, indicating customer requested a call back when the trouble is cleared.

The person information is modeled as a sequence of various optional elements with the condition that at least one of the optional elements should be present.

callBackInfoList ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.CallBackInfoList;

REGISTERED AS {trAttribute 19};

A.3.22 Called Number

The Called Number attribute specifies the number being called at the time of trouble detection.

calledNumber ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.CalledNumber;

MATCHES FOR

EQUALITY,

SUBSTRINGS;

REGISTERED AS {trAttribute 20};

A.3.23 Cancel Requested By Manager

The Cancel Requested By Manager attribute is a Boolean that indicates whether the manager has initiated the process to cancel a trouble report. When set to “TRUE”, the manager has requested that the trouble report be canceled.

cancelRequestedByManager ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.CancelRequestedByManager;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 21};

A.3.24 Close Out Narrative

The Close Out Narrative attribute specifies additional information about the problem. This field provides a place for the person who resolved the problem to document any additional information regarding the trouble report closure. This field will be copied into the Trouble History information.

closeOutNarr ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.CloseOutNarr;

REGISTERED AS {trAttribute 22};

A.3.25 Commitment Time

The Commitment Time attribute indicates either the on-site or trouble cleared time given to the customer. The agent provides a value when a trouble report is entered, but may update the value later.

commitmentTime ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.CommitmentTime;

MATCHES FOR

ORDERING;

REGISTERED AS {trAttribute 23};

A.3.26 Commitment Time Request

The Commitment Time Request attribute indicates either the on-site or trouble cleared time requested by the customer.

commitmentTimeRequest ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.CommitmentTime;

MATCHES FOR

ORDERING;

REGISTERED AS {trAttribute 24};

A.3.27 Contact Object Pointer List

The Contact Object Pointer List attribute points to instances of the Contact object that represents individuals in the agent's or manager's organization.

contactObjectPtrList ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.ContactObjectPtrList;

MATCHES FOR

EQUALITY,

SET-INTERSECTION,

SET-COMPARISON;

REGISTERED AS {trAttribute 25};

A.3.28 Close Out Verification

The Close Out Verification attribute indicates whether the manager has verified repair completion, denied repair completion, or taken no action.

```
closeOutVerification ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.CloseOutVerification;
   MATCHES FOR
      EQUALITY;
REGISTERED AS {trAttribute 26};
```

A.3.29 Current Problem List

This attribute is imported from the GNIM [see ATIS-0300240.1998(R2007)].

A.3.30 Customer Trouble Ticket Number

The Customer Trouble Ticket Number attribute contains the customer's internal trouble ticket number. It allows the customer to access troubles reported to the service provider with the local ticket number.

```
custTroubleTickNum ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.CustTroubleTickNum;
   MATCHES FOR
      EQUALITY,
      SUBSTRINGS;
REGISTERED AS {trAttribute 27};
```

A.3.31 Customer Work Center

The Customer Work Center attribute identifies the manager work center from which the trouble was entered.

```
customerWorkCenter ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.CustomerWorkCenter;
      MATCHES FOR
      EQUALITY,
      SUBSTRINGS;
REGISTERED AS {trAttribute 28};
```

A.3.32 Dialog

The Dialog attribute enables interaction to take place between the agent and the manager at each stage of the resolution of the trouble. This "dialog text" is free format text and a notification (attributeValueChange) is emitted each time it is modified. The contents are replaced by new "dialog text" as the dialog progresses during the trouble resolution. If the update is a response to previous text, the update may overwrite the current text.

```
dialog ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.Dialog;
   MATCHES FOR
      EQUALITY,
      SUBSTRINGS;
REGISTERED AS {trAttribute 29};
```

A.3.33 Entry Time

The Entry Time attribute indicates the time when the repair activity has been started.

```
entryTime ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.EntryTime;
   MATCHES FOR
      ORDERING;
REGISTERED AS {trAttribute 30};
```

A.3.34 Escalation List

A.3.34.1 General

The Escalation List attribute indicates whether escalation is requested by the manager or granted by the agent, or proactively provided by the agent. It optionally specifies the level of escalation and the person escalated to.

A.3.34.2 Mechanism for requesting and granting escalation

After a trouble report is created, the manager may request escalation by adding a “request” sequence to this multivalued escalationList attribute. The manager may indicate an escalation level (each level above 0 is another level higher in the agent organization). Typically the request would not include the person escalated to. The agent would respond by adding a “provided” sequence with the escalation time, and optionally the person and level escalated to.

```
escalationList ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.EscalationList;
REGISTERED AS {trAttribute 31};
```

A.3.35 Estimated Time Of Restoral

This attribute indicates the estimated time at which a subsystem of manager or agent is expected to be restored.

```
estimatedTimeOfRestoral ATTRIBUTE
  WITH ATTRIBUTE SYNTAX GNMTA.EstimatedTimeOfRestoral;
  MATCHES FOR EQUALITY;
REGISTERED AS {trAttribute 95};
```

A.3.36 Hand Off Center

The Hand Off Center attribute identifies the service provider’s control center to which a trouble report has been referred.

```
handOffCenter ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.HandOffCenter;
   MATCHES FOR
      EQUALITY,
      SUBSTRINGS;
REGISTERED AS {trAttribute 32};
```

A.3.37 Hand Off Location

The Hand Off Location attribute identifies the location within a service provider control center to which a trouble report has been referred.

```
handOffLocation ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.HandOffLocation;
   MATCHES FOR
      EQUALITY,
      SUBSTRINGS;
REGISTERED AS {trAttribute 33};
```

A.3.38 Hand Off Person Name

The Hand Off Person Name attribute identifies the person who is the head of the Hand Off Center where the trouble report has been referred. This center is on the agent side. Matching for equality means checking personNumber only in the sequence.

```
handOffPersonName ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.PersonReach;
   MATCHES FOR
      EQUALITY;
REGISTERED AS {trAttribute 34};
```

A.3.39 Hand Off Person Pointer

The Hand Off Person Pointer attribute identifies the person who is the head of the Hand Off Center where the trouble report has been referred. This center is on the agent side.

```
handOffPersonPtr ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.HandOffPersonPtr;
   MATCHES FOR
      EQUALITY;
REGISTERED AS {trAttribute 35};
```

A.3.40 Hand Off Time

The Hand Off Time attribute indicates the time at which a trouble was referred to the Hand Off Center. Matching for ordering is only applicable to GeneralizedTime.

```
handOffTime ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.HandOffTime;
   MATCHES FOR
      ORDERING;
REGISTERED AS {trAttribute 36};
```

A.3.41 Initiating Mode

The Initiating Mode attribute specifies the mode of initiation of the trouble report. This attribute can take the following integer values:

- managerDirect indicating that the manager caused the creation of the trouble report across the interoperable interface;
- managerIndirect indicating that the manager reported the trouble to the agent (other than through this interface) and the agent locally created the trouble report in the system.
    - managerIndirectEMail....(via email)....
    - managerIndirectFAX....(via Fax)....
    - managerIndirectPersonal....(personally)....
    - managerIndirectPhone....(via phone)....
- agentOriginated indicating that the agent detected a problem and locally created the trouble report in the system;
- alarmOriginated indicating that the trouble report was automatically created by the network/equipment because of an alarm.

initiatingMode ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.InitiatingMode;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 37};

A.3.42 Last Update Time

The Last Update Time attribute identifies the time and date of the most recent update made to the trouble report by either the manager or the agent. This attribute does not emit an attribute value change notification. The update is performed locally by the agent.

lastUpdateTime ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.LastUpdateTime;

MATCHES FOR

ORDERING;

REGISTERED AS {trAttribute 38};

A.3.43 A Location Access Address

The A Location Access Address attribute identifies the A address for which the respective A Location Access Hours attribute values are valid.

aLocationAccessAddress ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.LocationAddress;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 39};

A.3.44 Z Location Access Address

The Z Location Access Address attribute identifies the Z address for which the respective Z Location Access Hours attribute values are valid.

zLocationAccessAddress ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.LocationAddress;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 40};

A.3.45A Location Access Hours

The A Location Access Hours attribute defines the specific hours for each day of the week during which access to the A location is available. Same day may not be repeated in SET OF WeekMask syntax.

aLocationAccessHours ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.LocationAccessHours;

REGISTERED AS {trAttribute 41};

A.3.46Z Location Access Hours

The Z Location Access Hours attribute defines the specific hours for each day of the week during which access to the Z location is available. Same day may not be repeated in SET OF WeekMask syntax.

zLocationAccessHours ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.LocationAccessHours;

REGISTERED AS {trAttribute 42};

A.3.47A Location Access Person

The A Location Access Person attribute enables the manager to specify the details of the person at the A location. Matching for equality means checking personNumber only in the sequence.

aLocationAccessPerson ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.PersonReach;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 43};

A.3.48Z Location Access Person

The Z Location Access Person attribute enables the manager to specify the details of the person at the Z location. Matching for equality means checking personNumber only in the sequence.

zLocationAccessPerson ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.PersonReach;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 44};

A.3.49Maintenance Organization Contact Name

The Maintenance Organization Contact Name attribute describes the company or organization whose responsibility it is to perform maintenance on the "managed object instance". It is the agent who calls the Maintenance

Organization Contact Name and not the manager. Matching for equality means checking personNumber only in the sequence.

maintenanceOrgContactName ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.PersonReach;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 45};

A.3.50 Maintenance Organization Contact Pointer

The Maintenance Organization Contact Pointer attribute describes the company or organization whose responsibility it is to perform maintenance on the “managed object instance”. It is the agent who calls the Maintenance Organization Contact Name and not the manager (the agent is the service dedicated to “trouble management” within the company and centralizes all the trouble tickets for the company [CNE to CME interface]). Note that there is only one maintenance company for a given managed object instance, specified by contract.

maintenanceOrgContactPtr ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.MaintenanceOrgContactPtr;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 46};

A.3.51 Maintenance Organization Contact Time

The Maintenance Organization Contact Time attribute indicates the time at which the maintenance organization was contacted by the agent and requested to repair the trouble. Matching for ordering is only applicable to GeneralizedTime.

maintenanceOrgContactTime ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.MaintenanceOrgContactTime;

MATCHES FOR

ORDERING;

REGISTERED AS {trAttribute 47};

A.3.52 Maintenance of Service Charge

The Maintenance of Service Charge attribute indicates, once determined, whether the customer will be charged for repairs performed on the service.

maintServiceCharge ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.MaintServiceCharge;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 48};

A.3.53 Managed Object Access From Time

The Managed Object Access From Time attribute identifies the beginning of the time frame during which the service personnel can have access to the managed object. Access restrictions within this time frame can be

specified through the Managed Object Access Hours attribute. The managed object access time frame is service affecting by nature.

The relationship among Managed Object Access From Time, Managed Object Access To Time, and Managed Object Access Hours is explained via an example in Annex D.

managedObjectAccessFromTime ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.ManagedObjectAccessFromTime;

MATCHES FOR

ORDERING;

REGISTERED AS {trAttribute 49};

A.3.54 Managed Object Access Hours

The Managed Object Access Hours attribute defines the specific hours for each day of the week during which access to the managed object is available. This attribute further delimits the time frame defined by the attributes Managed Object Access From Time and Managed Object Access To Time by specifying the access availability intervals for each day of the week. Same day may not be repeated in “SET OF WeekMask” syntax.

The relationship among Managed Object Access From Time, Managed Object Access To Time, and Managed Object Access Hours is explained via an example in Annex D.

managedObjectAccessHours ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.ManagedObjectAccessHours;

REGISTERED AS {trAttribute 50};

A.3.55 Managed Object Access To Time

The Managed Object Access To Time attribute identifies the end of the time frame during which the service personnel can have access to the managed object. Access restrictions within this time frame can be specified through the Managed Object Access Hours attribute. The managed object access time frame is service affecting by nature.

The relationship among Managed Object Access From Time, Managed Object Access To Time, and Managed Object Access Hours is explained via an example in Annex D.

managedObjectAccessToTime ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.ManagedObjectAccessToTime;

MATCHES FOR

ORDERING;

REGISTERED AS {trAttribute 51};

A.3.56 Managed Object Instance

The Managed Object Instance attribute indicates the CNM Service object class instance or the GNIM telecommunications network resource instance associated with a particular trouble report instance.

managedObjectInstance ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.ManagedObjectInstance;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 52};

A.3.57 Managed Object Instance Alias List

The Managed Object Instance Alias List attribute identifies the managed object on which trouble has been reported by its alias(es). These aliases could be a Service Alias or a GNIM Managed Object alias.

managedObjectInstanceAliasList ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.ManagedObjectInstanceAliasList;

MATCHES FOR

EQUALITY,

SET-INTERSECTION,

SET-COMPARISON;

REGISTERED AS {trAttribute 53};

A.3.58 Manager Contact Person

The Manager Contact Person attribute identifies an individual in the manager's organization who can be contacted regarding the reported trouble. Matching for equality means checking personNumber only in the sequence.

managerContactPerson ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.PersonReach;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 54};

A.3.59 Manager Contact Object Pointer

The Manager Contact Object Pointer attribute points to a Contact object that identifies an individual in the manager's organization who can be contacted regarding the reported trouble.

managerContactObjectPtr ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.ManagerContactObjectPtr;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 55};

A.3.60 Manager Search Key 1

The Manager Search Key 1 attribute (single-valued) enables the manager to filter trouble reports, for example, by account or customerID. The use of GraphicString as a search key may not guarantee the desired results.

managerSearchKey1 ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.ManagerSearchKey;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 56};

A.3.61 Manager Search Key 2

The Manager Search Key 2 attribute (single-valued) enables the manager to filter trouble reports, for example, by account or customerID. The use of GraphicString as a search key may not guarantee the desired results.

```
managerSearchKey2 ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.ManagerSearchKey;
   MATCHES FOR
      EQUALITY;
REGISTERED AS {trAttribute 57};
```

A.3.62 Manager Search Key 3

The Manager Search Key 3 attribute (single-valued) enables the manager to filter trouble reports, for example, by account or customerID. The use of GraphicString as a search key may not guarantee the desired results.

```
managerSearchKey3 ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.ManagerSearchKey;
   MATCHES FOR
      EQUALITY;
REGISTERED AS {trAttribute 58};
```

A.3.63 Manager Search Key List

The Manager Search Key List attribute is used to filter and scope trouble reports, for example, by account or customerID. Use of a GraphicString as a search may not guarantee the desired results.

```
managerSearchKeyList ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.ManagerSearchKeyList;
   MATCHES FOR
      EQUALITY,
      SET-INTERSECTION,
      SET-COMPARISON;
REGISTERED AS {trAttribute 59};
```

A.3.64 Operational State

This attribute is imported from ISO/IEC 10164-2.

A.3.65 Outage Duration

The Outage Duration attribute, once determined, indicates the amount of time between the Trouble Report clearing time and the Trouble Report received time, excluding any times for delayed maintenance or any times the service could not be accessed by the service provider for repair.

```
outageDuration ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.OutageDuration;
REGISTERED AS {trAttribute 60};
```

A.3.66 Perceived Trouble Severity

The Perceived Trouble Severity attribute allows the manager to indicate the effect of the trouble on the managed object being reported.

```
perceivedTroubleSeverity ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.PerceivedTroubleSeverity;
   MATCHES FOR
      EQUALITY;
REGISTERED AS {trAttribute 61};
```

A.3.67 Preferred Priority

The Preferred Priority attribute defines the urgency with which the manager requires resolution of the problem.

```
preferredPriority ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.PreferredPriority;
   MATCHES FOR
      EQUALITY;
REGISTERED AS {trAttribute 62};
```

A.3.68 Received Time

The Received Time attribute indicates the date and time when a trouble report was entered.

```
receivedTime ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.ReceivedTime;
   MATCHES FOR
      ORDERING;
REGISTERED AS {trAttribute 63};
```

A.3.69 Related Trouble Report List

The Related Trouble Report List attribute identifies other associated trouble reports.

```
relatedTroubleReportList ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.RelatedTroubleReportList;
   MATCHES FOR
      EQUALITY,
      SET-INTERSECTION,
      SET-COMPARISON;
REGISTERED AS {trAttribute 64};
```

A.3.70 Repair Activity Identifier

The Repair Activity Identifier attribute is the distinguishing attribute of the Repair Activity managed object class.

```
repairActivityID ATTRIBUTE
   WITH ATTRIBUTE SYNTAX GNMTA.RepairActivityID;
```

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 65};

A.3.71 Repair Activity List

The Repair Activity List attribute contains parameters and text describing the specific repair functions performed, who performed them, and when they were performed. This attribute is intended to provide supporting details of repair activities for the purpose of tracking repair activity. Support of this optional attribute is determined by the policies of the administration performing the repair activities.

repairActivityList ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.RepairActivityList;

REGISTERED AS {trAttribute 66};

A.3.72 Repeat Report

The Repeat Report attribute code value indicates whether there has been a provisioning/installation or a trouble activity on the managed object in the recent past (established by local administrative procedures), e.g., within the last 30 days.

repeatReport ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.RepeatReport;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 67};

A.3.73 Responsible Person Name

The Responsible Person Name attribute indicates the person who has the overall responsibility for solving the problem indicated by the trouble report. He or she may not be the person who performs the repair activities, but is the one who is responsible for the trouble resolution process, which includes the tracking of the problem, the isolation of the problem, and the correction of the problem. Matching for equality means checking personNumber only in the sequence.

responsiblePersonName ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.PersonReach;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 68};

A.3.74 Responsible Person Pointer

The Responsible Person Pointer attribute indicates the person who has the overall responsibility for solving the problem indicated by the trouble report. He or she may not be the person who performs the repair activities, but is the one who is responsible for the trouble resolution process, which includes the tracking of the problem, the isolation of the problem, and the correction of the problem.

responsiblePersonPtr ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.ResponsiblePersonPtr;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 69};

A.3.75 Restored Time

The Restored Time attribute indicates when the trouble was cleared. The precise definition of cleared is outside the scope of this standard. Matches for ordering only applies to GeneralizedTime.

restoredTime ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.RestoredTime;

MATCHES FOR

ORDERING;

REGISTERED AS {trAttribute 70};

A.3.76 Service Alias List

The Service Alias List attribute identifies a CNM Service object instance by commonly used telecommunications terminology (e.g., telephone number, special services number). Administrations may specify additional structure for this graphic string.

serviceAliasList ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.ServiceAliasList;

MATCHES FOR

EQUALITY,

SET-INTERSECTION,

SET-COMPARISON;

REGISTERED AS {trAttribute 71};

A.3.77 Service Description

The Service Description attribute explains a particular instance of the CNM Service object in text format.

serviceDescription ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.ServiceDescription;

REGISTERED AS {trAttribute 72};

A.3.78 Service Location List

The Service Location List attribute identifies the locations where a service is used. Because there may be several locations on a particular service (e.g., a multipoint private line), it is multivalued.

serviceLocationList ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.ServiceLocationList;

REGISTERED AS {trAttribute 73};

A.3.79 Service Profile Description

The Service Profile Description attribute explains a particular instance of the Service Profile object in text format.

serviceProfileDescription ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.ServiceProfileDescription;

REGISTERED AS {trAttribute 74};

A.3.80 Service Profile Identifier

The Service Profile Identifier attribute is the distinguishing attribute of the Service Profile managed object class.

serviceProfileID ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.ServiceProfileID;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 75};

A.3.81 Service Profile Object Pointer

The Service Profile Object Pointer attribute points to an instance of the Service Profile object class.

serviceProfileObjectPtr ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.ServiceProfileObjectPtr;

MATCHES FOR EQUALITY;

REGISTERED AS {trAttribute 76};

A.3.82 Status Of System

This attribute indicates the status of a subsystem of manager or agent.

statusOfSystem ATTRIBUTE
WITH ATTRIBUTE SYNTAX GNMTA.StatusOfSystem;
MATCHES FOR EQUALITY;
REGISTERED AS {trAttribute 96};

A.3.83 Suspect Object List

This optional attribute indicates GNIM managed object instance(s) that may be the underlying cause of the trouble.

suspectObjectList ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.SuspectObjectList;

MATCHES FOR

EQUALITY,

SET-INTERSECTION,

SET-COMPARISON;

REGISTERED AS {trAttribute 77};

A.3.84System Monitor Id

Identifier for system monitor object instance representing status of agent or manager subsytem.

systemMonitorId ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.SystemMonitorId;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 97};

A.3.85Trouble Clearance Person

The Trouble Clearance Person attribute identifies an individual in the manager's organization who last modified either of the following two attributes:

- Cancel Requested By Manager
- CloseOut Verification

Matching for equallty means checking personNumber only in the sequence.

troubleClearancePerson ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.PersonReach;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 78};

A.3.86Trouble Detection Time

The Trouble Detection Time attribute indicates the time at which the trouble was detected. This may be different from the time at which the trouble report was created. Matching for ordering is only applicable to GeneralizedTime.

troubleDetectionTime ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.TroubleDetectionTime;

MATCHES FOR

ORDERING;

REGISTERED AS {trAttribute 79};

A.3.87Trouble Found

The Trouble Found attribute specifies an enumerated code value, which identifies the problem resolved. This field will be copied into the Trouble History information.

troubleFound ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.TroubleFound;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 80};

A.3.88 Trouble Location

The Trouble Location attribute indicates where the trouble is. This information could not be known at the time when the trouble report is created.

troubleLocation ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.TroubleLocation;

REGISTERED AS {trAttribute 81};

A.3.89 Trouble Report Constrained To Single Value Attribute ID List

The Trouble Report Constrained To Single Value Attribute ID List attribute specifies the trouble report set-valued attributes that are constrained by the agent to a single value.

tRConstrainedToSingleValueAttrIDList ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.AttributeIDList;

REGISTERED AS {trAttribute 82};

A.3.90 Trouble Report Format Object Pointer

The Trouble Report Format Object Pointer attribute indicates which instance of the Trouble Report Format Definition object class will be used for trouble reports for a particular CNM Service or GNIM managed object.

troubleReportFormatObjectPtr ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.TroubleReportFormatObjectPtr;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 83};

A.3.91 Trouble Report Format Identifier

The Trouble Report Format Identifier attribute is the distinguishing attribute of the Trouble Report Format Definition object class. It specifies the “format” of a trouble report.

tRFormatID ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.TRFormatID;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 84};

A.3.92 Trouble Report Identifier

The Trouble Report Identifier is the distinguishing attribute of the Trouble Report managed object class. It is assigned by the service provider at the time the trouble report is entered. The Trouble Report ID may include information that has been defined by the Trouble Report Number List attribute and/or the Service Alias List attribute.

troubleReportID ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.NamingString;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 85};

A.3.93Trouble Report Must Be Present Attribute ID List

The Trouble Report Must Be Present Attribute ID List attribute specifies the list of attributes in conditional packages in the Trouble Report object class (and its subclasses) that “must be present” in a particular instance of a trouble report according to a particular Trouble Report Format Definition.

tRMustBePresentAttrIDList ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.AttributeIDList;

REGISTERED AS {trAttribute 86};

A.3.94Trouble Report May Be Present Attribute ID List

The Trouble Report May Be Present Attribute ID List attribute specifies the list of attributes in conditional packages in the Trouble Report object class (and its subclasses) that “may be present” in a particular instance of a trouble report according to a particular Trouble Report Format Definition.

tRMayBePresentAttrIDList ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.AttributeIDList;

REGISTERED AS {trAttribute 87};

A.3.95Trouble Report Number List

The Trouble Report Number List attribute represents a list of internal trouble report alias identifiers presently being used to identify the trouble report within each of the agent’s internal systems working on a particular manager’s trouble. The trouble report number used by each internal agent system is not necessarily unique within the agent environment. However, by combining the trouble report number with an identifier for the agent’s internal system, a unique trouble report alias identifier can be constructed. These trouble report alias identifiers are needed by the manager when trouble resolution requires telephone conversations between the manager and a person working at a particular internal agent system (i.e., the trouble report object instances DN may not be available to uniquely identify the trouble report).

troubleReportNumberList ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.TroubleReportNumberList;

MATCHES FOR

EQUALITY,

SET-INTERSECTION,

SET-COMPARISON;

REGISTERED AS {trAttribute 88};

A.3.96Trouble Report State

The Trouble Report State attribute indicates the current state of a trouble report. A trouble report may be in the following states as defined below:

- *Queued* – A trouble report is in a queued state when it has been instantiated but the trouble resolution process has not yet been initiated;
- *Open/Active* – This is the active phase of the trouble report when appropriate actions to resolve the trouble are being carried;
- *Deferred* – This state indicates that the corrective action on the trouble report has been postponed. A deferred report can become Open/Active when specific conditions are met;
- *Cleared* – This state indicates that the trouble has been corrected. If the manager needs to verify that the trouble has been resolved, verification may optionally be awaited by the agent prior to closure of the trouble report;
- *Closed* – This state indicates that the trouble has been corrected, and a Trouble History Notification is emitted. Under specific conditions, a request to cancel a trouble report may be accepted from Queued, Open/Active, or Deferred states;
- *Disabled* – An instance of a trouble report exhibits a disabled value when its information cannot be updated due to local conditions. In the Disabled condition, only read operations can be performed on the trouble report object instance.

See Annex C (Trouble Report State Model) for a diagram detailing allowable state transitions. Annex C also provides examples of mapping trouble report status values into trouble report state values.

troubleReportState ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.TroubleReportState;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 89};

A.3.97 Trouble Report Status

The Trouble Report Status attribute indicates the current status of an active trouble report.

troubleReportStatus ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.TroubleReportStatus;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 90};

A.3.98 Trouble Report Status Time

The Trouble Report Status Time attribute identifies the last time at which the troubleReportStaus attribute was known to be changed.

troubleReportStatusTime ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.TroubleReportStatusTime;

MATCHES FOR

ORDERING;

REGISTERED AS {trAttribute 91};

A.3.99 Trouble Report Status Window

The trouble Report StatusWindow attribute specifies an interval of time within which a troubleReportProgress notification shall be provided. This interval is of a specified length. Initially, the interval is considered to begin from

the time of the create response and extend for the specified length of time. Subsequently, the interval is considered to begin from the event time of the last troubleReport Progress notification or troubleReportStatus attribute value change notification (whichever is latest), and extend for the specified length of time. Hence, if the troubleReportStatus attribute value changes within the interval, the beginning of the interval is redefined to be the time of the attribute value change event. The length of the interval does not change.

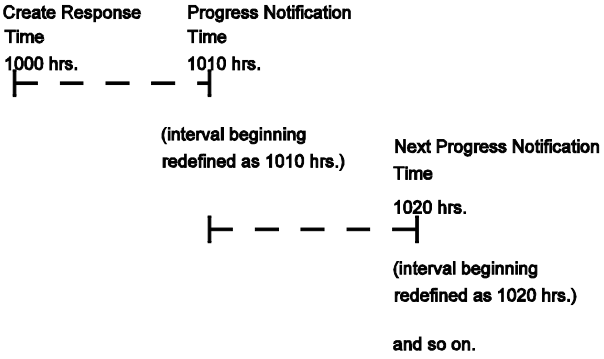
Thus, effectively, the next troubleReportProgress notification is deferred. If on the other hand, within the interval there is no change in the troubleReportStatus attribute value, then a troubleReportProgress notification is provided at the end of the interval, and the beginning of the interval is redefined as that time.

The notification, at the end of the interval, shall include the troubleReportStatus attribute and if the value of the status has not changed since last issued, it shall also include the additionalTrouble-StatusInfo attribute, i.e., a status narrative describing what progress has been made in resolving the trouble report.

If, within the interval, the value (length) of the interval is modified, the changed value becomes effective not later than the next redefinition of the beginning of the interval, that is not later than the next troubleReportStatus attribute value change notification or troubleReportProgress notification (whichever is earlier).

On a verification request (after the Trouble Report Status attribute value is changed to "clearedAwaitingCustVerification"), the progress notification will be stopped. The beginning of the interval will be redefined if that is desired.

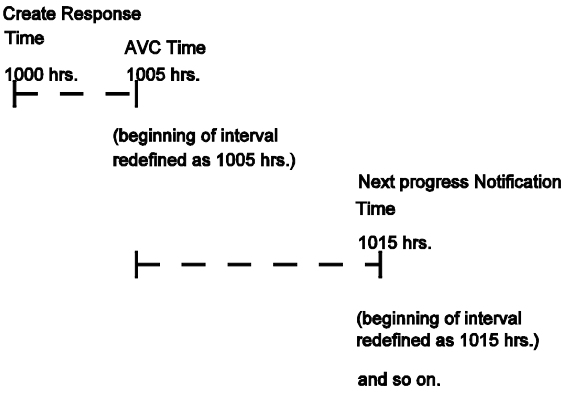
Example 1.
Value of Interval = 10 min.s .
No troubleReportStatus attribute value change takes place.



Example 2.

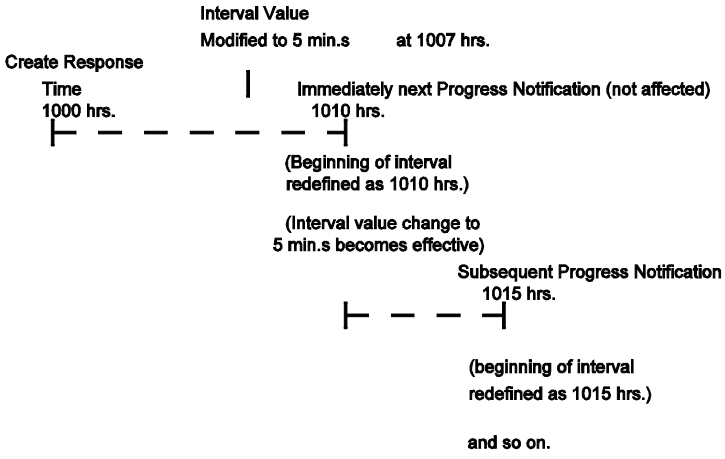
Interval Value = 10 min.s.

troubleReportStatus attribute value changes within the interval.



Example 3

Initial value of interval = 10 min.s.
Interval Value is modified within the interval.



NOTE 1 – The value of the interval is not modified as a result of redefining the beginning of the interval. An interval timer may be required to reflect the beginning and the end of an interval. This timer is not visible across the interface.

NOTE 2 – Although it is allowed to have “zero” value for any element of the sequence which indicates the overall value of the interval, not all elements of the sequence can be “zero” for this attribute, and a minimum value is implementation-dependent.

troubleReportStatusWindow ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.TroubleReportStatusWindow;

REGISTERED AS {trAttribute 92};

A.3.100 Trouble Type

The Trouble Type attribute identifies the category of trouble that is being reported on a CNM Service or GNIM managed object.

troubleType ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.TroubleType;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 93};

A.3.101 TSP Priority

The Telecommunication Service Priority (TSP) Priority attribute conveys TSP codes if applicable between the manager and the agent.

tspPriority ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.TspPriority;

MATCHES FOR

EQUALITY;

REGISTERED AS {trAttribute 94};

A.3.102 Customer Info Attribute

The customerInfo attribute identifies the data values necessary to further detail customer service information associated with the Managed Object Instance. The Manager will provide this information to the Agent to assist in the trouble resolution process. This attribute will be included in the telecommunicationsTroubleReportR1 Managed Object Class.

The attribute will provide customer information such as the PIC (Primary Interexchange Carrier), LPIC (Local Primary Intra-LATA Carrier), LRN (Location Routing Number), and OCN (Operating Company Number). The PIC and/or LPIC information is provided by the Manager when the trouble indicates possible switch translation errors. Troubles being reported by the Manager on a ported-out circuit (telephone number) require the OCN and/or LRN to further assist the Agent in troubles that are normally related to translation information associated with the LRN.

The customerInfo attribute will alleviate the need to provide the above information in the additionalTroubleInformationList as a string that has to be parsed by the Agent.

customerInfo ATTRIBUTE

WITH ATTRIBUTE SYNTAX GNMTA.CustomerInfo;

MATCHES FOR EQUALITY,

REGISTERED AS {trAttribute 98};

A.4 Error messages

A.4.1 Trouble Report Already Exists

troubleReportAlreadyExists PARAMETER

CONTEXT SPECIFIC-ERROR;

WITH SYNTAX GNMTA.TroubleReportAlreadyExists;

BEHAVIOUR troubleReportAlreadyExistsBehaviour BEHAVIOUR

DEFINED AS "This error is applicable only where

administrations restrict the number of

trouble reports per managed object. In such cases, the manager may use the additionalTroubleInfo attribute to include information on the new trouble. The error message returns the instance of the object, optionally the managed object class and the instance of the trouble report on which a trouble already exists.";;

REGISTERED AS {trParameter 1};

A.4.2 Fall Back Reporting

fallBackReporting PARAMETER

CONTEXT SPECIFIC-ERROR;

WITH SYNTAX GNMTA.FallBackReporting;

BEHAVIOUR callBackReportingBehaviour BEHAVIOUR

DEFINED AS "A trouble report object will not be

created (although the agent may accept the trouble report and process it manually). Fallback trouble reporting is defined to be outside the scope of the information model. Since a trouble report object will not exist, none of the other services normally associated with the Trouble Report object class are supported for fallback reporting. This error will be returned in the following two cases=

- Service pre-designated by agent to receive fallback reporting
- Agent partially failed or temporarily unavailable for receiving trouble

reports.";;

REGISTERED AS {trParameter 2};

A.4.3 Can Not Close

canNotClose PARAMETER

CONTEXT SPECIFIC-ERROR;

WITH SYNTAX GNMTA.CanNotClose;

BEHAVIOUR canNotCloseBehaviour BEHAVIOUR

DEFINED AS "This error message is sent to the manager

when the trouble report can not be closed by the agent because it is already cleared.";;

REGISTERED AS {trParameter 3};

A.4.4 Trouble Report Must Be Present Attribute Missing

tRMustBePresentAttributeMissing PARAMETER

CONTEXT SPECIFIC-ERROR;

WITH SYNTAX GNMTA.TRMustBePresentAttributeMissing;

BEHAVIOUR tRMustBePresentAttributeMissingBehaviour BEHAVIOUR

DEFINED AS "This error message is sent to the manager

by the agent when the manager fails to provide all required attributes identified in the attribute tRMustBePresentAttrID List,
the error message contains the attributeIDs of the missing attributes.";;

REGISTERED AS {trParameter 4};

A.4.5 Cannot Verify or Deny at This Time

cannotVerifyOrDenyAtThisTime PARAMETER

CONTEXT SPECIFIC-ERROR;

WITH SYNTAX GNMTA.CannotVerifyOrDenyAtThisTime;

BEHAVIOUR cannotVerifyOrDenyAtThisTimeBehaviour BEHAVIOUR

DEFINED AS "If the manager changes the value of the
CloseOut Verification attribute before the
Trouble Report Status value is
'clearedAwaitingCustVerification', the
agent system may optionally respond with
this error.";;

REGISTERED AS {trParameter 5};

A.4.6 Trouble Report Change Denied

troubleReportChangeDenied PARAMETER

CONTEXT SPECIFIC-ERROR;

WITH SYNTAX GNMTA.TroubleReportChangeDenied;

BEHAVIOUR troubleReportChangeDeniedBehaviour BEHAVIOUR

DEFINED AS "This error message is sent to the
manager when the manager attempts to
change a trouble report which is not in
an appropriate state to accept the
change.";;

REGISTERED AS {trParameter 6};

A.5 Extensibility Rules

As described in ISO 8824-1 Amendment 1 on extensibility rules, the productions that are of extensible types are to be indicated by including an ellipsis (...) in their type descriptions.

The productions of the following types will be indicated as having extensible types:

- ENUMERATED
- named BIT STRING
- SEQUENCE
- SET
- CHOICE

Under the rules of extensibility, new enumerations (for ENUMERATED type), new bit name assignments (for named BIT STRING type), new members (for SET type and SEQUENCE type), and new choices (for CHOICE type) may be added in the future versions of this standard.

In any interface implementation if any of the above (enumerations, bit name assignments, named numbers, etc.) are not recognized in a response to a request, a RORJ-U/ReturnResultProblem-“mistypedResult” will be issued. If an error response is not recognized in a returned error, then RORJ-U/ReturnErrorProblem-“mistypedParameter” will be issued.

A.6 Supporting Productions

For the productions shown in this subclause that have values of type INTEGER, BIT STRING, or ENUMERATED (e.g., troubleType, troubleFound codes), not all values have to be supported in all implementations. The subsets that will be supported will be determined by implementation agreements.

GNMTA {iso(1) member-body(2) usa(840) ansi-t1-227-1992(10015) trGNM(0) gnmta(0)}

DEFINITIONS IMPLICIT TAGS::=BEGIN

EXPORTS

everything;

IMPORTS

Time24

FROM Attribute-ASN1Module {joint-iso-ccitt ms(9) smi(3) part2(2) asn1Module(2) 1}

NameType, ObjectList

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

AttributeId, ObjectClass, ObjectInstance

FROM CMIP-1 {joint-iso-ccitt(2) ms(9) cmip(1) modules(0) protocol(3)};

– Only the Distinguished Name and Local

– Distinguished Name forms are supported.

trMObjectClass OBJECT IDENTIFIER::={iso(1) member-body(2) usa(840) ansi-t1-227-1992(10015) trGNM(0) objectClass(3)}

trPackage OBJECT IDENTIFIER::={iso(1) member-body(2) usa(840) ansi-t1-227-1992 (10015) trGNM(0) package(4)}

trParameter OBJECT IDENTIFIER::={iso(1) member-body(2) usa(840) ansi-t1-227-1992 (10015) trGNM(0) parameter(5)}

```
trNameBinding OBJECT IDENTIFIER::={iso(1) member-body(2) usa(840) ansi-t1-227-1992 (10015) trGNM(0) nameBinding(6)}

trAttribute OBJECT IDENTIFIER::={iso(1) member-body(2) usa(840) ansi-t1-227-1992 (10015) trGNM(0) attribute(7)}

activityDurationactivityDurationInitial ActivityDuration::= { } -- EMPTY SET OF

activityInfoActivityInfoDefault GraphicString::=""

afterHrsRepairAuthAfterHrsRepairAuthDefault BOOLEAN::=FALSE

authorizationAuthorizationDefault AuthorizationList::={{provided,'000000000'B}} -- (DEFAULT SEQUENCE)

closeOutNarrCloseOutNarrInitial GraphicString::=""

closeOutVerificationCloseOutVerificationInitial CloseOutVerification::=noAction

handOffCenterHandOffCenterInitial GraphicString::=""

handOffLocationHandOffLocationInitial GraphicString::=""

handOffPersonNameHandOffPersonNameInitial PersonReach::= {number"", name""}

handOffTimeHandOffTimeInitial NULL::=NULL

maintenanceOrgContactNameMaintenanceOrgContactNameInitial PersonReach::= {number"", name""}

maintenanceOrgContactTimeMaintenanceOrgContactTimeInitial NULL::=NULL

maintServiceChargeMaintServiceChargeInitial BOOLEAN::=FALSE

outageDurationOutageDurationInitial NULL::=NULL

repairActivityListRepairActivityListInitial RepairActivityList::={} -- EMPTY SET OF

restoredTimeRestoredTimeInitial NULL::=NULL

troubleClearancePersonTroubleClearancePersonDefault PersonReach::= {number"", name""}

troubleDetectionTimeTroubleDetectionTimeDefault NULL::=NULL

troubleFoundTroubleFoundInitial TroubleFound::= number:0

troubleReportCancelRequestedByManagerInitial BOOLEAN::=FALSE

AccountContactList::=SET OF PersonReach

AccountName::=GraphicString(SIZE(0..64))

ActivityCode::=CHOICE{
        number INTEGER{
                approved        (0),
                assign          (1),
                cancel          (2),
                clear           (3),
                close           (4),
                defer           (5),
                dispatch        (6),
                refer           (7),
                release         (8),
                re-open         (9),
```

```
                repair                  (10),
                test                    (11),
                transfer                (12)
                },
        identifier OBJECT IDENTIFIER
    }
ActivityDuration::=SET OF SEQUENCE{
        duration                        [0]TimeInterval,
        billable                        [1]BOOLEAN DEFAULT TRUE,
        type                            [2]ActivityType OPTIONAL,
        ...
        }
ActivityInfo::=GraphicString(SIZE(0..256))
ActivityPerson::=PersonReach
ActivityType::= BIT STRING {
                after-hours-repair  (0),
                standby         (1),
                after-hours-standby (2),
                test            (3),
                manager-initiated-test  (4),
                dispatch        (5),
                no-access       (6),
                delayed-maintenance  (7),
                release         (8),
                deregulatedWork  (9),
                authorize-to-work  (10),
                …
                }
AdditionalText::=GraphicString(SIZE(0..256))
AdditionalTroubleInfoList::=SET OF GraphicString
AdditionalTroubleStatusInfo::=SET OF GraphicString(SIZE(0..256))
AfterHrsRepairAuth::=BOOLEAN
AgentContactObjectPtr::=CHOICE{NULL,
                ObjectInstance,
                ...
                }
AlarmRecordPtrList::=SET OF ObjectInstance
AlternateManagerContactObjectPtr::=CHOICE{NULL,
```

```
                ObjectInstance,
                ...
                }
ApplicableManagedObjectClassList::=SET OF ObjectClass
ApplicableManagedObjectInstanceList::=SET OF ObjectInstance
AttributeIDList::=SET OF AttributeId
AuthorizationList::=SET OF SEQUENCE{
        state           RequestState,
        type            ActivityType,
        authTime        AuthorizationTime OPTIONAL,
        authPerson      PersonReach OPTIONAL,
        ...
        }
AuthorizationTime::=GeneralizedTime
CallBackInfoList::=SET OF CHOICE{--conformance to standard
        -- does not require support for all choices
        escalation                  [0]PersonReach,
        beforeAutoTest              [1]PersonReach,
        afterCleared                [2]PersonReach,
        ...
        }
CalledNumber::=GraphicString (SIZE(0..64))
CancelRequestedByManager::=BOOLEAN
CanNotClose::=INTEGER{
        alreadyCleared                  (0),

        }
CannotVerifyOrDenyAtThisTime::=NULL
ChangeDeniedReason::=ENUMERATED{
        waitingVerificationOfClosure    (1),
        troubleReportAlreadyClosed      (2),
        activityAuthorizationPending    (3),
        ...
        }
CloseOutNarr::=GraphicString (SIZE(0..256))
CloseOutVerification::=ENUMERATED{
        noAction                        (0),
        verified                        (1),
```

```
        denied                              (2),
        deniedActivityDurationDisputed      (3),
        deniedCloseOutNarrDisputed          (4),
        ...
        }
CommitmentTime::=CHOICE{
        onsiteTime                  [0] GeneralizedTime,
        clearedTime                 [1] GeneralizedTime,
        ...
        }
ContactObjectPtrList::=SET OF ObjectInstance
CustomerWorkCenter::=GraphicString(SIZE(0..64))
CustTroubleTickNum::=GraphicString(SIZE(0..64))

CustomerInfo::= SEQUENCE{
        pIC             [0] VisibleString (SIZE (0..64)) OPTIONAL,
        lPIC            [1] VisibleString (SIZE (0..64)) OPTIONAL,
        lRN             [2] VisibleString (SIZE (0..64)) OPTIONAL,
        oCN             [3] VisibleString (SIZE (0..64)) OPTIONAL,
        …
        }
Dialog::=GraphicString(SIZE(0..640))
EntryTime::=GeneralizedTime
EscalationList::=SET OF SEQUENCE{
        state                       RequestState,
        escTime                     EscalationTime,
        -- supplied by agent; the manager should supply the current time at the time of request. (The agent
        should update it).
        requestPerson               [0]PersonReach,
        level                       [1]OrgLevel OPTIONAL,
        escPerson                   [2]PersonReach OPTIONAL,
        ...
        }
EscalationTime::=GeneralizedTime
EstimatedTimeOfRestoral ::= GeneralizedTime
-- For working systems, use equivalent of January 1, 1900, 12:01 AM
-- For disabled systems for which no estimate is available, use equivalent of January 1, 3000, 12:01 AM
FallBackReporting::=CHOICE {NULL,
```

```
                GraphicString,
                ...
                }
HandOffCenter::=GraphicString(SIZE(0..64))
HandOffLocation::=GraphicString(SIZE(0..64))
HandOffPersonPtr::=CHOICE{NULL,
                ObjectInstance,
                ...
                }
HandOffTime::=CHOICE{NULL,
                GeneralizedTime,
                ...
                }
InitiatingMode::=INTEGER{
                --Integer values are to be registered in the standard.
                managerDirect                   (0),
                managerIndirect                 (1),
                agentOriginated                 (2),
                managerIndirectEMail            (4),
                managerIndirectFax              (5),
                managerIndirectPersonal         (6),
                managerIndirectPhone            (7)
                }
LastUpdateTime::=GeneralizedTime
LocationAddress::=SEQUENCE{PremisesName,
                 PremisesAddress,
                ...
                }
LocationAccessHours::=SET OF WeekMask
-- cannot repeat the same day
MaintenanceOrgContactPtr::=CHOICE{NULL,
                ObjectInstance,
                ...
                }
MaintenanceOrgContactTime::=CHOICE{NULL,
                GeneralizedTime,
                ...
                }
```

```
MaintServiceCharge::=BOOLEAN
ManagedObjectAccessFromTime::=GeneralizedTime
ManagedObjectAccessHours::=SET OF WeekMask
-- cannot repeat the same day
ManagedObjectAccessToTime::=StopTime
ManagedObjectInstance::=ObjectInstance
ManagedObjectInstanceAliasList::=SET OF GraphicString (SIZE(0..256))
ManagerContactObjectPtr::=CHOICE{NULL,
                ObjectInstance,
                ...
                }
ManagerSearchKey::=CHOICE{
                ManagerSearchString,
                ObjectInstance,
                ...
                }
ManagerSearchKeyList::=SET OF CHOICE{
                ManagerSearchString,
                ObjectInstance,
                ...
                }
ManagerSearchString::=GraphicString(SIZE(0..64))
NamingString::=GraphicString(SIZE(0..32))
OrgLevel::=INTEGER{
                no-escalation           (0),
                first-level             (1),
                second-level            (2),
                third-level             (3),
                fourth-level            (4),
                fifth-level             (5),
                sixth-level             (6)
                }
OutageDuration::=CHOICE{NULL,
                TimeInterval,
                ...
                }
PerceivedTroubleSeverity::=CHOICE{
        number INTEGER{
```

```
                -- administrations may provide additional
                -- values and/or restrict the values supported
                outOfService                (0),
                backInService               (1),
                serviceImpairment           (2),
                nonServiceAffectingTrouble  (3)
                },
            identifier OBJECT IDENTIFIER,
            ...
        }
PersonEmail::=GraphicString(SIZE(0..64))
PersonFax::=GraphicString(SIZE(0..64))
PersonLocation::=PremisesAddress
PersonName::=GraphicString(SIZE(0..64))
PersonNumber::=GraphicString(SIZE(0..64))
PersonPhone::=GraphicString(SIZE(0..64))
PersonRespon::=GraphicString(SIZE(0..64))
PersonReach::= SEQUENCE{
                number                      [0] PersonNumber DEFAULT "",
                name                        [1] PersonName DEFAULT "",
                phone                       [2] PersonPhone OPTIONAL,
                loc                         [3] PersonLocation OPTIONAL,
                email                       [4] PersonEmail OPTIONAL,
                fax                         [5] PersonFax OPTIONAL,
                respon                      [6] PersonRespon OPTIONAL,
                pager                       [7] PersonPhone OPTIONAL,
                …
                }
PreferredPriority::=ENUMERATED{
                undefined                   (0),
                minor                       (1),
                major                       (2),
                serious                     (3),
                ...
                }
PremisesName::=GraphicString(SIZE(0..64))
PremisesAddress::=SEQUENCE{
                civicAddress    GraphicString(SIZE(0..64)),
```

```
        city        GraphicString(SIZE(0..64)),
        state       GraphicString(SIZE(0..64)),
        zip         GraphicString(SIZE(0..64)),
        ...
        }
ReceivedTime::=GeneralizedTime
RelatedTroubleReportList::=SET OF ObjectInstance
RepairActivityID::=INTEGER
RepairActivityList::=SET OF SEQUENCE{
        entryTime                   GeneralizedTime,
        activityInfo                GraphicString,
        activityPerson              PersonReach OPTIONAL,
        activityCode                ActivityCode OPTIONAL,
        …
        }
RepeatReport::=ENUMERATED{
        unspecified                 (0),
        recentInstallation          (1),
        repeat                      (2),
        bothInstallationAndRepeat   (3),
        chronic                     (4),
        bothInstallationAndChronic  (5),
        ...
        }
RequestState::=ENUMERATED{
        requested                   (1),
        provided                    (2),
        denied                      (3),
        ...
        }
ResponsiblePersonPtr::=CHOICE{NULL,
         ObjectInstance,
         ...
         }
RestoredTime::=CHOICE{NULL,
         GeneralizedTime,
         ...
         }
```

```
ServiceAliasList::=SET OF GraphicString(SIZE(0..64))
ServiceDescription::=GraphicString(SIZE(0..256))
ServiceLocationList::=SET OF SEQUENCE{
                PremisesName,
                PremisesAddress,
                ...
                }
ServiceProfileID::=GraphicString(SIZE(0..64))
ServiceProfileDescription::=GraphicString(SIZE(0..256))
ServiceProfileObjectPtr::=ObjectInstance
StatusOfSystem::=INTEGER {
                 enabled              (0),
                 disabledMaintenance  (1),
                 disabledFailed       (2),
                 degraded             (3)
                 }
StopTime::=CHOICE{specific GeneralizedTime,
                continual NULL,
                ...
                }
SuspectObject ::= SEQUENCE {
                 suspectObjectClass     OBJECT IDENTIFIER,
                 suspectObjectInstance  ObjectInstance,
                 failureProbability     INTEGER OPTIONAL, -- in the range 1-100
                 ...
                 }
SuspectObjectList::=SET OF SuspectObject
SystemMonitorId::=INTEGER
TimeInterval ::= SEQUENCE {
                 day    [0]     INTEGER (0..31) DEFAULT 0,
                 hour   [1]     INTEGER (0..23) DEFAULT 0,
                 minute     [2]     INTEGER (0..59) DEFAULT 0,
                 second     [3]     INTEGER (0..59) DEFAULT 0,
                 msec   [4]     INTEGER (0..999) DEFAULT 0
                 }
TRFormatID::=INTEGER
TRMustBePresentAttributeMissing::=AttributeIDList
TroubleDetectionTime::=CHOICE{NULL,
```

```
                GeneralizedTime,
                ...
                }
TroubleFound::=CHOICE{
                number INTEGER{
                -- Integer values are to be registered in the
                -- standard. Administrations may restrict
                -- the values to be used.
                pending                         (0),
                cameClear                       (1),
                centralOffice                   (2),
                switchTrouble                   (3),
                customerProvidedEquipment       (4),
                facility                        (5),
                centralOfficeFacility           (6),
                iCfacility                      (7),
                interexchangeCarrier            (8),
                information                     (9),
                nonplanClassified               (10),
                nonplanClassifiedIC             (11),
                nonplanClassifiedEA             (12),
                noTroubleFound                  (13),
                station                         (14),
                stationProductData              (15),
                stationProductTerminal          (16),
                stationProductVideo             (17),
                stationProductVoice             (18),
                stationWiring                   (19),
                otherStationEquipment           (20),
                foundOKStation                  (21),
                servingBureau                   (22),
                testOK                          (23),
                publicServicesCoinSet           (24),
                customerOperatingInstructions   (25),
                testedOKVerifiedOK              (26),
                coFacilityTestedFoundOK         (27),
                outsideFacilityTestedFoundOK    (28),
                referredOutToOtherDept          (29),
```

```
        protectiveConnectingArrang      (30),
        cpeCustomerResponsibility       (31),
        preService                      (32),
        preServiceIC                    (33),
        preServiceEA                    (34),
        serviceNode                     (35),
        data                            (36),
        customerReferredToVendor        (37),
        exchangeAccess                  (38),
        international                   (39),
        otherProvidedAccess             (40),
        existingReport                  (41),
        cancelExclude                   (42),
        paBX                            (43),
        outsideWire                     (44),
        outsideTerminals                (45),
        outsidePlantEquipment           (46),
        outsidePlantFiberOptic          (47),
        ousidePlantOther                (48),
        coEquipmentOther                (49),
        coEquipmentFrames               (50),
        coConcentrator                  (51),
        receiverOffHook                 (52),
        cpeAuthorized                   (53),
        cpeTelcoMaintained              (54),
        independentCompany              (55),
        cpeCalledNumber                 (56),
        assigningProvisioning           (57),
        interServiceCenter              (58),
        referredOut                     (59),
        network                         (60)
        },
        identifier OBJECT IDENTIFIER
        }
TroubleLocation::=CHOICE{
        locationAddress                 [0] LocationAddress,
        locationPtr                     [1] ObjectInstance,
        ...
```

```
        }
TroubleReportAlreadyExists::=SEQUENCE{
        managedObjectClass          ObjectClass OPTIONAL,
        managedObjectInstance       ObjectInstance,
        troubleReportInstance       ObjectInstance,
        ...
        }
TroubleReportChangeDenied::=SEQUENCE {
        changeDeniedReason          ChangeDeniedReason,
        managedObjectClass          ObjectClass OPTIONAL,
        managedObjectInstance       ObjectInstance,
        troubleReportInstance       ObjectInstance,
        ...
        }
TroubleReportFormatObjectPtr::=ObjectInstance
TroubleReportNumberList::=SET OF GraphicString(SIZE(0..64))
TroubleReportState::=INTEGER {
        queued                  (0),
        openActive              (1),
        deferred                (2),
        cleared                 (3),
        closed                  (4),
        disabled                (5)
        } (0..255)
TroubleReportStatus::=CHOICE{
        number INTEGER{
        -- Integer values are to be registered in the
        -- standard. Administrations may restrict
        -- the values to be used.
        screening                   (1),
        testing                     (2),
        dispatchedIn                (3),
        dispatchedOut               (4),
        preassignedOut              (5),
        bulkDispatchedOut           (6),
        startRepair                 (7),
        pendingTest                 (8),
        pendingDispatch             (9),
```

```
        requestRepair                   (10),
        referMtceCenter                 (11),
        referVendor                     (12),
        noAccessOther                   (13),
        startNoAccess                   (14),
        stopNoAccess                    (15),
        startDelayedMtce                (16),
        stopDelayedMtce                 (17),
        troubleEscalated                (18),
        craftDispatched                 (19),
        temporaryOK                     (20),
        cableFailure                    (21),
        originatingEquipFailure         (22),
        backOrder                       (23),
        clearedCustNotAdvised           (24),
        clearedCustAdvised              (25),
        clearedAwaltingCustVerification (26),
        closedOut                       (27),
        closedOutByCustReq              (28),
        closedOutCustVerified           (29),
        closedOutCustDenied             (30),
        canceledPendingWorkInProgress       (31),
        canceledPendingTestCompletion       (32),
        canceledPendingDispatchCompl        (33),
        techOnSite                      (34),
        techLeftSite                    (35)
        },
        identifier OBJECT IDENTIFIER
        }
TroubleReportStatusTime::=GeneralizedTime
TroubleReportStatusWindow::=TimeInterval
        -- Although TimeInterval is allowed to have ‘zero’ value for any element of the
        -- sequence, not all
        -- elements of TimeInterval can be ‘zero’ for this attribute, and a minimum value is
        -- implementation-
        -- dependent.
TroubleType::= INTEGER
    -- Integer values are defined in ATIS-0300094.2008.
```

```
TspPriority::=GraphicString(SIZE(2))
        -- Assignment of these codes for repair shall be specified by TSP standard
        --ANS ATIS-0300211.2012, Information Interchange – Structure and Coded Representation of
        National Security and Emergency Preparedness (NS/EP) Telecommunications Service Priority
        (TSP) codes for the North American Telecommunications System only. The leftmost character of
        the TSP code
        -- represents the service provisioning priority and the rightmost character represents
        -- the restoration priority for the service. The leftmost character is restricted to be an
        -- E, 1, 2, 3, 4, 5, or 0. The rightmost character is restricted to be a 1, 2, 3, 4, 5, or
        -- 0. The meaning of these characters is as follows:
        --      0 - none
        --      1 - national security leadership
        --      2 - national security posture and U.S. population attack
        --      3 - public health, safety and maintenance of law and order
        --      4 - public welfare and maintenance of national economic posture 1
        --      5 - public welfare and maintenance of national economic posture 2
WeekMask::=SEQUENCE {
   daysOfWeek BIT STRING {
        sunday      (0),
        monday      (1),
        tuesday     (2),
        wednesday   (3),
        thursday    (4),
        friday      (5),
        saturday    (6),
        ...
        }
   DEFAULT '1111111'B,
   intervalsOfDay SET OF SEQUENCE {
        intervalStart Time24,
        intervalEnd  Time24,
        ...
        }
   DEFAULT{ { {0,0},{23,59}}},
   ...
   }
   -- If no time is allowed on a week day, the bit for that week day shall be set to zero, irrespective of what is in the
   -- IntervalsOfDay.
```

-- In some implementations, 23:59 may indicate the last minute of the week day.

END -- GNMTA

A.7 Object Class Inheritance

Some of the object classes in clause 6 were defined as “subclasses of another class”. A subclass is said to inherit the properties (attributes, events, actions, and behaviors) of the superclass from which it is refined. In other words, the subclass is an extension of the superclass in the sense that it has all properties of the superclass plus some additional ones.

The concept of inheritance is fundamental to the creation of a generic model. Generic objects (such as Service) can be subclassed to define new objects (such as CNM Service) based upon the customer network management service modeling concept. Subclassing can also be used to cover specific requirements, such as service-specific needs (e.g., ISDN CNM Service) or service-provider-specific needs (e.g., Company ABC CNM Service).

The succession of inheritance relationships (from generic to specific) can be represented by an inheritance hierarchy. Higher levels in this hierarchy contain more generic object classes that are refined to more specific ones located at lower levels of the hierarchy.

The inheritance hierarchy (implicitly defined in clause 7) of this document is illustrated in figure A.1. The arcs represent the subclassing relationships and the vertices represent the managed objects defined in this standard.

In this extension to the generic network information model, the inheritance hierarchy is rather “flat”. This is because priority was given to obtaining a generic model that would ensure a consistent approach for modeling communication networks. In future issues of this standard, it is expected that more classes will be specialized. It is also expected that these classes will be refined when considering different technologies.

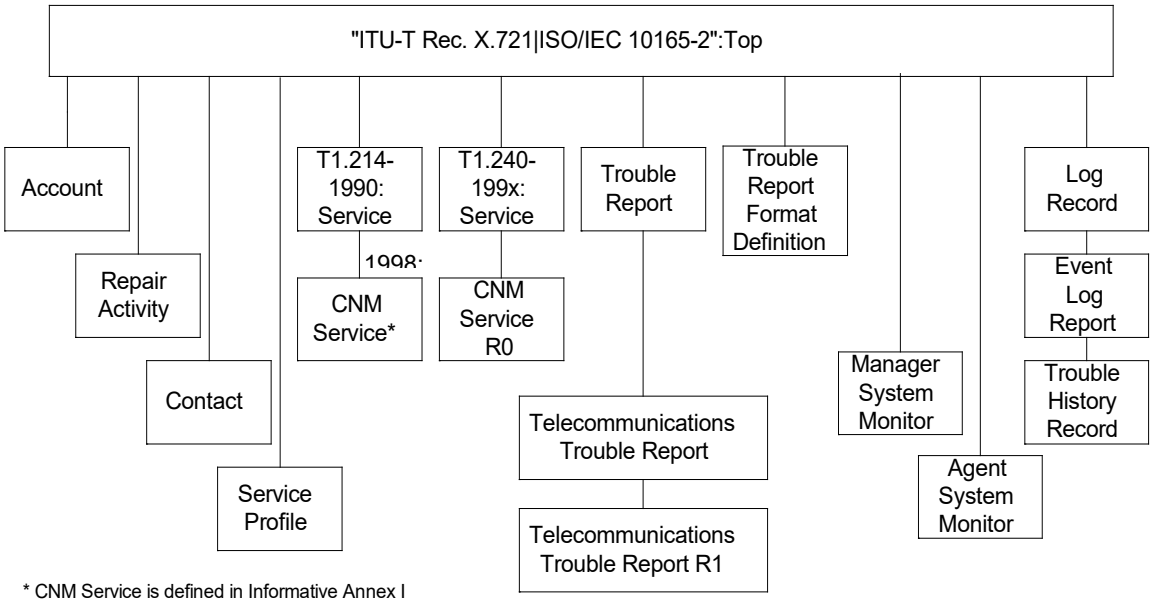


Figure A.1 – Inheritance hierarchy

CNM Service is defined in Informative Annex H.

A.8 Conformance & Compliance

A.8.1 Conformance

This standard, by itself, does not place any conformance requirements on the implementation of open systems. It is intended to provide an extension to the generic network information model to support the implementation of standardized messages for OS-OS CNM interfaces across jurisdictional boundaries. It is therefore intended to be used with the associated message specification standards (e.g., ATIS-0300228.2006 for Trouble Administration messages). The complete set of conformance requirements for standardized OS-OS messages is given in the conformance clause of those standards (e.g., clause 10 of ATIS-0300228.2006).

A.8.2 Compliance

Where other telecommunications standards import any of the definitions contained in this standard into attribute type or object class definitions via the referencing mechanism defined in ISO/IEC 10165-4, the attribute type or object class definitions that reference these definitions shall comply with the behavioral and syntactic aspects of those types to the extent specified in this standard.

Where other standards refine any of the object class definitions contained in this standard via the refinement and referencing mechanisms defined in ISO/IEC 10165-4, the object classes that refine these definitions shall comply with the behavioral and syntactic aspects of the superclass to the extent defined in this standard.

Annex B: Name Bindings

(normative)

B.1 General

Name bindings can be used to specify the form of subtrees of the MIT.

Figure B.1 illustrates name bindings between classes of objects defined in this standard. Object classes are drawn as rectangles. The arcs between them describe subordinate/superior relationships that can be used for naming.

B.2 Specification of Name Bindings

The following specifies the name bindings described in the previous clause:

account-account NAME BINDING

SUBORDINATE OBJECT CLASS account;

NAMED BY SUPERIOR OBJECT CLASS account;

WITH ATTRIBUTE accountName;

REGISTERED AS {trNameBinding 1};

account-network NAME BINDING

SUBORDINATE OBJECT CLASS account;

NAMED BY SUPERIOR OBJECT CLASS “Rec. M.3100 : 1992”:network AND SUBCLASSES;

WITH ATTRIBUTE accountName;

REGISTERED AS {trNameBinding 2};

agentSystemMonitor-network NAME BINDING

SUBORDINATE OBJECT CLASS agentSystemMonitor;

NAMED BY SUPERIOR OBJECT CLASS “Rec. M.3100 : 1992”:network AND SUBCLASSES;

WITH ATTRIBUTE systemMonitorId;

REGISTERED AS {trNameBinding 20};

contact-account NAME BINDING

SUBORDINATE OBJECT CLASS “OP1 Library Vol. R1:1992”: contact;

NAMED BY SUPERIOR OBJECT CLASS account;

WITH ATTRIBUTE “OP1 Library Vol. R1: 1992”: contactID;

REGISTERED AS {trNameBinding 3};

contact-network NAME BINDING

SUBORDINATE OBJECT CLASS “OP1 Library Vol. R1:1992”: contact;

NAMED BY SUPERIOR OBJECT CLASS “Rec. M.3100 : 1992”:network AND SUBCLASSES;

WITH ATTRIBUTE “OP1 Library Vol. R1: 1992”: contactID;

REGISTERED AS {trNameBinding 4};

contact-serviceR1 NAME BINDING

SUBORDINATE OBJECT CLASS “OP1 Library Vol. R1: 1992”:contact;

NAMED BY SUPERIOR OBJECT CLASS “T1.240-1998”:service AND SUBCLASSES;

```
    WITH ATTRIBUTE  “OP1 Library Vol. R1: 1992”:contactID;
REGISTERED AS {trNameBinding 23};
BINDING
    contact;
    1994”:service;
    contactID;
5};
contact-system NAME BINDING
    SUBORDINATE OBJECT CLASS “OP1 Library Vol. R1: 1992”: contact;
    NAMED BY SUPERIOR OBJECT CLASS “Rec. X.721|ISO/IEC 10165-2 : 1992”:system;
    WITH ATTRIBUTE “OP1 Library Vol. R1: 1992”: contactID;
REGISTERED AS {trNameBinding 6};
cnmServiceR0-account NAME BINDING
    SUBORDINATE OBJECT CLASS cnmServiceR0;
    NAMED BY SUPERIOR OBJECT CLASS account;
    WITH ATTRIBUTE “T1.240-1998”:serviceID;
REGISTERED AS {trNameBinding 30};
BINDING
    cnmService;
    account;
    WITH ATTRIBUTE “T1.240:1998”:serviceID;
7};
eventForwardingDiscriminator-account NAME BINDING
    SUBORDINATE OBJECT CLASS “Rec. X.734|ISO/IEC 10164 5:1993”:eventForwardingDiscriminator;
    NAMED BY SUPERIOR OBJECT CLASS account;
    WITH ATTRIBUTE discriminatorId;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {trNameBinding 19};
managerSystemMonitor-account NAME BINDING
    SUBORDINATE OBJECT CLASS managerSystemMonitor;
    NAMED BY SUPERIOR OBJECT CLASS account;
    WITH ATTRIBUTE systemMonitorId;
REGISTERED AS {trNameBinding 21};
telecommunicationsTroubleReport-account NAME BINDING
    SUBORDINATE OBJECT CLASS telecommunicationsTroubleReport;
```

```
    NAMED BY SUPERIOR OBJECT CLASS account;
    WITH ATTRIBUTE troubleReportID;
    CREATE
        WITH-AUTOMATIC-INSTANCE-NAMING,
        WITH-REFERENCE-OBJECT
        troubleReportAlreadyExists
        fallBackReporting
        tRMustBePresentAttributeMissing;
REGISTERED AS {trNameBinding 8};
telecommunicationsTroubleReport-cnmServiceR0 NAME BINDING
    SUBORDINATE OBJECT CLASS telecommunicationsTroubleReport;
    NAMED BY SUPERIOR OBJECT CLASS cnmServiceR0;
    WITH ATTRIBUTE troubleReportID;
    CREATE
        WITH-AUTOMATIC-INSTANCE-NAMING,
        WITH-REFERENCE-OBJECT
        troubleReportAlreadyExists
        fallBackReporting
        tRMustBePresentAttributeMissing;
REGISTERED AS {trNameBinding 29};
BINDING
    SUBORDINATE OBJECT CLASS telecommunicationsTroubleReport;
    cnmService;
    WITH ATTRIBUTE troubleReportID;
    CREATE
        WITH-AUTOMATIC-INSTANCE-NAMING,
        WITH-REFERENCE-OBJECT
        troubleReportAlreadyExists
        fallBackReporting
        tRMustBePresentAttributeMissing;
REGISTERED AS {trNameBinding 9};
telecommunicationsTroubleReport-network NAME BINDING
    SUBORDINATE OBJECT CLASS telecommunicationsTroubleReport;
    NAMED BY SUPERIOR OBJECT CLASS “Rec. M.3100 : 1992”:network AND SUBCLASSES;
    WITH ATTRIBUTE troubleReportID;
    CREATE
        WITH-AUTOMATIC-INSTANCE-NAMING,
        WITH-REFERENCE-OBJECT
        troubleReportAlreadyExists
```

```
        fallBackReporting
        tRMustBePresentAttributeMissing;
REGISTERED AS {trNameBinding 10}telecommunicationsTroubleReport-system NAME BINDING
    SUBORDINATE OBJECT CLASS telecommunicationsTroubleReport;
    NAMED BY SUPERIOR OBJECT CLASS “Rec. X.721|ISO/IEC 10165-2 : 1992”:system;
    WITH ATTRIBUTE troubleReportID;
    CREATE
        WITH-AUTOMATIC-INSTANCE-NAMING,
        WITH-REFERENCE-OBJECT
        troubleReportAlreadyExists
        fallBackReporting
        tRMustBePresentAttributeMissing;
REGISTERED AS {trNameBinding 11};
telecommunicationsTroubleReportR1-account NAME BINDING
    SUBORDINATE OBJECT CLASS telecommunicationsTroubleReportR1;
    NAMED BY SUPERIOR OBJECT CLASS account;
    WITH ATTRIBUTE troubleReportID;
    CREATE
        WITH-AUTOMATIC-INSTANCE-NAMING,
        WITH-REFERENCE-OBJECT
        troubleReportAlreadyExists
        fallBackReporting
        tRMustBePresentAttributeMissing;
REGISTERED AS {trNameBinding 24};
telecommunicationsTroubleReportR1-cnmServiceR0 NAME BINDING
    SUBORDINATE OBJECT CLASS telecommunicationsTroubleReportR1;
    NAMED BY SUPERIOR OBJECT CLASS cnmServiceR0;
    WITH ATTRIBUTE troubleReportID;
    CREATE
        WITH-AUTOMATIC-INSTANCE-NAMING,
        WITH-REFERENCE-OBJECT
        troubleReportAlreadyExists
        fallBackReporting
        tRMustBePresentAttributeMissing;
REGISTERED AS {trNameBinding 25};
telecommunicationsTroubleReportR1-network NAME BINDING
    SUBORDINATE OBJECT CLASS telecommunicationsTroubleReport;
    NAMED BY SUPERIOR OBJECT CLASS “Rec. M.3100 : 1992”:network AND SUBCLASSES;
```

```
    WITH ATTRIBUTE troubleReportID;
    CREATE
      WITH-AUTOMATIC-INSTANCE-NAMING,
      WITH-REFERENCE-OBJECT
      troubleReportAlreadyExists
      fallBackReporting
      tRMustBePresentAttributeMissing;
REGISTERED AS {trNameBinding 26};
telecommunicationsTroubleReportR1-system NAME BINDING
    SUBORDINATE OBJECT CLASS telecommunicationsTroubleReportR1;
    NAMED BY SUPERIOR OBJECT CLASS “Rec. X.721|ISO/IEC 10165-2 : 1992”:system;
    WITH ATTRIBUTE troubleReportID;
    CREATE
      WITH-AUTOMATIC-INSTANCE-NAMING,
      WITH-REFERENCE-OBJECT
      troubleReportAlreadyExists
      fallBackReporting
      tRMustBePresentAttributeMissing;
REGISTERED AS {trNameBinding 27};
repairActivity-telecommunicationsTroubleReportR1 NAME BINDING
    SUBORDINATE OBJECT CLASS repairActivity;
    NAMED BY SUPERIOR OBJECT CLASS telecommunicationsTroubleReportR1;
    WITH ATTRIBUTE repairActivityID;
REGISTERED AS {trNameBinding 28};

log-account NAME BINDING
    SUBORDINATE OBJECT CLASS “Rec. X.721|ISO/IEC 10165-2 : 1992”:log;
    NAMED BY SUPERIOR OBJECT CLASS account;
    WITH ATTRIBUTE “Rec. X.721|ISO/IEC 10165-2 : 1992”:logId;
REGISTERED AS {trNameBinding 12};
log-cnmServiceR0 NAME BINDING
    SUBORDINATE OBJECT CLASS “Rec. X.721|ISO/IEC 10165-2:1992”:log;
    NAMED BY SUPERIOR OBJECT CLASS cnmServiceR0;
    WITH ATTRIBUTE “Rec. X.721|ISO/IEC 10165-2:1992”:logId;
REGISTERED AS {trNameBinding 31};
BINDING
    1992”:log;
    cnmService;
```

```
    1992":logId;
13};
repairActivity-telecommunicationsTroubleReport NAME BINDING
    SUBORDINATE OBJECT CLASS repairActivity;
    NAMED BY SUPERIOR OBJECT CLASS telecommunicationsTroubleReport;
    WITH ATTRIBUTE repairActivityID;
REGISTERED AS {trNameBinding 14};
serviceProfile-network NAME BINDING
    SUBORDINATE OBJECT CLASS serviceProfile;
    NAMED BY SUPERIOR OBJECT CLASS "Rec. M.3100 : 1992":network AND SUBCLASSES;
    WITH ATTRIBUTE serviceProfileID;
REGISTERED AS {trNameBinding 15};
troubleHistoryRecord-log NAME BINDING
    SUBORDINATE OBJECT CLASS troubleHistoryRecord;
    NAMED BY SUPERIOR OBJECT CLASS "Rec. X.721|ISO/IEC 10165-2 : 1992":log;
    WITH ATTRIBUTE "Rec. X.721|ISO/IEC 10165-2 : 1992":logRecordId;
    DELETE;
REGISTERED AS {trNameBinding 16};
troubleReportFormatDefn-network NAME BINDING
    SUBORDINATE OBJECT CLASS troubleReportFormatDefn;
    NAMED BY SUPERIOR OBJECT CLASS "Rec. M.3100 : 1992":network AND SUBCLASSES;
    WITH ATTRIBUTE tRFormatID;
REGISTERED AS {trNameBinding 17};
troubleReportFormatDefn-system NAME BINDING
    SUBORDINATE OBJECT CLASS troubleReportFormatDefn;
    NAMED BY SUPERIOR OBJECT CLASS "Rec. X.721|ISO/IEC 10165-2 : 1992":system;
    WITH ATTRIBUTE tRFormatID;
REGISTERED AS {trNameBinding 18};
```

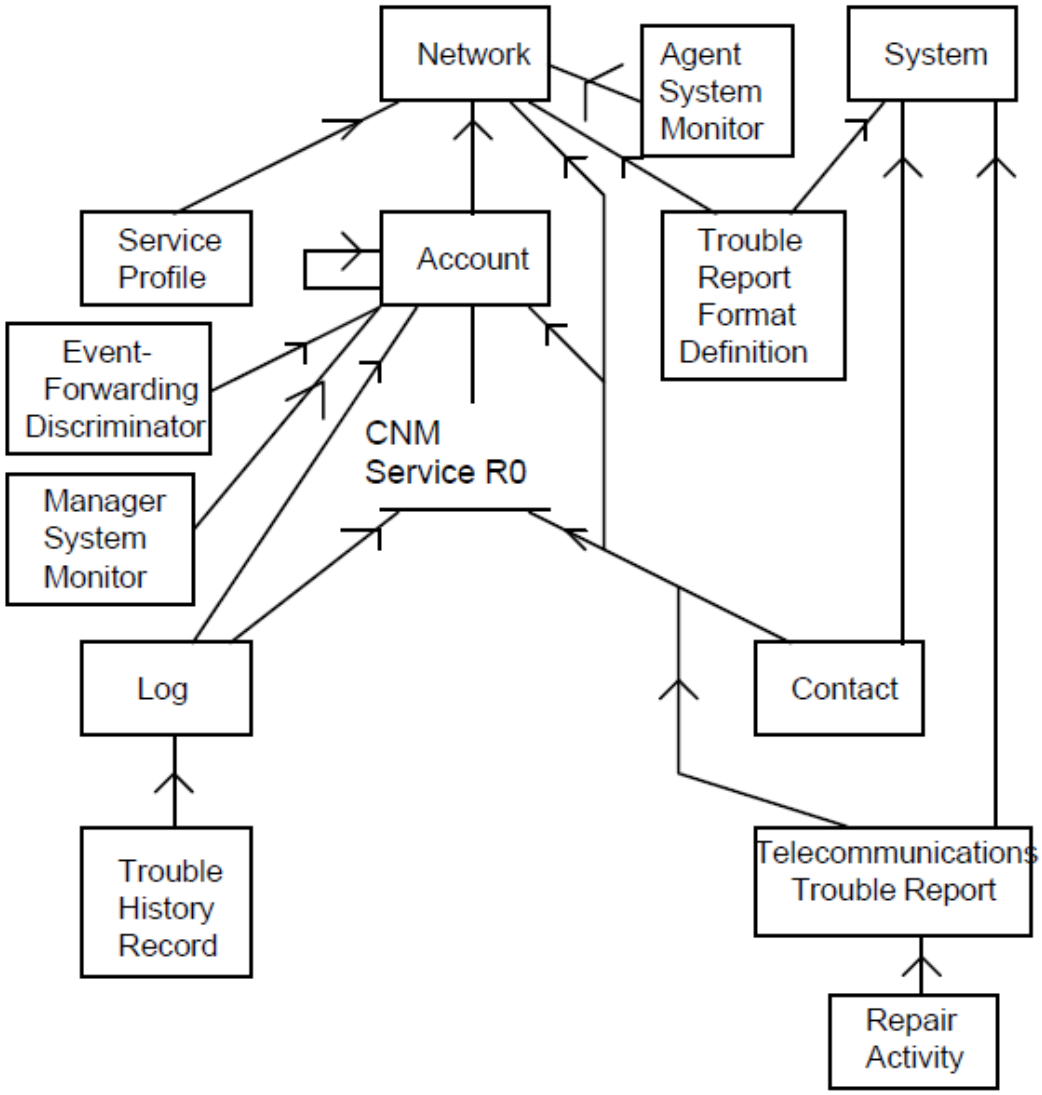


Figure B.1 – Suggested Name Bindings

NOTE – The "CNM Service R0" object (in 7.1.2) in Figure B.1 fully replaces the former cnmService object classes. The "Telecommunications Trouble Report" pertains to instances of both telecommunicationsTroubleReport and telecommunicationsTroubleReportR1 object classes.

Annex C: Objects from Network Management Forum used in the Standard

(informative)

For the convenience of the reader, this annex contains the details of the Contact object defined in ITU-T Recommendation X.790.

contact MANAGED OBJECT CLASS

DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;

CHARACTERIZED BY

contactPkg,

"Rec. M.3100 : 1992":attributeValueChangeNotificationPackage,

"Rec. M.3100 : 1992":createDeleteNotificationsPackage;

CONDITIONAL PACKAGES

contactCompanyPkg PACKAGE

BEHAVIOUR

contactCompanyPkgDefinition BEHAVIOUR

DEFINED AS "This package contains one attribute that specifies the company name that a

specific contact works for or is associated with.";,

contactCompanyPkgBehaviour BEHAVIOUR

DEFINED AS "If the attributeValueChange notification is defined for the managed object class

using this package, this notification is emitted when the contactCompany attribute changes value.";;

ATTRIBUTES

contactCompany

PERMITTED VALUES X790ASN1Module.GraphicString64 GET-REPLACE;

REGISTERED AS {x790Package 5};

PRESENT IF ! an instance supports it !,

contactFunctionPkg PACKAGE

BEHAVIOUR

contactFunctionPkgDefinition BEHAVIOUR

DEFINED AS "This package contains one attribute that provides information about the work

function performed by the contact person.";,

contactFunctionPkgBehaviour BEHAVIOUR

DEFINED AS "If the attributeValueChange notification is defined for the managed object class

using this package, this notification is emitted when the contactFunction attribute changes value.";;

ATTRIBUTES contactFunction GET-REPLACE;

REGISTERED AS {x790Package 6};

PRESENT IF ! an instance supports it !,

contactNamesPkg PACKAGE

ATTRIBUTES contactNames GET-REPLACE ADD-REMOVE;

REGISTERED AS {x790Package 7};

PRESENT IF ! an instance supports it !,

contactTypePkg PACKAGE
BEHAVIOUR
contactTypePkgDefinition BEHAVIOUR
DEFINED AS "This package contains one attribute that provides information about the type of contact.";,
contactTypePkgBehaviour BEHAVIOUR
DEFINED AS "If the attributeValueChange notification is defined for the managed object class using this package, this notification is emitted when the contactType attribute changes value.";;
ATTRIBUTES contactType GET-REPLACE;
REGISTERED AS {x790Package 8};
PRESENT IF ! an instance supports it !,
electronicMailAddressPkg PACKAGE
BEHAVIOUR
electronicMailAddressPkgDefinition BEHAVIOUR
DEFINED AS "This package contains one attribute that specifies the electronic mail address associated with an object.";,
electronicMailAddressPkgBehaviour BEHAVIOUR
DEFINED AS "If the attributeValueChange notification is defined for the managed object class using this package, this notification is emitted when the electronicMailAddress attribute changes value.";;ATTRIBUTES
electronicMailAddress
PERMITTED VALUES X790ASN1Module.ElectronicMailAddressRange
GET-REPLACE ADD-REMOVE;
REGISTERED AS {x790Package 9};
PRESENT IF ! an instance supports it !,
facsimileTelephoneNumberListPkg PACKAGE
BEHAVIOUR
facsimileTelephoneNumberListPkgDefinition BEHAVIOUR
DEFINED AS "This package contains one attribute that specifies telephone numbers for facsimile terminals associated with an object.";,
facsimileTelephoneNumberListPkgBehaviour BEHAVIOUR
DEFINED AS "If the attributeValueChange notification is defined for the managed object class using this package, this notification is emitted when the facsimileTelephoneNumberList attribute changes value.";;
ATTRIBUTES
facsimileTelephoneNumberList
PERMITTED VALUES X790ASN1Module.TelephoneNumberListRange
GET-REPLACE ADD-REMOVE;
REGISTERED AS {x790Package 10};
PRESENT IF ! an instance supports it !,
"Rec. M.3100 : 1992":locationNamePackage
PRESENT IF ! an instance supports it !,

typeTextPkg PACKAGE
ATTRIBUTES typeText GET-REPLACE;
REGISTERED AS {x790Package 11};
PRESENT IF ! an instance supports it !,
"Rec. M.3100 : 1992":userLabelPackage
PRESENT IF ! an instance supports it !;
REGISTERED AS {x790ObjectClass 3};
contactPkg PACKAGE
BEHAVIOUR
contactPkgDefinition BEHAVIOUR
DEFINED AS "The contact managed object class refers to a person or organization having responsibility for one or more managed object instances.";,
contactPkgBehaviour BEHAVIOUR
DEFINED AS "Attributes whose values are names of other managed object instances (e.g. locationPointer) must have names of managed objects which actually exist. The attributeValueChange notification is emitted when any of the following attributes change in value: contactDetails and telephoneNumberList. All attributeValueChange notifications shall include the Attribute Identifier List parameter. Conditions under which an attributeValueChange notification is emitted are stated in the behaviour of the appropriate package or attribute. In the absence of such a statement in the behaviour, the attribute does not cause an attribute ValueChange notification to be emitted.
A value for the contactID attribute can only be provided when the object is created. Furthermore, once the object is created, the value of contactID may not be modified (i.e. the instance cannot be renamed). The contact object is created locally by the agent. .";,
commonCreationBehaviour BEHAVIOUR
DEFINED AS "Unless otherwise specified, all attributes can be set by an M-CREATE.";;
ATTRIBUTES
contactDetails
PERMITTED VALUES X790ASN1Module.GraphicString128
GET-REPLACE,
contactID
PERMITTED VALUES X790ASN1Module.SimpleNameRange
GET,
telephoneNumberList
PERMITTED VALUES X790ASN1Module.TelephoneNumberListRange
GET-REPLACE ADD-REMOVE;
REGISTERED AS {x790Package 12};

1992”:top;
CHARACTERIZED BY contactPackage;
CONDITIONAL PACKAGES

```
    contactTypePkg PRESENT IF “an instance supports it”,
    contactCompanyPkg PRESENT IF “an instance supports.it”,
    contactFunctionPkg PRESENT IF “an instance supports it”,
    electronicMailAddressPkg PRESENT IF “an instance supports it”,
    facsimileTelephoneNumberListPkg PRESENT IF “an instance supports it”,

    and
        present”,
    IF
        and
        present”,

forumSecurityNotifier MANAGED OBJECT CLASS
 DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
 CHARACTERIZED BY forumSecurityNotifierPkg;
 CONDITIONAL PACKAGES
  "Rec. M.3100 : 1992":createDeleteNotificationsPackage PRESENT IF
   ! an instance supports it!,
  integrityViolationNotificationPkg PRESENT IF
   ! an instance supports it!,
  operationalViolationNotificationPkg PRESENT IF
   ! an instance supports it!,
  physicalViolationNotificationPkg PRESENT IF
   ! an instance supports it!,
  securityServiceOrMechanismViolationNotificationPkg PRESENT IF
   ! an instance supports it!,
  serviceReportNotificationPkg PRESENT IF
   ! an instance supports it!,
  timeDomainViolationNotificationPkg PRESENT IF
   ! an instance supports it!,
  usageReportNotificationPkg PRESENT IF
   ! an instance supports it!;
 REGISTERED AS {forum-objectClass 49};
forumSecurityNotifierPkg PACKAGE
 BEHAVIOUR forumSecurityNotifierPkgDefinition,
forumSecurityNotifierPkgBehaviour,
  commonCreationBehaviour;
```

```
 ATTRIBUTES forumSecurityNotifierId GET;
 ;
forumSecurityNotifierPkgDefinition BEHAVIOUR
 DEFINED AS !
   The forumSecurityNotifier managed object represents those elements of a
   system (e.g., security services and security mechanisms) that are required
   by a security policy to report on events of interest to security.
   The types of event that may be reported are potential security violations.
   The detection of such an event generally requires that a security alarm be
   issued; and other events that are defined by a security policy to be
   relevant to the support of a security audit. The detection of such an event
   generally requires that a security audit trail record be created and stored
   in an audit trail log for subsequent analysis.;
forumSecurityNotifierPkgBehaviour BEHAVIOUR
 DEFINED AS !
   When a security alarm notification is issued the forumSecurityNotifier
   shall identify itself as the securityAlarmDetector, and shall complete the
   serviceUser and serviceProvider parameters of the notification.
   A value for the forumSecurityNotifierId can only be provided when the
   object is created. Furthermore, once the object is created, the value of
   forumSecurityNotifierId may not be modified (i.e., the instance can not be
   renamed).;
FORUM-TYPES-ISGDMO-1 { iso member-body (2) 124 forum (360501) modules(0) types-ISGDMO-1(1)}
DEFINITIONS IMPLICIT TAGS::=BEGIN
        GraphicStringBase ::= GraphicString
        GraphicString64 ::= GraphicStringBase (SIZE(0..64))
        GraphicString64Range ::= GraphicString64
        ElectronicMailAddress ::= SET OF GraphicString -- from FORUM-ASN1-1 { forum-module release1 (11)}
        ContactCompany::=GraphicString
        ContactDetails::=GraphicString128
        ContactFunction::=ENUMERATED{
                customerMaintenanceManager      (0),
                providerMaintenanceManager      (1),
                customerAccountManager          (2),
                fieldServicesManager            (3),
                repairman                       (4),
                tester                          (5),
```

```
        screener                    (6),
        providerAccountManager      (7)
        }
    ContactType::=BIT STRING{
        other                       (0),
        contacts-for-equipment-related-activities   (1),
        contacts-for-location-related-activities    (2),
        contacts-for-circuit-related-activities     (3),
        contacts-for-provider-related-activities    (4),
        contacts-for-service-related-activities     (5),
        contacts-for-facility-related-activities    (6),
        contacts-for-customer-related-activities    (7),
        contacts-for-vendor-related-activities      (8),
        contacts-for-manufacturer-related-activities    (9),
        contacts-for-software-related-activities    (10),
        contacts-for-function-related-activities    (11)
        }
    ElectronicMailAddressRange::=SET SIZE (0..64) OF GraphicString(SIZE(0..64))
    GraphicString128::=GraphicString(SIZE(0..128))
    GraphicString128Range::=GraphicString128
    SimpleNameType::=GraphicString
    TelephoneNumber::=PrintableString (SIZE (0..32))
    TelephoneNumbers::=SET OF TelephoneNumber
    TelephoneNumbersRange::=SET SIZE (0..64) OF TelephoneNumber
END
```

Annex D: Trouble Report State Model

(informative)

Examples of Trouble Report Status attribute values corresponding to Trouble Report state values:

- Queued:

  screening

- Open/Active:

  testing, dispatchedIn, dispatchedOut, startRepair, pendingTest, pendingDispatch, requestRepair, referMtceCenter, referVendor, canceledPendingWorkInProgress, canceledPendingTestCompletion, canceledPendingDispatchCompletion

- Deferred:

  noAccessOther, startNoAccess, stopNoAccess, startDelayedMtce, stopDelayedMtce

- Cleared:

  temporaryOK, clearedCustNotAdvised, clearedCustAdvised, clearedAwaitingCustVerification

- Closed:

  closedOut, closedOutByCustReq, closedOutCustVerified, closedOutCustDenied

It should be noted however, that not all states shown in this Annex may be present in every implementation.

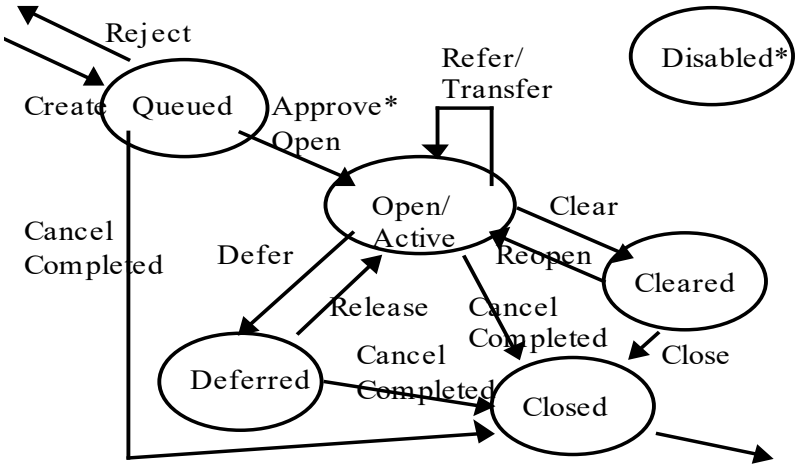


Figure D.1 – Trouble Report state model

* Can be entered from other states due to local conditions.

NOTE – Not all of the events that cause state transitions in this diagram translate to messages across the interface.

Annex E: Relationship between Managed Object Access To, Managed Object Access From, & Managed Object Access Hours Attributes of the Trouble Report

(informative)

This Annex provides an example of how the Managed Object Access To Time, Managed Access From Time, and Managed Object Access Hours attributes are related to one another to define the periods of time during which service personnel can have access to the managed object.

Assume that access to the managed object will be granted beginning on May 15, 1990, and end on July 16, 1990. Additionally, access will be granted only on Tuesday, Saturday, and Sunday. On Tuesday, access is available immediately after midnight until 7:30 am, then from 12:00 noon until 1:00 pm, and again from 5:00 pm until midnight.

This would be expressed in terms of the attribute values as shown in Figures E.1 and E.2:

```
mgdObjAccessFromTime ManagedObjectAccessFromTime ::= “199005150000.0”
mgdObjAccessToTime ManagedObjectAccessToTime ::= specific:”199007160000.0”
mgdObjAccessHours ManagedObjectAccessHours
    ::= {-- No access on mondays, wednesdays, thursdays, fridays ..
     -- Tuesday access intervals..
    { daysOfWeek '0010000'B,
    intervalsOfDay{
        {intervalStart {0,0},intervalEnd (7,30)},
        {intervalStart {12,00}, intervalEnd {13,00}},
        {intervalStart {17,00}, intervalEnd {24,00}} }
    },
    -- full access on WeekEnds..
    { daysOfWeek '1000001'B,
        intervalsofDay{
            {intervalStart {0,0}, intervalEnd {24,00}} }
    }
}
```

Figure E.1 – Managed Object access timings ASN.1 value notation

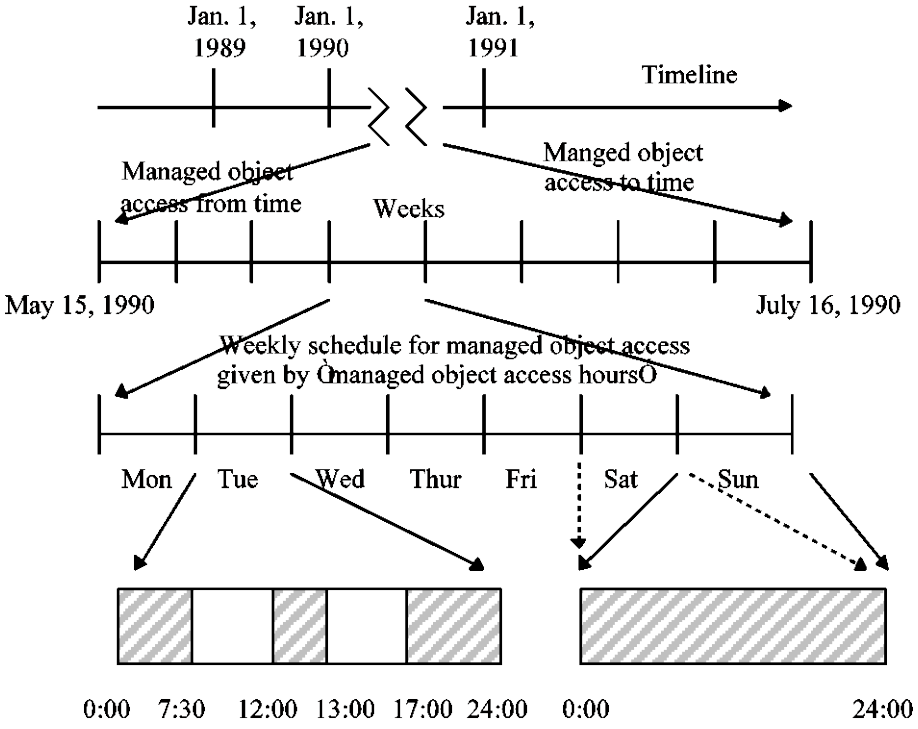


Figure E.2 – Managed Object access timings attributes and their relationships

NOTE – Shaded blocks of time represent Managed Object access hour intervals (i.e., times during which access is allowed). Also, in this example, Managed Object access hours intervals for Saturday and Sunday are identical (all day), and so in the diagram, they are represented by the same time block.

Annex F:Trouble Report Format Examples for a Typical Carrier

(informative)

This Annex provides additional explanation for the trouble report format concept. It shows what the Telecommunications Trouble Report and Trouble Report Format Definition objects look like for a typical exchange carrier with two trouble report formats based on near-term trouble reporting operations system capabilities.

The Trouble Report Format Definition object also provides a means for the evolution of the interface. The optional capabilities defined in the model, but not initially supported in an implementation can be added without requiring redefinition of the interface. An agent can offer a subset of the full range of functionality and still provide a coherent trouble administration environment. As the agent becomes capable of supporting more of the optional features (e.g., conditional packages) of the Telecommunications Trouble Report object, it will be able to advertise this additional capability.

The trouble administration standard allows the agent or service provider to specify the trouble report format for each managed object or managed object instance on which a trouble can be reported. The trouble report format specifies the optional trouble report attributes that are supported for that format. Each trouble report format will be defined by a trouble report format definition object containing the list of optional trouble report attributes that may or shall be present.

In addition, trouble report attributes that are set-valued may be constrained to sets of one member by listing them in the constraint to single member attribute of the trouble report format definition object. The instantiated trouble report object shall also contain attributes specified as mandatory (CHARACTERIZED BY LIST) in the trouble report object.

For this typical exchange carrier, there will be two instances of the Trouble Report Format Definition object (one for each trouble report format). Troubles are only allowed to be reported on CNM Service object instances in this typical exchange carrier. Therefore, for this typical exchange carrier, the trouble report format will be named in each CNM Service object, making it unnecessary to include either the Applicable Managed Object Classes or Applicable Managed Object Instances attribute in the Trouble Report Format Definition object. The “required” optional Telecommunication’s Trouble Report attributes supplied by the manager will be listed in the Trouble Report Must Supply Attribute ID List of the Trouble Report Format Definition object. The “not required” optional Telecommunications Trouble Report attributes, which could be supplied by the manager if it so chooses to, will be listed in the Trouble Report May Supply Attribute ID List of the Trouble Report Format Definition object.

Tables F.1 and F.2 show the Telecommunications Trouble Report attributes supported by each of the two example formats, show which are required (must supply attributes), and show who (manager or agent) will supply them at the time the object is created.

Table F.1 – Mandatory Trouble Report Format Definition objects

Mandatory Trouble Report attributes	Format	Required	Supplier
troubleReportID	1,2	X	Agent
additionalTroubleInfoList	1,2	X*	Manager
managedObjectInstance	1,2	X	Manager
receivedTime	1,2	X	Agent
troubleFound	1,2	X	Agent
troubleReportState	1,2	X	Agent
troubleReportStatus	1,2	X	Agent
troubleReportStatusTime	1,2	X	Agent
troubleType	1,2	X	Manager
*These normally set-valued attributes will be constrained to single-valued operation.			

Table F.2 – Conditional Trouble Report Format Definition objects

Conditional Trouble Report attributes	Format	Required	Supplier
afterHoursRepairAuth	2	X	Manager
callBackInfoLIst	2	X*	Manager
cancelRequestedByManager	1,2	X	Manager
commitmentTime	1	X	Agent
customerWorkCenter	2		Manager
custTroubleTickNum	1,2		Manager
handOffCenter	2	X	Agent
handOffLocation	2	X	Agent
aLocationAccessAddress	2	X	Manager
zLocationAccessAddress	2	X	Manager
aLocationAccessHours	2	X	Manager
zLocationAccessHours	2	X	Manager
maintServiceCharge	2	X	Agent

Conditional Trouble Report attributes	Format	Required	Supplier
managerContactPerson	1,2	X	Manager
outageDuration	2	X	Agent
perceivedTroubleSeverity	2	X	Manager
troubleClearancePerson	1,2	X	Manager
troubleReportFormatObjectPtr	1,2	X	Agent
troubleReportNumberList	1,2	X*	Agent
*These normally set-valued attributes will be constrained to single-valued operation.			

Annex G: Pointer Attribute Relationships between Objects

(informative)

This Annex gives a pictorial representation of the pointer attribute relationships between trouble administration objects defined in this document (see Figure G.1). The “CNM Service” in Figure G.1 pertains to cnmServiceR0 (in 7.1.2) object class. “Telecommunications Trouble Report” pertains to instances of telecommunicationsTroubleReport, telecommunicationsTroubleReportR1, and subclasses.

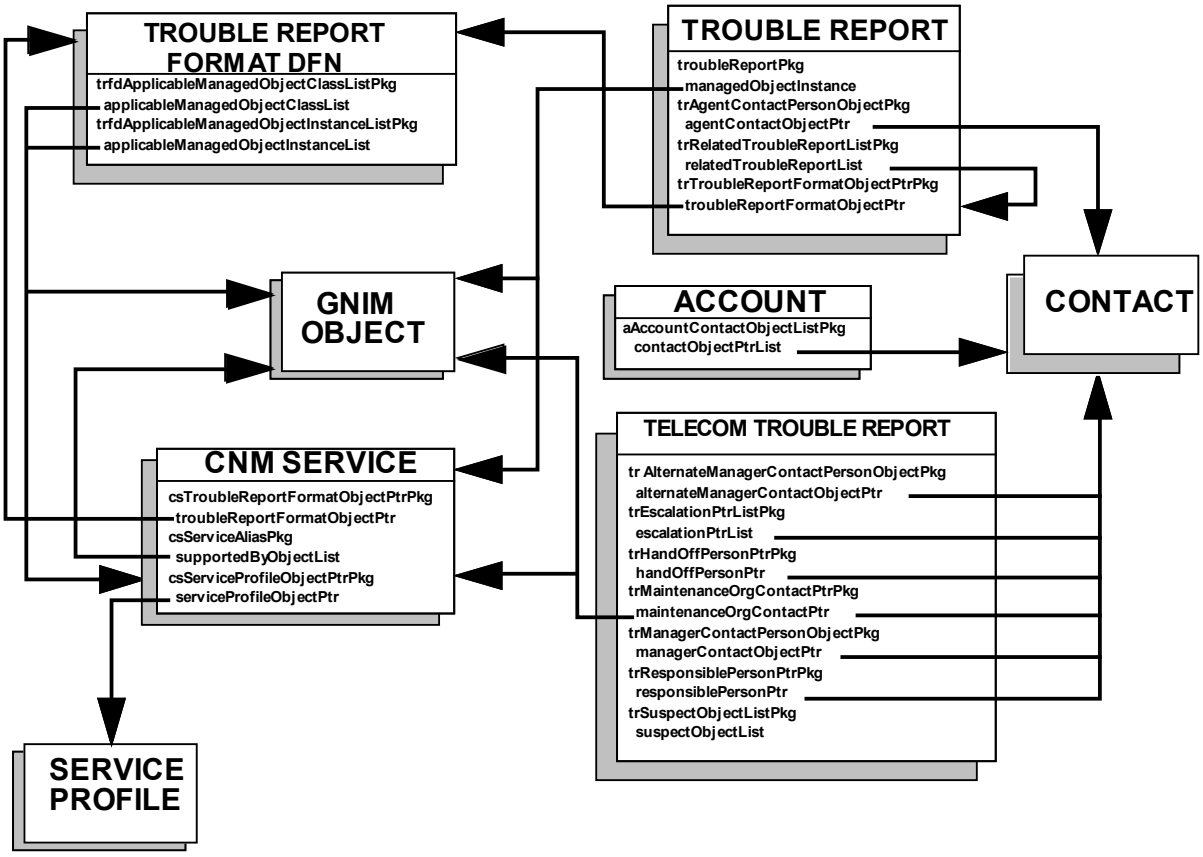


Figure G.1 – Pointer attribute relationships

Annex H: cnmService Managed Object Class

(informative)

H.1 Introduction

This GDMO template below describes the original *cnmService* object class, which was defined in the 1992 version of T1.227. While a new object class, *cnmServiceR0*, has been defined in the normative part of this standard, the original *cnmService* class is preserved here to support existing implementations as well as provide implementators an option to use the old object class in new implementations for consistency purposes.

cnmServiceR0 is subclassed from the T1.240-1998's *service* object class, and *cnmService* object class is subclassed from T1.214-1990's *service* object class. The two *service* object classes (T1.240 and T1.214) have slightly different syntax. Because T1.214-1990 is withdrawn, the *service* object class as well as all related attributes, ASN.1 productions, and name bindings were moved and are being preserved in an informative annex in T1.240-1998 [see also ATIS-0300240.1998(R2007)]. All object identifiers related to the T1.214 *service* object class remain unchanged.

H.2 Object Class Template

```
cnmService MANAGED OBJECT CLASS
   DERIVED FROM "T1.214: 1990": service;
   CHARACTERIZED BY cnmServicePkg PACKAGE
      ATTRIBUTES
         -- 2 mandatory attributes inherited from service:
         -- serviceID GET,
         -- serviceType GET-REPLACE,
         -- some implementations may not support a REPLACE on serviceType
         serviceLocationList GET,
         serviceDescription GET;;;
         -- 1 notification inherited from service:
         -- qualityOfServiceAlarm
         -- may need to be modified in this class
CONDITIONAL PACKAGES
            -- 7 optional attributes inherited from service:
            -- administrativeState GET-REPLACE,
            -- alarmState GET,
            -- currentProblemList GET,
            -- operationalState GET,
            -- supportedServiceNameList
            -- GET-REPLACE-ADD-REMOVE,
            -- supportedByObjectList
            -- GET-REPLACE-ADD-REMOVE,
            -- usageState GET,
```

csServiceAliasPkg PRESENT IF “an instance supports it.”,

csServiceProfileObjectPtrPkg PRESENT IF “an instance supports it.”,

csTroubleReportFormatObjectPtrPkg PRESENT IF “an instance supports it.”;

REGISTERED AS {trMObjectClass 2};

H.3 Name Binding Templates

cnmService-account NAME BINDING

SUBORDINATE OBJECT CLASS cnmService;

NAMED BY SUPERIOR OBJECT CLASS account;

WITH ATTRIBUTE “T1.214:1990”:serviceID;

REGISTERED AS {trNameBinding 7};

telecommunicationsTroubleReport-cnmService NAME BINDING

SUBORDINATE OBJECT CLASS telecommunicationsTroubleReport;

NAMED BY SUPERIOR OBJECT CLASS cnmService;

WITH ATTRIBUTE troubleReportID;

CREATE

WITH-AUTOMATIC-INSTANCE-NAMING,

WITH-REFERENCE-OBJECT

troubleReportAlreadyExists

fallBackReporting

tRMustBePresentAttributeMissing;

REGISTERED AS {trNameBinding 9};

telecommunicationsTroubleReportR1-cnmService NAME BINDING

SUBORDINATE OBJECT CLASS telecommunicationsTroubleReportR1;

NAMED BY SUPERIOR OBJECT CLASS cnmService;

WITH ATTRIBUTE troubleReportID;

CREATE

WITH-AUTOMATIC-INSTANCE-NAMING,

WITH-REFERENCE-OBJECT

troubleReportAlreadyExists

fallBackReporting

tRMustBePresentAttributeMissing;

REGISTERED AS {trNameBinding 22};

log-cnmService NAME BINDING

SUBORDINATE OBJECT CLASS “Rec. X.721|ISO/IEC 10165-2 : 1992”:log;

NAMED BY SUPERIOR OBJECT CLASS cnmService;

WITH ATTRIBUTE “Rec. X.721|ISO/IEC 10165-2 : 1992”:logId;

REGISTERED AS {trNameBinding 13};

contact-service NAME BINDING

SUBORDINATE OBJECT CLASS “OP1 Library Vol. R1: 1992”:contact;

NAMED BY SUPERIOR OBJECT CLASS “T1.214-1990”:service;

WITH ATTRIBUTE “OP1 Library Vol. R1: 1992”:contactID;

REGISTERED AS {trNameBinding 5};

Annex I: CMIP Services for Trouble Management & Accompanying Documentation

(informative)

Note – The text in Annex I was formerly contained in ATIS-0300228, which has since been withdrawn, as CMIP is generally no longer used for Trouble Management.

I.1 Functions of Fault Management

I.1.1 General

Fault Management functions enable the detection, isolation, and correction of abnormal operation of a service. These functions fall into three categories: 1) Alarms; 2) Trouble Administration; and 3) Testing.

Fault Management uses surveillance data collected on a set of services. The gathering and reporting of surveillance data is controlled by a set of generic functions. Fault Management functions for Customer Network Management interfaces include maintaining and examining error logs, reporting error conditions, managing trouble reports, and performing diagnostic tests.

I.1.2 Trouble Administration Functions

The following list describes the trouble administration functions of Fault Management:

a) *enter trouble report*: TMN gives notice to another TMN that a service provided by that TMN is in need of repair.
b) *request trouble report status:* TMN asks for status information on a previously entered trouble report.
c) *request trouble report format*: TMN requests another TMN to provide a template for a trouble report for a particular service or class of services. This allows the originating TMN to know what attributes of a trouble report are considered mandatory or optional by the receiving TMN.
d) *trouble history event*: TMN notifies the TMN that originated the trouble report that it has been closed out, or keeps the close out information in an internal log.
e) *review trouble history*: TMN asks for information about past troubles that it has reported.
f) *add trouble information*: TMN adds information to a trouble report that it has entered.
g) *trouble report status update*: TMN notifies the TMN that originated a trouble report that the status of that trouble report has changed.
h) *trouble report commitment time update*: TMN notifies the TMN that originated a trouble report that the commitment time for that trouble report has changed.
i) *trouble report attribute value change*: TMN notifies the TMN that originated a trouble report that other attributes of interest for that trouble report have changed.
j) *enrol trouble report*: TMN notifies the TMN that would normally originate a trouble report that a trouble report has been created, either as the result of a request or as a result of an internal action by the notifying TMN.
k) *deenrol trouble report*: TMN notifies the TMN that would normally originate a trouble report that a trouble report has been deleted, either as the result of a request or as a result of an internal action by the notifying TMN.
l) *verify repair completion*: This allows the TMN that originated a trouble report to verify that the repair has been completed to its satisfaction before the trouble report is closed out in the receiving TMN.
m) *modify other trouble report attributes*: TMN modifies writable attributes of a trouble report that are not specifically covered in other functions.
n) *enrol Trouble Report Format Definition*: TMN notifies the TMN that would normally originate a trouble report that a Trouble Report Format Definition has been created.
o) *deenrol Trouble Report Format Definition*: TMN notifies the TMN that would normally originate a trouble report that a Trouble Report Format Definition has been deleted.

p) *attribute value change Trouble Report Format Definition*: TMN notifies the TMN that would normally originate a trouble report that a Trouble Report Format Definition attribute of interest has changed.

q) *trouble report progress update*: TMN notifies the TMN that originated a trouble report about progress on resolving the trouble.

r) *cancel trouble report*: TMN gives notice to another TMN that a previously reported trouble is no longer of interest.

I.2 Object Classes & Attributes

A trouble report can be generated for any GNIM-managed object in ATIS-0300240.1998 (R2007), or for the CNM Service object defined in Annexes A-H of this document. Note that the CNM Service object class allows an abstraction of telecommunications network resources. Clause I.6 provides an overview of the service model approach.

The following support object classes (or their subclasses) and their associated attributes support the Fault Management functions (currently only trouble administration) specified in this standard:

- Event Forwarding Discriminator (defined in ISO/IEC 10164-5).
- Log (defined in ISO/IEC 10164-6).
- Telecommunications Trouble Report (defined in Annexes A-H).
- Trouble History Record (defined in Annexes A-H).
- Trouble Report Format Definition (defined in Annexes A-H).

I.3 Fault Management Functional Units

I.3.1 General

The services defined to support the Fault Management functions specified in Clause I.1 have been grouped into several “functional units” to allow negotiation of their use on an association (during association establishment), and to allow referencing by other standards. Functional unit negotiation shall be performed as described in ATIS-0300208.2008. Functional units for the trouble administration area of Fault Management are defined in I.3.2.

I.3.2 Trouble Administration Functional Units

Table I.1 lists the trouble administration functional units and their corresponding services. These services are defined in Clause I.4.

Table I.1 - Trouble Administration FUs and Services

Functional Unit	Service (s)
Kernel	PT-CREATE on Telecommunications Trouble Report object PT-GET on Telecommunications Trouble Report object
Request Trouble Report Format	PT-GET on Trouble Report Format Definition object
Trouble History Event Notification	Trouble History Event Notification
Review Trouble History Record	PT-GET on Trouble History Record object
Add Trouble Information	PT-SET on Additional Trouble Information List attribute in Telecommunications Trouble Report object
Trouble Report Status/Commitment Time Update Notification	Trouble Report Status/Commitment Time Update Notification
Verify Trouble Repair Completion	PT-SET on Close Out Verification attribute in Telecommunications Trouble Report object
Modify Trouble Administration Information	PT-SET on Telecommunications Trouble Report object
Trouble Administration Configuration	Attribute Value Change Notification

Functional Unit	Service (s)
Event Notification	Object Creation/Deletion Notification
Trouble Report Progress Notification	Trouble Report Progress Notification
Cancel Trouble Report	PT-SET on Cancel Requested by Manager attribute in Telecommunications Trouble Report object

In addition to these functional units, trouble administration events will make use of the event management control and log control functional units defined in ISO/IEC 10164-5 and 10164-6. Notifications from objects not included in functional units defined in this annex shall use the configuration event functional unit in ISO/IEC 10164-1.

The security alarm reporting functional unit and the security audit trail functional unit, defined in ITU-T Recommendation X.736 and CCITT Recommendation X.740, respectively, are also needed.

The functional units in Table I.1 do not support the modification of all the replaceable attributes in the Telecommunications Trouble Report object. Modifications of replaceable attributes not supported in clause I.4 are required only in the service provider to service provider interface. This document does not specify FUs required only by the service provider to service provider interface.

I.4 Trouble Administration Service Definitions

This clause defines the services needed to support the trouble administration category of Fault Management functions defined in clause I.1. Each service definition consists of:

- Behavior and purpose of the service;
- The CMISE service that it maps onto; and
- Restrictions (if any) on the usages of the CMIS parameters.

NOTE – Detailed parameter descriptions are not included when the SET and GET services are used.

The CMIS services, procedures, and parameters are defined in ISO/IEC 9595.

The mapping of the following services to the confirmed or unconfirmed mode of the supporting CMISE services, except where specified, is a local implementation issue and is not specified by this standard.

I.4.1 Kernel Functional Unit

The Kernel functional unit has the following capabilities:

- Entering Trouble Report; and
- Requesting Trouble Report Status.

I.4.1.1 Enter Trouble Report

The PT-CREATE service, as described in ISO/IEC 10164-1, is used to allow a manager to request that a trouble report be created by the agent with the appropriate information.

The PT-CREATE service request is issued by the manager with the Telecommunications Trouble Report object class as the managed object class parameter. If the manager chooses not to use the reference object option, the manager shall supply the following attributes as part of the create operation:

- Managed Object Instance;
- Trouble Type; and
- Additional Trouble Information List;

plus any manager-supplied attributes in conditional packages identified as “must be present”. The manager also has the option to include manager-supplied attributes in conditional packages identified as “may be present” attributes.

Administrations (Service Providers) shall state the required object classes on which troubles may be reported.

If the input information is correct, the agent will respond with the name (Trouble Report ID). The attribute list parameter is mandatory in the response for trouble reports that contain attributes (other than Trouble Report ID) supplied by the agent.

The Enter Trouble Report service supports the following Error messages defined in Annexes A-H:

1. *Trouble Report Already Exists*; and
2. *Fall Back Reporting*.

I.4.1.2 Request Trouble Report Status

The PT-GET service, as described in ISO/IEC 10164-1, is used to allow a manager to request status information on a trouble report.

The manager issues a PT-GET against the Trouble Report Status attribute and the Trouble Report Status Time attribute in the Telecommunications Trouble Report object class. Other “readable” Telecommunications Trouble Report attributes may also be present in the request.

For example, this service could also be used to query the status of more than one (potentially all) trouble reports for that manager. For example, a manager may wish to view all clearedAwaitingCustVerification trouble reports. By using the scoping and filtering capabilities of CMISE, the managing system could get the trouble report status of all its trouble reports where the state is not closed.

I.4.2 Request Trouble Report Format Functional Unit

The PT-GET service, as described in ISO/IEC 10164-1, is used to allow a manager to determine the format for reporting troubles on either a CNM Service or a GNIM object representing a telecommunications network resource. Based on the trouble report format, the manager supplies a different set of attributes when entering a trouble report (see Enter Trouble Report service).

The Telecommunications Trouble Report attributes that the manager must supply are determined by a PT-GET on the tRMustSupplyAttrIDList attribute in the appropriate instance of the Trouble Report Format Definition object. The Telecommunications Trouble Report attributes that the manager may choose to supply are determined by a PT-GET on the tRMaySupplyAttrIDList attribute in the appropriate instance of the Trouble Report Format Definition object.

The appropriate instance of the Trouble Report Format Definition object is determined by either:

a) A PT-GET on the Trouble Report Format Pointer attribute in the CNM Service object (when the format must be defined on an object instance basis);
b) A scoped and filtered PT-GET of the Trouble Report Format Definition object for values of the Applicable Managed Object Classes attribute that match the CNM Service or GNIM object class (when the format is the same for an entire object class); or
c) A scoped and filtered PT-GET of the Trouble Report Format Definition object for values of the Applicable Managed Object Instances attribute that match the GNIM object instance (when the format is specific to the object instance).

Table I.2 – Trouble History Event Notification Service Parameters

Parameter name	Req/Ind	Rsp/Cnf
Invoke Identifier	M	M(=)
Mode	M	-
Managed object class	M	U
Managed object instance	M	U
Event type	M	C(=)
Event time	M	-
Event information	M	-
Current time	-	U
Event reply	-	C
Errors	-	C

I.4.3 Trouble History Event Notification Functional Unit

I.4.3.1 Trouble History Event Notification

The Trouble History Event Notification service allows a managed system to report the trouble report close out information (when the trouble report state transitions to the closed value) to the managing system or log the information in the managed system. This service uses the CMIS M-EVENT-REPORT service and procedures defined in ISO/IEC 9595.

Some implementations may restrict the values of the discriminator construct so that all Trouble History Event Notifications are logged in the managed system and none are reported to the managing system.

Table I.2 shows the parameters used in the Trouble History Event Notification service.

I.4.3.2 Parameters

The following parameters are defined for use in this standard in the Trouble History Event Notification Service and are formally defined in ISO/IEC 9595:

- *Invoke identifier*: See ISO/IEC 9595.
- *Mode*: The mode shall have the value, “confirmed”.
- *Managed object class*: This parameter indicates the Telecommunications Trouble Report object class.
- *Managed object instance*: This parameter specifies an instance of the Telecommunications Trouble Report object class.
- *Event type*: This parameter identifies the Trouble History Event Notification. It may be included in the success confirmation and shall be included if the event reply parameter is included.
- *Event time*: This parameter is mandatory.
- *Event information*: This parameter includes the trouble report close-out information. This includes the following parameters (defined in I.9.2):
  - managed object instance
  - received time
  - trouble found
  - activity duration (optional)
  - additional trouble information list (optional)
  - authorization list (optional)
  - cancel requested by manager (optional)
  - close out narrative (optional)
  - close out verification (optional)
  - commitment time (optional)
  - customer trouble ticket number (optional)

  - perceived trouble severity (optional)
  - restored time (optional)
  - trouble clearance person (optional)
  - trouble report number list (optional)
  - trouble type (optional)
- *Current time:* See ISO/IEC 9595.
- *Event reply:* The inclusion of this parameter in the response is conditional upon the successful receipt of the event report. If included, it will be NULL.
- *Errors:* See ISO/IEC 9595.

I.4.4 Review Trouble History Functional Unit

The PT-GET service, described in ISO/IEC 10164-1, is used to allow a manager to request information about past troubles reported for a particular CNM service or GNIM object instance representing a telecommunications resource.

The PT-GET service request is issued by the manager with the trouble history record object as the managed object class parameter.

I.4.5 Add Trouble Information Functional Unit

The PT-SET service, described in ISO/IEC 10164-1, is used to allow a manager to provide additional descriptive text for an open trouble report. This additional information will be added to the description provided upon trouble entry. The Additional Trouble Information attribute is set-valued with an attribute syntax “graphic string”. A minimum of 256 octets shall be supported regardless of the number of values in the set. The manager can only add information, but not remove it. It is possible that the oldest information may be lost if an implementation has restrictions on the maximum size.

The PT-SET service is issued by the manager against the Additional Trouble Information attribute in the Telecommunications Trouble Report object class. Modification requests for other “writable” attributes may also be present in the same PT-SET if the associated functional units were successfully negotiated during the association establishment phase.

I.4.6 Trouble Report Status/Commitment Time Update Notification Functional Unit

This functional unit is currently defined to report changes in the values of the following trouble report attributes:

- Trouble Report Status; and
- Commitment Time.

NOTE – Although these are “read-only” attributes, the agent system may locally modify their attribute values in the process of addressing the trouble report.

I.4.6.1 Trouble Report Status/Commitment Time Update Notification

The Trouble Report Status/Commitment Time Update uses the Attribute Value Change Notification service defined in ISO/IEC 10164-1. In this FU, the Trouble Report Status/Commitment Time Update Notification service allows the agent to notify the manager of changes in the value(s) of a Trouble Report’s Status or Commitment Time attributes.

The Trouble Report Status/Commitment Time Update Notifications will be filtered by instances of the Event Forwarding Discriminator object in the agent system and if the discriminator criterion is satisfied result in the generation of an M-EVENT REPORT to the manager specified by the destination address.

I.4.7 Verify Repair Completion Functional Unit

The PT-SET service, described in ISO/IEC 10164-1, is used to allow the manager to verify that a repair has been completed to its satisfaction before the trouble report is permanently closed-out by the agent. This service only applies after the service provider has repaired the trouble and changes the Trouble Report Status attribute value to "clearedAwaitingCustVerification".[14]

The PT-SET service request is issued by the manager to change the value of the Close Out Verification attribute in the Telecommunications Trouble Report object to "verified". Once verified by the manager, the agent system will update the Trouble Report Status to "closedOutCustVerified". The PT-SET should also include a Trouble Clearance Person attribute value identifying the person verifying the trouble report.

If the manager determines that the trouble still exists, the manager will use the PT-SET to change the value of the Close Out Verification Attribute to "denied". The PT-SET should also include a Trouble Clearance Person Attribute value identifying the person verifying the trouble report. The agent will then either resume work on the trouble report or update the Trouble Report Status value to "closedOutCustDenied".

If after some time period (set by the local administration) the manager has not "verified" or "denied" the repair, the agent will update the trouble report status value to "closedOut".

If the manager changes the value of the Close Out Verification attribute before the Trouble Report status value is "clearedAwaitingCustVerification", the agent system may optionally respond to the PT-SET with a processing failure (e.g., "cannot verify/deny at this time").

Modification requests for other "writable" attributes may also be present in the same PT-SET if the associated functional units were successfully negotiated during the association establishment phase.

I.4.8 Modify Trouble Administration Information Functional Unit

The PT-SET service, as described in ISO/IEC 10164-1, is used to allow the manager to modify the "writable" attributes that are not covered for a similar function under other functional units.

NOTE – In Annexes A-H, the following attributes of the Telecommunications Trouble Report object are identified as applicable to this service:

- A Location Access Address
- A Location Access Hours
- A Location Access Person
- After Hours Repair Authorization
- Alternate Manager Contact Person
- Alternate Manager Contact Object Pointer
- Authorization List
- Callback Information List
- Commitment Time Request
- Escalation List
- Managed Object Access From Time
- Managed Object Access Hours
- Managed Object Access To Time
- Manager Contact Person
- Manager Contact Object Pointer
- Manager Search Key List
- Perceived Trouble Severity
- Trouble Clearance Person
- Trouble Report Status Window
- Z Location Access Address

[14] This service is meant to be a courtesy to the manager and is not intended to add an additional time element to a trouble report's open time. For purposes of tracking indices, "clearedAwaitingCustVerification" should be considered "closedOut".

- Z Location Access Hours
- Z Location Access Person

The PT-SET service request is issued by the manager to change the values of the “writable” attributes of the Telecommunications Trouble Report object, with the exceptions mentioned above. Modification requests for other “writable” attributes may also be present in the same PT-SET if the associated functional units were successfully negotiated during the association establishment phase.

I.4.9 Trouble Administration Configuration Event Notification Functional Unit

This FU allows the manager to be notified by the managed system when:

- The value of an attribute is changed in the Telecommunications Trouble Report or Trouble Report Format Definition object.
- A Telecommunications Trouble Report or Trouble Report Format Definition object instance is created or deleted by the agent.

Most Telecommunications Trouble Reports are created by the manager, but occasionally a Telecommunications Trouble Report could be opened by the agent on behalf of the managing system. Only the agent deletes Telecommunications Trouble Reports. The manager cannot create or delete Trouble Report Format Definitions.

I.4.9.1 Attribute Value Change Notification

The Attribute Value Change Notification service is defined in ISO/IEC 10164-1. In this FU, the Attribute Value Change Notification service allows the agent to notify the manager of changes in the value(s) of a Telecommunications Trouble Report’s or a Trouble Report Format Definition’s attributes. In general, this notification is used to report one or more of the following:

- The addition of one or more new members to one or more set-valued attributes;
- The removal of one or more members from one or more set-valued attributes;
- The replacement of the values of one or more attributes;
- The changing of the values of one or more attributes to their default value(s);

through either internal operation of the managed object or via management operation.

Attribute Value Change Notifications will be filtered by instances of the Event Forwarding Discriminator object in the agent system and, if the discriminator criterion is satisfied, result in the generation of an M-EVENT REPORT to the manager specified by the destination address. A manager may also log these events as instances of the Attribute Value Change Record object class.

I.4.9.2 Object Creation Notification

The Object Creation Notification service is defined in ISO/IEC 10164-1. In this FU, the Object Creation Notification service allows the agent to notify the manager when a Telecommunications Trouble Report or a Trouble Report Format Definition object is created through local administrative procedures.

Object Creation Notifications will be filtered by instances of the Event Forwarding Discriminator object in the agent system, and if the discriminator criterion is satisfied, results in the generation of an M-EVENT REPORT to the manager specified by the destination address. A manager may also log these events as instances of an Object Creation Record object class.

I.4.9.3 Object Deletion Notification

The Object Deletion Notification service is defined in ISO/IEC 10164-1. In this FU, the object Deletion Notification service allows the agent to notify the manager when a Telecommunications Trouble Report or a Trouble Report Format Definition object is deleted through local administrative procedures.

Object Deletion Notifications will be filtered by instances of the Event Forwarding Discriminator object in the agent system and, if the discriminator criterion is satisfied, results in the generation of an M-EVENT REPORT to the manager specified by the destination address. A manager may also log these events as instances of an Object Deletion Record object class.

I.4.10 Trouble Report Progress Notification Functional Unit

I.4.10.1 Trouble Report Progress Notification

The Trouble Report Progress Notification service allows an agent system to indicate progress made in resolving the trouble report. This notification is generated by the agent within the maximum time allotted by the troubleReportStatusWindow attribute if it is present in the instance of the Telecommunications Trouble Report object class. Once, either the troubleReportProgressNotification or troubleReportStatus attribute value change notification is sent, the beginning of the troubleReportStatusWindow is automatically redefined by the agent system as that event time. The length of the interval is not modified. The notification shall include the troubleReportStatus attribute and if the value of the status has not changed since last issued, it must also include the additionalTroubleStatusInfo attribute, indicating what progress has been made in resolving the Trouble Report. The service uses the CMIS M-EVENT-REPORT service and procedures defined in ISO/IEC 9595.

Table I.3 shows the parameters used in the Trouble Report Progress Notification service.

Table I.3 - Trouble Report Progress Notification service parameters

Parameter Name	Req/Ind	Rsp/Cnf
Invoke Identifier	M	M(=)
Mode	M	-
Managed object class	M	U
Managed object instance	M	U
Event type	M	C(=)
Event time	M	-
Event information	M	-
Current time	-	U
Event reply	-	C
Errors	-	C

I.4.10.2 Parameters

The following parameters are defined for use in this standard in the Trouble Report Progress Notification service and are formally defined in ISO/IEC 9595.

- *Invoke identifier*: See ISO/IEC 9595.
- *Mode*: The mode shall have the value confirmed.
- *Managed object class*: This parameter indicates the Telecommunications Trouble Report object class.
- *Managed object instance*: This parameter specifies an instance of the Telecommunications Trouble Report object class.
- *Event type*: This parameter identifies the Trouble Report Progress Notification. It may be included in the success confirmation and shall be included if the event reply parameter is included.
- *Event time*: This parameter is mandatory.
- *Event Information:* This parameter includes the trouble report progress information. This includes the following parameters (defined in I.9.2):
    - trouble report status; and
    - additional trouble status information (optional).

I.4.11 Cancel Trouble Report Functional Unit

The PT-SET service (see ISO/IEC 10164-1) is used to allow a manager to attempt to remove a trouble report from the agent. Typically, the manager made an error in reporting the trouble or has resolved the trouble and wants to abort the trouble report. In all cases, the agent will respect the manager's request.

The PT-SET service request is issued by the manager to change the value of the Cancel Requested By Manager attribute in the Telecommunications Trouble Report object to "True". The PT-SET should also include a Trouble Report Clearance Person attribute value identifying the person canceling the trouble report. When the request is accepted, the cancellation process begins. This process may have associated billing implications beyond the scope of this standard if work has already started on the trouble (e.g., testing dispatched). The cancellation process started by this PT-SET will eventually result in the update of the Trouble Report Status to "closedOutByCustReq" and, ultimately, in a Trouble History Record entry. A Can Not Close error message (defined in Annexes A-H) in response to the PT-SET implies that the trouble report is already cleared.

Table I.4 - Trouble Administration FU to required object class mapping

Functional Unit	Required object(s)
Kernel	Telecommunications Trouble Report
Request Trouble Report Format	Trouble Report Format Definition CNM Service (if format defined by instance of cnmService)
Trouble History Event Notification	Telecommunications Trouble Report Event Forwarding Discriminator
Review Trouble History Record	Log Trouble History Record
Add Trouble Information	Telecommunications Trouble Report
Trouble Report Status/Commitment Time Update Notification	Telecommunications Trouble Report Event Forwarding Discriminator
Verify Trouble Repair Completion	Telecommunications Trouble Report
Modify Trouble Administration Information	Telecommunications Trouble Report
Trouble Administration Configuration Event Notification	Telecommunications Trouble Report Event Forwarding Discriminator
Trouble Report Progress Notification	Telecommunications Trouble Report Event Forwarding Discriminator
Cancel Trouble Report	Telecommunications Trouble Report

I.5 Conformance

Any implementation claiming conformance to this standard shall comply with the following:

a) *Support as a minimum the Kernel functional unit as listed in clause I.3 (Table I.1) and defined in clause I.4 of this annex*. Each Fault Management service for which support is claimed by an agent or a manager implementation shall support the corresponding functional unit shown in Table I.2.

b) *Support for a given functional unit listed in clause 1.3 (Table I.1) and defined in clause I.4 of this annex requires support of the Object Class(es) associated with those functional units, as shown in Table I.4*.

c) *As part of the system conformance statement, implementations shall state the object classes supported across the trouble administration interface*. Any such object classes used shall conform to the ASN.1 type and value definitions given in T1.240-1998 (R2007) and Annexes A-H of this document, or associated addenda.

d) *Support of messages derived from Trouble Administration services not defined in this annex is permitted*. Also, support of object classes or attribute types not defined in ATIS-0300240.1998 (R2007) or Annexes A-H is permitted. However, the support of such messages, object classes, or attribute types shall not

preclude or hinder the interoperability of such a system with any other system that conforms to this standard.

e) *In order to ensure interoperability, any implementation claiming conformance to this standard that supports additional messages, object classes, or attribute types – as mentioned in (d) – shall be capable of operating as though only messages specified in this standard are being supported*.

Note that this version of the standard assumes a priori knowledge of some aspects of management information (for example, object classes, attributes, and the naming schemes) to be used on a given association is required to ensure interoperability. Definition of this set of management information is for further study.

Table I.4 shows the required object classes when supporting the various functional units specified in this standard.

I.6 Overview of the Service Model

This clause provides the rationale on why some administrations may require support for the CNM Service object class for the trouble administration interface.

Customers manage their services, not the components that the provider uses to provide the service. A customer's request to test a service should produce a result in a form useful and meaningful to a customer, while hiding that information about how the service is provided that may be proprietary to the service provider. Similarly reconfiguration requests or trouble reports should use a syntax and semantics that is friendly to the customer. The service should be managed in terms of how the service is perceived by the customer. Changes in the provider's underlying configuration that do not affect the customer's service should usually be hidden from the customer. Nor should a customer's Network Management systems or personnel need to interact with the details of the provider's system if they can be hidden by a more coherent customer-oriented service view.

Although services are relatively independent of network elements providing the service, they are not completely independent. Clearly, a customer will know of certain network elements and manage them directly (features of analog lines, for example). Also, it would be presumptuous of the service provider to assume complete ignorance of their architecture by their sophisticated customers. A model of customer network management needs to preserve this information where appropriate, if only to allow a common discourse between the customers and the service provider's telecommunications personnel who share a common training and language.

The service provider must map transactions from the customer about services to management activities performed in terms of network elements. For example, a trouble report on a customer's service may map into several trouble reports on distinct network elements. A model of customer network management must allow the definition of this mapping from the service-oriented customer view to the network-oriented internal view.

A service encapsulates common features that are not part of the network element world. These include such qualities as billing information and tariffs, customer contact information, Centrex Group relationships, customer-oriented naming, and ownership. Additionally, services that are provided by concrete network elements are abstracted from the network elements themselves. Identical services can be provided by different types of elements and different services are provided over the same elements (often simultaneously). Regardless of how a service is provided to the customer, its definition should present a coherent view of the product purchased from the provider.

A service may be made up of discrete elements that themselves are services – for example, a Centrex Group or a Private Virtual Network. Features may be shared among the discrete components. These include billing, tariff information, common operational features, common reports, common engineering criteria, common alarm formats, and common trouble reporting. To avoid redundancy, this information needs to be stored in a common object whose scope subtends the separate components of the service.

Customer Network Management, in which a customer interacts with the provider of a service to manage that service may need a partitioning of the "world" far different from that required to manage a network by the "network's" owner. Although the owner or operator of the network needs to view the components of the system and manage them as a complete entity providing many services to many customers, the user uses and manages the services provided by the network as services, not as a series of discreet components.

Services may be abstract concepts. The same network elements may define several different services for one or more customers. Or identical services could be provided by different sets of network elements at different times or for different customers. In either of these cases, the definition of a service shall describe the relationship between

the service offered the customer and the actual network elements that provide the service. The description of the service also will map the behavior of the service and the attributes that describe how it may be managed to the actual elements of the service provider.

A service provided to a customer may itself contain other services. For example, a “Basic Business Group” (Centrex Group) is a service composed of several services (voice lines, for example) provided to a customer as a coherent package and managed as a single entity. The individual components of the business group may themselves be services and be separately manageable. The Basic Business Group contains these component services and would be responsible for the definition and management of qualities that are common to all members of the Centrex Group.

Features and attributes of services to a customer may be common across many instances of the service. These common features or qualities may carry down to the specific instances of the services. A service profile object could model these shared features.

I.7 Negotiation of Functional Units

This specification assigns the following object identifier value:

{iso(1) member-body(2) usa (840) ansi-t1-228-1992(10016) troubleAdminFunctionPkg (2)}

as a value of the ASN.1 type functional unit package ID defined in CCITT Rec. X.701|ISO/IEC 10040 to use for negotiating the following functional units for the Telecommunications Trouble Report object class or any of its subclasses.

0 Kernel
1 Request Trouble Report Format
2 Trouble History Event Notification
3 Review Trouble History Record
4 Add Trouble Information
5 Trouble Report Status/Commitment Time Update Notification
6 Verify Trouble Repair Completion
7 Modify Trouble Administration Information
8 Trouble Administration Configuration Event Notification
9 Trouble Report Progress Notification
10 Cancel Trouble Report

I.8 Negotiation of Application Context

The following object identifier is assigned for the application context on the CNM interface:

{iso(1) member-body(2) usa (840) ansi-t1-228-1992(10016) app-context(1)}

In addition to the object identifier value specified above for the application context, the use of the system management application context specified in CCITT Recommendation X.701 (ISO/IEC 10040) is also applicable for use with the functions defined in this annex.

NOTE – This latter application context is appropriate in a context wider than telecommunications management.

I.9 Event Reports

This clause specifies the ASN.1 type and value definitions for Trouble Administration.

I.9.1 Events

troubleHistoryEventNotification NOTIFICATION

-- see clause I.4 for behavior

WITH INFORMATION SYNTAX TroubleModule.TroubleHistoryInfo;

REGISTERED AS {trNotification 1};

troubleReportProgressNotification NOTIFICATION

-- see clause I.4 for behavior

WITH INFORMATION SYNTAX TroubleModule.TroubleProgressInfo;

REGISTERED AS {trNotification 2};

I.9.2 Abstract Syntax Definitions

TroubleModule {iso(1) member-body(2) usa (840) ansi-t1-228-1992(10016) trGNM(0) troubleModule(1)}

DEFINITIONS IMPLICIT TAGS::=BEGIN

EXPORTS
trNotification;

IMPORTS

ActivityDuration, AdditionalTroubleInfoList, AdditionalTroubleStatusInfo, AuthorizationList,

CancelRequestedByManager, CloseOutVerification, CommitmentTime, PerceivedTroubleSeverity, PersonReach, TroubleFound, TroubleReportNumberList, TroubleReportStatus, TroubleType

FROM GNMTA {iso(1) member-body(2) usa (840) ansi-t1-227-1992(10015) trGNM(0) gnmta(0)}

ObjectInstance

FROM CMIP-1 {joint-iso-ccitt(2) ms(9) cmip(1) modules(0) protocol(3)}; -- Only the Distinguished Name and Local Distinguished Name forms are supported.

trNotification OBJECT IDENTIFIER::= {iso(1) member-body(2) usa (840) ansi-t1-228-1992(10016) trGNM(0)notification(10)}

-- Supporting Productions

TroubleHistoryInfo::= SEQUENCE{

managedObjectInstance [0] ObjectInstance,

receivedTime [1] GeneralizedTime,

troubleFound [2] TroubleFound,

activityDuration [3) ActivityDuration OPTIONAL,

additionalTroubleInfoList [4] AdditionalTroubleInfoList OPTIONAL,

authorizationList [5] AuthorizationList OPTIONAL,

cancelRequestedByManager [6] CancelRequestedByManager OPTIONAL,

closeOutNarr [7] GraphicString OPTIONAL,

closeOutVerification [8] CloseOutVerification OPTIONAL,

commitmentTime [9] CommitmentTime OPTIONAL,

custTroubleTickNum [10] GraphicString OPTIONAL,

perceivedTroubleSeverity [11] PerceivedTroubleSeverity OPTIONAL,

restoredTime [12) GeneralizedTime OPTIONAL,

troubleClearancePerson [13] PersonReach OPTIONAL,

troubleReportNumberList [14] TroubleReportNumberList OPTIONAL,

```
    troubleType[15] TroubleType OPTIONAL}

TroubleProgressInfo::= SEQUENCE{
    troubleReportStatus[0] TroubleReportStatus,
    additionalTroubleStatusInfo [1] AdditionalTroubleStatusInfo OPTIONAL}
END
```

I.10 Access Control

The following abstract syntax definitions and object identifiers are included in support of security requirements to facilitate implementation. Either of the syntax productions may be used for the access control parameter in CMIPUserInfo of A-ASSOCIATE requests and response PDUs and operations. Other approaches to providing security are not precluded. In the use proposed here, this parameter is to be used to authenticate the requester of A-ASSOCIATE requests, m-Get, m-Set, m-Create, m-Delete, m-Action operations, and sender of response PDUs.

I.10.1 Abstract Syntax 1

The following module for authentication (to be used for access control) is registered.

```
TroubleAdministration-AccessControl-Information {iso(1) member-body(2)
usa(840) ansi-t1-228-1992(10016) trGNM(0) accessControlModule(2)}
DEFINITIONS::= BEGIN
  AccessControl ::= SEQUENCE {
    entityIdentifier [0] IMPLICIT GraphicString (SIZE 1..64)),
    encryptedString [1] IMPLICIT OCTET STRING (SIZE (1..64))

    }
END
```

The following registration is for the abstract syntax:

{iso(1) member-body(2) usa(840) ansi-t1-228-1992(10016) accessControlAbstractSyntax(3)}

The encryption mechanism to be used is outside the scope of this approach. Depending on the algorithm, algorithm-specific parameters may have to be communicated using a mechanism outside the scope of this standard.

Depending on what is being encrypted, adequate security is provided only when a key is shared by exactly two parties.

I.10.2 Abstract Syntax 2

The following module for authentication (to be used for access control) is also registered.

```
TroubleAdministration-AccessControl-Information {iso(1) member-body(2)
```

usa(840) ansi-t1-228-1992(10016) trGNM(0) specificAccessControlModule(3) encryptionMethod(1)}

DEFINITIONS::= BEGIN

AccessControl ::= SEQUENCE {

entityIdentifier [0] IMPLICIT GraphicString (SIZE 1..64)),

initializationVector[1]IMPLICIT OCTET STRING (SIZE (8)),

keyIdentifier [2] IMPLICIT INTEGER,

encryptedString [3] IMPLICIT OCTET STRING (SIZE (8..64))

}

END

The entitytIdentifier identifies the manager or agent sending the Access Control field.

The initializationVector is used to initialize the Cipher Block Chaining (CBC) mode of the Data Encryption Standard (DES) algorithm which generates the encrypted string.

The keyIdentifier is the index value used to select the DES key. The key is selected from a common list of DES keys shared by the agent and manager.

The encryptedString contains the encrypted value of GeneralizedTime at which the PDU was transmitted.

The following object identifier has been registered for the above module.

TroubleAdministration-AccessControl-Information {iso(1) member-body(2)

usa(840) ansi-t1-228-1992(10016) trGNM(0) specificAccessControlModule(3) encryptionMethod(1)}

The following registration is for the abstract syntax:

{iso(1) member-body(2) usa(840) ansi-t1-228-1992(10016) specificAccessControlAbstractSyntax(4) encryptionMethod(1)}

While the encryption method is specified in this subclause, the description of the key management mechanism (i.e., how two parties come to share a common list of DES keys) remains outside the scope of this standard.

The access control mechanism described in this subclause provides adequate security only when a key is shared by exactly two parties.

Annex J: CORBA/IDL – Interface Design

(informative)

J.1 Interface Definitions

J.1.1 TroubleReportAdministrator

TroubleReportAdministrator provides a factory and an administrative interface for creating and locating trouble reports. The model does not restrict the number of the instances of the TroubleReportAdministrator interface; however, it is suggested that the interface supported by the provider be a service object (i.e., only one instance provides the service for a particular implementation). The single instance is called repetitively by the customer to satisfy request that may be unrelated.

J.1.1.1 Create

The customer uses ***create*** to initiate the creation of a trouble report on a provider's service, resulting in a CORBA TroubleReport object to manage the trouble report.

```
TroubleReport create(
            in  RequestIdType                   requestId,
            in  IdentityType                    customer,
            in  CreateRequestType               request,
            out CreateResponseType              response,
            inout TagValueListTypeOpt           appSpecificData
        )
        raises(
            FallBackReportingError,
            TroubleReportAlreadyExists,
            TRMustBePresentAttributeMissing,
            AccessDenied,
            MissingData,
            InvalidDataReceived,
            ProcessingFailureError
        );
```

Parameters:

requestId	Unique ID identifying the client's request. Intended for tracking across the X interface.
Customer	Identifies the end user at the company, account, or user level.
Request	Contains the initial values for the trouble report attributes specified by the customer.
Response	Contains the initial values of the trouble report attributes specified by the provider.
AppSpecificData	List of tags and associated values to be defined and employed on a per-implementation basis.

***ReturnValues*:**

TroubleReport	CORBA object reference for the object representing the provider's trouble report.

***Exceptions*:**

FallBackReportingError	A trouble report was not created, although the provider may accept the trouble report and process it manually.
TroubleReportAlreadyExists	A trouble report currently exists on the service identified in the request.
TRMustBePresentAttributesMissing	All required attributes, as defined by the Trouble Report Format Definition (TRFD) in the request, where not provided.
AccessDenied	The customer is not authorized to perform the requested transaction (e.g., the service does not belong to the customer).
MissingData	Required information was missing from the input parameters.
InvalidDataReceived	The input parameters contain invalid data.
ProcessingFailureError	A system-related error (e.g., system timed out, resources exceeded occurred during processing).

J.1.1.2 Locate

The customer uses ***locate*** to retrieve the CORBA object reference representing a provider's trouble report, identified by the passed troubleReporId.

```
TroubleReport locate(
        in  RequestIdType              requestId,
        in  IdentityType             customer,
        in  TroubleReportIdType          troubleReportId,
        inout TagValueListTypeOpt          appSpecificData
    )
    raises(
        NotFound,
        AccessDenied,
        ProcessingFailureError
    );
```

Parameters:

requestId	Unique ID identifying the client's request. Intended for tracking across the X interface.
Customer	Identifies the end user at the company, account, or user level.
TroubleReportId	Identifier for the provider's trouble report.
AppSpecificData	List of tags and associated values to be defined and employed on a per-implementation basis.

ReturnValues:

TroubleReport	CORBA object reference for the object representing the provider's trouble report.

***Exceptions*:**

NotFound	The trouble report was not found.
AccessDenied	The customer is not authorized to perform the requested transaction (e.g., the service on which the trouble report resides does not belong to the customer).
ProcessingFailureError	A system-related error (e.g., system timed out, resources exceeded occurred during processing).

J.1.1.3 LocateByServiceId

The customer uses ***locateByServiceId*** to retrieve the CORBA object reference representing a provider's trouble report on the specified service.

```
TroubleReport locateByServiceId(
        in  RequestIdType          requestId,
        in  IdentityType           customer,
        in  ServiceIdType          serviceId,
        inout TagValueListTypeOpt  appSpecificData
    )
    raises(
        NotFound,
        AccessDenied,
        ProcessingFailureError
    );
```

Parameters:

requestId	Unique ID identifying the client's request. Intended for tracking across the X interface.
Customer	Identifies the end user at the company, account, or user level.
ServiceId	Identifies the service.
AppSpecificData	List of tags and associated values to be defined and employed on a per-implementation basis.

ReturnValues:

TroubleReport	CORBA object reference for the object representing the provider's trouble report.

***Exceptions*:**

NotFound	The trouble report was not found.
AccessDenied	The customer is not authorized to perform the requested transaction (e.g., the service on which the trouble report resides does not belong to the customer).
ProcessingFailureError	A system-related error (e.g., system timed out, resources exceeded occurred during processing).

J.1.1.4 GetTroubleReports

The ***getTroubleReports*** method is used to retrieve a list of object references for all bonded trouble reports which are currently in one of the specified trouble report states on services "belonging" to the specified customer.

```
void getTroubleReports(
        in  RequestIdType                 requestId,
        in  IdentityType                  customer,
        in  TroubleReportStateListType    statesToGet,
        out TroubleReportListType         troubleReportList,
        inout TagValueListTypeOpt         appSpecificData
    )
    raises(
        ProcessingFailureError
    );
```

Parameters:

requestId	Unique ID identifying the client's request. Intended for tracking across the X interface.
Customer	Identifies the end user at the company, account, or user level. The returned object references must represent trouble reports "belonging" to this customer.
StatesToGet	List of trouble report state values (queued, open-active, deferred, cleared, etc.). All returned object references must represent trouble reports that are currently in on the specified states.
TroubleReportList	List of object references representing trouble reports matching the above selection criteria.
AppSpecificData	List of tags and associated values to be defined and employed on a per-implementation basis.

ReturnValues:

NONE

Exceptions:

ProcessingFailureError	A system-related error (e.g., system timed out, resources exceeded occurred during processing).

J.1.1.5 GetTroubleReportsAttributes

The customer uses ***getTroubleReportsAttributes*** to retrieve specific attribute values on all bonded trouble reports that are currently in one of the specified trouble report states, on services "belonging" to the specified customer.

```
Void getTroubleReportsAttributes(
        in  RequestIdType                requestId,
        in  IdentityType                 customer,
        in  AttributeIdListType          attributesToGet,
        in  TroubleReportStateListType   statesToGet,
        out GetResponseListType          getResponseList,
        inout TagValueListTypeOpt        appSpecificData
    )
    raises(
        InvalidDataReceived,
        ProcessingFailureError
    );
```

Parameters:

requestId	Unique ID identifying the client's request. Intended for tracking across the X interface.
Customer	Identifies the end user at the company, account, or user level. The trouble reports include in the getResponseList must "belong" to this customer.
AttributesToGet	List identifying the attribute values to return in the getResponseList.
StatesToGet	List of trouble report state values (queued, open-active, deferred, cleared, etc.). The trouble reports included in the getResponseList must currently be in one of these specified states.
GetResponseList	List, where each element holds the requested attribute values for a single trouble report.
AppSpecificData	List of tags and associated values to be defined and employed on a per-implementation basis.

ReturnValues:

NONE

Exceptions:

InvalidDataReceived	The input parameters contain invalid data.
ProcessingFailureError	A system-related error (e.g., system timed out, resources exceeded occurred during processing).

J.1.1.6 GetTroubleReportIdByServiceId

The customer uses ***getTroubleReportIdByServiceId*** to retrieve the troubleReportId for the existing trouble report on the specified service.

```
Void getTroubleReportIdByServiceId(
        in  RequestIdType              requestId,
        in  IdentityType               customer,
        in  ServiceIdType              serviceId,
        out TroubleReportIdType        troubleReportId,
        inout TagValueListTypeOpt      appSpecificData
    )
    raises (
        NotFound,
        AccessDenied,
        ProcessingFailureError
    );
```

Parameters:

requestId	Unique ID identifying the client's request. Intended for tracking across the X interface.
Customer	Identifies the end user at the company, account, or user level.
ServiceId	Identifies the service.
TroubleReportId	The returned troubleReportId of the trouble report.
AppSpecificData	List of tags and associated values to be defined and employed on a per-implementation basis.

ReturnValues:

NONE

***Exceptions*:**

NotFound	The trouble report was not found.
AccessDenied	The customer is not authorized to perform the requested transaction (e.g., the service on which the trouble report resides does not belong to the customer).
ProcessingFailureError	A system-related error (e.g., system timed out, resources exceeded occurred during processing).

J.1.1.7 GetServiceTroubleHistory

The ***getServiceTroubleHistory*** method returns a list of history records for prior trouble reports on the specified service.

```
Void getServiceTroubleHistory(
        in  RequestIdType                 requestId,
        in  IdentityType                  customer,
        in  ServiceIdType                 serviceId,
        out TroubleHistoryRecordListType  historyRecords,
        inout TagValueListTypeOpt         appSpecificData
    )
    raises (
        NotFound,
        AccessDenied,
        ProcessingFailureError );
```

requestId	Unique ID identifying the client's request. Intended for tracking across the X interface.
Customer	Identifies the end user at the company, account, or user level.
ServiceId	Identifies the service.
HistoryRecords	List of trouble report history records, one for each trouble report.
AppSpecificData	List of tags and associated values to be defined and employed on a per-implementation basis.

ReturnValues:

NONE

***Exceptions*:**

NotFound	The trouble report was not found.
AccessDenied	The customer is not authorized to perform the requested transaction (e.g., the service does not belong to the customer).
ProcessingFailureError	A system-related error (e.g., system timed out, resources exceeded occurred during processing).

J.1.1.8 BindToEventChannel

The customer uses ***bindToEventChannel*** to register a structured push consumer object for receiving event notifications from the provider.

```
Void bindToEventChannel(
        in RequestIdType          requestId,
        in IdentityType           customer,
        in ConstraintListTypeOpt  constraintList,
        in QoSPropertiesTypeOpt   qosList,
        in TroubleEventConsumer   troubleEventConsumer,
        inout TagValueListTypeOpt appSpecificData
    )
    raises (
      AlreadyBoundToEventChannel,
      UnsupportedQoS,
      InvalidConstraintList,
      ProcessingFailureError
);
```

Parameters:

requestId	Unique ID identifying the client's request. Intended for tracking across the X interface.
Customer	Identifies the end user at the company, account, or user level.
ConstraintList	A list of strings, each containing a boolean expression that is used to filter the events delivered to the push consumer. Each expression is expressed in default constraint grammar used by the OMG CosNotification service.
QosList	An optional CosNotification::QoSProperties sequence used to tailor the CosNotification QoS (Quality of Service) property settings for events delivered to the consumer.
TroubleEventConsumer	The object reference for the structured push consumer object that will receive the events.
AppSpecificData	List of tags and associated values to be defined and employed on a per-implementation basis.

ReturnValues:

NONE

***Exceptions*:**

AlreadyBoundToEventChannel	The consumer is already registered to receive events.
UnsupportedQos	One or more the the QoS property settings were invalid.
InvalidConstraintList	The constraint list is not valid.
ProcessingFailureError	A system-related error (e.g., system timed out, resources exceeded occurred during processing).

J.1.1.9 UnbindFromEventChannel

The customer uses ***unbindFromEventChannel*** to unregister the structured push consumer with the provider.

```
Void unbindFromEventChannel(
        in RequestIdType            requestId,
        in TroubleEventConsumer         troubleEventConsumer,
        inout TagValueListTypeOpt       appSpecificData
    )
    raises (
      NotFound,
      ProcessingFailureError
);
```

Parameters:

requestId	Unique ID identifying the client's request. Intended for tracking across the X interface.
TroubleEventConsumer	The object reference for the structured push consumer object to unregister.
AppSpecificData	List of tags and associated values to be defined and employed on a per-implementation basis.

ReturnValues:

NONE

***Exceptions*:**

NotFound	The consumer is not currently registered to receive events.
ProcessingFailureError	A system-related error (e.g., system timed out, resources exceeded occurred during processing).

J.1.2 TroubleReport

The ***TroubleReport*** interface represents a bonded trouble report on a service.

J.1.2.1 getAttributes

The customer uses ***getAttributes*** to retrieve specific attribute values (as identified in the attributesToGet list) for the trouble report.

```
void getAttributes(
        in  RequestIdType              requestId,
        in  IdentityType               customer,
        in  AttributeIdListType        attributesToGet,
        out GetResponseType            getResponse,
        inout TagValueListTypeOpt      appSpecificData
    )
    raises(
        AccessDenied,
        InvalidDataReceived,
        ProcessingFailureError
);
```

Parameters:

requestId	Unique ID identifying the client's request. Intended for tracking across the X interface.
customer	Identifies the end user at the company, account, or user level.
attributesToGet	List identifying the attribute values returned in the getResponse.
getResponse	Holds the requested attribute values for the trouble report.
appSpecificData	List of tags and associated values to be defined and employed on a per-implementation basis.

ReturnValues:

NONE

Exceptions:

AccessDenied	The customer is not authorized to perform the requested transaction (e.g., the service on which the trouble report resides does not belong to the customer).
InvalidDataReceived	The input parameters contain invalid data.
ProcessingFailureError	A system-related error (e.g., system timed out, resources exceeded occurred during processing).

J.1.2.2 modifyAttributes

The customer uses ***modifyAttributes*** to change the values for “customer-modifiable” trouble report attributes.

```
void modifyAttributes(
        in  RequestIdType              requestId,
        in  IdentityType               customer,
        in  ModifyRequestType          modifyRequest,
        out ModifyResponseType         modifyResponse,
        inout TagValueListTypeOpt      appSpecificData
    )
    raises(
        TroubleReportChangeDenied,
        AccessDenied,
        MissingData,
        InvalidDataReceived,
        ProcessingFailureError
);
```

Parameters:

requestId	Unique ID identifying the client’s request. Intended for tracking across the X interface.
customer	Identifies the customer at the company, account, or user level.
modifyRequest	Holds the new attribute values.
modifyResponse	List identifying the attributes that were modified.

ReturnValues:

NONE

Exceptions:

TroubleReportChangeDenied	Attempted modifications are not allowed for trouble reports in the current state.
AccessDenied	The customer is not authorized to perform the requested transaction (e.g., the service on which the trouble report resides does not belong to the customer).
MissingData	All expected values were not provided.
InvalidDataReceived	The input parameters contain invalid data.
ProcessingFailureError	A system-related error (e.g., system timed out, resources exceeded occurred during processing).

J.1.2.3 addInfo

The ***addInfo*** method allows text to be added to the additionalTroubleInfoList attribute of the trouble report.

```
void addInfo(
        in  RequestIdType                   requestId,
        in  IdentityType                  customer,
        in  AdditionalTroubleInfoListType     addInfoList,
        inout TagValueListTypeOpt           appSpecificData
    )
    raises(
        TroubleReportChangeDenied,
        AccessDenied,
        MissingData,
        InvalidDataReceived,
        ProcessingFailureError
);
```

Parameters:

requestId	Unique ID identifying the client's request. Intended for tracking across the X interface.
customer	Identifies the end user at the company, account, or user level.
addInfoList	Holds the text to add to the addtionalTroubleInfoList attribute.
appSpecificData	List of tags and associated values to be defined and employed on a per-implementation basis.

ReturnValues:

NONE

Exceptions:

TroubleReportChangeDenied	Attempted modifications are not allowed for trouble reports in the current state.
AccessDenied	The customer is not authorized to perform the requested transaction (e.g., the service on which the trouble report resides does not belong to the customer).
MissingData	All expected values were not provided.
InvalidDataReceived	The input parameters contain invalid data.
ProcessingFailureError	A system-related error (e.g., system timed out, resources exceeded occurred during processing).

J.1.2.4 verifyCompletion

The ***verifyCompletion*** method is used to grant or deny closeout of a trouble report in a cleared state.

```
void verifyCompletion(
        in  RequestIdType                   requestId,
        in  IdentityType                    customer,
        in  CloseOutVerificationType        closeOutVerification,
        in  AdditionalTroubleInfoListTypeOpt    additionalTroubleInfoList,
        in  PersonReachTypeOpt              troubleClearancePerson,
        inout TagValueListTypeOpt           appSpecificData
    )
    raises(
        CannotVerifyOrDenyAtThisTime,
        TroubleReportChangeDenied,
        AccessDenied,
        MissingData,
        InvalidDataReceived,
        ProcessingFailureError
);
```

Parameters:

requestId	Unique ID identifying the client's request. Intended for tracking across the X interface.
customer	Identifies the end user at the company, account, or user level.
closeOutVerification	Used to indicate if the customer allows or denies closeout of the trouble report.
additionalTroubleInfoList	Holds the customer remarks, explaining the closeOutVerification value.
troubleClearancePerson	Contact information for the user performing this method.
appSpecificData	List of tags and associated values to be defined and employed on a per-implementation basis.

ReturnValues:

NONE

Exceptions:

CannotVerifyOrDenyAtThisTime	The trouble report is not in a cleared state.
TroubleReportChangedDenied	The trouble report has been closed.
AccessDenied	The customer is not authorized to perform the requested transaction (e.g., the service on which the trouble report resides does not belong to the customer).
MissingData	All expected values were not provided.
InvalidDataReceived	The input parameters contain invalid data.
ProcessingFailureError	A system-related error (e.g., system timed out, resources exceeded occurred during processing).

J.1.2.5 authorize

The ***authorize*** method is used by the customer to grant and/or deny activities on the trouble report.

```
void authorize(
        in  RequestIdType              requestId,
        in  IdentityType               customer,
        in  AuthorizationListType      authorizationList,
        inout TagValueListTypeOpt      appSpecificData
    )
    raises(
        TroubleReportChangeDenied,
        AccessDenied,
        MissingData,
        InvalidDataReceived,
        ProcessingFailureError
);
```

Parameters:

requestId	Unique ID identifying the client's request. Intended for tracking across the X interface.
customer	Identifies the end user at the company, account, or user level.
authorizationList	Holds a list indicating which activities are denied or granted.
appSpecificData	List of tags and associated values to be defined and employed on a per-implementation basis.

ReturnValues:

NONE

Exceptions:

TroubleReportChangeDenied	Attempted modifications are not allowed for trouble reports in the current state.
AccessDenied	The customer is not authorized to perform the requested transaction (e.g., the service on which the trouble report resides does not belong to the customer).
MissingData	All expected values were not provided.
InvalidDataReceived	The input parameters contain invalid data.
ProcessingFailureError	A system-related error (e.g., system timed out, resources exceeded occurred during processing).

J.1.2.6 escalate

The ***escalate*** method is used by the customer to request escalation of the trouble report.

```
void escalate(
        in  RequestIdType              requestId,
        in  IdentityType               customer,
        in  EscalationListType         escalationList,
        inout TagValueListTypeOpt      appSpecificData
    )
    raises(
        TroubleReportChangeDenied,
        AccessDenied,
        MissingData,
        InvalidDataReceived,
        ProcessingFailureError
);
```

Parameters:

requestId	Unique ID identifying the client's request. Intended for tracking across the X interface.
customer	Identifies the end user at the company, account, or user level.
escalationList	Holds a list identifying which escalations are requested.
appSpecificData	List of tags and associated values to be defined and employed on a per-implementation basis.

ReturnValues:

NONE

Exceptions:

TroubleReportChangeDenied	Attempted modifications are not allowed for trouble reports in the current state.
AccessDenied	The customer is not authorized to perform the requested transaction (e.g., the service on which the trouble report resides does not belong to the customer).
MissingData	All expected values were not provided.
InvalidDataReceived	The input parameters contain invalid data.
ProcessingFailureError	A system-related error (e.g., system timed out, resources exceeded occurred during processing).

J.1.2.7 cancel

The customer uses ***cancel*** for requesting that the provider cancel the associated trouble report.

```
void cancel(
        in  RequestIdType                   requestId,
        in  IdentityType                    customer,
        in  CancelRequestedByCustomerType   cancelRequestedByCustomer,
        in  AdditionalTroubleInfoListTypeOpt  additionalTroubleInfoList,
        in  PersonReachTypeOpt              troubleClearancePerson,
        inout TagValueListTypeOpt           appSpecificData
    )
    raises(
        TroubleReportChangeDenied,
        AccessDenied,
        MissingData,
        InvalidDataReceived,
        ProcessingFailureError
);
```

Parameters:

requestId	Unique ID identifying the client's request. Intended for tracking across the X interface.
customer	Identifies the end user at the company, account, or user level.
cancelRequestedByCustomer	Boolean value indicating if the trouble report should be canceled.
additionalTroubleInfoList	Holds the customer remarks explaining the cancellation.
troubleClearancePerson	Contact information for the user performing this method.
appSpecificData	List of tags and associated values to be defined and employed on a per-implementation basis.

ReturnValues:

NONE

Exceptions:

TroubleReportChangeDenied	Attempted modifications are not allowed for trouble reports in the current state.
AccessDenied	The customer is not authorized to perform the requested transaction (e.g., the service on which the trouble report resides does not belong to the customer).
MissingData	All expected values were not provided.
InvalidDataReceived	The input parameters contain invalid data.
ProcessingFailureError	A system-related error (e.g., system timed out, resources exceeded occurred during processing).

J.1.2.8 getHistory

The ***getHistory*** method returns a list of events that have occurred on the trouble report.

```
void getHistory(
        in  RequestIdType              requestId,
        in  IdentityType               customer,
        in  DateTimeTypeOpt            oldestEventTime,
        out ChangeEventListType        changeEvents,
        inout TagValueListTypeOpt      appSpecificData
    )
    raises (
        AccessDenied,
        ProcessingFailureError
);
```

Parameters:

requestId	Unique ID identifying the client's request. Intended for tracking across the X interface.
customer	Identifies the end user at the company, account, or user level.
oldestEventTime	If specified, the returned events will have occurred at or after this data/time.
changeEvents	Resulting list of events.
appSpecificData	List of tags and associated values to be defined and employed on a per-implementation basis.

ReturnValues:

NONE

Exceptions:

AccessDenied	The customer is not authorized to perform the requested transaction (e.g., the service on which the trouble report resides does not belong to the customer).
ProcessingFailureError	A system-related error (e.g., system timed out, resources exceeded occurred during processing).

J.1.3 Notifications

The ***Notifications*** interface is used to document the layout of the structured events delivered by the provider. An implementation of this interface may be supported by the customer, but is not required.

The rules used to construct and read the events are as follows:

- The domain_name string in the fixed header of the structure is set to "telecommunications".
- The type_name string in the fixed header of the structure is set to the scoped name of the operation. For example, "Trouble::Notifications::troubleReportChanged".
- The event_name string in the fixed header of the structure is set to null.
- The optional header fields may be included to for QoS.

- Each parameter in the following operations is placed in a name-value pair in the filterable body portion of the notification. The fd_name string of the pair is set to the name of the paramenter, and the type placed in the associated fd_value will be the type specified for the parameter.
- The unfilterable portion of the event body will be set to null.

J.1.3.1 troubleReportCreated

A structured event of type troubleReportCreated will be delivered by the provider to when a new trouble report is created on one of the customer's services.

```
void troubleReportCreated(
        in  NotificationIdType      notificationId,
        in  RequestIdTypeOpt        requestId,
        in  UserIdTypeOpt           user,
        in  AccountTypeOpt          account,
        in  CompanyTypeOpt          company,
        in  TroubleReport           troubleReport,
        in  TroubleReportIdType     troubleReportId,
        in  DateTimeType            eventTime,
        in  ChangeEventType         eventData);
```

Parameters:

notificationId	Unique identifier for this notification.
requestId	The request ID of the customer's request that resulted in this notification (if any).
user	The optional user assigned to the trouble report.
account	The optional account assigned to the trouble report.
company	The optional company assigned to the trouble report.
troubleReport	Object reference for the CORBA object representing the trouble report.
troubleReportId	The trouble report ID value of the trouble report.
eventTime	The date and time that the event occurred.
eventData	Holds the values of the attributes that were modified.

J.1.3.2 troubleReportChanged

A structured event of type troubleReportChanged will be delivered by the provider whenever the attribute values of the trouble report are modified by the provider.

```
void troubleReportChanged(
        in  NotificationIdType      notificationId,
        in  RequestIdTypeOpt        requestId,
        in  UserIdTypeOpt           user,
        in  AccountTypeOpt          account,
        in  CompanyTypeOpt          company,
        in  TroubleReport           troubleReport,
        in  TroubleReportIdType     troubleReportId,
        in  DateTimeType            eventTime,
        in  ChangeEventType         eventData);
```

Parameters:

notificationId	Unique identifier for this notification.
requestId	The request ID of the customer's request that resulted in this notification (if any).
user	The optional user assigned to the trouble report.
account	The optional account assigned to the trouble report.
company	The optional company assigned to the trouble report.
troubleReport	Object reference for the CORBA object representing the trouble report.
troubleReportId	The trouble report ID value of the trouble report.
eventTime	The date and time that the event occurred.
eventData	Holds the values of the attributes that were modified.

J.2 IDL Modules

The following IDL are new to this document:

- ANSI T1 227 IDL Module
- Customer Types Module

The following IDL modules are included in prior standards, but with additional features:

- Base Module – additional data types added.
- TimeTypes Module – additional structures and enumerations added for length of time and day of week.
- Location Types Module – additional data types added and revised PersonReach types added.

J.2.1 ANSI T1 227 Module

```
/* This IDL code is meant to be stored in a file named "ansi_t1_227.idl"
 * located in the search path used by IDL compilers on your system.
 */

#ifndef ansi_t1_227_IDL
#define ansi_t1_227_IDL


#include <CosNotification.idl>
#include <CosNotifyFilter.idl>
#include <CosNotifyComm.idl>
#include <CosEventChannelAdmin.idl>

#include <Base.idl>
#include <TimeTypes.idl>
#include <LocationTypes.idl>
#include <CustomerTypes.idl>

#pragma prefix "t1.org"

//
// Imported Exceptions
//
#define AccessDenied            Base::AccessDenied
#define InvalidDataReceived     Base::InvalidData
#define MissingData             Base::MissingData
#define NotFound                Base::NotFound
#define ProcessingFailureError  Base::ProcessingFailureError

//
//
//  ansi_t1_227: Provides an interface for customer-provider trouble
//               administration. This interface is based on ANSI T1.227/228
//               (Trouble Administration).
//
//
module ansi_t1_227
{

   //================================================================
   //
   //                  T Y P E S
   //
   //================================================================
```

```
//
// Imports
//

typedef TimeTypes::WeekMaskList_t          AccessHoursType;
typedef CustomerTypes::AccountType         AccountType;
typedef CustomerTypes::AccountTypeOpt      AccountTypeOpt;
typedef Base::ElementName_t                AttributeIdType;
typedef Base::ElementNameList_t            AttributeIdListType;
typedef Base::BooleanOpt_t                 BooleanTypeOpt;
typedef CustomerTypes::CompanyType         CompanyType;
typedef CustomerTypes::CompanyTypeOpt      CompanyTypeOpt;
typedef CustomerTypes::CodesTypeOpt        CustomerCodesTypeOpt;
typedef TimeTypes::DateTime_t              DateTimeType;
typedef TimeTypes::DateTimeOpt_t           DateTimeTypeOpt;
typedef TimeTypes::Length_t                DurationType;
typedef TimeTypes::LengthOpt_t             DurationTypeOpt;
typedef CustomerTypes::IdentityType        IdentityType;
typedef CustomerTypes::IdentityTypeOpt     IdentityTypeOpt;
typedef LocationTypes::PersonReachR1_t     PersonReachType;
typedef LocationTypes::PersonReachR1Opt_t  PersonReachTypeOpt;
typedef Base::RequestId_t                  RequestIdType;
typedef Base::RequestIdOpt_t               RequestIdTypeOpt;
typedef Base::StringOpt_t                  StringTypeOpt;
typedef Base::TagValueListOpt_t            TagValueListTypeOpt;
typedef LocationTypes::UnfieldedAddress_t  UnfieldedAddressType;
typedef CustomerTypes::UserIdType          UserIdType;
typedef CustomerTypes::UserIdTypeOpt       UserIdTypeOpt;

//=====================================================================
//
//                  C O N S T A N T S
//
//=====================================================================

module ConstType {
   const string activityDurationList =        "activityDurationList";
   const string additionalTroubleInfoList =   "additionalTroubleInfoList";
   const string additionalTroubleStatusInfo = "additionalTroubleStatusInfo";
   const string aLocationAccessHours =        "aLocationAccessHours";
   const string aLocationAccessAddress =      "aLocationAccessAddress";
   const string aLocationAccessPerson =       "aLocationAccessPerson";
   const string alternateCustomerContactPerson
                              = "alternateCustomerContactPerson";
   const string authorizationList =           "authorizationList";
   const string callBackInfoList =            "callBackInfoList";
   const string calledNumber =                "calledNumber";
   const string cancelRequestedByCustomer =   "cancelRequestedByCustomer";
   const string closeOutNarr =                "closeOutNarr";
   const string closeOutVerification =        "closeOutVerification";
   const string commitmentTime =              "commitmentTime";
   const string commitmentTimeRequest =       "commitmentTimeRequest";
   const string customer=                     "customer";
   const string customerContactPerson =       "customerContactPerson";
   const string customerInfo =                "customerInfo";
   const string customerTroubleTickNum =      "customerTroubleTickNum";
   const string customerWorkCenter =          "customerWorkCenter";
   const string dialog =                      "dialog";
   const string escalationList =              "escalationList";
   const string handOffInfo =                 "handOffInfo";
   const string initiatingMode =              "initiatingMode";
   const string lastUpdateTime =              "lastUpdateTime";
   const string maintOrgContactPerson =       "maintOrgContactPerson";
   const string maintOrgContactTime =         "maintOrgContactTime";
```

```
    const string maintServiceCharge =           "maintServiceCharge";
    const string outageDuration =               "outageDuration";
    const string perceivedTroubleSeverity =     "perceivedTroubleSeverity";
    const string preferredPriority =            "preferredPriority";
    const string providerContactPerson =        "providerContactPerson";
    const string receivedTime =                 "receivedTime";
    const string relatedTroubleReportList =     "relatedTroubleReportList";
    const string repairActivityList =           "repairActivityList";
    const string repeatReport =                 "repeatReport";
    const string restoredTime =                 "restoredTime";
    const string responsiblePerson =            "responsiblePerson";
    const string serviceAccessHours =           "serviceAccessHours";
    const string serviceAccessFromTime =        "serviceAccessFromTime";
    const string serviceAccessToTime =          "serviceAccessToTime";
    const string serviceAliasList =             "serviceAliasList";
    const string serviceId =                    "serviceId";
    const string suspectObjectList =            "suspectObjectList";
    const string troubleClearancePerson =       "troubleClearancePerson";
    const string troubleDetectionTime =         "troubleDetectionTime";
    const string troubleFound =                 "troubleFound";
    const string troubleLocation =              "troubleLocation";
    const string troubleReportId =              "troubleReportId";
    const string troubleReportFormatId=         "troubleReportFormatId";
    const string troubleReportNumberList =      "troubleReportNumberList";
    const string troubleReportState =           "troubleReportState";
    const string troubleReportStatus =          "troubleReportStatus";
    const string troubleReportStatusTime =      "troubleReportStatusTime";
    const string troubleReportStatusWindow =    "troubleReportStatusWindow";
    const string troubleType =                  "troubleType";
    const string tspPriority =                  "tspPriority";
    const string zLocationAccessAddress =       "zLocationAccessAddress";
    const string zLocationAccessHours =         "zLocationAccessHours";
    const string zLocationAccessPerson =        "zLocationAccessPerson";
};


//
// Trouble Types
//

union AccessHoursTypeOpt switch(boolean) {
  case TRUE: AccessHoursType theValue;
};


// _AC appended to ActivityCodeType enum members to avoid conflict
// with names in other enum sets (an issue for some IDL compilers).

enum ActivityCodeType {
    Approved_AC,
    Assigned_AC,
    Cancel_AC,
    Clear_AC,
    Close_AC,
    Defer_AC,
    Dispatch_AC,
    Refer_AC,
    Release_AC,
    Reopen_AC,
    Repair_AC,
    Test_AC,
    Transfer_AC
};

union ActivityCodeTypeOpt switch(boolean) {
```

```
  case TRUE: ActivityCodeType theValue;
};

enum ActivityTypeType {
    AfterHoursRepair,
    Standby,
    AfterHoursStandby,
    Test,
    CustomerInitiatedTest,
    Dispatch,
    NoAccess,
    DelayedMaintenance,
    ActivityTypeRelease,
    DeregulatedWork,
    AuthorizeToWork,
    NUM_ACTIVITY_TYPES
};


union ActivityTypeTypeOpt switch (boolean) {
  case TRUE: ActivityTypeType theValue;
};


struct ActivityDurationType {
    DurationType                duration;
    BooleanTypeOpt              billable;
    ActivityTypeTypeOpt         type;
};


typedef sequence <ActivityDurationType> ActivityDurationListType;


union ActivityDurationListTypeOpt switch(boolean) {
  case TRUE: ActivityDurationListType theValue;
};


typedef sequence<string> AdditionalTroubleInfoListType;


union AdditionalTroubleInfoListTypeOpt switch(boolean) {
  case TRUE: AdditionalTroubleInfoListType theValue;
};

typedef sequence<string> AdditionalTroubleStatusInfoType;


union AdditionalTroubleStatusInfoTypeOpt switch(boolean) {
  case TRUE: AdditionalTroubleStatusInfoType theValue;
};



enum RequestStateType {
    Requested,
    Provided,
    Denied,
    NoState, // only used internally by TroubleMgmt
    NUM_REQUEST_STATES
};

struct AuthorizationType {
    RequestStateType            state;
```

```
    ActivityTypeType            type;
    DateTimeTypeOpt             authTime;
    PersonReachTypeOpt          authPerson;
};


typedef sequence<AuthorizationType> AuthorizationListType;


union AuthorizationListTypeOpt switch(boolean) {
  case TRUE: AuthorizationListType theValue;
};

enum BondedStateType {
    Bonded,
    Manual,
    NotBonded,
    UnKnownBondedState,
    NUM_BONDED_STATES
};

union BondedStateTypeOpt switch(boolean) {
  case TRUE: BondedStateType theValue;
};


enum CallBackInfoChoiceType {
    EscalationChoice,
    BeforeAutoTestChoice,
    AfterClearedChoice
};

union CallBackInfoType switch(CallBackInfoChoiceType) {
  case EscalationChoice:       PersonReachType escalation;
  case BeforeAutoTestChoice:   PersonReachType beforeAutoTest;
  case AfterClearedChoice:     PersonReachType afterCleared;
};

typedef sequence<CallBackInfoType> CallBackInfoListType;

union CallBackInfoListTypeOpt switch(boolean) {
  case TRUE: CallBackInfoListType theValue;
};


typedef string CalledNumberType;


union CalledNumberTypeOpt switch(boolean) {
  case TRUE: CalledNumberType theValue;
};


typedef boolean CancelRequestedByCustomerType;


union CancelRequestedByCustomerTypeOpt switch(boolean) {
  case TRUE: CancelRequestedByCustomerType theValue;
};


enum CanNotCloseType {
    alreadyCleared,
    NUM_CAN_NOT_CLOSES
};
```

```
enum ChangeDeniedReasonType {
    waitingVerificationOfClosure,
    troubleReportAlreadyClosed,
    activityAuthorizationPending,
    NUM_CHANGE_DENIED_REASONS
};


typedef string CloseOutNarrType;


union CloseOutNarrTypeOpt switch(boolean) {
  case TRUE: CloseOutNarrType theValue;
};


enum CloseOutVerificationType {
    NoAction,
    Verified,
    Denied_CloseOutVerification,
    DeniedActivityDurationDisputed,
    DeniedCloseOutNarrDisputed,
    NUM_CLOSE_OUT_VERIFICATIONS
};

union CloseOutVerificationTypeOpt switch(boolean) {
  case TRUE: CloseOutVerificationType theValue;
};

enum CommitmentTimeChoiceType {
    OnsiteTimeChoice,
    ClearedTimeChoice
};

union CommitmentTimeType switch(CommitmentTimeChoiceType) {
  case OnsiteTimeChoice:    DateTimeType onsiteTime;
  case ClearedTimeChoice:   DateTimeType clearedTime;
};

union CommitmentTimeTypeOpt switch(boolean) {
  case TRUE: CommitmentTimeType theValue;
};

typedef string CustomerWorkCenterType;


union CustomerWorkCenterTypeOpt switch(boolean) {
  case TRUE: CustomerWorkCenterType theValue;
};


typedef string CustomerTroubleTickNumType;


union CustomerTroubleTickNumTypeOpt switch(boolean) {
  case TRUE: CustomerTroubleTickNumType theValue;
};

enum OrgLevelType {
    NoEscalation,
    FirstLevel,
    SecondLevel,
    ThirdLevel,
```

```
    FourthLevel,
    FifthLevel,
    SixthLevel,
    NUM_ORG_LEVELS
};


union OrgLevelTypeOpt switch (boolean) {
  case TRUE: OrgLevelType theValue;
};


struct EscalationType  {
    RequestStateType         state;
    DateTimeType             escTime;

    // supplied by service provider
    PersonReachType                requestPerson;
    OrgLevelTypeOpt                level;
    PersonReachTypeOpt             escPerson;
};


typedef sequence<EscalationType> EscalationListType;


union EscalationListTypeOpt switch(boolean) {
  case TRUE: EscalationListType theValue;
};


// Fallback reporting is a combination of ECIC codes
// and a string. The string allows backward compatibility
// for any possible implements which do not use the ECIC
// codes.

enum FallBackReportingChoiceType {
    StringChoice,
    EnumChoice,
    NUM_FALL_BACK_REPORTING_CHOICES
};


enum FallBackReportingEnumType {
    CircuitMismatch,
    CreateInProgress,
    CircuitOwnership,
    InvalidStateCode,
    InvalidSegment,
    LeadingZeroes,
    NewServicePending,
    ServiceDisconnected,
    InvalidCompanyAssigningCode,
    InvalidCircuitFormat,
    ResoldServices,
    NUM_FALL_BACK_REPORTING_ENUMS
};


union FallBackReportingType switch(FallBackReportingChoiceType) {
  case StringChoice: string                  stringValue;
  case EnumChoice:   FallBackReportingEnumType   enumValue;
};

struct HandOffInfoType {
```

```
    DateTimeTypeOpt                 handOffTime;
    StringTypeOpt                   handOffCenter;
    StringTypeOpt                   handOffLocation;
    PersonReachTypeOpt              handOffPerson;
};

union HandOffInfoTypeOpt switch(boolean) {
  case TRUE: HandOffInfoType theValue;
};

enum InitiatingModeType {
    CustomerDirect,
    CustomerIndirect,
    ProviderOriginated,
    CustomerIndirectEMail,
    CustomerIndirectFax,
    CustomerIndirectPersonal,
    CustomerIndirectPhone,
    NUM_INITIATING_MODES
};


union InitiatingModeTypeOpt switch(boolean) {
  case TRUE: InitiatingModeType theValue;
};


struct LocationAddressType {
    string                  name;
    UnfieldedAddressType        address;
};


union LocationAddressTypeOpt switch(boolean) {
  case TRUE: LocationAddressType theValue;
};


typedef boolean MaintServiceChargeType;


union MaintServiceChargeTypeOpt switch(boolean) {
  case TRUE: MaintServiceChargeType theValue;
};

typedef unsigned long NotificationIdType;

typedef DurationType        OutageDurationType;


union OutageDurationTypeOpt switch(boolean) {
  case TRUE: OutageDurationType theValue;
};


enum PerceivedTroubleSeverityType {
    OutOfService,
    BackInService,
    ServiceImpairment,
    NonServiceAffectingTrouble,
    NUM_PERCEIVED_TROUBLE_SEVERITIES
};


union PerceivedTroubleSeverityTypeOpt switch(boolean) {
```

```
  case TRUE: PerceivedTroubleSeverityType theValue;
};


enum PreferredPriorityType {
    Undefined,
    Minor,
    Major,
    Serious,
    LifeThreatening,
    NUM_PREFERRED_PRIORITIES
};


union PreferredPriorityTypeOpt switch(boolean) {
  case TRUE: PreferredPriorityType theValue;
};

typedef string RepairActivityInfoType;

struct RepairActivityType {
    DateTimeType                entryTime;
    RepairActivityInfoType      activityInfo;
    PersonReachTypeOpt          activityPerson;
    ActivityCodeTypeOpt         activityCode;
};

typedef sequence<RepairActivityType> RepairActivityListType;

union RepairActivityListTypeOpt switch(boolean) {
  case TRUE: RepairActivityListType theValue;
};


enum RepeatReportType {
    Unspecified,
    RecentInstallation,
    Repeat,
    BothInstallationAndRepeat,
    Chronic,
    BothInstallationAndChronic,
    NUM_REPEAT_REPORTS
};


union RepeatReportTypeOpt switch(boolean) {
  case TRUE: RepeatReportType theValue;
};


typedef string ServiceIdType;

typedef StringTypeOpt ServiceIdTypeOpt;

typedef sequence<ServiceIdType> ServiceListType;

union ServiceListTypeOpt switch(boolean) {
    case TRUE: ServiceListType theValue;
};

struct SuspectObjectType {
    ServiceIdType           serviceId;
    long                    failureProbability;
};
```

```
typedef sequence<SuspectObjectType> SuspectObjectListType;

union SuspectObjectListTypeOpt switch(boolean) {
  case TRUE: SuspectObjectListType theValue;
};

enum TroubleFoundType {
    Pending,
    CameClear,
    CentralOffice,
    SwitchTrouble,
    CustomerProvidedEquipment,
    Facility,
    CentralOfficeFacility,
    ICfacility,
    InterexchangeCarrier,
    TroubleFoundInformation,
    NonplanClassified,
    NonplanClassifiedIC,
    NonplanClassifiedEA,
    NoTroubleFound,
    Station,
    StationProductData,
    StationProductTerminal,
    StationProductVideo,
    StationProductVoice,
    TroubleFoundStationWiring,
    OtherStationEquipment,
    FoundOKStation,
    ServingBureau,
    TestOK,
    PublicServicesCoinSet,
    CustomerOperatingInstructions,
    TestedOKVerifiedOK,
    CoFacilityTestedFoundOK,
    OutsideFacilityTestedFoundOK,
    ReferredOutToOtherDept,
    ProtectiveConnectingArrang,
    CpeCustomerResponsibility,
    PreService,
    PreServiceIC,
    PreServiceEA,
    ServiceNode,
    TroubleFoundData,
    CustomerReferredToVendor,
    ExchangeAccess,
    International,
    OtherProvidedAccess,
    ExistingReport,
    CancelExclude,
    PaBX,
    OutsideWire,
    OutsideTerminals,
    OutsidePlantEquipment,
    OutsidePlantFiberOptic,
    OutsidePlantOther,
    CoEquipmentOther,
    CoEquipmentFrames,
    CoConcentrator,
    ReceiverOffHook,
    CpeAuthorized,
    CpeTelcoMaintained,
    IndependentCompany,
    CpeCalledNumber,
    AssigningProvisioning,
```

```
    InterServiceCenter,
    ReferredOut,
    Network_TroubleFound,
    NUM_TROUBLE_FOUNDS
};

union TroubleFoundTypeOpt switch(boolean) {
  case TRUE: TroubleFoundType theValue;
};

typedef short TroubleReportFormatIdType;

union TroubleReportFormatIdTypeOpt switch(boolean) {
  case TRUE: TroubleReportFormatIdType theValue;
};

typedef string TroubleReportIdType;

typedef StringTypeOpt       TroubleReportIdTypeOpt;

typedef sequence<TroubleReportIdType> TroubleReportIdListType;

union TroubleReportIdListTypeOpt switch(boolean) {
  case TRUE: TroubleReportIdListType theValue;
};

// forward declaration
interface TroubleReport;

typedef sequence<TroubleReport> TroubleReportListType;


enum TroubleReportStateType {
    UnknownTroubleState,
    Queued,
    OpenActive,
    Deferred,
    Cleared,
    Closed,
    Disabled,
    NUM_TROUBLE_REPORT_STATES
};

typedef sequence<TroubleReportStateType> TroubleReportStateListType;

union TroubleReportStateTypeOpt switch(boolean) {
  case TRUE: TroubleReportStateType theValue;
};


enum TroubleReportStatusType {
    Screening,
    Testing,
    DispatchedIn,
    DispatchedOut,
    PreassignedOut,
    BulkDispatchedOut,
    StartRepair,
    PendingTest,
    PendingDispatch,
    RequestRepair,
    ReferMtceCenter,
    ReferVendor,
    NoAccessOther,
    StartNoAccess,
```

```
    StopNoAccess,
    StartDelayedMtce,
    StopDelayedMtce,
    TroubleEscalated,
    CraftDispatched,
    TemporaryOK,
    CableFailure,
    OriginatingEquipFailure,
    BackOrder,
    ClearedCustNotAdvised,
    ClearedCustAdvised,
    ClearedAwaitingCustVerification,
    ClosedOut,
    ClosedOutByCustReq,
    ClosedOutCustVerified,
    ClosedOutCustDenied,
    CanceledPendingWorkInProgress,
    CanceledPendingTestCompletion,
    CanceledPendingDispatchCompl,
    NUM_TROUBLE_REPORT_STATUSES
  };


  union TroubleReportStatusTypeOpt switch(boolean) {
    case TRUE: TroubleReportStatusType theValue;
  };

  typdef short TroubleTypeType
 /**
Integer values are defined in ATIS-0300094.2008.
*/

  union TroubleTypeTypeOpt switch(boolean) {
    case TRUE: TroubleTypeType theValue;
  };


  typedef string TspPriorityType;


  union TspPriorityTypeOpt switch(boolean) {
    case TRUE: TspPriorityType theValue;
  };


  //=================================================================
  //
  //              P A R A M E T E R   T Y P E S
  //
  //=================================================================


  //-----------------------------------------------------------------
  //
  // CreateRequestType :
  //       in parameter on TroubleReportAdministrator::create()
  //
  //-----------------------------------------------------------------

  struct CreateRequestType
  {
    TroubleReportFormatIdType          troubleReportFormatId;
    AdditionalTroubleInfoListTypeOpt   additionalTroubleInfoList;
    LocationAddressTypeOpt             aLocationAccessAddress;
    AccessHoursTypeOpt                 aLocationAccessHours;
```

```
   PersonReachTypeOpt                aLocationAccessPerson;
   PersonReachTypeOpt                alternateCustomerContactPerson;
   AuthorizationListTypeOpt          authorizationList;
   CallBackInfoListTypeOpt           callBackInfoList;
   CalledNumberTypeOpt               calledNumber;
   CommitmentTimeTypeOpt             commitmentTimeRequest;
   IdentityTypeOpt                   customer;
   PersonReachTypeOpt                customerContactPerson;
   CustomerCodesTypeOpt              customerInfo;
   CustomerTroubleTickNumTypeOpt     customerTroubleTickNum;
   CustomerWorkCenterTypeOpt         customerWorkCenter;
   StringTypeOpt                     dialog;
   EscalationListTypeOpt             escalationList;
   PerceivedTroubleSeverityTypeOpt   perceivedTroubleSeverity;
   PreferredPriorityTypeOpt          preferredPriority;
   RepeatReportTypeOpt               repeatReport;
   ServiceIdTypeOpt                  serviceId;
   AccessHoursTypeOpt                serviceAccessHours;
   DateTimeTypeOpt                   serviceAccessFromTime;
   DateTimeTypeOpt                   serviceAccessToTime;
   ServiceListTypeOpt                serviceAliasList;
   SuspectObjectListTypeOpt          suspectObjectList;
   DateTimeTypeOpt                   troubleDetectionTime;
   DurationTypeOpt                   troubleReportStatusWindow;
   TroubleTypeTypeOpt                troubleType;
   TspPriorityTypeOpt                tspPriority;
   LocationAddressTypeOpt            zLocationAccessAddress;
   AccessHoursTypeOpt                zLocationAccessHours;
   PersonReachTypeOpt                zLocationAccessPerson;
};


//-------------------------------------------------------------------
//
// CreateResponseType :
//        out parameter on TroubleReportAdministrator::create()
//
//-------------------------------------------------------------------

struct CreateResponseType
{
   InitiatingModeTypeOpt             initiatingMode;
   DateTimeType                      receivedTime;
   TroubleReportIdType               troubleReportId;
   TroubleReportStateType            troubleReportState;
   TroubleReportStatusType           troubleReportStatus;
   DateTimeType                      troubleReportStatusTime;
   PersonReachTypeOpt                providerContactPerson;
   CommitmentTimeTypeOpt             commitmentTime;
   TspPriorityTypeOpt                tspPriority;
};

//-------------------------------------------------------------------
//
// GetResponseType : out parameter on
//       TroubleReport::getAttributes()
//
//-------------------------------------------------------------------

struct GetResponseType
{
   ActivityDurationListTypeOpt       activityDurationList;
   AdditionalTroubleInfoListTypeOpt  additionalTroubleInfoList;
   AdditionalTroubleStatusInfoTypeOpt  additionalTroubleStatusInfo;
   AccessHoursTypeOpt                aLocationAccessHours;
```

```
    LocationAddressTypeOpt                  aLocationAccessAddress;
    PersonReachTypeOpt                      aLocationAccessPerson;
    PersonReachTypeOpt                      alternateCustomerContactPerson;
    AuthorizationListTypeOpt                authorizationList;
    CallBackInfoListTypeOpt                 callBackInfoList;
    CalledNumberTypeOpt                     calledNumber;
    CancelRequestedByCustomerTypeOpt        cancelRequestedByCustomer;
    CloseOutNarrTypeOpt                     closeOutNarr;
    CloseOutVerificationTypeOpt             closeOutVerification;
    CommitmentTimeTypeOpt                   commitmentTime;
    CommitmentTimeTypeOpt                   commitmentTimeRequest;
    IdentityTypeOpt                         customer;
    PersonReachTypeOpt                      customerContactPerson;
    CustomerCodesTypeOpt                    customerInfo;
    CustomerTroubleTickNumTypeOpt           customerTroubleTickNum;
    CustomerWorkCenterTypeOpt               customerWorkCenter;
    StringTypeOpt                           dialog;
    EscalationListTypeOpt                   escalationList;
    HandOffInfoTypeOpt                      handOffInfo;
    InitiatingModeTypeOpt                   initiatingMode;
    DateTimeTypeOpt                         lastUpdateTime;
    PersonReachTypeOpt                      maintOrgContactPerson;
    DateTimeTypeOpt                         maintOrgContactTime;
    MaintServiceChargeTypeOpt               maintServiceCharge;
    OutageDurationTypeOpt                   outageDuration;
    PerceivedTroubleSeverityTypeOpt         perceivedTroubleSeverity;
    PreferredPriorityTypeOpt                preferredPriority;
    PersonReachTypeOpt                      providerContactPerson;
    DateTimeTypeOpt                         receivedTime;
    TroubleReportIdListTypeOpt              relatedTroubleReportList;
    RepairActivityListTypeOpt               repairActivityList;
    RepeatReportTypeOpt                     repeatReport;
    PersonReachTypeOpt                      responsiblePerson;
    DateTimeTypeOpt                         restoredTime;
    AccessHoursTypeOpt                      serviceAccessHours;
    DateTimeTypeOpt                         serviceAccessFromTime;
    DateTimeTypeOpt                         serviceAccessToTime;
    ServiceListTypeOpt                      serviceAliasList;
    ServiceIdTypeOpt                        serviceId;
    SuspectObjectListTypeOpt                suspectObjectList;
    PersonReachTypeOpt                      troubleClearancePerson;
    DateTimeTypeOpt                         troubleDetectionTime;
    TroubleFoundTypeOpt                     troubleFound;
    LocationAddressTypeOpt                  troubleLocation;
    TroubleReportIdTypeOpt                  troubleReportId;
    TroubleReportFormatIdTypeOpt            troubleReportFormatId;
    TroubleReportIdListTypeOpt              troubleReportNumberList;
    TroubleReportStateTypeOpt               troubleReportState;
    TroubleReportStatusTypeOpt              troubleReportStatus;
    DateTimeTypeOpt                         troubleReportStatusTime;
    DurationTypeOpt                         troubleReportStatusWindow;
    TroubleTypeTypeOpt                      troubleType;
    TspPriorityTypeOpt                      tspPriority;
    LocationAddressTypeOpt                  zLocationAccessAddress;
    AccessHoursTypeOpt                      zLocationAccessHours;
    PersonReachTypeOpt                      zLocationAccessPerson;
};

typedef sequence<GetResponseType> GetResponseListType;

//---------------------------------------------------------------------
//
// ModifyRequestType : in parameter on TroubleReport::modifyAttributes()
//
//---------------------------------------------------------------------
```

```
struct ModifyRequestType
{
   AdditionalTroubleInfoListTypeOpt        additionalTroubleInfoList;
   LocationAddressTypeOpt                  aLocationAccessAddress;
   AccessHoursTypeOpt                      aLocationAccessHours;
   PersonReachTypeOpt                      aLocationAccessPerson;
   PersonReachTypeOpt                      alternateCustomerContactPerson;
   AuthorizationListTypeOpt                authorizationList;
   CallBackInfoListTypeOpt                 callBackInfoList;
   CommitmentTimeTypeOpt                   commitmentTimeRequest;
   PersonReachTypeOpt                      customerContactPerson;
   StringTypeOpt                           dialog;
   EscalationListTypeOpt                   escalationList;
   PerceivedTroubleSeverityTypeOpt         perceivedTroubleSeverity;
   PreferredPriorityTypeOpt                preferredPriority;
   RepeatReportTypeOpt                     repeatReport;
   AccessHoursTypeOpt                      serviceAccessHours;
   DateTimeTypeOpt                         serviceAccessFromTime;
   DateTimeTypeOpt                         serviceAccessToTime;
   DateTimeTypeOpt                         troubleDetectionTime;
   DurationTypeOpt                         troubleReportStatusWindow;
   LocationAddressTypeOpt                  zLocationAccessAddress;
   AccessHoursTypeOpt                      zLocationAccessHours;
   PersonReachTypeOpt                      zLocationAccessPerson;
};

//-------------------------------------------------------------------
//
// ModifyResponseType : out parameter on TroubleReport::modifyAttributes()
//
//-------------------------------------------------------------------

struct ModifyResponseType
{
   BooleanTypeOpt                          additionalTroubleInfoList;
   BooleanTypeOpt                          aLocationAccessAddress;
   BooleanTypeOpt                          aLocationAccessHours;
   BooleanTypeOpt                          aLocationAccessPerson;
   BooleanTypeOpt                          alternateCustomerContactPerson;
   BooleanTypeOpt                          authorizationList;
   BooleanTypeOpt                          callBackInfoList;
   BooleanTypeOpt                          commitmentTimeRequest;
   BooleanTypeOpt                          customerContactPerson;
   BooleanTypeOpt                          dialog;
   BooleanTypeOpt                          escalationList;
   BooleanTypeOpt                          perceivedTroubleSeverity;
   BooleanTypeOpt                          preferredPriority;
   BooleanTypeOpt                          repeatReport;
   BooleanTypeOpt                          serviceAccessHours;
   BooleanTypeOpt                          serviceAccessFromTime;
   BooleanTypeOpt                          serviceAccessToTime;
   BooleanTypeOpt                          troubleDetectionTime;
   BooleanTypeOpt                          troubleReportStatusWindow;
   BooleanTypeOpt                          zLocationAccessAddress;
   BooleanTypeOpt                          zLocationAccessHours;
   BooleanTypeOpt                          zLocationAccessPerson;
};

//-------------------------------------------------------------------
//
// TroubleHistoryRecord types : out parameter on
//       TroubleReportAdministrator::getServiceTroubleHistory()
//
//-------------------------------------------------------------------
```

```
struct TroubleHistoryRecordType
{
   ActivityDurationListTypeOpt             activityDurationList;
   AdditionalTroubleInfoListTypeOpt        additionalTroubleInfoList;
   AuthorizationListTypeOpt                authorizationList;
   CancelRequestedByCustomerTypeOpt        cancelRequestedByCustomer;
   CloseOutNarrTypeOpt                     closeOutNarr;
   CloseOutVerificationTypeOpt             closeOutVerification;
   CommitmentTimeTypeOpt                   commitmentTime;
   IdentityType                         customer;
   CustomerTroubleTickNumTypeOpt           customerTroubleTickNum;
   PerceivedTroubleSeverityTypeOpt         perceivedTroubleSeverity;
   DateTimeType                         receivedTime;
   DateTimeTypeOpt                         restoredTime;
   PersonReachTypeOpt                      troubleClearancePerson;
   TroubleFoundType                        troubleFound;
   DateTimeType                         troubleReportCloseOutTime;
   TroubleReportIdTypeOpt                  troubleReportId;
   TroubleReportIdListTypeOpt              troubleReportNumberList;
   TroubleTypeTypeOpt                      troubleType;
};

union TroubleHistoryRecordTypeOpt switch(boolean) {
  case TRUE: TroubleHistoryRecordType theValue;
};

typedef sequence<TroubleHistoryRecordType> TroubleHistoryRecordListType;

union TroubleHistoryRecordListTypeOpt switch(boolean) {
 case TRUE: TroubleHistoryRecordListType theValue;
};


//============================================================================
//
//               E V E N T   T Y P E S
//
//============================================================================


// ConstraintList and QoSProperties types used to tailor event delivery
// by the notification service on a per-client basis.

typedef string Constraint;
typedef sequence<Constraint> ConstraintListType;

union ConstraintListTypeOpt switch(boolean) {
 case TRUE: ConstraintListType theValue;
};

union QoSPropertiesTypeOpt switch(boolean) {
 case TRUE: CosNotification::QoSProperties theValue;
};

typedef CosNotifyComm::StructuredPushConsumer TroubleEventConsumer;


// ChangeEventType
//     Represents events that occur based on a change in the
//     trouble report's state.  If requestId is present, then
//     it identifies the request that triggered the change
//     event.

struct ChangeEventType {
```

```
    NotificationIdType                  notificationId;
    RequestIdTypeOpt                    requestId;
    DateTimeType                        eventTime;
    TroubleReportIdType                 troubleReportId;
    IdentityType                        customer;
    ActivityDurationListTypeOpt         activityDurationList;
    AdditionalTroubleStatusInfoTypeOpt  additionalTroubleStatusInfo;
    AuthorizationListTypeOpt            authorizationList;
    CloseOutNarrTypeOpt                 closeOutNarr;
    CommitmentTimeTypeOpt               commitmentTime;
    StringTypeOpt                       dialog;
    EscalationListTypeOpt               escalationList;
    HandOffInfoTypeOpt                  handOffInfo;
    DateTimeTypeOpt                     lastUpdateTime;
    PersonReachTypeOpt                  maintOrgContactPerson;
    DateTimeTypeOpt                     maintOrgContactTime;
    MaintServiceChargeTypeOpt           maintServiceCharge;
    OutageDurationTypeOpt               outageDuration;
    PersonReachTypeOpt                  providerContactPerson;
    TroubleReportIdListTypeOpt          relatedTroubleReportList;
    RepairActivityListTypeOpt           repairActivityList;
    RepeatReportTypeOpt                 repeatReport;
    PersonReachTypeOpt                  responsiblePerson;
    DateTimeTypeOpt                     restoredTime;
    ServiceListTypeOpt                  serviceAliasList;
    TroubleFoundTypeOpt                 troubleFound;
    LocationAddressTypeOpt              troubleLocation;
    TroubleReportIdListTypeOpt          troubleReportNumberList;
    TroubleReportStateTypeOpt           troubleReportState;
    TroubleReportStatusTypeOpt          troubleReportStatus;
    DateTimeTypeOpt                     troubleReportStatusTime;
};

union ChangeEventTypeOpt switch(boolean) {
  case TRUE: ChangeEventType theValue;
};

typedef sequence<ChangeEventType> ChangeEventListType;

union ChangeEventListTypeOpt switch(boolean) {
  case TRUE: ChangeEventListType theValue;
};


//=================================================================
//
//                 E X C E P T I O N S
//
//=================================================================

exception AlreadyBoundToEventChannel {};

exception CannotVerifyOrDenyAtThisTime {};

exception InvalidConstraintList {};

exception UnsupportedQoS {};

exception TroubleReportChangeDenied {
    ServiceIdType               serviceId;
    TroubleReportIdType         troubleReportId;
    ChangeDeniedReasonType      changeDeniedReason;
};
```

```
exception TroubleReportAlreadyExists {
   ServiceIdType              serviceId;
   TroubleReportIdType        troubleReportId;
};


exception TRMustBePresentAttributeMissing {
   AttributeIdListType  missingAttributes;
};


exception FallBackReportingError {
   FallBackReportingType  fallBackReporting;
};

//===================================================================
//
//                TroubleReport Interface
//
// TroubleReport: Represents a trouble that was reported on a
//              telecommunications service.
//
//===================================================================

interface TroubleReport  {

   //
   // getAttributes(): Allows the retrieval of trouble report information.
   //

   void getAttributes(
      in  RequestIdType                  requestId,
      in  IdentityType                   customer,
      in  AttributeIdListType            attributesToGet,
      out GetResponseType                getResponse,
      inout TagValueListTypeOpt          appSpecificData
   )
   raises(
      AccessDenied,
      InvalidDataReceived,
      ProcessingFailureError
   );

   //
   // addInfo() : Allows additionalTroubleInfo to be added to the trouble
   //           report.
   void addInfo(
      in  RequestIdType                  requestId,
      in  IdentityType                   customer,
      in  AdditionalTroubleInfoListType  addInfoList,
      inout TagValueListTypeOpt          appSpecificData
   )
   raises(
      TroubleReportChangeDenied,
      AccessDenied,
      MissingData,
      InvalidDataReceived,
      ProcessingFailureError
   );



   //
   // modifyAttributes(): Allows certain attributes on the trouble report
   //              to be modified.
```

```
//
void modifyAttributes(
   in  RequestIdType                    requestId,
   in  IdentityType                     customer,
   in  ModifyRequestType                modifyRequest,
   out ModifyResponseType               modifyResponse,
   inout TagValueListTypeOpt            appSpecificData
)
raises(
   TroubleReportChangeDenied,
   AccessDenied,
   MissingData,
   InvalidDataReceived,
   ProcessingFailureError
);



//
// verifyCompletion(): Allows the customer to verify the completion
//                  of a trouble repair.
//
void verifyCompletion(
   in  RequestIdType                    requestId,
   in  IdentityType                     customer,
   in  CloseOutVerificationType         closeOutVerification,
   in  AdditionalTroubleInfoListTypeOpt    additionalTroubleInfoList,
   in  PersonReachTypeOpt               troubleClearancePerson,
   inout TagValueListTypeOpt            appSpecificData
)
raises(
   CannotVerifyOrDenyAtThisTime,
   TroubleReportChangeDenied,
   AccessDenied,
   MissingData,
   InvalidDataReceived,
   ProcessingFailureError
);



//
// authorize(): Allows a customer to approve or deny authorization.
//
void authorize(
   in  RequestIdType                    requestId,
   in  IdentityType                     customer,
   in  AuthorizationListType            authorizationList,
   inout TagValueListTypeOpt            appSpecificData
)
raises(
   TroubleReportChangeDenied,
   AccessDenied,
   MissingData,
   InvalidDataReceived,
   ProcessingFailureError
);


//
// escalate(): Allows a customer to request escalation.
//
void escalate(
   in  RequestIdType                    requestId,
   in  IdentityType                     customer,
```

```
        in  EscalationListType              escalationList,
        inout TagValueListTypeOpt           appSpecificData
    )
    raises(
        TroubleReportChangeDenied,
        AccessDenied,
        MissingData,
        InvalidDataReceived,
        ProcessingFailureError
    );


    //
    // cancel(): Allows a customer to request that the trouble report
    //           be cancelled.
    void cancel(
        in  RequestIdType                   requestId,
        in  IdentityType                    customer,
        in  CancelRequestedByCustomerType   cancelRequestedByCustomer,
        in  AdditionalTroubleInfoListTypeOpt  additionalTroubleInfoList,
        in  PersonReachTypeOpt              troubleClearancePerson,
        inout TagValueListTypeOpt           appSpecificData
    )
    raises(
        TroubleReportChangeDenied,
        AccessDenied,
        MissingData,
        InvalidDataReceived,
        ProcessingFailureError
    );


    //
    //  getHistory: Retrieves a list of change events that occurred
    //              after a specified time for this trouble report.
    //
    //
    void getTroubleHistory(
        in  RequestIdType           requestId,
        in  IdentityType            customer,
        in  DateTimeTypeOpt         oldestEventTime,
        out ChangeEventListType     changeEvents,
        inout TagValueListTypeOpt   appSpecificData
    )
    raises (
        AccessDenied,
        ProcessingFailureError
    );

}; // TroubleReport

//============================================================================
//
//            TroubleReportAdministrator Interface
//
// TroubleReportAdministrator: Provides a "factory" and administrative
//                         interface for creating and locating
//                         trouble reports.
//
//============================================================================

interface TroubleReportAdministrator {

    //
    // create(): Creates a new trouble report object.
```

```
//
TroubleReport create(
   in  RequestIdType              requestId,
   in  IdentityType               customer,
   in  CreateRequestType          request,
   out CreateResponseType         response,
   inout TagValueListTypeOpt      appSpecificData
)
raises(
   FallBackReportingError,
   TroubleReportAlreadyExists,
   TRMustBePresentAttributeMissing,
   AccessDenied,
   MissingData,
   InvalidDataReceived,
   ProcessingFailureError
);


//
// locate(): Returns an existing trouble report object reference
//          identified by the troubleReportId.
//
TroubleReport locate(
   in  RequestIdType                 requestId,
   in  IdentityType                  customer,
   in  TroubleReportIdType           troubleReportId,
   inout TagValueListTypeOpt         appSpecificData
)
raises(
   NotFound,
   AccessDenied,
   ProcessingFailureError
);

//
// locateByServiceId(): Returns the object reference for an
//        existing trouble report on the specified service.
//
//
TroubleReport locateByServiceId(
   in  RequestIdType              requestId,
   in  IdentityType               customer,
   in  ServiceIdType              serviceId,
   inout TagValueListTypeOpt      appSpecificData
)
raises (
   NotFound,
   AccessDenied,
   ProcessingFailureError
);

//
//  getTroubleReports: Returns a list of TroubleReport references for
//                all trouble reports in one of the specified states
//                for the specified customer. The customer can be
//                a company, an account, or a particular user.
//
//
void getTroubleReports(
   in  RequestIdType              requestId,
   in  IdentityType               customer,
   in  TroubleReportStateListType statesToGet,
   out TroubleReportListType      troubleReportList,
   inout TagValueListTypeOpt      appSpecificData
```

```
)
raises(
   ProcessingFailureError
);

//
//  getTroubleReportsAttributes: Retrieves the requested attributes for
//                  all trouble reports in one of the specified states
//                  for the specified customer. The customer can be
//                  a company, an account, or a particular user.
//
//
void getTroubleReportsAttributes(
   in  RequestIdType                  requestId,
   in  IdentityType                   customer,
   in  AttributeIdListType            attributesToGet,
   in  TroubleReportStateListType     statesToGet,
   out GetResponseListType            getResponseList,
   inout TagValueListTypeOpt          appSpecificData
)
raises(
   InvalidDataReceived,
   ProcessingFailureError
);

//
//  getTroubleReportIdByServiceId: retrieves trouble report id on
//         specified service id.
//
//
void getTroubleReportIdByServiceId(
   in  RequestIdType                  requestId,
   in  IdentityType                   customer,
   in  ServiceIdType                  serviceId,
   out TroubleReportIdType            troubleReportId,
   inout TagValueListTypeOpt          appSpecificData
)
raises (
   NotFound,
   AccessDenied,
   ProcessingFailureError
);

//
// getServiceTroubleHistory: Returns a list of history records
//     representing prior trouble reports (currently closed)
//     that have resided on the specified service.
//

void getServiceTroubleHistory(
   in  RequestIdType                  requestId,
   in  IdentityType                   customer,
   in  ServiceIdType                  serviceId,
   out TroubleHistoryRecordListType   historyRecords,
   inout TagValueListTypeOpt          appSpecificData
)
raises (
   NotFound,
   AccessDenied,
   ProcessingFailureError
);

//
// bindToEventChannel: Performs the following:
//
```

```
    //      - Locate or create an event channel.
    //
    //      - Call <eventChannel>.new_for_consumers() to create a new
    //        consumerAdmin object for the customer (using AND sematics
    //        for the combination of the filters associated with the
    //        admin object and supplier proxy object(s)).
    //
    //      - Add filters to the admin object to provide for the delivery
    //        of all events for the customers trouble reports.
    //
    //      - Make appropriate QoS property settings to the admin object.
    //
    //      - Call <adminObject>.obtain_notification_push_supplier() to
    //        create a StructuredProxyPushSupplier object.
    //
    //      - Add new filters for all constrains in the constraintList to
    //        the structuredProxyPushSupplier object.
    //
    //      - Set all QoS values in the qosList on the
    //        structuredProxyPushSupplier object.
    //
    //      - Call <structuredProxyPushSupplier>.
    //              connect_structure_push_consumer(troubleEventConsumer)
    //        to connect to troubleEventConsumer (a push consumer) to
    //        the proxy supplier.
    //

    void bindToEventChannel(
          in RequestIdType                requestId,
          in IdentityType                 customer,
          in ConstraintListTypeOpt        constraintList,
          in QoSPropertiesTypeOpt         qosList,
          in TroubleEventConsumer         troubleEventConsumer,
          inout TagValueListTypeOpt       appSpecificData
    )
    raises (
       AlreadyBoundToEventChannel,
       UnsupportedQoS,
       InvalidConstraintList,
       ProcessingFailureError
    );

    //
    // unbindFromEventChannel:
    //
    // Disconnects the consumer from the event channel by calling
    // disconnect_structured_push_supplier() on the associated
    // structProxyPushSupplier object and then destroying that object.
    //
    void unbindFromEventChannel(
       in RequestIdType             requestId,
       in TroubleEventConsumer      troubleEventConsumer,
       inout TagValueListTypeOpt    appSpecificData
    )
    raises (
       NotFound,
       ProcessingFailureError
    );

}; // TroubleReportAdministrator

interface Notifications {

    /* The "typed" notifications are defined here so that the notification
     * structure can be documented in the IDL.  Notification users may
```

```
        * wish to support this interface.
        *
        * The rules used to construct and readthe structured events are as follows:
        *
        *    1. The domain_name string in the fixed header of the structure
        *       is set to "telecommunications."
        *
        *    2. The type_name string in the fixed header of the structure
        *       is set to the scoped name of the operation.  For
        *       example, "Trouble::Notifications::troubleReportChanged."
        *
        *    3. The event_name string in the fixed header of the structure
        *       is set to null.
        *
        *    4. Optional header fields may be included to for QoS.
        *
        *    5. Each parameter in the operation is placed in a
        *       name-value pair in the filterable body portion of the
        *       notification.  The fd_name string of the pair is set to
        *       the name of the paramenter, and the type placed in the
        *       associated fd_value will be the type specified for the
        *       parameter.
        *
        *    6. The unfilterable portion of the event body is set to null.
        */

       void troubleReportCreated(
          in  NotificationIdType        notificationId,
          in  RequestIdTypeOpt            requestId,
          in  UserIdTypeOpt             user,
          in  AccountTypeOpt              account,
          in  CompanyTypeOpt              company,
          in  TroubleReport             troubleReport,
          in  TroubleReportIdType         troubleReportId,
          in  DateTimeType              eventTime,
          in  ChangeEventType             eventData );

       void troubleReportChanged(
          in  NotificationIdType        notificationId,
          in  RequestIdTypeOpt            requestId,
          in  UserIdTypeOpt             user,
          in  AccountTypeOpt              account,
          in  CompanyTypeOpt              company,
          in  TroubleReport             troubleReport,
          in  TroubleReportIdType         troubleReportId,
          in  DateTimeType              eventTime,
          in  ChangeEventType             eventData );

    }; // Notifications

}; // ansi_t1_227

#pragma prefix ""


#endif // ansi_t1_227_IDL
```

J.2.1 Customer Types Module

```
#ifndef CustomerTypes_idl

#define CustomerTypes_idl
```

```
#include <Base.idl>

#pragma prefix "t1.org"

// Customer: Types associated with customers.
Module CustomerTypes {

   //---------------------------------------------------------
   // TYPES
   //---------------------------------------------------------
   typedef Base::StringOpt_t StringTypeOpt;

   typedef string AccountType;
   typedef string CompanyType;
   typedef string UserIdType;

   union AccountTypeOpt switch(boolean) {
     case TRUE: AccountType theValue;
   };

   union CompanyTypeOpt switch(boolean) {
     case TRUE: CompanyType theValue;
   };

   union UserIdTypeOpt switch(boolean) {
     case TRUE: UserIdType theValue;
   };

   enum IdentityChoiceType {
       UserChoice,
       AccountChoice,
       CompanyChoice
   };

   union IdentityType switch(IdentityChoiceType) {
     case UserChoice:    UserIdType user;
     case AccountChoice: AccountType account;
     case CompanyChoice: CompanyType company;
   };

   union IdentityTypeOpt switch(boolean) {
```

```
    case TRUE: IdentityType theValue;
  };

  struct CodesType {
    StringTypeOpt pIC;
    StringTypeOpt lPIC;
    StringTypeOpt lRN;
    StringTypeOpt oCN;
  };

  union CodesTypeOpt switch(boolean) {
    case TRUE: CodesType theValue;
  };
};

#pragma prefix ""

#endif // CustomerTypes_idl
```

J.2.2 Base IDL Module

```
#ifndef Base_idl
#define Base_idl

#pragma prefix "t1.org"

//============================================================
//
// Base: General use types and exceptions.
//
//============================================================

module Base {

  //-----------------------------------------------------
  // TYPES
  //-----------------------------------------------------

  // Null
  typedef char Null_t;
```

```
//Optional any type
union AnyOpt_t switch (boolean) {
   case TRUE: any theValue;
};


//Optional short type
union ShortOpt_t switch (boolean) {
   case TRUE: short theValue;
};


//Optional long type
union LongOpt_t switch (boolean) {
   case TRUE: long theValue;
};

//Optional boolean type
union BooleanOpt_t switch (boolean) {
   case TRUE: boolean theValue;
};

//Optional string type
union StringOpt_t switch (boolean) {
   case TRUE: string theValue;
};


//Quantity
typedef long  Quantity_t;

union QuantityOpt_t switch (boolean) {
   case TRUE: Quantity_t   theValue;
};


// RequestId_t - A unique id for the operation to provide audit at
//               the application layer (IIOP request id is not available
//               to programmer).
typedef string RequestId_t;
```

```
union RequestIdOpt_t switch(boolean) {
   case TRUE: RequestId_t theValue;
};


// InquiryId - A unique id for an inquiry that provides audit
//             at the application layer. (OBF Data Element)
typedef RequestId_t  InquiryId_t;

union InquiryIdOpt_t switch(boolean) {
   case TRUE: InquiryId_t theValue;
};

// ElementName  - data element name types
//
typedef string ElementName_t;
typedef sequence<ElementName_t> ElementNameList_t;

union ElementNameListOpt_t switch(boolean) {
  case TRUE: ElementNameList_t theValue;
};


//
// TagValueList - A list of tag-value that can be used to extend
//                certain IDL types without breaking existing
//                code (a hack, but is useful in certain situations).
//
struct TagValue_t {
   string tag;
   string value;
};

union TagValueOpt_t switch(boolean) {
   case TRUE: TagValue_t theValue;
};

typedef sequence<TagValue_t> TagValueList_t;

union TagValueListOpt_t switch(boolean) {
```

```
    case TRUE: TagValueList_t theValue;
};




//---------------------------------------------------------
// Exceptions
//---------------------------------------------------------



//---------------------------------------------------------
// MissingData: A required data element is missing.
//              The data element names are the IDL
//              field names that are missing (i.e.,
//              not populated).
//---------------------------------------------------------

exception MissingData {
    ElementNameList_t   missingElements;
};



//---------------------------------------------------------
// InvalidData: A data element is invalid. The data
//              element names are the IDL field names
//              that are invalid. The reason describes
//              the error (e.g., "too long",
//              "out-of-range").
//---------------------------------------------------------

struct ErrorDescriptor_t {
    ElementName_t   elementName;
    string          reason;
};

typedef sequence<ErrorDescriptor_t> ErrorDescriptorList_t;

exception InvalidData {
    ErrorDescriptorList_t  errorList;
};
```

```
    //-------------------------------------------------------
    // ProcessingFailure: A non-application error occurred
    //                    (i.e., the error is system related).
    //                    The code and description give
    //                    details on the error that occurred.
    //-------------------------------------------------------
    exception ProcessingFailureError {
        string            code;
        StringOpt_t       description;
    };


    //-------------------------------------------------------
    // AccessDenied: The caller does not have access to the
    //               requested resource.
    //-------------------------------------------------------
    exception AccessDenied {
        string            code;
        StringOpt_t       description;
        StringOpt_t       additionalInformation;
    };


    //-------------------------------------------------------
    // NotFound: The requested resource was not found
    //           requested resource.
    //-------------------------------------------------------
    exception NotFound {
        string         code;
        StringOpt_t    description;
        StringOpt_t    additionalInformation;
    };


};

#pragma prefix ""

#endif // Base_idl
```

J.2.3 Time Types Module

```
#ifndef TimeTypes_idl
#define TimeTypes_idl

#pragma prefix "t1.org"

//============================================================
//
// TimeTypes: Types associated with time.
//
//============================================================

module TimeTypes {

   //
   // Date
   //
   enum Month_t {
      January,
      February,
      March,
      April,
      May,
      June,
      July,
      August,
      September,
      October,
      November,
      December
   };

   typedef unsigned short Year_t;
   typedef unsigned short DayOfMonth_t;

   struct Date_t {
      Year_t          year; // at least four digits
      Month_t         month;
      DayOfMonth_t    dayOfMonth;
```

```
};

union DateOpt_t switch (boolean) {
  case TRUE: Date_t theValue;
};

typedef sequence<Date_t> DateList_t;

union DateListOpt_t switch (boolean) {
  case TRUE: DateList_t theValue;
};




//
// Time: Instance of Time
//
typedef unsigned short Hour_t;
typedef unsigned short Minute_t;
typedef unsigned short Seconds_t;
typedef unsigned short MilliSeconds_t;
typedef short UtcOffset_t;

struct Time_t {
    Hour_t           hour;       // 24 hours clock (i.e., 0-23)
    Minute_t         minute;
    Seconds_t        seconds;
    MilliSeconds_t   msecs;
    UtcOffset_t      UtcOffset;  // number of hours from UTC
};

union TimeOpt_t switch (boolean) {
  case TRUE: Time_t theValue;
};

typedef sequence<Time_t> TimeList_t;

union TimeListOpt_t switch (boolean) {
  case TRUE: TimeList_t theValue;
};
```

```
//
// DateTime: Instance of time including date.
//
struct DateTime_t {
    Date_t   date;
    Time_t   time;
};

union DateTimeOpt_t switch (boolean) {
  case TRUE: DateTime_t theValue;
};

typedef sequence<DateTime_t> DateTimeList_t;

union DateTimeListOpt_t switch (boolean) {
  case TRUE: DateTimeList_t theValue;
};




//
// Length: Length of Time.
//
typedef unsigned short Day_t;
struct Length_t {
    Year_t        years;  // at least four digits
    short         months;
    Day_t         days;
    Hour_t        hours;
    Minute_t      minutes;
    Seconds_t     seconds;
    MilliSeconds_t msecs;
};


union LengthOpt_t switch (boolean) {
  case TRUE: Length_t theValue;
};

typedef sequence <Length_t> LengthList_t;
```

```
union LengthListOpt_t switch (boolean) {
  case TRUE: LengthList_t theValue;
};

typedef sequence <LengthListOpt_t> LengthOptList_t;



//
// Interval: A period of time (from/to).
//
struct Interval_t {
    Time_t start;
    Time_t end;
};

union IntervalOpt_t switch (boolean) {
  case TRUE: Interval_t theValue;
};

typedef sequence <Interval_t> IntervalList_t;

union IntervaListOpt_t switch (boolean) {
  case TRUE: IntervalList_t theValue;
};



//
// Day Of Week
//
enum DayOfWeek_t {
    Sunday,
    Monday,
    Tuesday,
    Wednesday,
    Thursday,
    Friday,
    Saturday
};

union DayOfWeekOpt_t switch (boolean) {
  case TRUE: DayOfWeek_t theValue;
```

```
    };

    typedef sequence <DayOfWeek_t> DayOfWeekList_t;

    union DayOfWeekListOpt_t switch (boolean) {
      case TRUE: DayOfWeekList_t theValue;
    };

    typedef sequence<DayOfWeekOpt_t> DayOfWeekOptList_t;



    //
    // WeekMask: Periods of time within a week.
    //
    struct WeekMask_t {
        DayOfWeekList_t   daysOfWeek;
        IntervalList_t    intervalsOfDay;
    };

    union WeekMaskOpt_t switch (boolean) {
      case TRUE: WeekMask_t theValue;
    };

    typedef sequence <WeekMask_t> WeekMaskList_t;

    union WeekMaskListOpt_t switch (boolean) {
      case TRUE: WeekMaskList_t theValue;
    };

    typedef sequence<WeekMaskOpt_t> WeekMaskOptList_t;
};

#pragma prefix ""

#endif // TimeTypes_idl
```

J.2.4 Location Types Module

```
#ifndef LocationTypes_idl
#define LocationTypes_idl

#include <Base.idl>
```

```
#pragma prefix "t1.org"

//=============================================================
//
//  LocationTypes: Types associated with Locations (e.g.,
//                 Addresses, Location Codes, etc).
//
//=============================================================

module LocationTypes {

   //
   // NumberedAddress
   //
   struct NumberedAddress_t {
      Base::StringOpt_t     numberPrefix;
      string                number;
      Base::StringOpt_t     numberSuffix;
      Base::StringOpt_t     streetDirection;
      string                streetName;
      Base::StringOpt_t     streetSuffix;
      Base::StringOpt_t     streetType;
      Base::StringOpt_t     room;
      Base::StringOpt_t     floor;
      Base::StringOpt_t     building;
      string                city;
      string                stateOrProvince;
      Base::StringOpt_t     country;
      string                postalCode;
   };


   //
   // UnnumberedAddress
   //
   struct UnnumberedAddress_t {
      Base::StringOpt_t     streetDirection;
      string                streetName;
      Base::StringOpt_t     streetSuffix;
      Base::StringOpt_t     streetType;
      Base::StringOpt_t     room;
      Base::StringOpt_t     floor;
      Base::StringOpt_t     building;
      string                city;
      string                stateOrProvince;
      Base::StringOpt_t     country;
      string                postalCode;
   };


   //
   // Descriptive - Added for consistency with Local Ordering.
   //
   typedef UnnumberedAddress_t DescriptiveAddress_t;


   //
   // UnfieldedAddress - Added for consistency with Local Ordering.
   //
   struct UnfieldedAddress_t {
      string                address;
      string                city;
      string                stateOrProvince;
      Base::StringOpt_t     country;
      Base::StringOpt_t     postalCode;
```

```
};


//
// HouseNumberRangeAddress - Added for consistency with Local
// Ordering.
//
struct HouseNumberRange_t {
    string fromHouseNumber;
    string toHouseNumber;
};

typedef sequence<HouseNumberRange_t> HouseNumberRangeList_t;

struct HouseNumberRangeAddress_t {
    HouseNumberRangeList_t  houseNumberRangeList;
    Base::StringOpt_t       streetDirection;
    string                  streetName;
    Base::StringOpt_t       streetSuffix;
    Base::StringOpt_t       streetType;
    Base::StringOpt_t       room;
    Base::StringOpt_t       floor;
    Base::StringOpt_t       building;
    string                  city;
    string                  stateOrProvince;
    Base::StringOpt_t       country;
    string                  postalCode;
};


//
// CivicAddress
//
enum CivicAddressChoice_t {
    descriptiveAddressChoice,
    numberedAddressChoice,
    unnumberedAddressChoice

};
union CivicAddress_t switch (CivicAddressChoice_t) {
   case descriptiveAddressChoice:
      DescriptiveAddress_t       descriptive;
   case numberedAddressChoice:
      NumberedAddress_t          numbered;
   case unnumberedAddressChoice:
      UnnumberedAddress_t        unnumbered;
};

union CivicAddressOpt_t switch (boolean) {
    case TRUE: CivicAddress_t theValue;
};

typedef sequence<CivicAddress_t> CivicAddressList_t;

union CivicAddressListOpt_t switch(boolean) {
    case TRUE:  CivicAddressList_t theValue;
};


enum AltCivicAddressChoice_t {
    altDescriptiveAddressChoice,
    altNumberedAddressChoice,
    altUnnumberedAddressChoice,
    altUnfieldedAddressChoice,
    altHouseNumberRangeAddressChoice
```

```
};

union AltCivicAddress_t switch (AltCivicAddressChoice_t) {
   case altDescriptiveAddressChoice:
      DescriptiveAddress_t      descriptive;
   case altNumberedAddressChoice:
      NumberedAddress_t         numbered;
   case altUnnumberedAddressChoice:
      UnnumberedAddress_t       unnumbered;
   case altUnfieldedAddressChoice:
      UnfieldedAddress_t        unfielded;
   case altHouseNumberRangeAddressChoice:
      HouseNumberRangeAddress_t  houseNumberRange;
};

union AltCivicAddressOpt_t switch (boolean) {
   case TRUE: AltCivicAddress_t theValue;
};

// Person Reach
struct PersonReach_t {
   string       number;
   string       name;
};

// PersonReach Revision 1
struct PersonReachR1_t {
   string                        number;
   string                        name;
   Base::StringOpt_t                phone;
   UnfieldedAddress_t                 address;
   Base::StringOpt_t                email;
   Base::StringOpt_t                fax;
   Base::StringOpt_t                respon;
   Base::StringOpt_t                pager;
};

union PersonReachR1Opt_t switch (boolean) {
  case TRUE: PersonReachR1_t theValue;
};


//
// LocationCode (size is 8 or 11)
//
typedef string<11> LocationCode_t;

union LocationCodeOpt_t switch (boolean) {
  case TRUE: LocationCode_t theValue;
};


//
// LocalServingOffice
//
typedef string<6> LocalServingOffice_t; //NPA-NXX

union LocalServingOfficeOpt_t switch (boolean) {
  case TRUE: LocalServingOffice_t theValue;
};


//
// Locations - A & Z locations
//
```

```
    struct Locations_t {
        LocationCode_t location1;
        LocationCode_t location2;
    };

    union LocationsOpt_t switch (boolean) {
       case TRUE: Locations_t theValue;
    };


};

#pragma prefix ""

#endif // LocationTypes_idl
```

Annex K: Bibliography

(informative)

ATIS-0300211, *Information Interchange – Structure and Coded Representation of National Security and Emergency Preparedness (NS/EP) Telecommunications Service Priority (TSP) codes for the North American Telecommunications System.*[15]

ISO/IEC 9595:1991, *Information technology – Open systems interconnection – Common management information service definition.*[16]

ISO/IEC 9596-1:1991, *Information technology – Open systems interconnection – Common management information protocol – Part 1: Specification.* [16]

ISO/IEC 10040:1992, *Information technology – Open systems interconnection – Systems management overview.* [16]

ITU-T Recommendation X.200, *Reference model of open systems interconnection for CCITT applications.*[17]

ITU-T Recommendation X.208, *Specification of abstract syntax notation one.* [17]

ITU-T Recommendation X.730|ISO/IEC 10164-1:1992, *Information technology – Open systems interconnection – Systems management – Part 1: Object management function.* [17]

ITU-T Recommendation X.790, *Data Networks and Open System Communications – OSI Management – Trouble Management Function for ITU-T Applications.*[17]

ITU-T Rec. X.722|ISO/IEC 10165-4: 1992, *Information technology – Open systems interconnection – Structure of management information – Part 4: Guidelines for the definition of managed objects.*[17]

[15] This document is available from the Alliance for Telecommunications Industry Solutions (ATIS), 1200 G Street N.W., Suite 500, Washington, DC 20005. < https://www.atis.org/docstore/product.aspx?id=26795 >

[16] This document is available from the International Electrotechnical Commission. < http://www.iec.ch />

[17] This document is available from the International Telecommunications Union. < http://www.itu.int/ITU-T/ >