



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.790

(11/95)

**DATA NETWORKS
AND OPEN SYSTEM COMMUNICATIONS
OSI MANAGEMENT**

**TROUBLE MANAGEMENT FUNCTION
FOR ITU-T APPLICATIONS**

ITU-T Recommendation X.790

(Previously "CCITT Recommendation")

FOREWORD

The ITU-T (Telecommunication Standardization Sector) is a permanent organ of the International Telecommunication Union (ITU). The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1 (Helsinki, March 1-12, 1993).

ITU-T Recommendation X.790 was prepared by ITU-T Study Group 7 (1993-1996) and was approved under the WTSC Resolution No. 1 procedure on the 21st of November 1995.

NOTE

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

© ITU 1996

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

(February 1994)

ORGANIZATION OF X-SERIES RECOMMENDATIONS

Subject area	Recommendation Series
PUBLIC DATA NETWORKS	
Services and Facilities	X.1-X.19
Interfaces	X.20-X.49
Transmission, Signalling and Switching	X.50-X.89
Network Aspects	X.90-X.149
Maintenance	X.150-X.179
Administrative Arrangements	X.180-X.199
OPEN SYSTEMS INTERCONNECTION	
Model and Notation	X.200-X.209
Service Definitions	X.210-X.219
Connection-mode Protocol Specifications	X.220-X.229
Connectionless-mode Protocol Specifications	X.230-X.239
PICS Proformas	X.240-X.259
Protocol Identification	X.260-X.269
Security Protocols	X.270-X.279
Layer Managed Objects	X.280-X.289
Conformance Testing	X.290-X.299
INTERWORKING BETWEEN NETWORKS	
General	X.300-X.349
Mobile Data Transmission Systems	X.350-X.369
Management	X.370-X.399
MESSAGE HANDLING SYSTEMS	X.400-X.499
DIRECTORY	X.500-X.599
OSI NETWORKING AND SYSTEM ASPECTS	
Networking	X.600-X.649
Naming, Addressing and Registration	X.650-X.679
Abstract Syntax Notation One (ASN.1)	X.680-X.699
OSI MANAGEMENT	X.700-X.799
SECURITY	X.800-X.849
OSI APPLICATIONS	
Commitment, Concurrency and Recovery	X.850-X.859
Transaction Processing	X.860-X.879
Remote Operations	X.880-X.899
OPEN DISTRIBUTED PROCESSING	X.900-X.999

CONTENTS

	<i>Page</i>
1 Scope	1
1.1 Functionality	1
1.2 Field of application	1
1.3 Document structure	2
2 Normative References	2
2.1 Identical Recommendations International Standards	2
2.2 Paired Recommendations International Standards equivalent in technical content	3
2.3 Additional References	3
3 Definitions	4
4 Abbreviations	6
5 Conventions	7
5.1 Use of graphic string syntax	7
5.2 Use of list in attribute labels	7
5.3 Labelling conditional packages	7
5.4 Describing parameters in primitives	7
6 Requirements	7
6.1 Trouble management report creation	7
6.2 Tracking trouble management reports	9
6.3 Management of trouble management reports	9
6.4 Trouble management report clearing and closure	10
7 Model description	11
7.1 Introduction	11
7.1.1 Instantiation of troubles reports	11
7.1.2 An overview of the trouble management objects model	11
7.1.3 Trouble report states and status	15
7.2 Model components generic definitions	16
7.2.1 Object class definitions	17
7.2.2 Attribute type definitions	23
7.2.3 Error messages	34
8 Service description	34
8.1 Introduction	34
8.2 Kernel functional unit	36
8.2.1 Enter Trouble Report	36
8.2.2 Request Trouble Report Status	36
8.3 Request Trouble Report Format Functional Unit	36
8.4 Trouble History Event Notification Functional Unit	37
8.4.1 Trouble History Event Notification	37
8.4.2 Parameters	37
8.5 Review Trouble History Functional Unit	38
8.6 Add Trouble Information Functional Unit	38
8.7 Trouble Report Status/Commitment Time Update Notification Functional Unit	38
8.7.1 Trouble Report Status/Commitment Time Update Notification	38
8.8 Verify Repair Completion Functional Unit	39
8.9 Modify Trouble Administration Information Functional Unit	39

	<i>Page</i>
8.10 Trouble Administration Configuration Event Notification Functional Unit.....	40
8.10.1 Attribute Value Change Notification	40
8.10.2 Object CreationNotification	40
8.10.3 Object Deletion Notification.....	41
8.11 Trouble Report Progress Notification Functional Unit.....	41
8.11.1 Trouble Report Progress Notification	41
8.11.2 Parameters.....	41
8.12 Cancel Trouble Report Functional Unit.....	41
8.13 Extended Modify Trouble Administration Information Functional Unit	42
8.14 Delete Telecommunications Trouble Report functional unit.....	42
8.15 Refer Telecommunications Trouble Report functional unit	42
8.16 Transfer Telecommunications Trouble Report functional unit.....	42
8.17 Update State and Status functional unit	43
8.18 Repair Activity Object Functional Unit.....	43
8.19 Provider Trouble Report Control Functional Unit.....	43
8.20 Summary of functional units.....	43
9 Service mapping to Protocol	45
9.1 Element of procedure.....	45
9.1.1 Trouble History Event Notification Service procedures	45
9.1.2 Trouble Report Progress Notification Service procedures.....	47
9.2 List of items having templates in Annex A and Recommendation X.721	50
9.2.1 Objects	50
9.2.2 Attributes.....	50
9.2.3 Notifications.....	53
9.2.4 Actions	54
9.3 Negotiation of functional units	54
10 Relationship with other Standards.....	55
11 Conformance	55
11.1 Static conformance	55
11.2 Dynamic conformance	55
11.3 Management implementation conformance statement requirements.....	55
11.4 Conformance to the general purpose platform profile for CCITT Rec. X.733 ISO/IEC 10164-4..	56
11.5 Conformance to support managed object definition	56
Annex A – Managed objects	57
A.1 Managed object class and package definitions	57
A.1.1 Account.....	57
A.1.2 cnmService.....	57
A.1.3 contact	58
A.1.4 providerTroubleReport	60
A.1.5 repairActivity	61
A.1.6 service	62
A.1.7 telecommunicationsTroubleReport	63
A.1.8 troubleHistoryRecord.....	68
A.1.9 troubleReport	70
A.1.10 troubleReportFormatDefn.....	72
A.2 Attribute definitions.....	73
A.2.1 Account contact list.....	73
A.2.2 Account name	73
A.2.3 Activity code.....	73
A.2.4 Activity duration	73
A.2.5 Activity info	73

	<i>Page</i>
A.2.6 Activity person.....	73
A.2.7 Additional text	73
A.2.8 Additional trouble info list.....	73
A.2.9 Additional trouble status info.....	74
A.2.10 After hrs repair auth.....	74
A.2.11 Agent contact person.....	74
A.2.12 Agent contact object ptr	74
A.2.13 Alarm record ptr list.....	74
A.2.14 Alternate manager contact person.....	74
A.2.15 Alternate manager contact object ptr	74
A.2.16 Applicable managed object class list	75
A.2.17 Applicable managed object instance list	75
A.2.18 Authorization list	75
A.2.19 Begin time	75
A.2.20 Call back info list.....	75
A.2.21 Called number.....	75
A.2.22 Cancel requested by manager	75
A.2.23 Close out narr	75
A.2.24 Commitment time.....	75
A.2.25 Commitment time request.....	76
A.2.26 Contact company.....	76
A.2.27 Contact details.....	76
A.2.28 Contact function.....	76
A.2.29 Contact id.....	76
A.2.30 Contact names.....	76
A.2.31 Contact object ptr list	76
A.2.32 Contact type	76
A.2.33 Close out verification	76
A.2.34 Cust trouble tick num.....	77
A.2.35 Customer work center	77
A.2.36 Dialog.....	77
A.2.37 Electronic mail address	77
A.2.38 End time.....	77
A.2.39 Entry time.....	77
A.2.40 Escalation list	77
A.2.41 Facsimile telephone number list.....	77
A.2.42 Hand off center	77
A.2.43 Hand off location	78
A.2.45 Hand off person name.....	78
A.2.46 Hand off person ptr	78
A.2.47 Hand off time	78
A.2.48 Initiating mode	78
A.2.49 Last update time.....	78
A.2.50 A location access address.....	78
A.2.51 Z location access address	78
A.2.52 A location access hours.....	78
A.2.53 Z location access hours	79
A.2.54 A location access person	79
A.2.55 Z location access person	79
A.2.56 Maintenance org contact name.....	79
A.2.57 Maintenance org contact ptr.....	79
A.2.58 Maintenance org contact time	79
A.2.59 Maint service charge	79
A.2.60 Managed object access from time	79
A.2.61 Managed object access hours	79

	<i>Page</i>
A.2.62 Managed object access to time.....	80
A.2.63 Managed object instance	80
A.2.64 Managed object instance alias list.....	80
A.2.65 Manager contact person	80
A.2.66 Manager contact object ptr.....	80
A.2.67 Manager search key1	80
A.2.68 Manager search key2	80
A.2.69 Manager search key3	80
A.2.70 Manager search key list.....	81
A.2.71 Outage duration.....	81
A.2.72 perceived trouble severity	81
A.2.73 Preferred priority.....	81
A.2.74 Received time.....	81
A.2.75 Related trouble report list.....	81
A.2.76 Repair activity	81
A.2.77 Repair activity list	81
A.2.78 Repeat report.....	82
A.2.79 Responsible person name.....	82
A.2.80 Responsible person ptr	82
A.2.81 Restored time	82
A.2.82 Service alias list	82
A.2.83 Service description.....	82
A.2.84 Service location list.....	82
A.2.85 Service Id	82
A.2.86 Service type.....	83
A.2.87 Supported service name list	83
A.2.88 Suspect object list	83
A.2.89 Telephone number list.....	83
A.2.90 Trouble clearance person	83
A.2.91 Trouble detection time	83
A.2.92 Trouble found	83
A.2.93 Trouble location.....	83
A.2.94 Tr constrained to single value attr id list	84
A.2.95 Trouble report format object ptr.....	84
A.2.96 Tr format id.....	84
A.2.97 Trouble report id	84
A.2.98 Tr must be present attr id list	84
A.2.99 Tr may be present attr id list	84
A.2.100 Trouble report number list	84
A.2.101 Trouble report state	84
A.2.102 Trouble report status	84
A.2.103 Trouble report status time	85
A.2.104 Trouble report status window	85
A.2.105 Trouble type	85
A.2.106 Tsp priority.....	85
A.2.107 Type text	85
A.2.108 Unavailable service ptr.....	85
A.3 Error messages.....	85
A.3.1 Trouble Report Already Exists	85
A.3.2 Fallback Reporting.....	85
A.3.3 Can Not close.....	86
A.3.4 Trouble Report Must Be Present Attribute Missing.....	86
A.3.5 Cannot Verify or Deny at This Time	86
A.3.6 Trouble Report Change Denied	86

	<i>Page</i>
A.4 Events	86
A.5 Name Bindings	86
A.6 Abstract syntax	89
A.6.1	89
A.7 Rules of extensibility	102
Appendix I – Scenarios	103
I.1 Introduction	103
I.1.1 Trouble management environment	103
I.1.2 Trouble management roles.....	103
I.2 Creation of a trouble report.....	104
I.3 Processing and tracking of the trouble report	105
I.3.1 Clearing of the trouble report.....	105
I.3.2 Closure of the trouble report	105
I.3.3 Cancelling of the trouble report	105
I.4 History notification.....	105
Appendix II – Further work.....	106
II.1 Additional requirements not met in this Recommendation.....	106
II.2 Future work.....	106
Appendix III – Overview of the service model	106
Appendix IV – Trouble report format examples	108
Appendix V	108

SUMMARY

This Recommendation is concerned with the management of malfunction in systems and communications networks from the perspective of a provider of service and user of that service. In the Recommendation these malfunctions are referred to as “troubles”. A report format is defined to allow a user to report a trouble, which will then be progressed to resolution by a provider. During the resolution by the service provider, the service user may determine the current state of resolution by issuing a request for this information. When cleared the provider may notify the user. Particular types of troubles are included; however, the use of this Recommendation by a particular application may require trouble types specific to that application to be used – this is catered for in this Recommendation.

TROUBLE MANAGEMENT FUNCTION FOR ITU-T APPLICATIONS

(Geneva, 1995)

1 Scope

From time to time all systems, including communications networks, develop problems or malfunctions referred to in this Recommendation as “troubles”. A “trouble” in a communications network is a problem that has an adverse effect on the quality of service perceived by network users. When a trouble is detected, possibly as a result of an alarm report, a trouble report may be entered by a user or the system may raise a report automatically. Management of that trouble report is necessary to ensure that it receives attention and that the trouble is cleared to restore the service to its previous level of capability.

At the time of a trouble, a network may have been interworking with another network to provide a service, and the problem or malfunction may be due to the other network. Therefore it may be necessary to exchange trouble management information between management systems across interfaces which may be client to service provider or service provider to service provider interfaces and may represent inter-jurisdictional as well as intra-jurisdictional boundaries. In addition to exchanging information on trouble that has already been detected, advance information on service inaccessibility may also need to be exchanged. Thus, a service provider may need to inform a customer of future service inaccessibility (because of planned maintenance, for example). The scope of this Recommendation includes all of the above processes for exchange of management information.

1.1 Functionality

This Recommendation specifies the Trouble Management functionality for:

- Reporting of troubles on services or resources on a managed network or system;
- Tracking the progress of trouble to resolution;
- Clearing and closure of trouble.

In a network environment this encompasses computer networks, data networks and telephony networks. This Recommendation defines a trouble management model, application services and a number of objects and their attributes that are necessary for trouble management.

Requirements for the detection of a trouble, that is to say any cause that may lead to or contribute to a manager perceiving a degradation in the quality of service of one or more network services or one or more network resources being managed, are outside the scope of this Recommendation.

1.2 Field of application

In general, trouble management is the trouble reporting and tracking between Conformant Management Entities (CMEs) interoperating co-operatively towards resolution of a trouble. (No distinction is made between inter-jurisdictional and intra-jurisdictional interfaces.) A trouble report gets instantiated for this purpose. In cases where CME's are interoperating co-operatively towards the resolution of a trouble, it means that both manager and agent CME may have a shared responsibility for resolution of trouble.

The trouble management function Recommendation is one that may be used by a CME acting in:

- 1) **a manager role** to manage trouble(s) and any corresponding trouble report(s) that have been raised to an agent role CME for resolution;

- 2) **an agent role** responsible for resolving a trouble(s) and any corresponding trouble report(s) that have been raised to it by a manager role CME;
- 3) **both an agent and manager role** to manage trouble(s) and any corresponding trouble report(s) that have been raised internally, (i.e. to the part performing the agent role), by another part of itself performing the manager role. In this case the CME itself is responsible for resolving the trouble, but in addition, the CME may also inform other manager role CMEs that are part of the co-operating networks, if they are liable to be affected by the trouble or can help with trouble resolution.

The actual means and methods by which the manager perceives the trouble in the first place and the means and methods employed by the agent for detection and identification of the trouble are outside the scope of this Recommendation.

1.3 Document structure

Clause 1 provides the scope of the Recommendation. Clause 2 provides the normative references. Clause 3 lists the definitions of terms used. Clause 4 gives expansions of the acronyms used. Clause 5 provides conventions used. A description of the requirements for trouble management to meet the needs of communications network service providers, vendors and implementors of network management solutions, operators and users is found in clause 6. Clause 7 describes the trouble management model. Clause 8 describes the services used and clause 9 describes the protocols. Clause 10 gives the relationships of this Recommendation with other Standards while the conformance requirements are covered in clause 11.

Annex A provides the supporting ASN.1 productions for this Recommendation.

Appendix I describes trouble management scenarios.

Appendix II identifies requirements and further work items not covered in this Recommendation.

Appendix III provides an overview of the service model.

Appendix IV gives examples of Trouble Report Format.

Appendix V provides a pictorial representation of pointer attribute relationships between the trouble administration objects defined in this Recommendation.

2 Normative References

The following Recommendations | International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations | Standards are subject to revision, and parties to agreements based on this Recommendation are encouraged to investigate the possibility of applying the most recent editions of the Recommendations | Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation X.210 (1993) | ISO/IEC 10731:1994, *Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services*.
- ITU-T Recommendation X.217 (1995) | ISO/IEC 8649:1996, *Information technology – Open Systems Interconnection – Service definition for the Association Control Service Element*.
- ITU-T Recommendation X.680 (1994)/Amd. 1 (1995) | ISO/IEC 8824-1:1995/Amd. 1:1995, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation – Rules of extensibility*.

- CCITT Recommendation X.701 (1992) | ISO/IEC 10040:1992, *Information technology – Open Systems Interconnection – Systems management overview plus Technical Corrigendum 1 (1994) plus Technical Corrigendum 2 (1995)*.
- CCITT Recommendation X.721 (1992) | ISO/IEC 10165-2:1992, *Information technology – Open Systems Interconnection – Structure of management information: Definition of management information plus Technical Corrigendum 1 (1994)*.
- CCITT Recommendation X.722 (1992) | ISO/IEC 10165-4:1992, *Information technology – Open Systems Interconnection – Structure of management information: Guidelines for the definition of managed objects*.
- ITU-T Recommendation X.724 (1993) | ISO/IEC 10165-6:1994, *Information technology – Open Systems Interconnection – Structure of management information: Requirements and guidelines for implementation conformance statement proformas associated with OSI management*.
- CCITT Recommendation X.730 (1992) | ISO/IEC 10164-1:1993, *Information Technology – Open Systems Interconnection – Systems management: Object management function*.
- CCITT Recommendation X.731 (1992) | ISO/IEC 10164-2:1993, *Information technology – Open Systems Interconnection – Systems management: State management function plus Technical Corrigendum 1 (1995)*.
- CCITT Recommendation X.734 (1992) | ISO/IEC 10164-5:1993, *Information technology – Open Systems Interconnection – Systems management: Event report management function plus Technical Corrigendum 1 (1994)*.
- CCITT Recommendation X.735 (1992) | ISO/IEC 10164-6:1993, *Information technology – Open Systems Interconnection – Systems management: Log control function*.
- ITU-T Recommendation X.741 (1995) | ISO/IEC 10164-9:1995, *Information technology – Open Systems Interconnection – Systems management: Objects and attributes for access control*.

2.2 Paired Recommendations | International Standards equivalent in technical content

- CCITT Recommendation X.209 (1988), *Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1)*.

ISO/IEC 8825:1990, *Information technology – Open Systems Interconnection – Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)*.
- CCITT Recommendation X.710 (1991), *Common management information service definition for CCITT applications*.

ISO/IEC 9595:1991, *Information technology – Open Systems Interconnection – Common management information service definition*.

2.3 Additional References

- CCITT Recommendation M.3100 (1992), *Generic network information model*.
- ISO/IEC ISP 11183-1:1992, *Information technology – International Standardized Profiles AOM1n OSI Management – Management Communications – Part 1: Specification of ACSE, presentation and session Protocols for the use of ROSE and CMISE*.
- ISO/IEC ISP 11183-2:1992, *Information technology – International standardized profiles AOM1n OSI Management – Management Communications – Part 2: CMISE/Rose for AOM12 – Enhanced Management Communications*.
- ISO/IEC ISP 12059-0:1992, *Information technology – International Standardized Profiles – OSI Management – Common Information Management Functions – Part 0: Common definitions for management function profiles*.
- ISO/IEC ISP 12059-1:1992, *Information technology – International Standardized Profiles – OSI Management – Common Information for Management Functions – Part 1: Object Management*.

3 Definitions

For the purposes of this Recommendation, the following definitions apply:

- 3.1 agent:** As defined in Recommendation X.701, the Systems Management Overview (SMO), but with the following restriction. With respect to a particular telecommunications service (or resource) instance, it shall be possible to manage the service with one system playing the manager role, and the other playing the agent role.
- 3.2 alias:** Another name, besides the object identifier, by which a trouble report may be known, referenced or identified (usually by the customer).
- 3.3 application entity:** The aspects of an application process pertinent to OSI.
- 3.4 application association:** A co-operative relationship between two application entities, formed by their exchange of application protocol control information through their use of presentation services.
- 3.5 application context:** An explicitly identified set of application service elements, related options, and any other necessary information for the interworking of application entities on an application association.
- 3.6 associated alarms:** Alarms directly related to a given identified trouble.
- 3.7 attribute:** Information concerning a managed object used to describe (either in part or in whole) that managed object. This information consists of an attribute type and its corresponding attribute value (single-valued) or values (multi-valued).
- 3.8 attribute type:** The component of an attribute that indicates the class of information given by that attribute.
- 3.9 attribute value:** A particular instance of the class of information indicated by an attribute type.
- 3.10 cancelled:** A manager can request the agent to “cancel” a trouble report. The manager wants to abort this trouble report (either because it was entered in error or because there is no longer any trouble condition). Under certain conditions (e.g. the trouble has not been dispatched or tested), the agent will “cancel” the trouble report by updating its status to “closed-out by customer request.” “Cancelling” a trouble report may also have business ramifications beyond the scope of this Recommendation (e.g. whether the customer must pay for the trouble report).
- 3.11 clearing trouble reports:** An assertion by an agent that actions which are identified in the trouble report or the repair activity object instances have been satisfactorily performed to resolve the trouble, or that such actions are no longer necessary, such that in either case the trouble report is a candidate for closure.
- 3.12 client:** User of a service provided by a system or a network.
- 3.13 closed-out:** A trouble report is considered “closed-out” when the agent determines that the reported trouble has either been cleared or no longer exists, and the agent updates the trouble report status to indicate the trouble report is “closed-out”. Only an agent can change the trouble report status to “closedOut”. The status of a trouble report might change to “closedOutByCustReq” as a result of a request to cancel the trouble report from the manager.
- 3.14 closing trouble reports:** An assertion by an Agent that the trouble is resolved such that the cleared trouble report may only be processed further to generate a trouble history record and/or be deleted.
- 3.15 conformant management entity:** A real open system which supports the interoperable interface defined in this Recommendation.
- 3.16 contact:** A person who can provide additional information about the trouble on behalf of the manager or the agent.
- 3.17 customer:** The customer is a user of telecommunications services provided by a service provider. Specifically, in the context of this Recommendation, the customer is a user who chooses to use the OS (Operations System) -to-OS OSI interface for network management across jurisdictions in order to achieve control of the telecommunications services (or resources) being used. The customer (or customer representative) acts in the manager role.

There is no requirement that the interface be confined to cases where there is a traditional telecommunication service customer to service provider relationship between the parties. Two telecommunications service providers (carriers) may use this interface to exchange trouble reports in situations where their networks interwork in order to provide service to an end user. In that case, the Customer role may change from situation to situation. However, in any particular situation, one carrier will be the customer and will act in the manager role, while the other will be the supplier and will act in the agent role.

- 3.18 defer:** To postpone work on, or set aside, a trouble report until such time as when appropriate conditions are met and it can be progressed further.
- 3.19 event:** An instantaneous occurrence that changes the global status of an object. This status change may be persistent or temporary, thus allowing for surveillance, monitoring, and performance measurement functionality, etc. Events may or may not generate reports; they may be spontaneous or planned; they may trigger other events or may be triggered by one or more other events.
- 3.20 escalating a trouble report:** Identifying a trouble report which is to receive urgent and immediate supervisory attention to resolve the trouble.
- 3.21 fault management:** Fault Management consists of a set of functions that enable the detection, isolation, and correction of abnormal operation of the telecommunications network and its environment.
- 3.22 jurisdiction:** This refers to the functional separation of telecommunications networks. A jurisdiction is one of the following four types:
- a) Local Exchange Carrier Network;
 - b) Interexchange Carrier Network;
 - c) End User Network;
 - d) Some combination of the above.
- 3.23 manager:** As defined in Recommendation X.701, the Systems Management Overview (SMO), but with the following restriction. With respect to a particular telecommunications service (or resource) instance, it shall be possible to manage the service with one system playing the manager role, and the other playing the agent role.
- 3.24 outage:** Unavailability of a service or resource.
- 3.25 perceived severity:** The seriousness of the problem as seen by the person reporting the trouble.
- 3.26 priority:** The degree of urgency with which the manager requires resolution of the problem.
- 3.27 service:** This term represents telecommunications capabilities that the customer buys or leases from a service provider. Service is an abstraction of the network-element-oriented or equipment-oriented view. Identical services can be provided by different network elements, and different services can be provided by the same network elements.
- 3.28 service provider:** A system or a network that provides a telecommunication service to a customer. In the context of this document, a service provider is specifically a provider of telecommunications services who offers the OS-to-OS OSI interface to allow a customer the capability for network management across jurisdictions in order to control the services (or resources) being provided (See *Customer*). A service provider acts in the agent role.
- There is no requirement that the interface be confined to cases where there is a traditional telecommunication service customer to telecommunication service provider relationship between the parties. It is certainly possible that two telecommunications carriers, whose networks interwork to provide a telecommunications service to an end user, may use this interface. In that case, the customer and service provider roles may change from situation to situation. However, in any particular situation, one carrier will be the customer and have the manager role, while the other will be the supplier, and will have the agent role.
- 3.29 status of a trouble report:** The stage that has been reached by a trouble report since its instantiation/creation while the trouble is being resolved.
- 3.30 time-stamp:** A time value used to indicate when a particular activity, action or an occurrence of an event took place.
- 3.31 trouble:** Any cause that may lead to or contribute to a manager perceiving a degradation in the quality of service of one or more network services or one or more network resources being managed.
- 3.32 trouble administration:** Trouble Administration consists of a set of functions that enable troubles to be reported and their status tracked. Trouble Administration services include request trouble report format, enter trouble report, add trouble information, cancel trouble report, request trouble report status, review trouble history, attribute value

change notification (e.g. trouble report status/commitment time), object creation/deletion (trouble report), verify trouble repair completion, and modify trouble administration information.

3.33 trouble history record: A record of selected information from a trouble report that is retained for historical purposes after the trouble report is closed.

3.34 trouble management: The trouble reporting and tracking between CMEs interoperating co-operatively towards resolution of a trouble. (No distinction is made between inter-jurisdictional or intra-jurisdictional interfaces.)

3.35 trouble reporting: The act of communicating that a trouble has been detected so that trouble management may be used in its resolution.

3.36 trouble resolution: It is the process of diagnosis and repair action required to clear a problem. It includes the process of assigning specific work items or overall responsibility for clearing and closing the trouble report.

3.37 trouble tracking: The ability to follow the progress of a trouble report from its creation through to its closure.

3.38 trouble type: The description or category of the trouble that was detected.

4 Abbreviations

For the purposes of this Recommendation, the following abbreviations are used.

ANSI	American National Standards Institute
ASN.1	Abstract Syntax Notation One
CME	Conformant Management Entity
CMIS	Common Management Information Service
CMISE	Common Management Information Service Element
CNM	Customer Network Management
DN	Distinguished Name
FU	Functional Unit
GNM	General Network Model
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
MAPDU	Management Application Protocol Data Unit
NE	Network Element
NOC	Network Operations Centre
OAM&P	Operations, Administration, Maintenance & Provisioning
OS	Operations System
OSI	Open Systems Interconnection
PTR	Provider Trouble Report
RDN	Relative Distinguished Name
SMAPM	System Management Application Protocol Machine
SMO	Systems Management Overview
TMN	Telecommunications Management Network
TSP	Telecommunication Service Priority
TTR	Telecommunications Trouble Report

5 Conventions

This clause explains the conventions used throughout this Recommendation.

5.1 Use of graphic string syntax

By convention, GraphicString may have country or language specific limitations.

5.2 Use of list in attribute labels

By convention, attributes with a “List” suffix indicate that the attribute is set valued.

5.3 Labelling conditional packages

By convention, conditional packages are prefixed with an abbreviation of the object class, e.g. Package1 in the Trouble Report object would be labeled x790Package1.

5.4 Describing parameters in primitives

The definition of certain Fault Management services in this Recommendation includes a table that lists the parameters of its primitives. For a given primitive, the presence of each parameter is described by one of the following values:

- M The parameter is mandatory.
- (=) The value of the parameter is equal to the value of the parameter in the column to the left.
- U Use of the parameter is a service–user option.
- The parameter is not present in the interaction.
- C The parameter is conditionally present.
The condition(s) are defined by the text that describes the parameter.

The service definitions in clause 9 are described using the service definition conventions specified in Recommendation X.210.

6 Requirements

6.1 Trouble management report creation

- 1) A trouble report may be created on a resource or service on a managed network (e.g. a circuit, a dialed number, a switched access number or service identifier), or a computer system, when:
 - an agent role CME detects a trouble and automatically generates a trouble report, for instance due to threshold limits being exceeded or alarm notifications;
 - a person in the manager’s organization reports a trouble to the agent by a procedure other than using the interoperable interface, and as a result the agent creates a trouble report;
 - a manager role CME requests that a trouble report be created by the agent role CME, i.e. using the interoperable interface;
 - an agent role CME wants to create a trouble report to specifically notify the manager role CME that planned maintenance will be carried out at a given time and that all or parts of the service(s), resource(s), network or system will be inaccessible during that time. In this case the trouble management function is used to notify the manager that planned maintenance action is scheduled.
- 2) A trouble report used during the trouble resolution procedures may have associated with it one or more:
 - other trouble reports used for reporting the trouble; or
 - alarms triggered by a single common problem.

- 3) Correlation of alarm information is outside the scope of this Recommendation.
- 4) The agent should be able to select the format of the trouble report with the exception of any compulsory information common to all trouble reports which must be present. The agent may select the format based on the managed object instance, or the managed object class of the managed object instance, against which the trouble is reported. The manager should be able to retrieve the applicable trouble report format to be used, prior to reporting a trouble.
- 5) It should be possible to have multiple distinct trouble reports created against the same managed object instance.
- 6) A trouble report shall be time-stamped at creation.
- 7) It should be possible for the trouble report originator, either agent or manager, to include in the trouble report the trouble type and additional information about the trouble. For example, the nature of the problem being reported, the source of the trouble report (external, internal, customer, etc.) and descriptive information in free format text.
- 8) The trouble report originator, either agent or manager, should be able to supply the identity of the person who originated the trouble report and/or the person who last modified information in the trouble report.
- 9) The trouble report originator, either agent or manager, should be able to supply the perceived trouble severity in terms of the effect on the managed object instance (either a service or a resource on a managed network).
- 10) The trouble report contains the identity of the service or network resource against which the trouble is being reported and may contain additional suspect objects and the location of the trouble.
- 11) The trouble report originator, either agent or manager, should be able to supply the priority to indicate the urgency for resolution of the trouble as seen by the originator relative to other outstanding trouble reports it has reported.
- 12) The manager should be able to identify a contact person(s) and also an alternative contact person(s) who can provide additional information to the agent about the trouble on the managed object instance.
- 13) The agent may also provide a contact who can provide additional information about the trouble.
- 14) It should be possible to have alias(es) for the trouble report to provide backwards compatibility to existing trouble handling systems, e.g. the trouble report identity derived using existing internal trouble management practices may be used.
- 15) The trouble report may contain information about associated fault reports, e.g. alarms, other trouble reports and test results, to assist the trouble resolution process.
- 16) For the purpose of trouble reporting, tracking or monitoring it should be possible for a manager to associate several trouble reports together to indicate a common problem. (The criteria for grouping or regrouping trouble reports is outside the scope of this Recommendation.)
- 17) It should be possible for either the agent or manager to attach comments in free format text to the trouble report.
- 18) The trouble report originator, either agent or manager, should be able to specify the time at which the trouble was detected, as it is likely to be different from the time at which the trouble report was created.
- 19) An agent should be able to provide the manager with an identifier for a created trouble report such that the manager can subsequently use it to uniquely identify the trouble report for the purpose of reporting, tracking or monitoring trouble and requesting event forwarding.
- 20) In some cases it is possible that the agent system may not be able to update the trouble report information. In such a situation, provided the manager has requested event forwarding on events from the trouble report, the manager could be informed that the trouble report is disabled and its information cannot be updated.

- 21) The manager may be able to indicate that the managed object against which the trouble report is being generated was recently created or another trouble was reported on this object in the recent past. (The mechanism used for the association is outside the scope of this Recommendation.)
- 22) A trouble report should contain sufficient information to enable it to be directed to the appropriate repair and/or maintenance support location, also commonly referred to as a Work Centre, for trouble resolution.

6.2 Tracking trouble management reports

- 1) A manager should be able to track the progress of a trouble report to resolution by requesting event forwarding. This includes the ability to request the trouble report state and or status from the agent.
- 2) The state and or status of the trouble report should be updated as the problem it relates to is progressed to resolution. The manager should be notified about the change in state or status of the trouble report if it has requested the forwarding of events from the agent. The time-stamp for the status change may be retained by the trouble report to facilitate subsequent analysis by the manager or agent.
- 3) The time, at which the trouble report was last updated, should be recorded. The identity of the manager, agent or person who did the last modification should also be recorded.
- 4) The trouble report should maintain a running account of information on the activities taken to diagnose, test and repair the trouble, the repair type, the length of time spent on each activity, whether the activity is billable or not and the equipment involved in each activity. This information may be provided by either the manager or the agent or both. In addition to the running account it should be possible to summarise key information, such as, how long was spent on an activity, whether the work is billable or not, etc.
- 5) The current agent/person working to resolve the problem should be identified in the trouble report.
- 6) The agent should allow the manager to:
 - view specific trouble reports;
 - view a specific subset of trouble reports, e.g. active (not closed), cleared, updated trouble reports;
 - view historical trouble information for an object if it is still available.
- 7) It should be possible to view (scope and filter) trouble reports by customer and or service type.
- 8) The agent should be able to provide autonomous trouble report status reports to the manager over the interoperable interface within the intervals requested by the manager.

6.3 Management of trouble management reports

- 1) It should be possible for the agent to allocate the person/agent with overall responsibility for resolving the trouble.
- 2) It should be possible for the manager to request escalation of a trouble report under certain circumstances, for example, if the trouble has not been resolved in a given period of time, known as time based escalation. Subsequent to receiving a request to escalate a trouble report, the agent should be able to indicate the person the trouble report was escalated to. (Business level agreements, which may be required to define the rules for escalation, are outside the scope of this Recommendation.)
- 3) It should be possible to automatically notify an appropriate contact within the agent role conformant management entity, based on a time based escalation.
- 4) A manager should be able to request the cancellation of a trouble report by the agent and indicate in the trouble report the person making the request.

- 5) It should be possible to defer work on a trouble report. For example, it may be necessary to suspend repair work because access to the faulty resource has been prevented by the manager. The deferral time may be limited by a time-out value or a duration specified by the manager. The agent should notify the manager when a trouble report has been deferred.
- 6) In order to protect the information in a trouble report from being updated and/or modified by unauthorised users, suitable access control mechanisms should be used. Thus, a manager could be permitted to modify only certain specific attributes in an open trouble report.
- 7) The manager should be able to request the following information:
 - an indication when an instance of a trouble report is created;
 - an indication when an instance of a trouble report is deleted;
 - an indication when a trouble report's information is changed, for example, the escalation of a trouble report, a trouble report status change, or a commitment time update, etc.
- 8) The trouble report should contain information to enable the quality of service of the repairer to be determined, for example, information such as, the repairer's name, the time at which the repairer was called and the time at which the repairer arrived to attend to the trouble.
- 9) The manager should be able to provide the required repair completion time and obtain from the agent the expected repair time, and subsequently the actual repair time to resolve the trouble.
- 10) The manager should be able to update an active trouble report with new information, e.g. additional trouble related information, contact change, etc.
- 11) The manager should be able to authorise repairs to be made at the manager's location outside of normal business hours, and also be able to identify the permitted access times for a given location.
- 12) It should be possible to perform statistical processing on trouble reports, for example, to obtain statistics on the mean time between troubles for a service, instance of a service, network resource or an instance of a resource or the mean time to repair a trouble, etc. (The actual algorithms for statistical analysis are outside the scope of this Recommendation.)
- 13) Upon receiving a request for authorisation related to specific repair activities from the agent, a manager should be able to provide authorisation over the interoperable interface.

6.4 Trouble management report clearing and closure

- 1) It should be possible to compute/determine the outage duration of the equipment/service from the repair activity information held in the trouble report or the Repair Activity object instance if present. This outage duration may then be added to the trouble report at the time of closure. (The algorithm for computing/determining the outage duration is outside the scope of this Recommendation.)
- 2) The manager may be allowed to verify closing can proceed, prior to closure of the trouble report by the agent. When the clearing of the problem related to the trouble report is verified by the manager, the time of verification and the identity of the person doing the verification should be captured in the trouble report. In the event that the manager denies or delays closure verification, then the agent should have the option of closing the trouble report.
- 3) After receiving the appropriate notification for trouble reports that require verification by the manager before closure, and the manager fails to respond and/or the trouble report clearance is not verified then the agent should be able to deny the manager the ability to modify some of the information in the trouble report. (The details of how the agent ensures that the appropriate notification has been received by the manager and the time period within which he has to respond to the request are beyond the scope of this Recommendation.)

- 4) In cases where CMEs are interoperating co-operatively towards the resolution of a trouble, it should be possible for a manager to update the trouble state and status and related information where the manager has shared responsibility towards trouble resolution.
- 5) When a trouble report is closed it may be retained for a period in the agent and a subset of the trouble report's information should be logged by the agent to provide a historical record. It should also be possible to report this information to the manager across the interoperable interface and for the manager to review these trouble history records.
- 6) It should be possible for the agent to delete a trouble history record from the log after a period specified by business agreement.
- 7) The actual cause of the trouble found, together with the solution provided and other relevant factors, should be recorded in the trouble report.

7 Model description

7.1 Introduction

The trouble management model describes the objects in the trouble management agent CME, and the relationship between them.

Trouble management is initiated by a request to the agent to create a trouble report (trouble report is used here, and elsewhere in this Recommendation, in a generic sense to represent the different types of instantiable report objects, whereas Trouble Report refers to the non-instantiable super-class in the trouble management model).

The trouble report contains information necessary for a manager to manage and track the trouble and the agent to manage and resolve the trouble in a client to service provider environment. In a service provider to service provider environment, a manager may co-operate in the trouble resolution process and take specific steps in its area of responsibility to resolve a trouble. The information contained within a trouble report and that concerned with the management of it may need to pass across the interoperable interface between two CMEs. Once created the trouble report progresses from queued through to clearance and closure states as a result of actions carried out normally by the CME acting in the agent role during the course of resolving the trouble. Status and state transitions may also occur as a result of intervention by the manager role CME in the service provider to service provider environment.

7.1.1 Instantiation of trouble reports

The model allows a trouble report, to be generated by an agent role CME as a consequence of:

- a trouble detected, automatically or otherwise, by the agent role CME itself;
- a trouble reported by means other than over an interoperable interface, by a person;
- a trouble reported over the interoperable interface, by a manager role CME;
- scheduled maintenance action which the agent role CME wishes to communicate to the manager role CME.

Refer to Figure 7-1.

7.1.2 An overview of the trouble management objects model

The Trouble Management Model defines a non-instantiable super-class (Trouble Report Object class), and two instantiable sub-classes (Provider Trouble Report, and Telecommunications Trouble Report). A Trouble Report Format Definition object class has also been defined, as well as a Trouble History Record object class and a Repair Activity object class. The inheritance relationship between these objects is shown in Figure 7-2. Figure 7-3 provides a diagrammatic summary of the model.

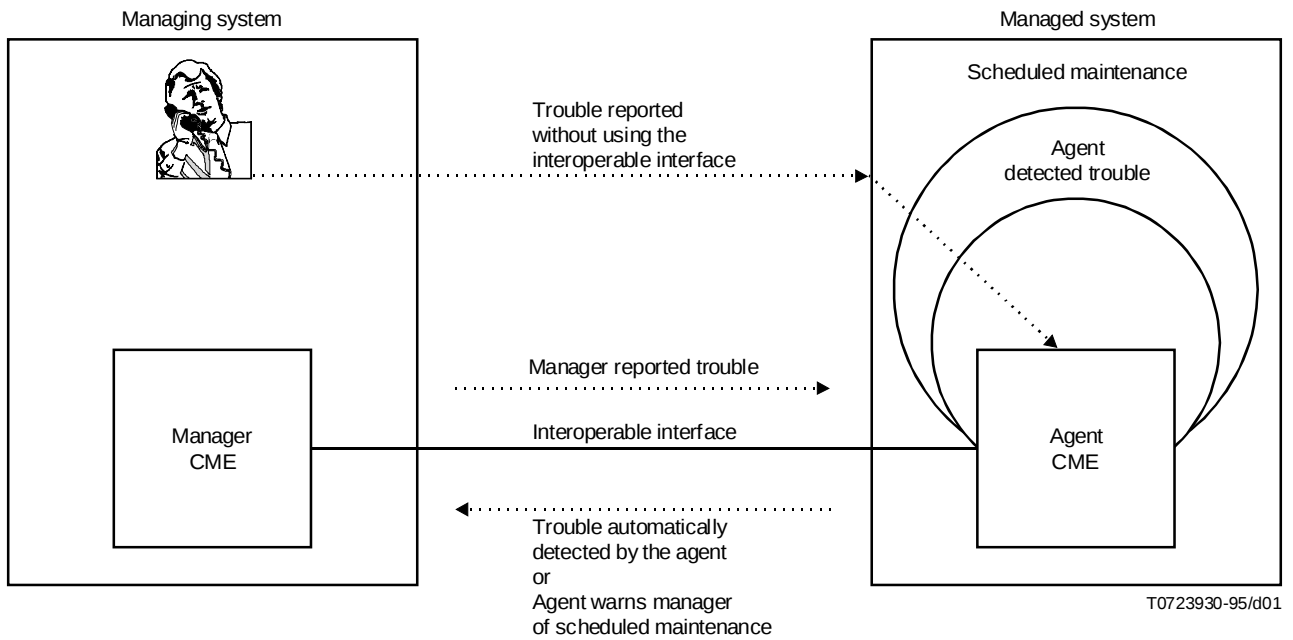


FIGURE 7-1/X.790
Trouble management report creation

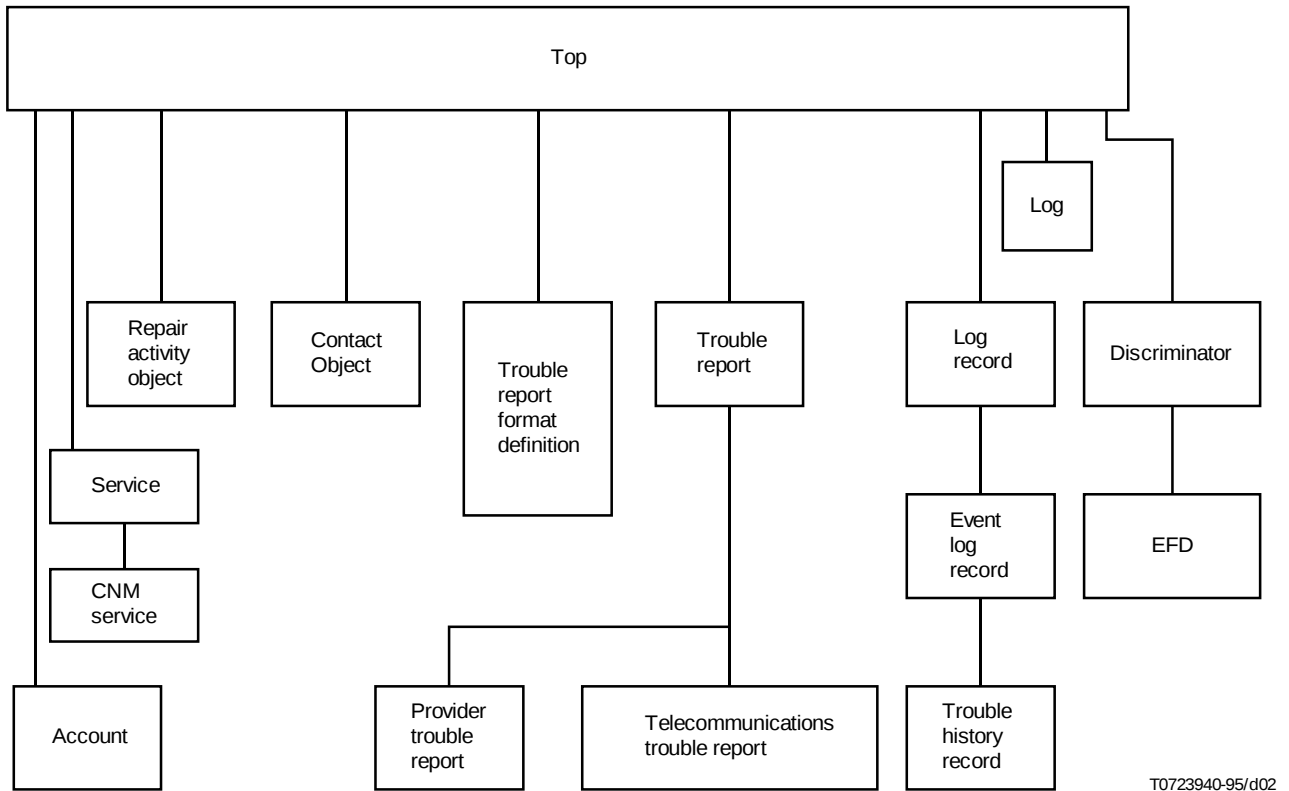


FIGURE 7-2/X.790
Inheritance hierarchy

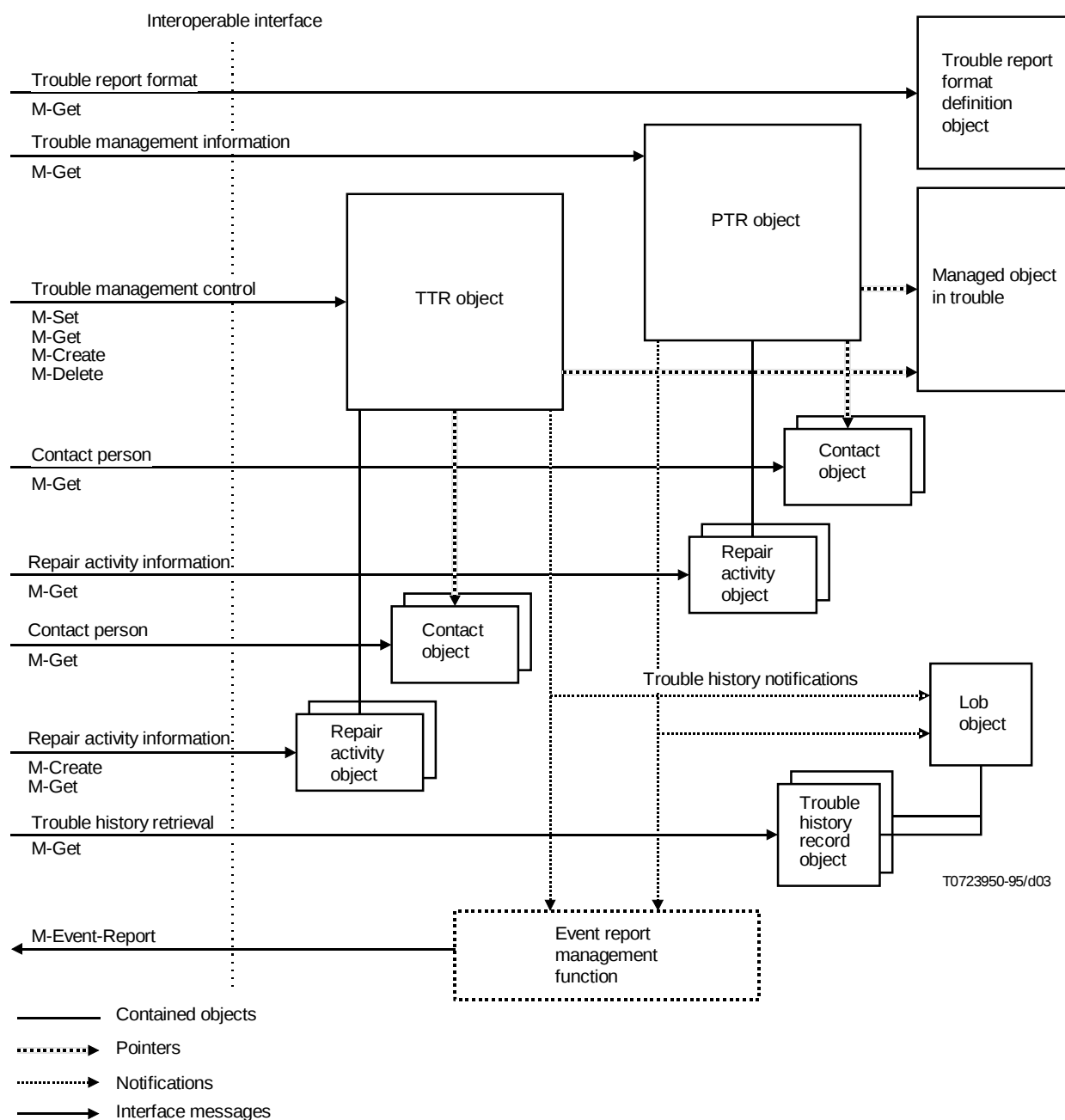


FIGURE 7-3/X.790
Trouble management conceptual model overview

NOTE – In addition to the Telecommunications Trouble Report instantiation shown in above, a TTR object instance may be locally instantiated by the agent.

The scope of the Recommendations in this model is targeted at interfaces between jurisdictional boundaries (i.e. Local Exchange Carrier Network, Interexchange Carrier Network, End User Network, or some combination of these). In other words, the scope of the work is focussed primarily on information that is made visible to a client by a service provider. The Recommendations do not specifically include interfaces between management systems within jurisdictions; however, its use in such environments is not precluded.

The trouble management function Recommendation models a client to service provider relationship and a service provider to service provider relationship with the Telecommunications Trouble Report object and covers trouble management over an interoperable interface between two CMEs whether constrained by jurisdictional boundaries or not. In order to allow for the differing requirements of the client to service provider interface and the service provider to service provider interface however, separate profiles of the Telecommunications Trouble Report are necessary.

In the client to service provider case, a client may be able to request certain information related to a trouble report while the service provider has responsibility for trouble resolution functions. In this case, the distinction between the agent CME and the manager CME is that the trouble report managed object instance resides in the agent CME, and the agent CME performs trouble resolution functions, while the manager CME is only allowed to perform trouble tracking functions (but not trouble resolution functions).

In the service provider to service provider case, both service providers may be able to perform the same set of trouble resolution functions on the same trouble report. The only distinction that can be drawn between the agent role CME and the manager role CME in this case is that the trouble report managed object instance resides in the agent role CME. Note that the functions that can be performed by a manager and those that can be performed by an agent may be constrained using the Security services to control:

- access to the object, as well as;
- the permission to modify specific individual attributes.

However, such distinctions between manager and agent with respect to the allowable trouble management functions that can be performed is beyond the scope of this trouble management function Recommendation.

The Telecommunications Trouble Report is a superset of the information necessary for both the client to service provider and a service provider to service provider relationships. The distinction between a client to service provider relationship and a service provider to a service provider relationship on an association will be by the selection of a suitable profile and negotiation of appropriate functional units.

The Provider Trouble Report object has been defined to address specific additional requirements covered in clause 2. The Provider Trouble Report object primarily covers the case where the service provider wants to make visible troubles covering routine maintenance to a customer. The Provider Trouble Report object is created by the agent role CME to specifically notify the manager role CME that planned maintenance will be carried out at a given time and all or parts of the service(s), resource(s), network or system will be inaccessible during that time. In this case, the trouble management function is used to notify the manager that planned maintenance action is scheduled to prevent future trouble.

The Trouble History Record object is used to log selected information of reported instances of troubles that have been closed.

Instances of the Repair Activity object capture the activities performed on the individual reported trouble instance as it progresses from creation to closure. The Repair Activity List attribute, which provides an alternative mechanism for storing repair information, is not present in a trouble report instance, when Repair Activity objects are instantiated.

During the trouble resolution process, human actions may be required to perform specific functions. Some of this person-related information may be unique to a trouble report instance, and hence best represented as an attribute (e.g. Z Location Access Person Attribute). Other person-related information may be applicable to many trouble reports and stable over an extended period, in which case it may be more appropriate to represent it via an object instance and point to this object from a trouble report instance (e.g. Responsible Person Ptr.). The Contact object is used for this purpose.

The trouble management model allows for multiple trouble report formats. Each trouble report format is a predefined combination of trouble report attributes. The trouble report applicable to a particular service or resource instance can be dynamically specified by the service provider through the Trouble Report Format Definition object. The appropriate instance of the Trouble Report Format Definition object to be used to report a trouble on a service/resource can be selected on an object class basis or on an object instance basis.

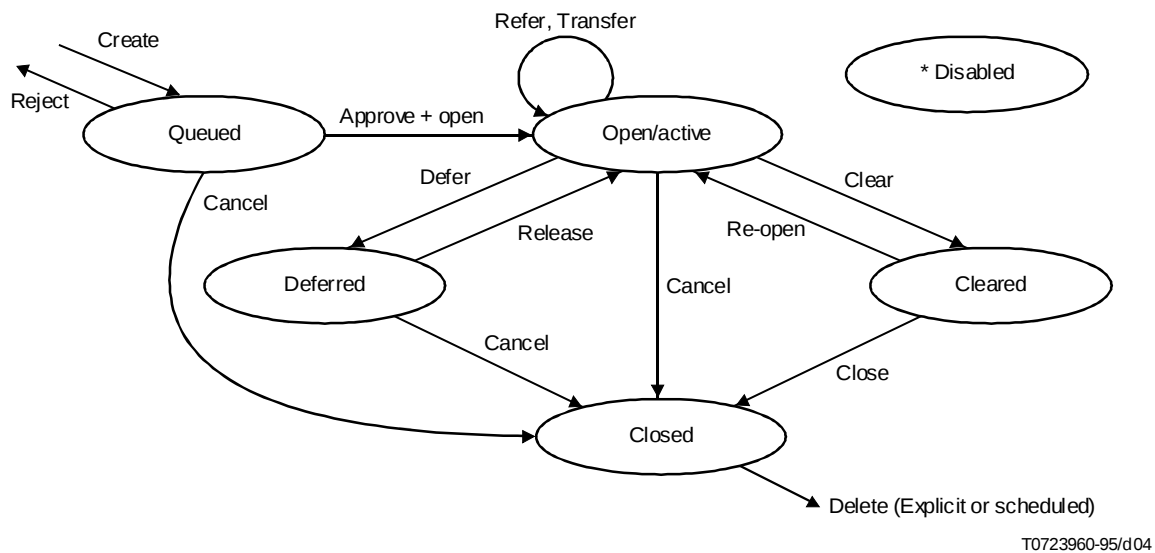
When the trouble report format is explicitly defined through the Trouble Report Format Definition object for a service or a resource, a trouble report instance is composed of:

- attributes that are specified as mandatory by the trouble report object class Recommendation;
- attributes in conditional packages of the trouble report which are specified as “must be present” by the appropriate instance of the Trouble Report Format Definition object;
- and optionally attributes in conditional packages of the trouble report which are specified as “may be present” by the appropriate instance of the Trouble Report Format Definition object.

7.1.3 Trouble report states and status

Referring to the State transition diagram in Figure 7-4, a trouble report may go through any of six states during its life cycle. In addition, a Trouble Status attribute is defined which qualifies the state (finer granularity) e.g. cleared awaiting customer verification. The time at which the status attribute change is also captured in the trouble report.

This subclause defines the trouble report states.



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

NOTE – In this model escalation is not considered to be a separate state, rather it is modelled as a function that applies to a trouble which is in the open/active state.

FIGURE 7-4/X.790
State transition diagram for trouble reports

7.1.3.1 Queued

A trouble report is in a queued state when it has been instantiated but the trouble resolution process has not yet been initiated.

A trouble report which is in the queued state may be cancelled by the manager. The agent on receiving such a request will attempt to close the trouble report.

7.1.3.2 Open/active

The trouble report becomes “open/active” when appropriate actions to resolve the trouble are initiated.

An “open/active” trouble report may be “referred” to another Hand-off Person, or “transferred” to another Responsible Person for further processing. The state however remains unchanged as “open/active”.

A trouble report in the open/active state may be cancelled by the manager. The agent on receiving such a request will attempt to close the trouble report.

7.1.3.3 Deferred

This state indicates that corrective action to resolve the trouble has been postponed. This can occur when the faulty resource is inaccessible for a period and repair activity cannot proceed.

A deferred Telecommunications Trouble Report may become “open/active” again, or move directly to the “closed” state if it is cancelled for some reason.

A trouble report in the deferred state may be cancelled by the manager. The agent on receiving such a request will attempt to close the trouble report.

7.1.3.4 Cleared

A trouble report is moved by the agent to the “cleared” state when it determines that the trouble has been resolved. 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.

7.1.3.5 Closed

This state indicates that the trouble resolution process is complete. Upon closure, the trouble report attributes are captured in a historical event generated at trouble report closure which may then be stored in a log of trouble history records, for future reference. The trouble report may then be eliminated at the agent’s convenience. However, the agent may be required to maintain such records for a period of time as per business agreements.

7.1.3.6 Disabled

A “disabled” value is exhibited when a trouble report’s information cannot be updated due to local conditions. In the “disabled” condition only read operations can be performed.

7.2 Model components generic definitions

This subclause contains the definitions of the object classes and attribute types that form the basis for the extensions to the model needed for interfaces between OSs across jurisdictional boundaries. The various object classes and attribute types are described in 7.2.1-7.2.3, as follows:

Subclause	Object class/Attribute type
7.2.1	Object classes
7.2.2	Attribute types using the template notation described in Recommendation X.722
7.2.3	Specific error messages identified for the interface
Annex A	Type definitions in the Abstract Syntax NotationOne (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.

NOTE – Being a trouble management Recommendation, billing and service charging are outside the scope of this Recommendation. The inclusion of objects such as 'Account' and 'CNM service' is purely to facilitate the optionality required by those existing implementations that use name bindings based on 'account' and 'CNM Service' objects and not for the purpose of accounting at all.

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 Recommendation X.217, and the Access Control object described in Recommendation X.741.

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.

7.2.1 Object class definitions

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

For the convenience of the reader, Appendix V gives a pictorial representation of the pointer attribute relationships between objects in this Recommendation. Appendix V does not represent normative information.

The remainder of this subclause contains the text definition of the object classes specified in this Recommendation. For each object, in addition to the text definition, a template is also provided in this Recommendation, as specified in Recommendation X.722. The templates for the objects as well as name bindings are supplied in Annex A.

7.2.1.1 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.

7.2.1.2 CNM Service

The Customer Network Management (CNM) Service object class is a specialization of the Service object class. These refinements are necessary to support the service modelling 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.

7.2.1.3 Contact

The Contact object class refers to a person or organization having responsibility for one or more managed object instances.

7.2.1.4 Provider Trouble Report

The trouble management function can be used to notify the manager that planned preventative maintenance action is scheduled to prevent future trouble.

The Provider Trouble Report object is created by the agent role CME to notify the manager role CME that planned maintenance will be carried out at a given time and that all or part(s) of the service(s), resource(s), network or system will be inaccessible during that time.

NOTE – In addition to the attributes listed below, the Provider Trouble Report object also includes the attributes inherited from the Trouble Report.

The information contained in the Provider Trouble Report object class is described below:

Begin Time – Indicates the beginning of the time frame when the service will be unavailable.

End Time – Indicates the end of the time frame for which the service will be unavailable.

Location Ptr – Indicates the location of the managed object instance against which the trouble report is created.

Unavailable Service Ptr – Indicates which service is affected.

7.2.1.5 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 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 – managed object, a Telecommunications Trouble Report instance for that CNM Service or – managed object shall consist of:

- the mandatory attributes of the Telecommunications Trouble Report object class as defined in the Telecommunications Trouble Report object class definition;
- 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 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
troubleDetectionTime
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:

activityDuration¹⁾
additionalTroubleStatusInfo
agentContactPerson
agentContactObjectPtr
alarmRecordPtrList
commitmentTime
lastUpdateTime
relatedTroubleReportList

¹⁾ These attributes are required to be updatable by the manager in the service-provider-to-service-provider interface.

responsiblePersonName²⁾
responsiblePersonPtr²⁾
troubleLocation²⁾
troubleReportNumberList
troubleReportState²⁾
troubleReportStatus²⁾
troubleReportStatusTime

- The following attributes can be set to default by the agent and only updated by the agent:
closeOutNarr²⁾
handOffCenter
handOffLocation
handOffPersonName²⁾
handOffPersonPtr²⁾
handOffTime
maintenanceOrgContactName²⁾
maintenanceOrgContactPtr²⁾
maintenanceOrgContactTime²⁾
maintServiceCharge
outageDuration
repairActivityList²⁾
restoredTime²⁾
troubleFound²⁾
- The following attributes can be set to default by the agent and only updated by the manager:
afterHrsRepairAuth
cancelRequestedByManager
closeOutVerification
troubleClearancePerson
- The following attribute 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:
authorizationList
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.

Manager deletion of Telecommunications Trouble Reports is not supported on inter-jurisdictional interfaces. A manager may request that a Telecommunications Trouble Report be cancelled, 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.

²⁾ These attributes are required to be updatable by the manager in the service-provider-to-service-provider interface.

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.

NOTES

1 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 Recommendation X.735.

2 If an additional trouble type defined as an integer is considered necessary, then a subclass of the TTR object can be specified to provide context specific trouble type attribute.

7.2.1.6 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.

7.2.1.7 Trouble History Record

The Trouble History Record object class is a refinement of the Log Record object class in Recommendation X.721 and is used to log the Trouble History Event notifications from the Trouble Report object and its sub-classes. The Trouble History Record object is a repository for selected information from a Trouble Report object and its sub-classes. Instantiated sub-classes 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.

7.2.1.8 Trouble Report

The Trouble Report object is the super-class of the Telecommunications Trouble Report object. The Trouble Report object is not instantiated. Instances of the Telecommunications Trouble Report sub-class 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 –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 managed object, a Telecommunications Trouble Report instance for that CNM-Service-managed or managed object shall consist of:

- the mandatory attributes of the Telecommunications Trouble Report, as defined in the Telecommunications Trouble Report object definition;
- 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 an 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 (sub-class 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 cancelled, 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 notifications 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 Recommendation X.735.

7.2.1.9 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 sub-classes.

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 sub-classes.

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 an 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 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 telecommunications resource in an `applicableManagedObjectInstanceList` attribute in the Trouble Report Format Definition object (when the format is specific to the object instance).

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

NOTES

1 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.

2 Trouble Report format examples for a typical Exchange Carrier are provided in Appendix IV which does not contain normative information.

7.2.2 Attribute type definitions

This subclause contains the text definitions of the attribute types specified in this Recommendation. For each attribute type definition, in addition to the text definition, this Recommendation also provides a template, as specified in Recommendation X.722. The attribute templates are in Annex A.

The ability to read and write attributes belonging to object classes defined in this Recommendation can generally be accomplished using the PT-GET and PT-SET services of Recommendation X.730. 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 Recommendation (e.g. CNM Service). These pointers may also reference objects belonging to sub-classes of the specified object classes (e.g. sub-classes of CNM Service). When these sub-classes are not known to the

peer systems, objects of these sub-classes are to be treated as members of an object class specified in this Recommendation (e.g. CNM Service if the pointer refers to a sub-class of CNM Service, etc.).

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

NOTE – For the convenience of the reader, Appendix V gives a pictorial representation of the pointer attribute relationships between objects in this Recommendation. Appendix V does not represent normative information.

7.2.2.1 Account Contact List

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

7.2.2.2 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.

7.2.2.3 Activity Code

The Activity Code attribute identifies a general repair activity category.

7.2.2.4 Activity Duration

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

7.2.2.5 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.

7.2.2.6 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.

7.2.2.7 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.

7.2.2.8 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.

7.2.2.9 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.

7.2.2.10 Administrative State

This attribute is imported from Recommendation X.731.

7.2.2.11 After Hours Repair Authorisation

The After Hours Repair Authorisation attribute indicates whether the customer has given the OK to repair the service outside normal business hours (e.g. 9 a.m.-5 p.m., Monday through Friday).

7.2.2.12 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.

7.2.2.13 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.

7.2.2.14 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.

7.2.2.15 Alarm State

This attribute is shown in detail in Annex A.

7.2.2.16 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.

7.2.2.17 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.

7.2.2.18 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.

7.2.2.19 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.

7.2.2.20 Authorisation List

7.2.2.20.1 General

The Authorisation List attribute identifies whether authorisation is requested by the agent and granted by the manager. It also specifies the type of activities that are authorised, and optionally the authorising person, and the time of authorisation.

7.2.2.20.2 Mechanism for requesting and granting authorisation

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 authorisations provided by the manager. In the former case, no activities are authorised by the manager. In the latter case, the authorisationList attribute would include a requestedState with a value of "provided(2)" (since authorisation is always provided by the manager and may be requested by the agent). It would also include an activityType where the bitString would include a "1" for each bit representing an activity that is authorised. It may also include an authTime and an authPerson indicating who in the manager's organization has authorised the specific activities.

Subsequently, after the troubleReport object instance has been created the agent may request authorisation 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 authorisation is required shall be indicated,

even if previously authorised). The agent would not include an authTime or a value for the authPerson in the sequence. This change (request for authorisation) would be indicated to the manager via an attribute ValueChange notification.

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

7.2.2.21 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 modelled as a sequence of various optional elements with the condition that at least one of the optional elements should be present.

7.2.2.22 Called Number

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

7.2.2.23 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 cancelled.

7.2.2.24 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.

7.2.2.25 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.

7.2.2.26 Commitment Time Request

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

7.2.2.27 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.

7.2.2.28 Close Out Verification

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

7.2.2.29 Current Problem List

This attribute is shown in detail in Annex A.

7.2.2.30 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.

7.2.2.31 Customer Work Center

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

7.2.2.32 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.

7.2.2.33 Entry Time

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

7.2.2.34 Escalation List

7.2.2.34.1 General

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

7.2.2.34.2 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.

7.2.2.35 Event Time

This attribute is shown in detail in Annex A.

7.2.2.36 Hand Off Center

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

7.2.2.37 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.

7.2.2.38 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.

7.2.2.39 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.

7.2.2.40 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.

7.2.2.41 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.

7.2.2.42 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.

7.2.2.43 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.

7.2.2.44 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.

7.2.2.45 A 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.

7.2.2.46 Z 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.

7.2.2.47 A 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.

7.2.2.48 Z 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.

7.2.2.49 Maintenance Organization Contact Name

The Maintenance Organization Contact Name attribute describes the company or organization whose responsibility 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.

7.2.2.50 Maintenance Organization Contact Pointer

The Maintenance Organization Contact Pointer attribute describes the company or organization whose responsibility 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 (CME to CME interface)]. Note that there is only one maintenance company for a given managed object instance, specified by contract.

7.2.2.51 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.

7.2.2.52 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.

7.2.2.53 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.

7.2.2.54 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.

7.2.2.55 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.

7.2.2.56 Managed Object Instance

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

7.2.2.57 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 Managed Object alias.

7.2.2.58 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.

7.2.2.59 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.

7.2.2.60 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.

7.2.2.61 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.

7.2.2.62 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.

7.2.2.63 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.

7.2.2.64 Operational State

This attribute is imported from Recommendation X.731.

7.2.2.65 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.

7.2.2.66 Perceived Trouble Severity

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

7.2.2.67 Preferred Priority

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

7.2.2.68 Received Time

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

7.2.2.69 Related Trouble Report List

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

7.2.2.70 Repair Activity Identifier

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

7.2.2.71 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.

7.2.2.72 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.

7.2.2.73 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.

7.2.2.74 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.

7.2.2.75 Restored Time

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

7.2.2.76 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.

7.2.2.77 Service Description

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

7.2.2.78 Service Identifier

This attribute is shown in detail in Annex A.

The Service Identifier is the distinguishing attribute of the CNM Service managed object class. It is assigned by the service provider at the time the service is delivered to the customer. The Service ID may include a combination of the Service Alias attribute and the Service Type attribute (or some other attribute) to guarantee that the Service ID is unique.

7.2.2.79 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.

7.2.2.80 Service Profile Description

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

7.2.2.81 Service Profile Identifier

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

7.2.2.82 Service Profile Object Pointer

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

7.2.2.83 Service Type

The Service Type attribute identifies the category of service (e.g. POTS, CENTREX, private line).

7.2.2.84 Supported Service Name List

This attribute specifies the services supported by a given managed object.

7.2.2.85 Suspect Object List

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

7.2.2.86 Trouble 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 equality means checking personNumber only in the sequence.

7.2.2.87 Trouble 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.

7.2.2.88 Trouble 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.

7.2.2.89 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.

7.2.2.90 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.

7.2.2.91 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 managed object.

7.2.2.92 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.

7.2.2.93 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.

7.2.2.94 Trouble 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 sub-classes) that "must be present" in a particular instance of a trouble report according to a particular Trouble Report Format Definition.

7.2.2.95 Trouble 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 sub-classes) that "may be present" in a particular instance of a trouble report according to a particular Trouble Report Format Definition.

7.2.2.96 Trouble 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).

7.2.2.97 Trouble 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.

7.2.2.98 Trouble Report Status

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

7.2.2.99 Trouble Report Status Time

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

7.2.2.100 Trouble Report Status Window

The Trouble Report Status Window attribute specifies a sliding window during which a troubleReportProgress notification is expected. This notification shall include the Trouble Report Status attribute and if the value of the status has not changed since last issued, it shall also include the Additional Trouble Status Info attribute, i.e. a status narrative describing what progress has been made in resolving the trouble report. The sliding window begins at the event time for the most recent notification that includes an Additional Trouble Status Info attribute describing progress on the trouble.

7.2.2.101 Trouble Type

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

7.2.2.102 TSP Priority

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

7.2.2.103 Usage state

This attribute is imported from Recommendation X.731.

7.2.3 Error messages

7.2.3.1 Trouble Report Already Exists

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.

7.2.3.2 Fallback Reporting

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.

7.2.3.3 Can Not close

canNotClose PARAMETER

This error message is sent to the manager when the trouble report cannot be closed by the agent because it is already cleared.

7.2.3.4 Trouble Report Must Be Present Attribute Missing

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.

7.2.3.5 Cannot Verify or Deny at This Time

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.

7.2.3.6 Trouble Report Change Denied

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.

8 Service description

8.1 Introduction

This clause describes the details of Trouble Management Services and functional units. Subclauses 8.4 and 8.11 provide details of additional functional units.

To provide additional functionality, this Recommendation makes use of PT-GET, PT-SET, PT-CREATE, and PT-DELETE which map into CMIS M-GET, CMIS M-SET, CMIS M-CREATE and CMIS M-DELETE services. Object creation reporting, object deletion reporting and attribute value change reporting services are also utilised.

In addition to the Trouble History Event Notification and the Telecommunications Trouble Report Progress Notification Services, the Trouble Management Function provides the capabilities such as:

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.

This subclause defines the services needed to support the trouble administration category of Fault Management functions defined in clause 6. Each service definition consists of:

- behavior and purpose of the service;
- the CMISE service that it maps onto;
- 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 the CMIS parameters are defined in Recommendation X.710.

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 Recommendation.

8.2 Kernel functional unit

The Kernel functional unit has the following capabilities:

- Entering Trouble Report;
- Requesting Trouble Report Status.

8.2.1 Enter Trouble Report

The PT-CREATE service, as described in Recommendation X.730 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;
- 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 error parameters are listed in A.3.

This error will be returned in the following two cases:

- Service predesignated by agent to receive fallback reporting;
- Agent partially failed or temporarily unavailable for receiving trouble reports.

8.2.2 Request Trouble Report Status

The PT-GET service, as described in Recommendation X.730 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.

8.3 Request Trouble Report Format Functional Unit

The PT-GET service, as described in Recommendation X.730 is used to allow a manager to determine the format for reporting troubles on either a CNM Service or an 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 tRMustBePresentAttrIDList 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 tRMayBePresentAttrIDList 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); or
- 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 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 object instance (when the format is specific to the object instance).

8.4 Trouble History Event Notification Functional Unit

8.4.1 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 Recommendation X.710.

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.

8.4.2 Parameters

The following parameters are defined for use in this Recommendation in the Trouble History Event Notification Service and are formally defined in Recommendation X.710:

- *Invoke identifier* – See Recommendation X.710.
- *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 7.2.2):
 - managed object instance;
 - received time;
 - trouble found;
 - activity duration (optional);
 - additional trouble information list (optional);
 - authorisation 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 Recommendation X.710.
 - *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 Recommendation X.710.

8.5 Review Trouble History Functional Unit

The PT-GET service, described in Recommendation X.730, is used to allow a manager to request information about past troubles reported for a particular CNM service or 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.

8.6 Add Trouble Information Functional Unit

The PT-SET service, described in Recommendation X.730, 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.

8.7 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;
- 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.

8.7.1 Trouble Report Status/Commitment Time Update Notification

The Trouble Report Status/Commitment Time Update uses the Attribute Value Change Notification service defined in Recommendation X.730. 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.

8.8 Verify Repair Completion Functional Unit

The PT-SET service, described in Recommendation X.730, is used to allow the manager to verify that 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”³⁾.

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.

8.9 Modify Trouble Administration Information Functional Unit

The PT-SET service, as described in Recommendation X.730, is used to allow the manager to modify the “writable” attributes that are not covered for a similar function under other functional units.

NOTE – 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 Authorisation;
- Alternate Manager Contact Person;
- Alternate Manager Contact Object Pointer;
- Authorisation 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;

³⁾ 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 incides, “clearedAwaitingCustVerification” should be considered “closedOut”.

- Perceived Trouble Severity;
- Trouble Clearance Person;
- Trouble Report Status Window;
- Z Location Access Address;
- 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.

8.10 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.

8.10.1 Attribute Value Change Notification

The Attribute Value Change Notification service is defined in Recommendation X.730. 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.

8.10.2 Object Creation Notification

The Object Creation Notification service is defined in Recommendation X.730. 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, 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 an Object Creation Record object class.

8.10.3 Object Deletion Notification

The object Deletion Notification service is defined in Recommendation X.730. 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, 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 an Object Deletion Record object class.

8.11 Trouble Report Progress Notification Functional Unit

8.11.1 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 the troubleReportProgressNotification is sent, the value of troubleReportStatusWindow attribute is automatically reset by the agent system and begins to count down once more. 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 troubleReport. The service uses the CMIS M-EVENT-REPORT service and procedures defined in Recommendation X.710.

8.11.2 Parameters

The following parameters are defined for use in this Recommendation in the Trouble Report Progress Notification service and are formally defined in Recommendation X.710.

- *Invoke identifier* – See Recommendation X.710.
- *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 Annex A):
 - trouble report status;
 - additional trouble status information (optional).

8.12 Cancel Trouble Report Functional Unit

The PT-SET service (see Recommendation X.730) 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 cancelling the trouble report. When the request is accepted, the cancellation process begins. This process may have associated billing implications beyond the scope of this Recommendation 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 in response to the PT-SET implies that the trouble report is already cleared.

8.13 Extended Modify Trouble Administration Information Functional Unit

The Extended Modify Trouble Administration Information Functional Unit allows the manager, in both client to service provider and service provider to service provider application environments, to update the following attributes:

- Dialog;
- Manager Search Key 1;
- Manager Search Key 2;
- Manager Search Key 3;
- Preferred Priority;
- Repeat Report;
- Trouble Detection Time.

The Extended Modify Trouble Administration Information capability is invoked by the manager using the PT-SET service to change one or more of the above mentioned attributes.

8.14 Delete Telecommunications Trouble Report functional unit

The Delete Telecommunications Trouble Report functional unit allows the manager, in a service provider to service provider application environment, to delete “closed” Telecommunications Trouble Reports. Consequently an Object Deletion Notification will be emitted. If the manager attempts to delete a telecommunications trouble report which is not in a “closed” state then “trouble ReportChangeDenied” specific error is returned.

8.15 Refer Telecommunications Trouble Report functional unit

The Refer Telecommunications Trouble Report functional unit allows the manager, in a service provider to service provider application environment, to delegate problem resolution to a hand-off person.

The Refer Telecommunications Trouble Report capability is invoked by the manager using PT-SET service (confirmed mode) to change either the Hand-off person name or the Hand-off person pointer attribute value, only if the telecommunications trouble report is in the “open/active” state. If the manager attempts to set this attribute in other than “open/active” state, then a CMIS error Processing Failure with “specific Error Info” value as “miscellaneousError” is returned.

8.16 Transfer Telecommunications Trouble Report functional unit

The Transfer Telecommunications Trouble Report functional unit allows the manager, in a service provider to service provider application environment, to either assign a Responsible Person, or re-assign to another Responsible person.

The Transfer Telecommunications Trouble Report capability is invoked by the manager using the PT-SET service to change either the Responsible Person Name or the Responsible Person Pointer attribute value, only if the telecommunications trouble report is in the “open/active” state. If the manager attempts to set this attribute in other than “open/active” state, then a CMIS error Processing Failure with “specific Error Info” value as “miscellaneousError” is returned.

8.17 Update State and Status functional unit

The Update State and Status functional unit requires the support of PT-SET service.

The Update State and Status capability is invoked by the manager using PT-SET service in a service provider to service provider application environment to modify the Trouble Report State, Trouble Report Status and other associated attributes that need to be modified as the trouble resolution process progresses.

Attributes that may be set when this functional unit is invoked are:

- Activity Duration;
- Close-Out Narrative;
- Maintenance Organization Contact Name;
- Maintenance Organization Contact Ptr;
- Maintenance Organization Contact Time;
- Repair Activity List;
- Restored Time;
- Trouble Found;
- Trouble Location;
- Trouble Report State;
- Trouble Report Status.

8.18 Repair Activity Object Functional Unit

The Repair Activity Object Functional Unit allows the manager, in a service provider to service provider application environment, to update repair and maintenance related information about a trouble report by creating an instance of a Repair Activity object named by the Telecommunications Trouble Report.

To create a Repair Activity object, the manager uses the PT-CREATE service. Note that in order for the Repair Activity object to be created (and thus named by a Telecommunications Trouble Report instance), the Repair Activity List attribute must not be in the Telecommunications Trouble Report instance.

8.19 Provider Trouble Report Control Functional Unit

The Provider Trouble Report Control Functional Unit allows an agent to inform the manager of planned maintenance. The agent system uses object creation reporting service to notify the manager of a newly created Provider Trouble Report or of any attribute value changes (e.g. Trouble Report State) through the use of the attribute change notification. The object deletion reporting service is used to notify deletion of Provider Trouble Reports.

The manager uses the PT-GET services to view specific attributes of a Provider Trouble Report.

8.20 Summary of functional units

Table 8-20 below provides a summary of the functional units for the Trouble Management Function and the related objects required for conformance.

TABLE 8-20/X.790

Functional units, services and objects

Functional unit	Services	Object(s) required for FU conformance
Kernel	PT-CREATE PT-GET	Telecommunications Trouble Report
Request Trouble Report Format	PT-GET	Trouble Report Format Definition
Trouble History Event Notification	Trouble History Event Notification	Telecommunications Trouble Report Event Forwarding Discriminator
Review Trouble History Record	PT-GET	Log Trouble History Record
Add Trouble Information	PT-SET	Telecommunications Trouble Report
Trouble Report Status/Commitment Time Update Notification	Attribute Value Change Reporting	Telecommunications Trouble Report Event Forwarding Discriminator
Verify Trouble Repair Completion	PT-SET	Telecommunications Trouble Report
Modify Trouble Administration Information	PT-SET	Telecommunications Trouble Report
Trouble Administration Configuration Event Notification	Object Creation Reporting Object Deletion Reporting Attribute Value Change Reporting	Event Forwarding Discriminator Trouble Report Format Definition and/or Telecommunications Trouble Report
Trouble Report Progress Notification	Trouble Report Progress Notification	Telecommunications Trouble Report Event Forwarding Discriminator
Cancel Trouble Report	PT-SET	Telecommunications Trouble Report
Extended Modify Trouble Administration Information	PT-SET	Telecommunications Trouble Report
Delete Telecommunications Trouble Report	PT-DELETE	Telecommunications Trouble Report
Refer Telecommunications Trouble Report	PT-SET	Telecommunications Trouble Report
Transfer Telecommunications Trouble Report	PT-SET	Telecommunications Trouble Report
Update State and Status	PT-SET	Telecommunications Trouble Report
Repair activity Object	PT-CREATE	Repair Activity
Provider Trouble Report Control	PT-GET, Object Creation Reporting Object Deletion Reporting Attribute Value Change Reporting	Provider Trouble Report Event Forwarding Discriminator

9 Service mapping to Protocol

This clause provides the mapping of the services specifically defined by the trouble management function to the M-EVENT-REPORT CMIS-service. The PT-services map directly onto the CMIS-services as defined in the Object Management Function, Recommendation X.730. The application context is defined in Systems Management Overview, Recommendation X.701. For the CMIS-services to be transferred between an agent role CME and a manager role CME, an association established by using the Association Control Service Element and a Remote Operations Service Element in the Application Layer of the Open Systems Interconnection Model must be available.

9.1 Element of procedure

Trouble Management Function defines two new services Trouble History Event Notification and Trouble Report Progression Notification specified in the services description in 8.4 and 8.11 respectively. Clause 8, the services description, provides references to other Recommendations whose services are needed for the Trouble Management Function.

Trouble History Event Notification element of procedure is specified in 9.1.1.

Trouble Report Progress Notification element of procedure is specified in 9.1.2.

9.1.1 Trouble History Event Notification Service procedures

For the Trouble History Event Notification Service, the agent role and manager role procedures are as follows.

9.1.1.1 Agent role

9.1.1.1.1 Invocation

See Figure 9-1.

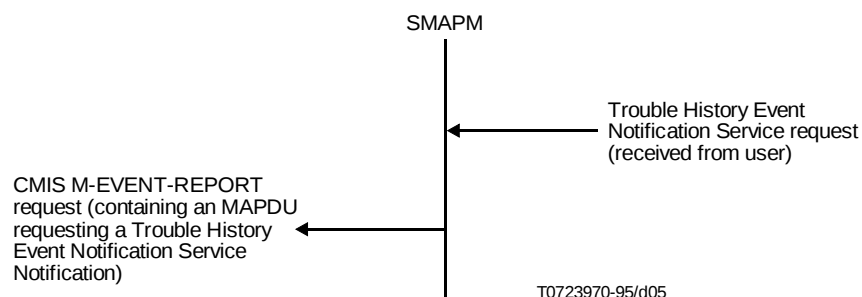


FIGURE 9-1/X.790

Invocation (Agent role)

The Trouble History Event Notification Service procedures are initiated by the Trouble History Event Notification Service request primitive. On receipt of a Trouble History Event Notification Service request primitive, the System Management Application Protocol Machine (SMAPM) shall construct a Management Application Protocol Data Unit (MAPDU) and issue a CMIS M-EVENT-REPORT request service primitive with parameters derived from the Trouble History Event Notification Service request primitive.

9.1.1.1.2 Receipt of response

See Figure 9-2.

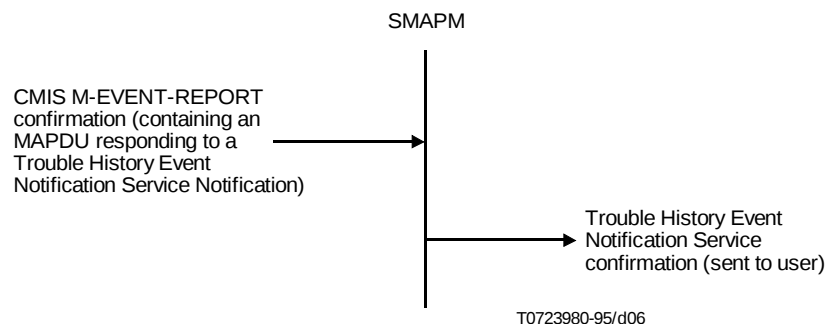


FIGURE 9-2/X.790
Receipt of response (Agent role)

On receipt of a CMIS M-EVENT-REPORT confirm service primitive containing an MAPDU responding to a Trouble History Event Notification Service notification, the SMAPM shall issue a Trouble History Event Notification Service confirmation primitive to the Trouble History Event Notification Service user with parameters derived from the CMIS M-EVENT-REPORT confirm service primitive, thus completing the Trouble History Event Notification Service procedure.

NOTE – The SMAPM shall ignore all errors in the received MAPDU. The Trouble History Event Notification Service user may ignore such errors, or abort the association as a consequence of such errors.

9.1.1.2 Manager role

9.1.1.2.1 Receipt of request

See Figure 9-3.

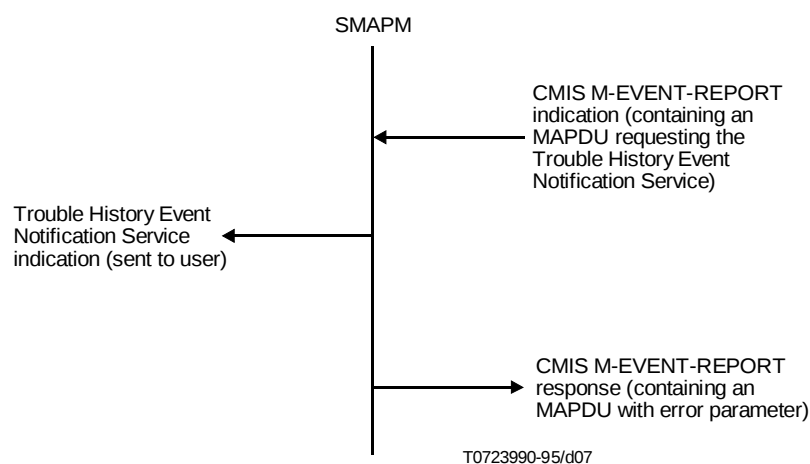


FIGURE 9-3/X.790
Receipt of request (Manager role)

On receipt of a CMIS M-EVENT-REPORT indication service primitive containing an MAPDU requesting the Trouble History Event Notification Service, the SMAPM shall, if the MAPDU is well formed, issue a Trouble History Event Notification Service indication primitive to the Trouble History Event Notification Service user with parameters derived from the CMIS M-EVENT-REPORT indication service primitive.

Otherwise, the SMAPM shall construct an appropriate MAPDU containing notification of the error, and shall issue a CMIS M-EVENT-REPORT response service primitive with an error parameter present.

9.1.1.2.2 Response

The SMAPM shall accept Trouble History Event Notification Service response primitive and shall construct an MAPDU confirming the notification and issue a CMIS M-EVENT-REPORT response service primitive with parameters derived from the Trouble History Event Notification Service response primitive. See Figure 9-4.

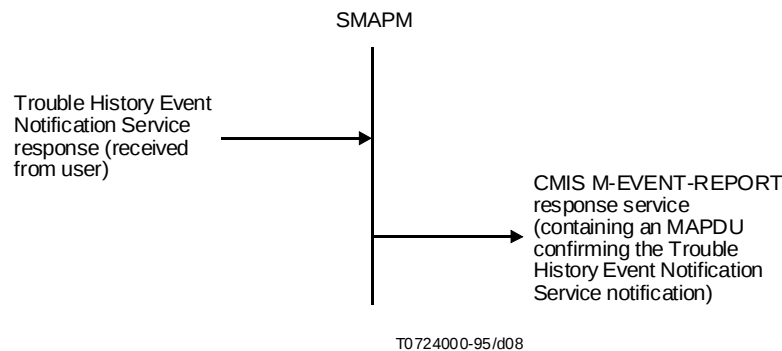


FIGURE 9-4/X.790
Response (Manager role)

9.1.2 Trouble Report Progress Notification Service procedures

For the Trouble Report Progress Notification Service, the agent role and manager role procedures are as follows.

9.1.2.1 Agent role

9.1.2.1.1 Invocation

See Figure 9-5.

The Trouble Report Progress Notification Service procedures are initiated by the Trouble Report Progress Notification Service request primitive. On receipt of a Trouble Report Progress Notification Service request primitive, the System Management Application Protocol Machine (SMAPM) shall construct a Management Application Protocol Data Unit (MAPDU) and issue a CMIS M-EVENT-REPORT request service primitive with parameters derived from the Trouble Report Progress Notification Service request primitive.

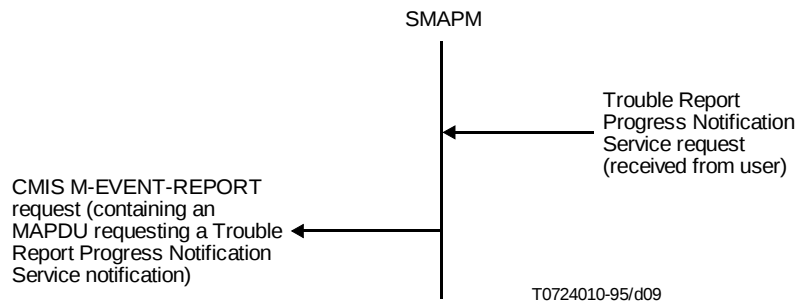


FIGURE 9-5/X.790

Invocation (Agent role)

9.1.2.1.2 Receipt of response

See Figure 9-6.

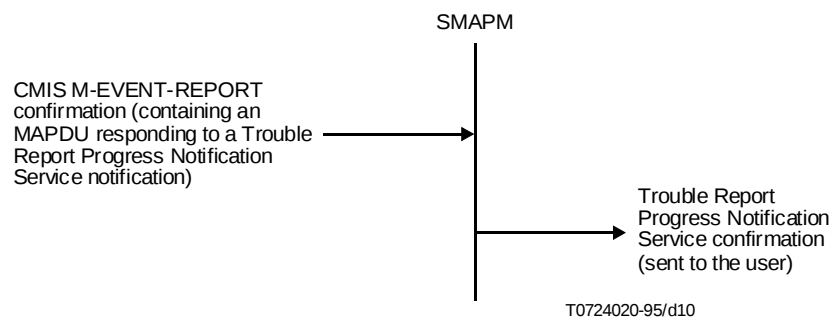


FIGURE 9-6/X.790

Receipt of response (Agent role)

On receipt of a CMIS M-EVENT-REPORT confirm service primitive containing an MAPDU responding to a Trouble Report Progress Notification Service notification, the SMAPM shall issue a Trouble Report Progress Notification Service confirmation primitive to the Trouble Report Progress Notification Service user with parameters derived from the CMIS M-EVENT-REPORT confirm service primitive, thus completing the Trouble Report Progress Notification Service procedure.

NOTE – The SMAPM shall ignore all errors in the received MAPDU. The Trouble Report Progress Notification Service user may ignore such errors, or abort the association as a consequence of such errors.

9.1.2.2 Manager role

9.1.2.2.1 Receipt of request

See Figure 9-7.

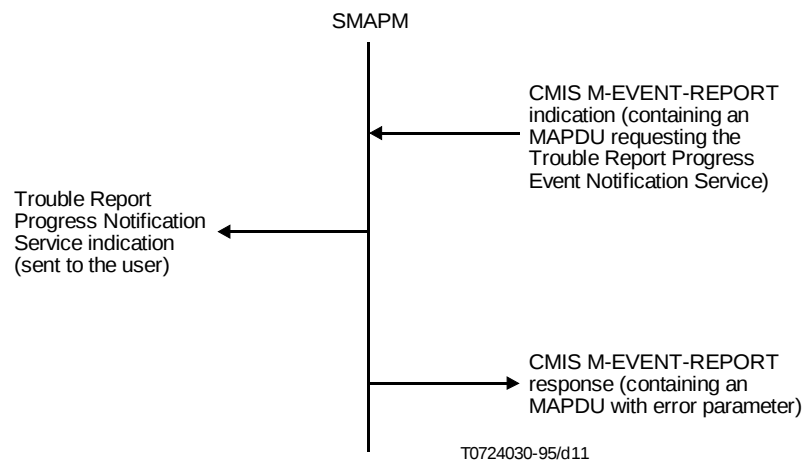


FIGURE 9-7/X.790
Receipt of request (Manager role)

On receipt of a CMIS M-EVENT-REPORT indication service primitive containing an MAPDU requesting the Trouble Report Progress Notification Service, the SMAPM shall, if the MAPDU is well formed, issue a Trouble Report Progress Notification Service indication primitive to the Trouble Report Progress Notification Service user with parameters derived from the CMIS M-EVENT-REPORT indication service primitive.

Otherwise, the SMAPM shall construct an appropriate MAPDU containing notification of the error, and shall issue a CMIS M-EVENT-REPORT response service primitive with an error parameter present.

9.1.2.2.2 Response

See Figure 9-8.

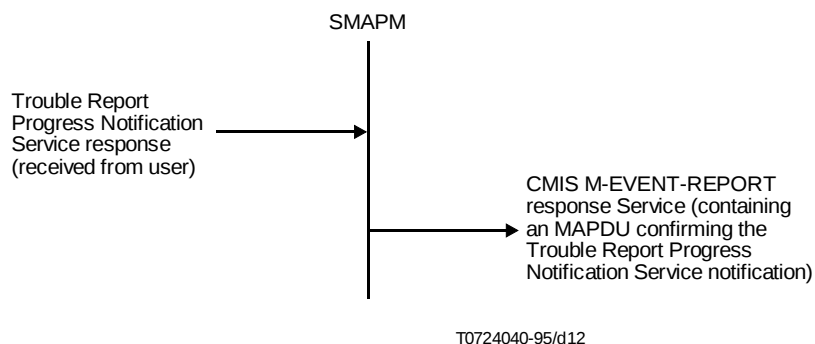


FIGURE 9-8/X.790
Response (Manager role)

The SMAPM shall accept Trouble Report Progress Notification Service response primitive and shall construct an MAPDU confirming the notification and issue a CMIS M-EVENT-REPORT response service primitive with parameters derived from the Trouble Report Progress Notification Service response primitive.

9.2 List of items having templates in Annex A and Recommendation X.721

9.2.1 Objects

This Recommendation defines the following object classes for which the templates are specified in A.1:

- Repair Activity;
- Telecommunications Trouble Report;
- Trouble History Record;
- Trouble Report;
- Provider Trouble Report;
- Trouble Report Format Definition;
- Contact;
- Account;
- CnmService;
- Service.

This Recommendation references the following objects for which the templates are specified in Recommendation X.721:

- Event Log Record;
- Log Record;
- Top;
- Event Forwarding Discriminator;
- Discriminator;
- Log.

9.2.2 Attributes

This Recommendation defines the following management attributes for which the templates are specified in A.2. See Table 9-1.

TABLE 9-1/X.790

Annex A.2 trouble management attribute labels

	Annex A.2 trouble management attribute labels
1	beginTime
2	endTime
3	unavailable ServicePtr

This Recommendation also references the following management attributes for which the templates are specified in A.2, and Recommendation X.721. See Table 9-2.

TABLE 9-2/X.790

**Attribute labels of trouble management attributes in Annex A.2 and
Recommendation X.721**

	Annex A.2 and Recommendation X.721 attributes
1	activityCode
2	activityDuration
3	activityInfo
4	activityPerson
5	additionalInformation
6	“Rec. X.721”: additionalText
7	additionalTroubleInfoList
8	additionalTroubleStatusInfo
9	afterHrsRepairAuth
10	agentContactPerson
11	agentContactObjectPtr
12	alarmRecordName
13	alarmRecordPtrList
14	“Rec. X.721”: allomorphs
15	alternateManagerContactPerson
16	alternateManagerContactObjectPtr
17	aLocationAccessAddress
18	aLocationAccessHours
19	aLocationAccessPerson
20	applicableManagedObjectClassList
21	applicableManagedObjectInstanceList
22	authorizationList
23	callBackInfoList
24	calledNumber
25	cancelRequestedByManager
26	closeOutNarr
27	closeOutVerification
28	commitmentTime
29	commitmentTimeRequest
30	“Rec. X.721”: correlated Notifications
31	custTroubleTickNum
32	customerWorkCenter
33	dialogue
34	entryTime

TABLE 9-2/X.790 (cont.)

**Attribute labels of trouble management attributes in Annex A.2 and
Recommendation X.721**

	Annex A.2 and Recommendation X.721 attributes
35	escalationList
36	“Rec. X.721”: eventTime
37	“Rec. X.721”: eventType
38	handOffCenter
39	handOffLocation
40	handOffPersonName
41	handOffPersonPtr
42	handOffTime
43	initiatingMode
44	lastUpdateTime
45	“Rec. X.721”: loggingTime
46	“Rec. X.721”: logRecordId
47	maintenanceOrgContactName
48	maintenanceOrgContactPtr
49	maintenanceOrgContactTime
50	:maintServiceCharge
51	managedObjectAccessFromTime
52	managedObjectAccessHours
53	managedObjectAccessToTime
54	managedObjectInstance
55	managedObjectInstanceAliasList
56	managerContactPerson
57	managerContactObjectPtr
58	managerSearchKey1
59	managerSearchKey2
60	managerSearchKey3
61	managerSearchKeyList
62	“Rec. X.721”: nameBinding
63	“Rec. X.721”: notificationIdentifier
64	“Rec. X.721”: objectClass
65	outageDuration
66	“Rec. X.721”: packages
67	perceivedTroubleSeverity
68	preferredPriority

TABLE 9-2/X.790 (end)

Attribute labels of trouble management attributes in Annex A.2 and Recommendation X.721

	Annex A.2 and Recommendation X.721 attributes
69	receivedTime
70	relatedTroubleReportList
71	repairActivityID
72	repairActivityList
73	repeatReport
74	responsiblePersonName
75	responsiblePersonPtr
76	restoredTime
77	suspectObjectList
78	troubleClearancePerson
79	troubleDetectionTime
80	troubleFound
81	troubleLocation
82	tRConstrainedToSingleValueAttrIDList
83	tRFormatID
84	tRMayBePresentAttrIDList
85	tRMustBePresentAttrIDList
86	troubleReportFormatObjectPtr
87	troubleReportID
88	troubleReportNumberList
89	troubleReportState
90	troubleReportStatus
91	troubleReportStatusTime
92	troubleReportStatusWindow
93	troubleType
94	tspPriority
95	zLocationAccessAddress
96	zLocationAccessHours
97	zLocationAccessPerson

9.2.3 Notifications

This Recommendation references the following notifications defined in A.4:

- troubleHistoryEventNotification;
- troubleReportProgressNotification.

This Recommendation references the following notifications defined in ITU-T Rec. X.721 | ISO/IEC 10165-2:

- attributeValueChange;
- objectCreation;
- objectDeletion.

9.2.4 Actions

No actions are defined or referenced by this Recommendation.

9.3 Negotiation of functional units

9.3.1 The following object identifier value {itu-t(0) recommendation(0) x(24) x790(790) managementApplicationsSupport(2) functionalUnitPackage(1) troubleManagement(1)} has been assigned as a value of the ASN.1 type FunctionalUnitsPackageId defined in CCITT Rec. X.701 | ISO /IEC 10040 to use for negotiating the following functional units. See Table 9-3.

TABLE 9-3/X.790

Functional units to be referenced by the object identifier in 9.3.1

	Functional unit
0	Extended Modify Trouble Administration Information
1	Delete Telecommunications Trouble Report
2	Refer Telecommunications Trouble Report
3	Transfer Telecommunications Trouble Report
4	Update State and Status
5	Repair Activity Object
6	Provider Trouble Report Control

9.3.2 The Trouble Management function uses the functional units in Table 9-3 in conjunction with those in Table 9-4 below. This Recommendation assigns the following object identifier value {itu-t(0) recommendation(0) x(24) x790(790) managementApplicationsSupport(2) functionalUnitPackage(1) 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 sub-classes.

TABLE 9-4/X.790

Functional units to be referenced specifically using object identifier in 9.3.2

	Functional unit
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

10 Relationship with other Standards

The interface recommendation in this Recommendation is interoperable with the following standards:

- ANSI T1.227 – American National Standard for Telecommunications: Operations, Administration, Maintenance and Provisioning (OAM&P) – Extension to Generic Network Model for Interfaces between Operations Systems across Jurisdictional Boundaries to Support Fault Management (Trouble Administration), September 22, 1992. This together with ANSI T1.228 constitute the North American standard for telecommunication trouble management.
- ANSI T1.228 – American National Standard for Telecommunications: Operations, Administration, Maintenance and Provisioning (OAM&P) – Services for Interfaces between Operations Systems across Jurisdictional Boundaries to Support Fault Management (Trouble Administration), October 12, 1992.
- Network Management Forum: Forum 024, Application Services: Trouble Management Function, Issue 1.0, August 1992.

11 Conformance

An implementation claiming to conform to this Recommendation shall comply with the requirements as defined in the following subclauses. In addition, this Recommendation specifies a general purpose platform profile; the supplier of the implementation shall state whether conformance is also claimed to this general purpose platform.

11.1 Static conformance

The implementation shall conform to the requirements of this Recommendation in the manager role, the agent role, or both roles. A claim of conformance to at least one role shall be made in Annex B. If a claim of conformance is made for support in the manager role, the implementation shall support at least one of the notifications or the managed object described in Annex B. A claim of conformance in the manager role requires the support of at least one management operation or notification as specified by those management definitions.

If a claim of conformance is made for support in the agent role, the implementation shall support at least one of the notifications or the managed object described Annex B. A claim of conformance in the agent role requires the support of all the mandatory operations and mandatory notifications specified by those management definitions.

The implementation shall support the transfer syntax derived from the encoding rules specified in CCITT Rec. X.209 | ISO/IEC 8825 and named {joint-iso-itu-t asn1(1) basicEncoding(1)}, for the abstract data types referenced by the definitions for which support is claimed.

NOTE – Annex B will be provided later as an amendment to this Recommendation.

11.2 Dynamic conformance

Implementations claiming to conform to this Recommendation shall support the elements of procedure and definitions of semantics corresponding to the definitions for which support is claimed.

11.3 Management implementation conformance statement requirements

Any MCS proforma, PICS proforma, MOCS proforma, and MIDS proforma which conforms to this Recommendation shall be technically identical to the proformas specified in Annex B preserving table numbering and the index numbers of items, and differing only in pagination and page headers.

The supplier of an implementation which is claimed to conform to this Recommendation shall complete a copy of the Management Conformance Summary (MCS) provided in Annex B as part of the conformance requirements together with any other ICS proformas referenced as applicable from that MCS. An MCS, MIDS, MOCS, MRCS and PICS which conforms to this Recommendation shall:

- describe an implementation which conforms to this Recommendation;
- have been completed in accordance with the instructions for completion given in ITU-T Rec. X.724 | ISO/IEC 10165-6;
- include the information necessary to uniquely identify both the supplier and the implementation.

Claims of conformance to the management information defined in this Recommendation in managed object classes defined elsewhere shall include the requirements of the MIDS proforma in the MOCS proforma for the managed object class.

11.4 Conformance to the general purpose platform profile for CCITT Rec. X.733 | ISO/IEC 10164-4

The general purpose platform profile for CCITT Rec. X.733 | ISO/IEC 10164-4 identifies a useful collection of the functionality defined in this Recommendation. A claim of conformance to this general purpose platform profile in the manager role, the agent role, or both roles, shall be made by conforming to Annex B. Annex B and other referenced annexes identify the requirements of this general purpose platform profile. A system claiming conformance to this general purpose platform profile shall support this function for all managed object classes that import the management information defined in this Recommendation.

11.5 Conformance to support managed object definition

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

- The objects supported shall comply with the behaviour specified in clause 7 and the syntax specified in Annex A.
- Support as a minimum, the Kernel functional unit as defined in clause 8 and the Telecommunication Trouble Report object.
- Support for a given functional unit defined in clause 8 requires support of the Object Class(es) associated with those functional units, as shown in Table 8-20.

Annex A

Managed objects

(This annex forms an integral part of this Recommendation)

A.1 Managed object class and package definitions

A.1.1 Account

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 {x790Package 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 {x790Package 2};

PRESENT IF "an instance supports it and accountContactAttributeListPkg is not present.";

REGISTERED AS {x790ObjectClass 1};

A.1.2 cnmService

cnmService MANAGED OBJECT CLASS

DERIVED FROM service;

CHARACTERIZED BY cnmServicePkg PACKAGE

ATTRIBUTES

serviceLocationList GET,

serviceDescription GET;;;

CONDITIONAL PACKAGES

csServiceAliasPkg PACKAGE

ATTRIBUTES

serviceAliasList GET;

-- Administrations may specify structure

-- of serviceAliasList GraphicString

REGISTERED AS {x790Package 3};

PRESENT IF "an instance supports it.",

csTroubleReportFormatObjectPtrPkg PACKAGE

ATTRIBUTES

troubleReportFormatObjectPtr GET;

REGISTERED AS {x790Package 4};

PRESENT IF "an instance supports it.";

REGISTERED AS {x790ObjectClass 2};

A.1.3 contact

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};

A.1.4 providerTroubleReport

providerTroubleReport MANAGED OBJECT CLASS

DERIVED FROM troubleReport;

CHARACTERIZED BY

providerTroubleReportPkg,

trAttributeValueChangedPkg,

trObjectCreationDeletionPkg;

CONDITIONAL PACKAGES

ptrUnavailableServicePtrPkg PACKAGE

ATTRIBUTES

unavailableServicePtr GET;

REGISTERED AS {x790Package 13};

PRESENT IF "an instance supports it and if a service is impacted";

REGISTERED AS {x790ObjectClass 4};

providerTroubleReportPkg PACKAGE

BEHAVIOUR providerTroubleReportBehaviour,

providerTroubleReportPkgDefinition;

ATTRIBUTES

beginTime GET,

endTime GET,

troubleLocation GET;

;

providerTroubleReportPkgDefinition BEHAVIOUR

DEFINED AS !The providerTroubleReport object is created by the agent role CME to specifically notify the manager role CME that planned maintenance will be carried out at a given time and that all or parts of the service(s), resource(s), network or system will be inaccessible during that time. In this case the trouble management function is used to notify the manager that planned maintenance action is scheduled to prevent future trouble.!

providerTroubleReportBehaviour BEHAVIOUR

DEFINED AS !In the following text, the term "agent" refers to a system operating in the agent role, and "manager" refers to a system operating in the manager role.

Only the agent is allowed to create or delete an instance of the Provider Trouble Report managed object class. The manager cannot create or delete an instance of the Provider Trouble Report managed object class. If the manager attempts to create or delete an instance of the Provider Trouble Report managed object class, then the "accessDenied" CMIS error is returned by the agent.

When the agent creates an instance of the Provider Trouble Report managed object class, the "objectCreation" notification is emitted.

When the agent deletes an instance of the Provider Trouble Report managed object class, the "objectDeletion" notification is emitted.

When the agent updates one or several attributes of an instance of the Provider Trouble Report managed object class, the "attributeValueChanged" notification is emitted.

If no specific additional information relative to the Provider Trouble Report managed object instance is known at creation time, then the additionalTroubleInfoList attribute must be set to GraphicString::="".

The receivedTime attribute is the creation time of the Provider Trouble Report managed object instance.

If the maintenance has already started at creation time, then the Provider Trouble Report managed object instance must be created with the state set to "open/active". If the maintenance has not started yet at creation time, then the Provider Trouble Report managed object instance must be created with the state set to "queued". In this case the Provider Trouble Report managed object instance will enter the "open/active" state when the maintenance will start. When the maintenance is done, then the Provider Trouble Report managed object instance must enter the "cleared" state. When the Provider Trouble Report managed object instance enters the cleared state, then the "troubleFound" attribute can be set for example to "information" if no specific problems have been detected during the maintenance. The Provider Trouble Report managed object instance will then enter the "closed" state.

The attributes that can be updated by the agent during the processing of a Provider Trouble Report managed object instance are :

activityDuration -- when the PTR is in the "cleared" or "closed" state

additionalTroubleInfoList -- at any time
agentContactPerson or **agentContactObjectPtr** -- at any time
beginTime -- when the PTR is in the "queued" state
closeOutNarr -- when the PTR is in the "cleared" or "closed" state
endTime -- when the PTR is not in the "closed" state
managedObjectInstanceAliasList -- at any time
relatedTroubleReportList -- at any time
repairActivityList -- at any time
restoredTime -- when the PTR is in the "cleared" or "closed" state
troubleClearancePerson -- according to the attribute's behaviour
troubleFound -- when the PTR is in the "cleared" or "closed" state
troubleLocation -- at any time
troubleReportNumberList -- at any time
troubleReportState -- according to the attribute's behaviour
troubleReportStatus -- according to the attribute's behaviour
troubleReportStatusTime -- when the "troubleReportStatus" is updated.

The manager cannot update attributes of a Provider Trouble Report managed object instance. If the manager attempts to update one or several attributes of a Provider Trouble Report managed object instance, then the "accessDenied" CMIS error is returned by the agent.

A value for the troubleReportID attribute can only be provided when the object is created. Furthermore, once the object is created, the value of troubleReportID may not be modified (i.e. the instance cannot be renamed).!;

trAttributeValueChangedPkg PACKAGE
 NOTIFICATIONS
 "Rec. X.721 | ISO/IEC 10165-2 : 1992":attributeValueChanged;
 REGISTERED AS {x790Package 14};

trObjectCreationDeletionPkg PACKAGE
 NOTIFICATIONS
 "Rec. X.721 | ISO/IEC 10165-2 : 1992":objectCreation,
 "Rec. X.721 | ISO/IEC 10165-2 : 1992":objectDeletion;
 REGISTERED AS {x790Package 15};

A.1.5 repairActivity

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 X790ASN1Module.activityInfoActivityInfoDefault GET;;;

CONDITIONAL PACKAGES
raActivityPersonPkg PACKAGE
 ATTRIBUTES
activityPerson GET;
 REGISTERED AS {x790Package 16};
 PRESENT IF "an instance supports it.",
raActivityCodePkg PACKAGE
 ATTRIBUTES
activityCode GET;
 REGISTERED AS {x790Package 17};
 PRESENT IF "an instance supports it.",
raObjectCreationPkg PACKAGE
 NOTIFICATIONS
 "Rec. X.721 | ISO/IEC 10165-2 : 1992":objectCreation;
 REGISTERED AS {x790Package 18};
 PRESENT IF "an instance supports it.";

REGISTERED AS {x790ObjectClass 5};

A.1.6 service

service MANAGED OBJECT CLASS
 DERIVED FROM "Recommendation X.721 : 1992":top;

**CHARACTERIZED BY servicePackage PACKAGE
BEHAVIOUR**

serviceBehaviour BEHAVIOUR

DEFINED AS "The service object class is a class of managed objects that represents offerings from a provider that supplies specific network functionality to one or more customers. A service may or may not be tarified. Services may be nested, thereby creating a containment relationship. The Service Type identifies the distinguishing characteristics of the Service. The Supported Service Name List identifies other Services supported by the Service, and the Supported By Object List identifies the objects that support the Service.";;

ATTRIBUTES

serviceID GET,

serviceType GET-REPLACE;

NOTIFICATIONS

"Rec. X.721 | ISO/IEC 10165-2" : qualityofServiceAlarm;

;;

CONDITIONAL PACKAGES

"Rec. M.3100 : 1992" : createDeleteNotificationsPackage

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

"Rec. M.3100 : 1992" : attributeValueChangeNotificationPackage

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

"Rec. M.3100 : 1992" : stateChangeNotificationPackage

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

"Rec. M.3100 : 1992" : administrativeOperationalStatesPackage

PRESENT IF "an instance supports it.",

usageStatePackage **PRESENT IF** "an instance supports it.",

alarmStatusPackage **PRESENT IF** "an instance supports it.",

"Rec. M.3100 : 1992" : currentProblemListPackage

PRESENT IF "an instance supports it.",

supportedServiceNameListPackage **PRESENT IF** "an instance supports it.",

supportedByObjectListPackage **PRESENT IF** "an instance supports it.";

REGISTERED AS {x790ObjectClass 6};

supportedByObjectListPackage PACKAGE

ATTRIBUTES

"Rec. M.3100 : 1992" : supportedByObjectList GET-REPLACE ADD-REMOVE;

REGISTERED AS {x790Package 19};

supportedServiceNameListPackage PACKAGE

ATTRIBUTES

supportedServiceNameList GET-REPLACE ADD-REMOVE;

REGISTERED AS {x790Package 20};

alarmStatusPackage PACKAGE

ATTRIBUTES

"Rec. M.3100 : 1992" : alarmStatus GET;

REGISTERED AS { x790Package 21};

usageStatePackage PACKAGE

ATTRIBUTES

"Rec. X.721 | ISO/IEC 10165-2" : usageState GET;

REGISTERED AS {x790Package 22};

A.1.7 telecommunicationsTroubleReport

telecommunicationsTroubleReport MANAGED OBJECT CLASS

DERIVED FROM troubleReport;

CONDITIONAL PACKAGES

trAfterHrsRepairAuthPkg **PACKAGE**

ATTRIBUTES

afterHrsRepairAuth

DEFAULT VALUE X790ASN1Module.afterHrsRepairAuthAfterHrsRepairAuthDefault
 GET-REPLACE;
 REGISTERED AS {x790Package 23};
 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 {x790Package 24};
 PRESENT IF "an instance supports it.",

trAlternateManagerContactPersonAttributePkg PACKAGE
 ATTRIBUTES
 alternateManagerContactPerson GET-REPLACE;
 REGISTERED AS {x790Package 25};
 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 {x790Package 26};
 PRESENT IF "an instance supports it and trAlternateManagerContactPersonAttributePkg
 is not present.",

trAuthorizationListPkg PACKAGE
 ATTRIBUTES
 authorizationList
 DEFAULT VALUE X790ASN1Module.authorizationAuthorizationDefault
 GET-REPLACE ADD-REMOVE troubleReportChangeDenied;
 REGISTERED AS {x790Package 27};
 PRESENT IF "an instance supports it and trAfterHrsRepairAuthPkg is not present.",

trCallBackInfoListPkg PACKAGE
 ATTRIBUTES
 callBackInfoList GET-REPLACE ADD-REMOVE;
 REGISTERED AS {x790Package 28};
 PRESENT IF "an instance supports it.",

trCalledNumberPkg PACKAGE
 ATTRIBUTES
 calledNumber GET;
 REGISTERED AS {x790Package 29};
 PRESENT IF "an instance supports it.",

trCancelRequestedByManagerPkg PACKAGE
 ATTRIBUTES
 cancelRequestedByManager
 DEFAULT VALUE X790ASN1Module.troubleReportCancelRequestedByManagerDefault
 GET-REPLACE troubleReportChangeDenied canNotClose;
 REGISTERED AS {x790Package 30};
 PRESENT IF "an instance supports it.",

trCloseOutVerificationPkg PACKAGE
 ATTRIBUTES
 closeOutVerification
 DEFAULT VALUE X790ASN1Module.closeOutVerificationCloseOutVerificationDefault
 GET-REPLACE cannotVerifyOrDenyAtThisTime;
 REGISTERED AS {x790Package 31};
 PRESENT IF "an instance supports it.",

trCommitmentTimePkg PACKAGE
 ATTRIBUTES
 commitmentTime GET;
 REGISTERED AS {x790Package 32};

PRESENT IF "an instance supports it.",

trCommitmentTimeRequestPkg PACKAGE
ATTRIBUTES
commitmentTimeRequest GET-REPLACE troubleReportChangeDenied;
REGISTERED AS {x790Package 33};
PRESENT IF "an instance supports it.",

trCustomerWorkCenterPkg PACKAGE
ATTRIBUTES
customerWorkCenter GET SET-BY-CREATE;
REGISTERED AS {x790Package 34};
PRESENT IF "an instance supports it.",

trCustTroubleTickNumPkg PACKAGE
ATTRIBUTES
custTroubleTickNum GET SET-BY-CREATE;
REGISTERED AS {x790Package 35};
PRESENT IF "an instance supports it.",

trDialogPkg PACKAGE
ATTRIBUTES
dialog GET-REPLACE;
REGISTERED AS {x790Package 36};
PRESENT IF "an instance supports it.",

trEscalationListPkg PACKAGE
ATTRIBUTES
escalationList GET ADD troubleReportChangeDenied;
REGISTERED AS {x790Package 37};
PRESENT IF "an instance supports it.",

trHandOffCenterPkg PACKAGE
ATTRIBUTES
handOffCenter
INITIAL VALUE X790ASN1Module.handOffCenterHandOffCenterInitial
GET;
REGISTERED AS {x790Package 38};
PRESENT IF "an instance supports it.",

trHandOffLocationPkg PACKAGE
ATTRIBUTES
handOffLocation
INITIAL VALUE X790ASN1Module.handOffLocationHandOffLocationInitial
GET;
REGISTERED AS {x790Package 39};
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 X790ASN1Module.handOffPersonNameHandOffPersonNameInitial
GET-REPLACE;
REGISTERED AS {x790Package 40};
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 {x790Package 41};
PRESENT IF "an instance supports it and trHandOffPersonNamePkg is not present.",

trHandOffTimePkg PACKAGE
ATTRIBUTES

handOffTime
 INITIAL VALUE X790ASN1Module.handOffTimeHandOffTimeInitial
 GET;
 REGISTERED AS {x790Package 42};
 PRESENT IF "an instance supports it.",

trInitiatingModePkg PACKAGE
 ATTRIBUTES
 initiatingMode GET;
 REGISTERED AS {x790Package 43};
 PRESENT IF "an instance supports it.",

trLastUpdateTimePkg PACKAGE
 ATTRIBUTES
 lastUpdateTime GET;
 REGISTERED AS {x790Package 44};
 PRESENT IF "an instance supports it.",

trALocationPkg PACKAGE
 ATTRIBUTES
 aLocationAccessAddress GET-REPLACE;
 REGISTERED AS {x790Package 45};
 PRESENT IF "an instance supports it.",

trZLocationPkg PACKAGE
 ATTRIBUTES
 zLocationAccessAddress GET-REPLACE;
 REGISTERED AS {x790Package 46};
 PRESENT IF "an instance supports it.",

trALocationAccessHoursPkg PACKAGE
 ATTRIBUTES
 aLocationAccessHours GET-REPLACE ADD-REMOVE;
 REGISTERED AS {x790Package 47};
 PRESENT IF "an instance supports it.",

trZLocationAccessHoursPkg PACKAGE
 ATTRIBUTES
 zLocationAccessHours GET-REPLACE ADD-REMOVE;
 REGISTERED AS {x790Package 48};
 PRESENT IF "an instance supports it.",

trALocationAccessPersonPkg PACKAGE
 ATTRIBUTES
 aLocationAccessPerson GET-REPLACE;
 REGISTERED AS {x790Package 49};
 PRESENT IF "an instance supports it.",

trZLocationAccessPersonPkg PACKAGE
 ATTRIBUTES
 zLocationAccessPerson GET-REPLACE;
 REGISTERED AS {x790Package 50};
 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
 X790ASN1Module.maintenanceOrgContactNameMaintenanceOrgContactNameInitial
 GET-REPLACE;
 REGISTERED AS {x790Package 51};
 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 {x790Package 52};
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
X790ASN1Module.maintenanceOrgContactTimeMaintenanceOrgContactTimeInitial
GET-REPLACE;
REGISTERED AS {x790Package 53};
PRESENT IF "an instance supports it.",

trMaintServiceChargePkg PACKAGE
ATTRIBUTES
maintServiceCharge
INITIAL VALUE X790ASN1Module.maintServiceChargeMaintServiceChargeInitial
GET;
REGISTERED AS {x790Package 54};
PRESENT IF "an instance supports it.",

trManagedObjectAccessHoursPkg PACKAGE
ATTRIBUTES
managedObjectAccessHours GET-REPLACE ADD-REMOVE;
REGISTERED AS {x790Package 55};
PRESENT IF "an instance supports it.",

trManagedObjectAccessFromTimePkg PACKAGE
ATTRIBUTES
managedObjectAccessFromTime GET-REPLACE;
REGISTERED AS {x790Package 56};
PRESENT IF "an instance supports it.",

trManagedObjectAccessToTimePkg PACKAGE
ATTRIBUTES
managedObjectAccessToTime GET-REPLACE;
REGISTERED AS {x790Package 57};
PRESENT IF "an instance supports it.",

trManagerContactPersonAttributePkg PACKAGE
ATTRIBUTES
managerContactPerson GET-REPLACE;
REGISTERED AS {x790Package 58};
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 {x790Package 59};
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 {x790Package 60};
PRESENT IF "an instance supports it and trManagerSearchKeyListPkg is not present.",

trManagerSearchKeyListPkg PACKAGE
ATTRIBUTES
managerSearchKeyList GET-REPLACE ADD-REMOVE;
REGISTERED AS {x790Package 61};

PRESENT IF "an instance supports it and trManagerSearchKeyPkg is not present.",

trOutageDurationPkg PACKAGE
ATTRIBUTES
outageDuration
INITIAL VALUE X790ASN1Module.outageDurationOutageDurationInitial
GET;
REGISTERED AS {x790Package 62};
PRESENT IF "an instance supports it.",

trPerceivedTroubleSeverityPkg PACKAGE
ATTRIBUTES
perceivedTroubleSeverity **GET-REPLACE;**
REGISTERED AS {x790Package 63};
PRESENT IF "an instance supports it.",

trPreferredPriorityPkg PACKAGE
ATTRIBUTES
preferredPriority **GET-REPLACE;**
REGISTERED AS {x790Package 64};
PRESENT IF "an instance supports it.",

trRepeatReportPkg PACKAGE
ATTRIBUTES
repeatReport **GET-REPLACE;**
REGISTERED AS {x790Package 65};
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 {x790Package 66};
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 {x790Package 67};
PRESENT IF "an instance supports it and trResponsiblePersonNamePkg is not present.",

trSuspectObjectListPkg PACKAGE
ATTRIBUTES
suspectObjectList **GET SET-BY-CREATE;**
REGISTERED AS {x790Package 68};
PRESENT IF "an instance supports it.",

trTroubleDetectionTimePkg PACKAGE
ATTRIBUTES
troubleDetectionTime
DEFAULT VALUE X790ASN1Module.troubleDetectionTimeTroubleDetectionTimeDefault
GET-REPLACE;
REGISTERED AS {x790Package 69};
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 {x790Package 70};
PRESENT IF "an instance supports it.",

trTroubleReportStatusWindowPkg PACKAGE
 ATTRIBUTES
 additionalTroubleStatusInfo GET,
 troubleReportStatusWindow GET-REPLACE;
 NOTIFICATIONS
 troubleReportProgressNotification;
 REGISTERED AS {x790Package 71};
 PRESENT IF "an instance supports it.",

trTspPriorityPkg PACKAGE
 ATTRIBUTES
 tspPriority GET SET-BY-CREATE;
 REGISTERED AS {x790Package 72};
 PRESENT IF "an instance supports it.";

REGISTERED AS {x790ObjectClass 7};

A.1.8 troubleHistoryRecord

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 SET-BY-CREATE,
 -- Copied from the corresponding trouble report object.
 -- Refers to instance of CNM Service
 -- or GNM object representing a
 -- telecommunications resource.
 receivedTime GET,
 -- indicates trouble report creation time
 troubleFound GET;;;
 CONDITIONAL PACKAGES

thrActivityDurationPkg PACKAGE
 ATTRIBUTES
 activityDuration GET;
 REGISTERED AS {x790Package 73};
 PRESENT IF "an instance supports it.",

thrAdditionalTroubleInfoListPkg PACKAGE
 ATTRIBUTES
 additionalTroubleInfoList GET;
 REGISTERED AS {x790Package 74};
 PRESENT IF "an instance supports it.",

thrAuthorizationPkg PACKAGE
 ATTRIBUTES
 authorizationList GET;
 REGISTERED AS {x790Package 75};
 PRESENT IF "an instance supports it.",

thrCancelRequestedByManagerPkg PACKAGE
 ATTRIBUTES
 cancelRequestedByManager GET;
 REGISTERED AS {x790Package 76};
 PRESENT IF "an instance supports it.",

thrCloseOutNarrPkg PACKAGE
 ATTRIBUTES
 closeOutNarr GET;
 REGISTERED AS {x790Package 77};
 PRESENT IF "an instance supports it.",

thrCloseOutVerificationPkg PACKAGE
 ATTRIBUTES
 closeOutVerification GET;
 REGISTERED AS {x790Package 78};
 PRESENT IF "an instance supports it.",

thrCommitmentTimePkg PACKAGE

ATTRIBUTES
commitmentTime GET;
REGISTERED AS {x790Package 79};
PRESENT IF "an instance supports it.",

thrCustTroubleTickNumPkg PACKAGE
ATTRIBUTES
custTroubleTickNum GET;
REGISTERED AS {x790Package 80};
PRESENT IF "an instance supports it.",

thrPerceivedTroubleSeverityPkg PACKAGE
ATTRIBUTES
perceivedTroubleSeverity GET;
REGISTERED AS {x790Package 81};
PRESENT IF "an instance supports it.",

thrRestoredTimePkg PACKAGE
ATTRIBUTES
restoredTime GET;
REGISTERED AS {x790Package 82};
PRESENT IF "an instance supports it.",

thrTroubleClearancePersonPkg PACKAGE
ATTRIBUTES
troubleClearancePerson GET;
REGISTERED AS {x790Package 83};
PRESENT IF "an instance supports it.",

thrTroubleReportNumberListPkg PACKAGE
ATTRIBUTES
troubleReportNumberList GET;
REGISTERED AS {x790Package 84};
PRESENT IF "an instance supports it.",

thrTroubleTypePkg PACKAGE
ATTRIBUTES
troubleType GET SET-BY-CREATE;
REGISTERED AS {x790Package 85};
PRESENT IF "an instance supports it.";

REGISTERED AS {x790ObjectClass 8};

A.1.9 troubleReport

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,
receivedTime GET,
troubleFound INITIAL VALUE X790ASN1Module.troubleFoundTroubleFoundInitial
GET-REPLACE,
troubleReportID GET,
troubleReportState GET-REPLACE,
troubleReportStatus GET-REPLACE,
troubleReportStatusTime GET,
troubleType GET;;;

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 GET ADD;
 REGISTERED AS {x790Package 86};
 PRESENT IF "an instance supports it.",

trAgentContactPersonAttributePkg PACKAGE
 ATTRIBUTES
agentContactPerson GET;
 REGISTERED AS {x790Package 87};
 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 {x790Package 88};
 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 X790ASN1Module.closeOutNarrCloseOutNarrInitial
 GET-REPLACE;
 REGISTERED AS {x790Package 89};
 PRESENT IF "an instance supports it.",

trManagedObjectInstanceAliasListPkg PACKAGE
 ATTRIBUTES
managedObjectInstanceAliasList GET;
 REGISTERED AS {x790Package 90};
 PRESENT IF "an instance supports it.",

x790NumberListPkg PACKAGE
 ATTRIBUTES
troubleReportNumberList GET;
 REGISTERED AS {x790Package 91};
 PRESENT IF "an instance supports it.",

trRelatedTroubleReportListPkg PACKAGE
 ATTRIBUTES
relatedTroubleReportList GET;
 REGISTERED AS {x790Package 92};
 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 X790ASN1Module.repairActivityListRepairActivityListInitial
 GET ADD;
 -- Support for repairActivityList determined by policies of administration performing repair.
 REGISTERED AS {x790Package 93};
 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 X790ASN1Module.restoredTimeRestoredTimeInitial
 GET-REPLACE;

REGISTERED AS {x790Package 94};
PRESENT IF "an instance supports it.",

trTroubleClearancePersonAttributePkg PACKAGE
ATTRIBUTES
troubleClearancePerson
DEFAULT VALUE X790ASN1Module.troubleClearancePersonTroubleClearancePersonDefault
GET-REPLACE;
REGISTERED AS {x790Package 95};
PRESENT IF "an instance supports it.",

trTroubleReportFormatObjectPtrPkg PACKAGE
ATTRIBUTES
troubleReportFormatObjectPtr GET;
REGISTERED AS {x790Package 96};
PRESENT IF "an instance supports it.",

x790AttributeValueChangePkg PACKAGE
NOTIFICATIONS
"Rec. X.721|ISO/IEC 10165-2 : 1992":attributeValueChange;
REGISTERED AS {x790Package 97};
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 {x790Package 98};
PRESENT IF "an instance supports it.",

trHistoryEventPkg PACKAGE
NOTIFICATIONS
troubleHistoryEventNotification;
REGISTERED AS {x790Package 99};
PRESENT IF "an instance supports it.";

REGISTERED AS {x790ObjectClass 9};

A.1.10 troubleReportFormatDefn

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 {x790Package 100};
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 an
-- object class
REGISTERED AS {x790Package 101};
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 {x790Package 102};
PRESENT IF "an instance supports it.",

trfdTrMayBePresentAttrIDListPkg PACKAGE
ATTRIBUTES

```

tRMayBePresentAttrIDList GET;
REGISTERED AS {x790Package 103};
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 {x790Package 104};
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 {x790Package 105};
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 {x790Package 106};
PRESENT IF "an instance supports it.";

REGISTERED AS {x790ObjectClass 10};

```

A.2 Attribute definitions

A.2.1 Account contact list

```

accountContactList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.AccountContactList;
MATCHES FOR
EQUALITY,
SET-INTERSECTION,
SET-COMPARISON;
REGISTERED AS {x790Attribute 1};

```

A.2.2 Account name

```

accountName ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.AccountName;
MATCHES FOR
EQUALITY,
SUBSTRINGS;
REGISTERED AS {x790Attribute 2};

```

A.2.3 Activity code

```

activityCode ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ActivityCode;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 3};

```

A.2.4 Activity duration

```

activityDuration ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ActivityDuration;
REGISTERED AS {x790Attribute 4};

```

A.2.5 Activity info

```

activityInfo ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ActivityInfo;
MATCHES FOR
EQUALITY,

```

SUBSTRINGS;
REGISTERED AS {x790Attribute 5};

A.2.6 Activity person

activityPerson ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ActivityPerson;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 6};

A.2.7 Additional text

additionalText ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.AdditionalText;
MATCHES FOR
EQUALITY,
SUBSTRINGS;
REGISTERED AS {x790Attribute 7};

A.2.8 Additional trouble info list

additionalTroubleInfoList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.AdditionalTroubleInfoList;
MATCHES FOR
EQUALITY,
SET-INTERSECTION,
SET-COMPARISON;
REGISTERED AS {x790Attribute 8};

A.2.9 Additional trouble status info

additionalTroubleStatusInfo ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.AdditionalTroubleStatusInfo;
MATCHES FOR
EQUALITY,
SET-INTERSECTION,
SET-COMPARISON;
REGISTERED AS {x790Attribute 9};

-- administrativeState attribute is imported from Recommendation X.721

A.2.10 After hrs repair auth

afterHrsRepairAuth ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.AfterHrsRepairAuth;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 10};

A.2.11 Agent contact person

agentContactPerson ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.PersonReach;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 11};

A.2.12 Agent contact object ptr

agentContactObjectPtr ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.AgentContactObjectPtr;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 12};

A.2.13 Alarm record ptr list

alarmRecordPtrList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.AlarmRecordPtrList;
MATCHES FOR
EQUALITY,

SET-INTERSECTION,
SET-COMPARISON;
REGISTERED AS {x790Attribute 13};

-- *alarmStatus* attribute is imported from Recommendation M.3100

A.2.14 Alternate manager contact person

alternateManagerContactPerson ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.PersonReach;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 14};

A.2.15 Alternate manager contact object ptr

alternateManagerContactObjectPtr ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.AlternateManagerContactObjectPtr;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 15};

A.2.16 Applicable managed object class list

applicableManagedObjectClassList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ApplicableManagedObjectClassList;
MATCHES FOR
EQUALITY,
SET-INTERSECTION,
SET-COMPARISON;
REGISTERED AS {x790Attribute 16};

A.2.17 Applicable managed object instance list

applicableManagedObjectInstanceList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ApplicableManagedObjectInstanceList;
MATCHES FOR
EQUALITY,
SET-INTERSECTION,
SET-COMPARISON;
REGISTERED AS {x790Attribute 17};

A.2.18 Authorization list

authorizationList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.AuthorizationList;
REGISTERED AS {x790Attribute 18};

A.2.19 Begin time

beginTime ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.BeginTime;
MATCHES FOR ORDERING;
REGISTERED AS {x790Attribute 19};

A.2.20 Call back info list

callBackInfoList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.CallBackInfoList;
REGISTERED AS {x790Attribute 20};

A.2.21 Called number

calledNumber ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.CalledNumber;
MATCHES FOR
EQUALITY,
SUBSTRINGS;
REGISTERED AS {x790Attribute 21};

A.2.22 Cancel requested by manager

cancelRequestedByManager ATTRIBUTE

WITH ATTRIBUTE SYNTAX X790ASN1Module.CancelRequestedByManager;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 22};

A.2.23 Close out narr

closeOutNarr ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.CloseOutNarr;
REGISTERED AS {x790Attribute 23};

A.2.24 Commitment time

commitmentTime ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.CommitmentTime;
MATCHES FOR
ORDERING;
REGISTERED AS {x790Attribute 24};

A.2.25 Commitment time request

commitmentTimeRequest ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.CommitmentTime;
MATCHES FOR
ORDERING;
REGISTERED AS {x790Attribute 25};

A.2.26 Contact company

contactCompany ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ContactCompany;
MATCHES FOR EQUALITY, SUBSTRINGS;
REGISTERED AS {x790Attribute 26};

A.2.27 Contact details

contactDetails ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ContactDetails;
MATCHES FOR EQUALITY, SUBSTRINGS;
REGISTERED AS {x790Attribute 27};

A.2.28 Contact function

contactFunction ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ContactFunction;
MATCHES FOR EQUALITY;
REGISTERED AS {x790Attribute 28};

A.2.29 Contact id

contactID ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.SimpleNameType;
MATCHES FOR EQUALITY;
REGISTERED AS {x790Attribute 29};

A.2.30 Contact names

contactNames ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.Names;
MATCHES FOR SET-COMPARISON, SET-INTERSECTION;
REGISTERED AS {x790Attribute 30};

A.2.31 Contact object ptr list

contactObjectPtrList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ContactObjectPtrList;
MATCHES FOR
EQUALITY,
SET-INTERSECTION,
SET-COMPARISON;
REGISTERED AS {x790Attribute 31};

A.2.32 Contact type

contactType ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ContactType;
MATCHES FOR EQUALITY, SUBSTRINGS;
REGISTERED AS {x790Attribute 32};

A.2.33 Close out verification

closeOutVerification ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.CloseOutVerification;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 33};

-- currentproblemList attribute is imported from Recommendation M.3100

A.2.34 Cust trouble tick num

custTroubleTickNum ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.CustTroubleTickNum;
MATCHES FOR
EQUALITY,
SUBSTRINGS;
REGISTERED AS {x790Attribute 34};

A.2.35 Customer work center

customerWorkCenter ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.CustomerWorkCenter;
MATCHES FOR
EQUALITY,
SUBSTRINGS;
REGISTERED AS {x790Attribute 35};

A.2.36 Dialog

dialog ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.Dialog;
MATCHES FOR
EQUALITY,
SUBSTRINGS;
REGISTERED AS {x790Attribute 36};

A.2.37 Electronic mail address

electronicMailAddress ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ElectronicMailAddress;
MATCHES FOR SET-COMPARISON, SET-INTERSECTION;
REGISTERED AS {x790Attribute 37};

A.2.38 End time

endTime ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.EndTime;
MATCHES FOR ORDERING;
REGISTERED AS {x790Attribute 38};

A.2.39 Entry time

entryTime ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.EntryTime;
MATCHES FOR ORDERING;
REGISTERED AS {x790Attribute 39};

A.2.40 Escalation list

escalationList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.EscalationList;
REGISTERED AS {x790Attribute 40};

A.2.41 Facsimile telephone number list

facsimileTelephoneNumberList ATTRIBUTE
DERIVED FROM telephoneNumberList;
REGISTERED AS {x790Attribute 41};

-- The eventTime attribute is imported from Recommendation X.721

A.2.42 Hand off center

handOffCenter ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.HandOffCenter;
MATCHES FOR
EQUALITY,
SUBSTRINGS;
REGISTERED AS {x790Attribute 42};

A.2.43 Hand off location

handOffLocation ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.HandOffLocation;
MATCHES FOR
EQUALITY,
SUBSTRINGS;
REGISTERED AS {x790Attribute 43};

A.2.45 Hand off person name

handOffPersonName ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.PersonReach;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 45};

A.2.46 Hand off person ptr

handOffPersonPtr ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.HandOffPersonPtr;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 46};

A.2.47 Hand off time

handOffTime ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.HandOffTime;
MATCHES FOR
ORDERING;
REGISTERED AS {x790Attribute 47};

A.2.48 Initiating mode

initiatingMode ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.InitiatingMode;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 48};

A.2.49 Last update time

lastUpdateTime ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.LastUpdateTime;
MATCHES FOR
ORDERING;
REGISTERED AS {x790Attribute 49};

A.2.50 A location access address

aLocationAccessAddress ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.LocationAddress;
MATCHES FOR
EQUALITY;

REGISTERED AS {x790Attribute 50};

A.2.51 Z location access address

zLocationAccessAddress ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.LocationAddress;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 51};

A.2.52 A location access hours

aLocationAccessHours ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.LocationAccessHours;
REGISTERED AS {x790Attribute 52};

A.2.53 Z location access hours

zLocationAccessHours ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.LocationAccessHours;
REGISTERED AS {x790Attribute 53};

A.2.54 A location access person

aLocationAccessPerson ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.PersonReach;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 54};

A.2.55 Z location access person

zLocationAccessPerson ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.PersonReach;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 55};

A.2.56 Maintenance org contact name

maintenanceOrgContactName ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.PersonReach;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 56};

A.2.57 Maintenance org contact ptr

maintenanceOrgContactPtr ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.MaintenanceOrgContactPtr;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 57};

A.2.58 Maintenance org contact time

maintenanceOrgContactTime ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.MaintenanceOrgContactTime;
MATCHES FOR
ORDERING;
REGISTERED AS {x790Attribute 58};

A.2.59 Maint service charge

maintServiceCharge ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.MaintServiceCharge;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 59};

A.2.60 Managed object access from time

managedObjectAccessFromTime ATTRIBUTE

WITH ATTRIBUTE SYNTAX X790ASN1Module.ManagedObjectAccessFromTime;
MATCHES FOR
ORDERING;
REGISTERED AS {x790Attribute 60};

A.2.61 Managed object access hours

managedObjectAccessHours ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ManagedObjectAccessHours;
REGISTERED AS {x790Attribute 61};

A.2.62 Managed object access to time

managedObjectAccessToTime ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ManagedObjectAccessToTime;
MATCHES FOR
ORDERING;
REGISTERED AS {x790Attribute 62};

A.2.63 Managed object instance

managedObjectInstance ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ManagedObjectInstance;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 63};

A.2.64 Managed object instance alias list

managedObjectInstanceAliasList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ManagedObjectInstanceAliasList;
MATCHES FOR
EQUALITY,
SET-INTERSECTION,
SET-COMPARISON;
REGISTERED AS {x790Attribute 64};

A.2.65 Manager contact person

managerContactPerson ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.PersonReach;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 65};

A.2.66 Manager contact object ptr

managerContactObjectPtr ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ManagerContactObjectPtr;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 66};

A.2.67 Manager search key1

managerSearchKey1 ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ManagerSearchKey;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 67};

A.2.68 Manager search key2

managerSearchKey2 ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ManagerSearchKey;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 68};

A.2.69 Manager search key3

managerSearchKey3 ATTRIBUTE

WITH ATTRIBUTE SYNTAX X790ASN1Module.ManagerSearchKey;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 69};

A.2.70 Manager search key list

managerSearchKeyList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ManagerSearchKeyList;
MATCHES FOR
EQUALITY,
SET-INTERSECTION,
SET-COMPARISON;
REGISTERED AS {x790Attribute 70};

-- operationalState attribute is imported from Recommendation X.721

A.2.71 Outage duration

outageDuration ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.OutageDuration;
REGISTERED AS {x790Attribute 71};

A.2.72 perceived trouble severity

perceivedTroubleSeverity ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.PerceivedTroubleSeverity;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 72};

A.2.73 Preferred priority

preferredPriority ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.PreferredPriority;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 73};

A.2.74 Received time

receivedTime ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ReceivedTime;
MATCHES FOR
ORDERING;
REGISTERED AS {x790Attribute 74};

A.2.75 Related trouble report list

relatedTroubleReportList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.RelatedTroubleReportList;
MATCHES FOR
EQUALITY,
SET-INTERSECTION,
SET-COMPARISON;
REGISTERED AS {x790Attribute 75};

A.2.76 Repair activity

repairActivityID ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.RepairActivityID;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 76};

A.2.77 Repair activity list

repairActivityList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.RepairActivityList;
REGISTERED AS {x790Attribute 77};

A.2.78 Repeat report

repeatReport ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.RepeatReport;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 78};

A.2.79 Responsible person name

responsiblePersonName ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.PersonReach;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 79};

A.2.80 Responsible person ptr

responsiblePersonPtr ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ResponsiblePersonPtr;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 80};

A.2.81 Restored time

restoredTime ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.RestoredTime;
MATCHES FOR
ORDERING;
REGISTERED AS {x790Attribute 81};

A.2.82 Service alias list

serviceAliasList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ServiceAliasList;
MATCHES FOR
EQUALITY,
SET-INTERSECTION,
SET-COMPARISON;
REGISTERED AS {x790Attribute 82};

A.2.83 Service description

serviceDescription ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ServiceDescription;
REGISTERED AS {x790Attribute 83};

A.2.84 Service location list

serviceLocationList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ServiceLocationList;
REGISTERED AS {x790Attribute 84};

-- The Service Identifier is the distinguishing attribute of the CNM Service managed object class. It
-- is assigned by the service provider at the time the service is delivered to the customer. The
-- Service ID may include a combination of the Service Alias attribute and the Service Type attribute
-- (or some other attribute) to guarantee that the Service ID is unique.

A.2.85 Service Id

serviceId ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ServiceId;
MATCHES FOR EQUALITY;
BEHAVIOUR
serviceIDBehaviour BEHAVIOUR
DEFINED AS "The Service ID is an attribute type whose distinguished value can be used as an RDN when naming an instance
of the Management Operations Schedule object class";
REGISTERED AS { x790Attribute 85};

-- The Service Type attribute identifies the category of service (e.g. POTS, CENTREX, private line)

A.2.86 Service type

serviceType ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ServiceType;
MATCHES FOR EQUALITY;
BEHAVIOUR
serviceTypeBehaviour BEHAVIOUR
DEFINED AS "The service Type attribute identifies the distinguishing characteristics of a Service";;
REGISTERED AS {x790Attribute 86};

-- The Supported Service Name List attribute type specifies the services supported by a given managed object --

A.2.87 Supported service name list

supportedServiceNameList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.ObjectList;
MATCHES FOR EQUALITY;
BEHAVIOUR
supportedServiceNameListBehaviour BEHAVIOUR
DEFINED AS "The Supported Service Name List attribute type specifies the services supported by a given managed object";;
REGISTERED AS {x790Attribute 87};

-- supportedByObjectList attribute is imported from Recommendation M.3100

A.2.88 Suspect object list

suspectObjectList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.SuspectObjectList;
MATCHES FOR
EQUALITY,
SET-INTERSECTION,
SET-COMPARISON;
REGISTERED AS {x790Attribute 88};

A.2.89 Telephone number list

telephoneNumberList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.TelephoneNumberList;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
REGISTERED AS {x790Attribute 89};

A.2.90 Trouble clearance person

troubleClearancePerson ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.PersonReach;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 90};

A.2.91 Trouble detection time

troubleDetectionTime ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.TroubleDetectionTime;
MATCHES FOR
ORDERING;
REGISTERED AS {x790Attribute 91};

A.2.92 Trouble found

troubleFound ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.TroubleFound;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 92};

A.2.93 Trouble location

troubleLocation ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.TroubleLocation;
REGISTERED AS {x790Attribute 93};

A.2.94 Tr constrained to single value attr id list

tRConstrainedToSingleValueAttrIDList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.AttributeIDList;
REGISTERED AS {x790Attribute 94};

A.2.95 Trouble report format object ptr

troubleReportFormatObjectPtr ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.TroubleReportFormatObjectPtr;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 95};

A.2.96 Tr format id

tRFormatID ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.TRFormatID;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 96};

A.2.97 Trouble report id

troubleReportID ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.NamingString;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 97};

A.2.98 Tr must be present attr id list

tRMustBePresentAttrIDList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.AttributeIDList;
REGISTERED AS {x790Attribute 98};

A.2.99 Tr may be present attr id list

tRMayBePresentAttrIDList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.AttributeIDList;
REGISTERED AS {x790Attribute 99};

A.2.100 Trouble report number list

troubleReportNumberList ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.TroubleReportNumberList;
MATCHES FOR
EQUALITY,
SET-INTERSECTION,
SET-COMPARISON;
REGISTERED AS {x790Attribute 100};

A.2.101 Trouble report state

troubleReportState ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.TroubleReportState;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 101};

A.2.102 Trouble report status

troubleReportStatus ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.TroubleReportStatus;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 102};

A.2.103 Trouble report status time

troubleReportStatusTime ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.TroubleReportStatusTime;
MATCHES FOR

ORDERING;
REGISTERED AS {x790Attribute 103};

A.2.104 Trouble report status window

troubleReportStatusWindow ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.TroubleReportStatusWindow;
REGISTERED AS {x790Attribute 104};

A.2.105 Trouble type

troubleType ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.TroubleType;
MATCHES FOR
EQUALITY;
REGISTERED AS {x790Attribute 105};

A.2.106 Tsp priority

tspPriority ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.tspPriority;
MATCHES FOR EQUALITY;
REGISTERED AS {x790Attribute 106};

A.2.107 Type text

typeText ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.typeText;
REGISTERED AS {x790Attribute 107};

A.2.108 Unavailable service ptr

unavailableServicePtr ATTRIBUTE
WITH ATTRIBUTE SYNTAX X790ASN1Module.UnavailableServicePtr;
MATCHES FOR EQUALITY;
REGISTERED AS {x790Attribute 108};

A.3 Error messages

A.3.1 Trouble Report Already Exists

troubleReportAlreadyExists PARAMETER
CONTEXT SPECIFIC-ERROR;
WITH SYNTAX X790ASN1Module.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 {x790Parameter 1};

A.3.2 Fallback Reporting

fallBackReporting PARAMETER
CONTEXT SPECIFIC-ERROR;
WITH SYNTAX X790ASN1Module.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 predesignated by agent to receive fallback reporting
-- Agent partially failed or temporarily unavailable for receiving trouble reports.";;
REGISTERED AS {x790Parameter 2};

A.3.3 Can Not close

canNotClose PARAMETER
CONTEXT SPECIFIC-ERROR;
WITH SYNTAX X790ASN1Module.CanNotClose;

BEHAVIOUR canNotCloseBehaviour BEHAVIOUR
DEFINED AS "This error message is sent to the manager when the trouble report cannot be closed by the agent because it is already cleared.";;
REGISTERED AS {x790Parameter 3};

A.3.4 Trouble Report Must Be Present Attribute Missing

tRMustBePresentAttributeMissing PARAMETER
CONTEXT SPECIFIC-ERROR;
WITH SYNTAX X790ASN1Module.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 {x790Parameter 4};

A.3.5 Cannot Verify or Deny at This Time

cannotVerifyOrDenyAtThisTime PARAMETER
CONTEXT SPECIFIC-ERROR;
WITH SYNTAX X790ASN1Module.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 {x790Parameter 5};

A.3.6 Trouble Report Change Denied

troubleReportChangeDenied PARAMETER
CONTEXT SPECIFIC-ERROR;
WITH SYNTAX X790ASN1Module.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 {x790Parameter 6};

A.4 Events

troubleHistoryEventNotification NOTIFICATION
WITH INFORMATION SYNTAX X790ASN1Module.TroubleHistoryInfo;
REGISTERED AS {x790Notification 1};

troubleReportProgressNotification NOTIFICATION
WITH INFORMATION SYNTAX X790ASN1Module.TroubleProgressInfo;
REGISTERED AS {x790Notification 2};

A.5 Name Bindings

account-account NAME BINDING
SUBORDINATE OBJECT CLASS account;
NAMED BY SUPERIOR OBJECT CLASS account;
WITH ATTRIBUTE accountName;
REGISTERED AS {x790NameBinding 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 {x790NameBinding 2};

contact-account NAME BINDING
SUBORDINATE OBJECT CLASS contact;
NAMED BY SUPERIOR OBJECT CLASS account;
WITH ATTRIBUTE contactID;
REGISTERED AS {x790NameBinding 3};

contact-network NAME BINDING
SUBORDINATE OBJECT CLASS contact;
NAMED BY SUPERIOR OBJECT CLASS "Rec. M.3100 : 1992":network AND SUBCLASSES;
WITH ATTRIBUTE contactID;

REGISTERED AS {x790NameBinding 4};

contact-service NAME BINDING
 SUBORDINATE OBJECT CLASS contact;
 NAMED BY SUPERIOR OBJECT CLASS service;
 WITH ATTRIBUTE contactID;
 REGISTERED AS {x790NameBinding 5};

contact-system NAME BINDING
 SUBORDINATE OBJECT CLASS contact;
 NAMED BY SUPERIOR OBJECT CLASS "Rec. X.721|ISO/IEC 10165-2 : 1992":system;
 WITH ATTRIBUTE contactID;
 REGISTERED AS {x790NameBinding 6};

cnmService-account NAME BINDING
 SUBORDINATE OBJECT CLASS cnmService;
 NAMED BY SUPERIOR OBJECT CLASS account;
 WITH ATTRIBUTE serviceID;
 REGISTERED AS {x790NameBinding 7};

eventForwardingDiscriminator-account NAME BINDING
 SUBORDINATE OBJECT CLASS "Rec. X.721|ISO/IEC 10165-2 : 1992":eventForwardingDiscriminator;
 NAMED BY SUPERIOR OBJECT CLASS account;
 WITH ATTRIBUTE "Rec. X.721|ISO/IEC 10165-2 : 1992":discriminatorId;
 CREATE
 WITH-REFERENCE-OBJECT,
 WITH-AUTOMATIC-INSTANCE-NAMING;
 DELETE
 ONLY-IF-NO-CONTAINED-OBJECTS;
 REGISTERED AS {x790NameBinding 8};

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 {x790NameBinding 9};

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 {x790NameBinding 10};

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 {x790NameBinding 11};

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 {x790NameBinding 12};

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 {x790NameBinding 13};

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 {x790NameBinding 14};

repairActivity-telecommunicationsTroubleReport NAME BINDING
SUBORDINATE OBJECT CLASS repairActivity;
NAMED BY SUPERIOR OBJECT CLASS telecommunicationsTroubleReport;
WITH ATTRIBUTE repairActivityID;
REGISTERED AS {x790NameBinding 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 {x790NameBinding 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 {x790NameBinding 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 {x790NameBinding 18};

providerTroubleReport-network NAME BINDING
SUBORDINATE OBJECT CLASS providerTroubleReport;
NAMED BY SUPERIOR OBJECT CLASS "Rec. M.3100 : 1992":network AND SUBCLASSES;
WITH ATTRIBUTE troubleReportID;
REGISTERED AS {x790NameBinding 19};

```

A.6 Abstract syntax

A.6.1

```

X790ASN1Module {itu-t(0) recommendation(0) x(24) x790(790) informationModel(0) asn1module(2)}
DEFINITIONS IMPLICIT TAGS::=BEGIN
-- EXPORTS everything
IMPORTS

AttributeId, ObjectClass, ObjectInstance, GraphicString64
FROM CMIP-1 {joint-iso-itu-t(2) ms(9) cmip(1) modules(0) protocol(3)}
-- Only the Distinguished Name and Local Distinguished Name forms are supported

-- From Recommendation X.721
Time24, AdministrativeState, EventTime, OperationalState, UsageState, SimpleNameType
FROM Attribute-ASN1Module {joint-iso-itu-t ms(9) smi(3) part2(2) asn1Module(2) 1}

-- From Recommendation M.3100
ObjectList, AlarmStatus, CurrentProblemList, SupportedByObjectList
FROM ASN1DefinedTypesModule {itu-t recommendation m(13) gnm(3100)
informationModel(0) asn1Modules(2) asn1DefinedTypesModule(0)};

```

```

x790ObjectClass OBJECT IDENTIFIER::={itu-t(0) recommendation(0) x(24)
x790(790)informationModel(0) managedObjectClass(3)}

x790Package OBJECT IDENTIFIER::={itu-t(0) recommendation(0) x(24)
x790(790)informationModel(0) package(4)}

x790Parameter OBJECT IDENTIFIER::={itu-t(0) recommendation(0) x(24)
x790(790)informationModel(0) parameter(5)}

x790NameBinding OBJECT IDENTIFIER::={itu-t(0) recommendation(0) x(24)
x790(790)informationModel(0) nameBinding(6)}

x790Attribute OBJECT IDENTIFIER::={itu-t(0) recommendation(0) x(24)
x790(790)informationModel(0) attribute(7)}

x790Notification OBJECT IDENTIFIER::={itu-t(0) recommendation(0) x(24)
x790(790)informationModel(0) notification(10)}

activityInfoActivityInfoDefault GraphicString::=""
afterHrsRepairAuthAfterHrsRepairAuthDefault BOOLEAN::=FALSE
authorizationAuthorizationDefault AuthorizationList::={{provided,'000000000'B}}
closeOutNarrCloseOutNarrInitial GraphicString::=""
closeOutVerificationCloseOutVerificationDefault 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::={{"000000000000.0",""}}
restoredTimeRestoredTimeInitial NULL::=NULL
troubleClearancePersonTroubleClearancePersonDefault PersonReach::={number "",name ""}
troubleDetectionTimeTroubleDetectionTimeDefault NULL::=NULL
troubleFoundTroubleFoundInitial TroubleFound::=number:0
troubleReportCancelRequestedByManagerDefault 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, -- OPTIONAL,
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)
...
}
AdditionalText::=GraphicString(SIZE(0..256))

AdditionalTroubleInfoList::=SET OF GraphicString
-- A minimum of 256 octets should be supported
-- if a Japanese operator set of Chinese characters is used for this attribute
-- 256 characters cannot be accommodated.

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
BeginTime::=GeneralizedTime
CallBackInfoList::=SET OF CHOICE{
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
...
}
ContactCompany::=GraphicStringBase
ContactDetails::=GraphicStringBase

```

```

ContactFunction::=ENUMERATED{
    other                (0),
    customerMaintenanceManager (1),
    providerMaintenanceManager (2),
    customerAccountManager (3),
    providerAccountManager (4),
    fieldServiceManager (5),
    repairman (6),
    tester (7),
    screener (8)
...
}
ContactObjectPtrList::=SET OF ObjectInstance
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)
...
}
CustomerWorkCenter::=GraphicString(SIZE(0..64))
CustTroubleTickNum::=GraphicString(SIZE(0..64))
Dialog::=GraphicString(SIZE(0..64))
ElectronicMailAddress::=SET OF IA5String(SIZE(0..64))
ElectronicMailAddressRange::=SET OF GraphicString
EndTime::=GeneralizedTime
EntryTime::=GeneralizedTime
EscalationList::=SET OF SEQUENCE{
    state      RequestState,
    escTime    EscalationTime,
    -- supplied by agent
    requestPerson [0]PersonReach,
    level         [1]OrgLevel OPTIONAL,
    escPerson     [2]PersonReach OPTIONAL
...
}

EscalationTime::=GeneralizedTime
FallBackReporting::=CHOICE{ NULL, GraphicString
...
}
GraphicString128::=GraphicStringBase(SIZE(0..128))
GraphicStringBase::=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))
NameType::=CHOICE{
    number    INTEGER,
    pString   PrintableString
    ...
}
NamingString::=GraphicString(SIZE(0..32))
Name::=CHOICE{
    null      NULL,
    classAndInstance SEQUENCE{
        managedObjectClass      ObjectClass,
        managedObjectInstance    ObjectInstance
        ...
    }
    ...
}
Names::=SET OF Name

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
...
}

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
RelatedObject::=CHOICE{
noObject     NULL,
object       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)
...
}

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
...
}

ServiceId::=NameType
ServiceType::=CHOICE{
    INTEGER,
    PrintableString,
    OBJECT IDENTIFIER
    ...
}

SimpleNameRange ::= CHOICE{
    number    INTEGER,
    string    GraphicString64
...
}

StopTime::=CHOICE{ specific GeneralizedTime, continual NULL
...
}
SuspectObjectList TelephoneNumber::=SET OF SuspectObject
TelephoneNumberListRange::=SET SIZE(0..64) OF TelephoneNumber
TRFormatID::=INTEGER
TelephoneNumber::=GraphicString(SIZE(0..32))

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),
outsidePlantOther	(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	
...	
}	
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
...	
}	
TroubleLocation::=CHOICE{	
locationAddress	[0] LocationAddress,
locationPtr	[1] ObjectInstance
...	
}	

```

TroubleProgressInfo::=SEQUENCE{
    troubleReportStatus      [0] TroubleReportStatus,
                             additionalTroubleStatusInfo
    [1] AdditionalTroubleStatusInfo OPTIONAL
    ...
}
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(0..255){
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),
        clearedAwaitingCustVerification (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

TroubleType::=CHOICE{

number INTEGER{

-- Integer values are to be registered in the
 -- standard. Administrations may restrict
 -- the values to be used.

noDialToneGroup (100),
 noDialTone (101),
 slowDialTone (102),
 circuitDead (103),

-- No audible tone when the telephone receiver is off-hook

canNotCallOutGroup (200),
 canNotCallOut (201),
 canNotBreakDialTone (203),
 dialToneAfterDialing (204),
 highAndDry (205),
 canNotRaise (206),
 allAccessBusy (207),
 canNotCallOut2 (208),
 canNotCallLongDistance (209),
 canNotCallOverseas (210),
 speedCall (211),
 canNotBeCalledGroup (300),
 canNotBeCalled (301),
 canNotBeCalledBusy (302),
 doNotGetCalled (303),
 canNotTripRing (304),
 falseRings (305),
 doNotAnswer (306),
 reachRecording (307),
 canNotRaiseAStation (308),
 canNotRaiseADrop (309),
 canNotRaiseACircuitLocation (310),
 ringNoAnswer (311),
 reorder (312),
 alwaysBusy (313),
 bellDoesNotRing (314),
 bellDoesNotRing2 (315),
 bellRingsCanNotAnswer (316),
 bellRingsAfterAnswer (317),
 noRingNoAnswer (318),
 otherRingTrouble (319),
 receivesCallsForWrongNumber (320),
 recordingOnLine (321),
 canNotBeHeardGroup (400),
 canNotBeHeard (401),
 canNotHear (402),
 fading (403),
 distant (404),
 reachedWrongNumberGroup (500),
 wrongNumber (501),
 circuitOperationGroup (600),
 open (601),
 falseDisconnect (602),
 grounded (603),

canNotBeSignalled	(604),
canNotSignal	(605),
permanentSignal	(606),
improperSupervision	(607),
supervision	(608),
canNotMeet	(609),
canNotReleaseCircuit	(610),
hungUp	(611),
noWinkStart	(612),
noSF	(613),
lowSF	(614),
noContinuity	(615),
cutCable	(616),
openToDEMARC	(617),
noRingGenerator	(618),
badERL	(619),
echo	(620),
hollow	(621),
circuitDead	(622),

-- No activity on circuit at all. Circuit shows no sign of continuity with service provider

circuitDown	(623),
failingCircuit	(624),
noSignal	(625),
seizureOnCircuit	(626),
lossEPSCSorSwitchedServices	(627),
monitorCircuit	(628),
newServiceNotWorking	(629),
openEPSCSorSwitchedServices	(630),
otherVoiceDescribeAdditInfo	(631),
cutOffsGroup	(700),
cutsOff	(701),
noiseProblemGroup	(800),
intermittentNoise	(801),
noisy	(802),
foreignTone	(803),
clipping	(804),
crossTalk	(805),
staticOnLine	(806),
groundHum	(807),
hearsOtherOnLine	(808),
humOnLine	(809),
clicking	(810),
noiseEPSCSorSwitchedServices	(811),
levelTroublesGroup	(900),
lowLevels	(901),
highLevels	(902),
longLevels	(903),
hotLevels	(904),
highEndRollOff	(905),
lowEndRollOff	(906),
needsEqualized	(907),
lineLoss	(908),
doesNotPassFreqResponse	(909),
miscellaneousTroubleGroup	(1000),
hiCapDown	(1001),
carrierDown	(1002),
biPolarViolations	(1003),
frameErrorsHiCap	(1004),
outOfFrame	(1005),
lossOfSync	(1006),
frameSlips	(1007),
noLoopback	(1008),
canNotLoopbackDEMARC	(1009),
recordingOnCircuit	(1010),
linesNeedTagging	(1011),
outwatsRingingin	(1012),
remoteAccess	(1013),
other	(1014),

alarm	(1015),
memoryServiceProblemGroup	(1100),
dataTroubleGroup	(1200),
canNotReceiveData	(1201),
canNotSendData	(1202),
canNotTransmitCanNotReceive	(1203),
noReceive	(1204),
noResponse	(1205),
delay	(1206),
impulseNoise	(1207),
phaseJitter	(1208),
harmonicDistortion	(1209),
highDistortion	(1210),
noDataLoopback	(1211),
noCarrier	(1212),
notPolling	(1213),
dataFramingErrors	(1214),
dropOuts	(1215),
hits	(1216),
noAnswerBack	(1217),
streamer	(1218),
outOfSpecification	(1219),
canNotRunToCSU	(1220),
canNotRunToOSU	(1221),
deadDataCircuit	(1222),
circuitInLoopback	(1223),
errors	(1224),
garbledData	(1225),
invalidData	(1226),
crossModulation	(1227),
slowResponse	(1228),
otherDataDescribeAdditInfo	(1229),
gettingAllOnes	(1230),
slip	(1231),
stationTroubleGroup	(1300),
voiceEquipment	(1301),
dataEquipment	(1302),
videoEquipment	(1303),
otherEquipment	(1304),
stationWiring	(1305),
physicalTroubleGroup	(1400),
lightBurnedOut	(1401),
dataset	(1402),
ttySet	(1403),
highSpeedPrinter	(1404),
aNI	(1405),
aLI	(1406),
canNotActivatePC	(1407),
modem	(1408),
cathodeRayTube	(1409),
looseJack	(1410),
offHook	(1411),
physicalProblem	(1412),
processorDead	(1413),
wiringProblem	(1414),
wireBrokeSetBrokePoleDown	(1415),
noRegister	(1416),
stuckSender	(1417),
otherStationTrouble	(1418),
otherCaseGroup	(1500),
callTransferProblem	(1501),
callWaitingProblem	(1502),
customCallFeatureDoNotWork	(1503),
information	(1504),
threeWayCallingProblem	(1505),
orderWork	(1506),
releaseCktRequestedByIC	(1507),
releaseCktRequestedByEC	(1508),
releaseFacilityRequestedByIC	(1509),

```

releaseFacilityRequestedByEC      (1510),
requestForRoutine                  (1511),
--release                          (1512)
release                            (1512),
requestDispatch                    (1513),
requestMonitorOfCircuit            (1514),
routineTestFailure                 (1515),
lostTimerReport                   (1516),
historicalReports                  (1517),
switchOrTrunkRelated              (1518),
testAssist                         (1519)

...
},
identifier OBJECT IDENTIFIER
...
}

Text128::=PrintableString(SIZE(0..128))
TspPriority::=GraphicString(SIZE(2))

SuspectObject::=SEQUENCE {
    suspectObjectClass      OBJECT
                           IDENTIFIER,
    suspectObjectInstance   ObjectInstance,
    failureProbability      INTEGER
                           OPTIONAL
-- in the range 1-100
...
}

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
...
}
-- TimeInterval shall be non-zero

TypeText::=IA5String(SIZE(0..32))
UnavailableServicePtr::=RelatedObject

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)}
DEFAULT {{{0,0},{23,59}}}
...
}

```

END

A.7 Rules of extensibility

-- Extensibility rules

-- As per Recommendation X.680 on extensibility rules, the productions that are of extensible types

-- are to be indicated by including three (3) ellipses (...) in their type descriptions.

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

--- ENUMERATED

--- named INTEGER

--- 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 named numbers (for named INTEGER type),

-- new members (for SET type and SEQUENCE type) and new choices (for CHOICE type) may

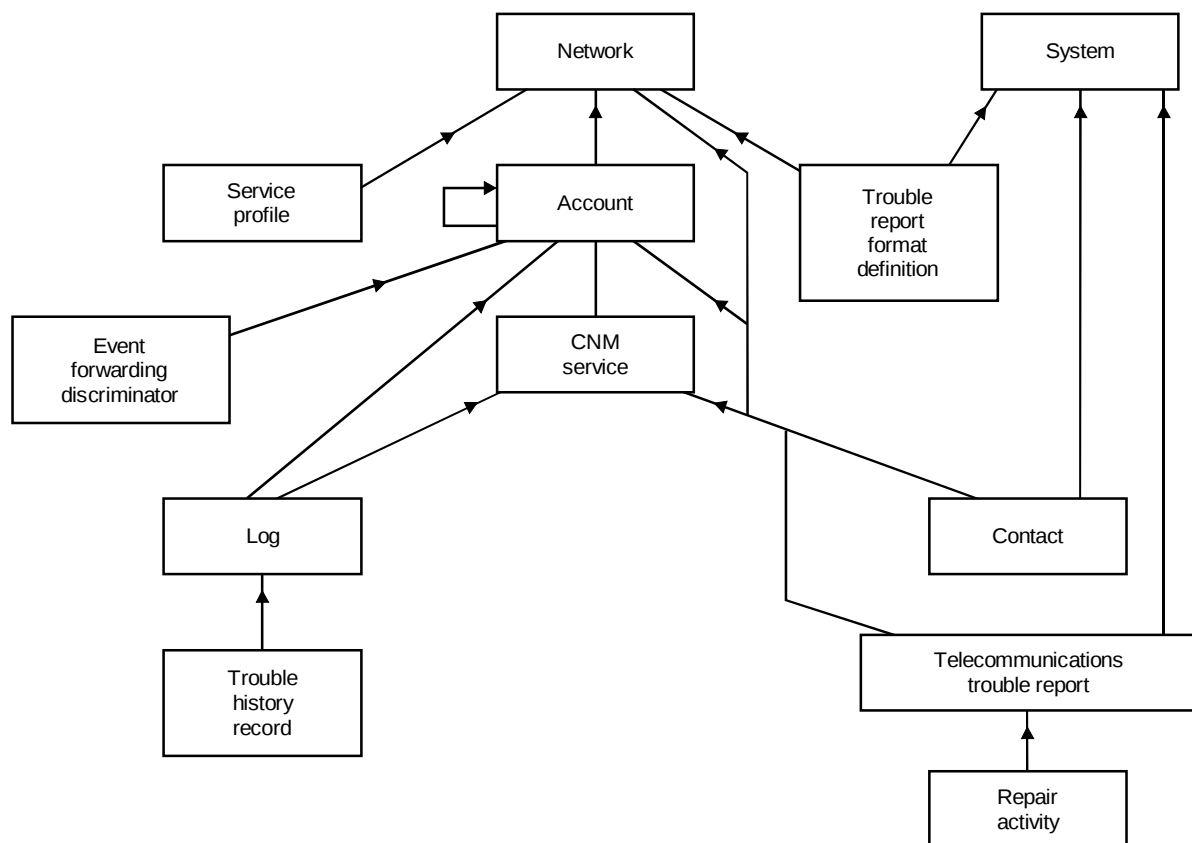
-- be added in the future versions of this Recommendation.

-- 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/ReturnError Problem "mistypedParameter" will be issued.



T0724050-95/d13

FIGURE A.1/X.790
Suggested Name Bindings

Appendix I

Scenarios

(This appendix does not form an integral part of this Recommendation)

I.1 Introduction

This informative appendix defines sample scenarios to explain the trouble management functions and various aspects of the Trouble Management model.

In general, the process of trouble management spans several distinct steps:

- Creation of a Trouble Report;
- Processing and Tracking of the Trouble Report;
- Trouble Report History Notification.

The scenarios contained in this appendix describe how trouble management can be applied to either a service provider to service provider interface, or a client to service provider interface. In general, client to service provider is a restricted subset of service provider to service provider functionality. Therefore, in the scenarios described below, some functionality will apply to both client to service provider and service provider to service provider while other functionality will only be applicable to service provider to service provider interface.

In the service provider to service provider case, both service providers may be able to perform the same set of trouble resolution functions on the same trouble report. The only distinction that can be drawn between the agent role CME and the manager role CME in this case is that the trouble report managed object instance resides in the agent role CME. Note that the functions that can be performed by a manager and those that can be performed by an agent may be constrained using the Security services; however, such distinctions between manager and agent with respect to the allowable trouble management functions that can be performed is beyond the scope of this trouble management Recommendation.

In the client to service provider case, a client may be able to request certain information related to a trouble report while the service provider has responsibility for trouble resolution functions. In this case, the distinction between the agent CME and the manager CME is that, the trouble report managed object instance resides in the agent CME and the agent CME performs trouble resolution functions; while the manager CME is only allowed to perform trouble tracking functions (but not trouble resolution functions).

I.1.1 Trouble management environment

In the client to service provider case, the division of functionality between manager CME and agent CME is straightforward, i.e. the rules that dictate which CME has control over which functionality are clearly defined. In service provider to service provider, some type of Business Level Agreement must be made in order to allocate functional control domains between the manager CME and the agent CME. A framework to help guide these Business Level Agreements is provided in I.1.2. Two examples of the flexibility inherent in the service provider to service provider case are provided below.

Each CME could represent a single work area. Within each of these work areas, trouble report repositories could be defined to represent different work force skills or different telecommunications technologies to provide a logical grouping independent of the CME.

In another configuration, a common problem may need to be worked on through a common CME. A trouble report repository could be set up for common problems between the two CMEs (work areas), since the problem could affect both work areas but the actual source of the problem can only be fixed from one work area.

I.1.2 Trouble management roles

In the client to service provider case, all of the trouble management organizational roles discussed below are assigned by the agent CME. In the service provider to service provider case, however, these organizational roles can be assigned by the manager CME, the agent CME, or both. Business Level Agreements are necessary in the service provider to service provider case to help determine how these roles are assigned in any specific implementation of the service provider to service provider interface.

The person or organization designated to progress a newly created trouble report through its various states is called the “responsible person” for the trouble report. Other people or organization(s) may be delegated responsibility for the trouble report at various times and stages during the transition of the trouble report to its final status. In the client to service provider case, tracking of these changes in responsibilities is a local matter; however, a manager CME could be allowed by the agent CME to know the identity of the responsible person. In the service provider to service provider case, the manager CME may be allowed to change the “responsible person” using PT-SET.

The Trouble Management model will support different generic organizational levels. The levels that have been designated are: “responsible person” and “hand-off centre person”. Although a maintenance manager has not been explicitly defined and is not visible across the interoperable interface, one could be defined to be responsible for the overall performance of one or several Networks. For example, a maintenance manager may be responsible for verifying that each trouble report is closed within a certain period of time or for reviewing the statistics of the previous month’s trouble report resolution performance. This person could be assigned a wide and/or limited responsibility (i.e. network region) of network and managed objects using the security services. One maintenance manager could be assigned to each CME. An example of a maintenance manager would be a supervisor in charge of a Network Operations Centre (NOC). An example of how these levels could be used with the Trouble Management model provide the following different levels of responsibility:

- A responsible person could be defined who would be responsible for the resolution of each problem related to the individual trouble reports. This person could be defined with limited responsibility (created or defined by a maintenance manager) of network and managed objects using a set of services related to managed objects and trouble report tracking. This person may also encompass quality of service managers who may have a wider scope of network responsibility related to overall network supervision for a particular service. The same responsible person could then be assigned to multiple trouble reports in multiple CMEs. Normally, the responsible person would be an operator working within an NOC reporting to the maintenance manager.
- The hand-off centre person is the person who is currently working on resolving a problem. Overall ownership of the resolution of each problem is still assigned to the responsible person but the responsible person may delegate work to be done related to the problem to another person (the secondary responsible).

NOTE – These examples represent generic examples of the logical levels of responsibility for problem solving and tracking in a typical telecommunications network. There are no special constraints in the security services on the assignment of names to each of the previously listed examples, so that the same name may be used for all of them or different names may be used.

I.2 Creation of a trouble report

In response to a detected problem (e.g. an alarm report), two possibilities exist. The agent CME could automatically create a trouble report via some internal algorithm and relate it to the alarm report that was received. The information contained in the alarm report could form the basis for the information included in the newly created trouble report or the alarm report instance could be related to the newly created trouble report. Or a PT-CREATE request under the Kernel Functional Unit could be sent from the manager role CME to the agent role CME to ask for a trouble report to be created.

The trouble report object may be related to other concurrently existing or subsequently created alarm reports that arrive as a result of a common problem. Different trouble report objects may be created for alarm reports that arrived concurrently as a result of different identifiable problems. These differences may only be apparent differences at the time of creation of the trouble reports and subsequently may prove to be related to a common problem. For example, the failure of a fibre optic cable would have a ripple effect that would cause services dependent on the fibre cable to fail.

I.3 Processing and tracking of the trouble report

Once a trouble report has been created, a manager CME may track its progression to resolution by requesting its current status from the agent CME at any time. In addition, this request may cover one or more other trouble reports. In the service provider to service provider case, the manager CME may also change the current status of a trouble report using the PT-SET service provided by the Update State and Status functional unit.

A queued trouble report will change to “open/active” when it has been assigned to a particular responsible person, who has been given the responsibility for resolving the problem. In the service provider to service provider case, responsibility for an “open/active” trouble report may be transferred to another responsible person using the service provided by the Telecommunications Trouble Report Transfer functional unit.

A responsible person may delegate problem resolution to a subordinate responsible person after retrieving a trouble report. In the service provider to service provider case, the service provided by the Refer Telecommunications Trouble Report functional unit may be used.

Once a trouble report has been created, a manager CME may request escalation on it for more urgent attention. The escalation is based on rules and conditions which are beyond the scope of this Recommendation to define.

For an “open/active” trouble report, an agent CME may defer working towards its resolution for an arbitrary length of time. In the service provider to service provider case, the manager CME may also defer the trouble report via a PT-SET request sent to the agent CME. The deferral rules and conditions are beyond the scope of this Recommendation.

I.3.1 Clearing of the trouble report

A trouble report may be entered into a “cleared” state when the hand-off centre person has resolved the trouble. The repair activity list is updated to capture the appropriate information.

A “cleared” trouble report may enter the “open/active” state again if the trouble is subsequently found to be unresolved; however, this implies that verification by the manager CME is necessary. If the manager CME does not confirm or deny the verification in a timely fashion, the agent CME may enter the trouble report into a “closed” state in any case. If the trouble is subsequently found to persist by the manager CME, since the closed trouble report cannot re-enter the “open/active” state, the only means available to the manager CME to get the trouble resolved is by creating a new trouble report for the trouble. While the manager CME verification functionality is not precluded in the service provider to service provider case, it was really intended by its definition to be used for the client to service provider case.

When no verification by the manager CME is necessary, the trouble report could enter the closed state immediately after it enters the “cleared” state; however, depending on the local conditions and availability of the resources, the trouble report may spend an indeterminate time in the “cleared” state.

I.3.2 Closure of the trouble report

A trouble report may be closed by the responsible person from either the agent CME, or (in the service provider to service provider case) from the manager CME using the PT-SET service.

I.3.3 Cancelling of the trouble report

A trouble report may be cancelled prior to achieving a closed status if the agent CME, the manager CME, or both decide that this is warranted. In the client to service provider case, cancellation will only take place if there is concurrence by the agent CME that the cancellation is warranted. For example, configuration changes in the network may invalidate existing trouble reports.

I.4 History notification

When a trouble report is closed, a trouble report history notification will be generated. This notification can then be logged to archive a subset of the information contained within the trouble report.

Appendix II

Further work

(This appendix does not form an integral part of this Recommendation)

The purpose of this informative appendix is to identify items of further work on trouble management.

II.1 Additional requirements not met in this Recommendation

Requirements not met in this Recommendation are as follows:

II.1.1 The capability to include the Outage Duration information in the Trouble History Event Notification and Trouble History record. Refer to requirement 1) in 6.4.

The intent was to capture this information, which is computed locally using the information available in the closed trouble report. Upon being logged, that is to say after closure, some of the information required to compute the outage duration gets discarded making it impossible to generate this information in the future for the particular trouble concerned.

II.1.2 The capability to include pointers to related test results in the trouble report. Refer to requirement 15) in 6.1.

The intent was to capture this information if it was available as it can be very useful in the trouble resolution. Further, if the manager has a capability to carry out the tests then this information can be included in the trouble report at the time of creation, or after initial diagnostic testing by the agent.

II.1.3 The capability to capture the information about the identity of the manager, agent, or person who did the last modification should also be recorded. Refer to requirement 3) in 6.2.

II.2 Future work

Currently, client to service provider and service provider to service provider profiles are defined by this Recommendation. The scenarios in Appendix I will be further expanded if a specific work item is initiated on Trouble Management Ensembles and/or when additional profiles are developed.

Appendix III

Overview of the service model

(This appendix does not form an integral part of this Recommendation)

III.1 This subclause 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 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 to 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.

Appendix IV

Trouble report format examples

(This appendix does not form an integral part of this Recommendation)

IV.1 This appendix 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 must 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 must 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 IV.1 and IV.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 IV.1/X.790

Mandatory trouble report format definition objects

Mandatory trouble report attributes	Format	Required	Supplier
troubleReportID	1.2	X	Agent
additionalTroubleInfoList	1.2	X (Note)	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
NOTE – These normally set-valued attributes will be constrained to single-valued operation.			

TABLE IV.2/X.790

Conditional trouble report format definition objects

Conditional trouble report attributes	Format	Required	Supplier
afterHoursRepairAuth	2	X	Manager
callBackInfoList	2	X (Note)	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
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 (Note)	Agent
NOTE – These normally set-valued attributes will be constrained to single-valued operation.			

Appendix V

(This appendix does not form an integral part of this Recommendation)

This appendix gives a pictorial representation of the pointer attribute relationships between trouble administration objects defined in this Recommendation (see Figure V.1).

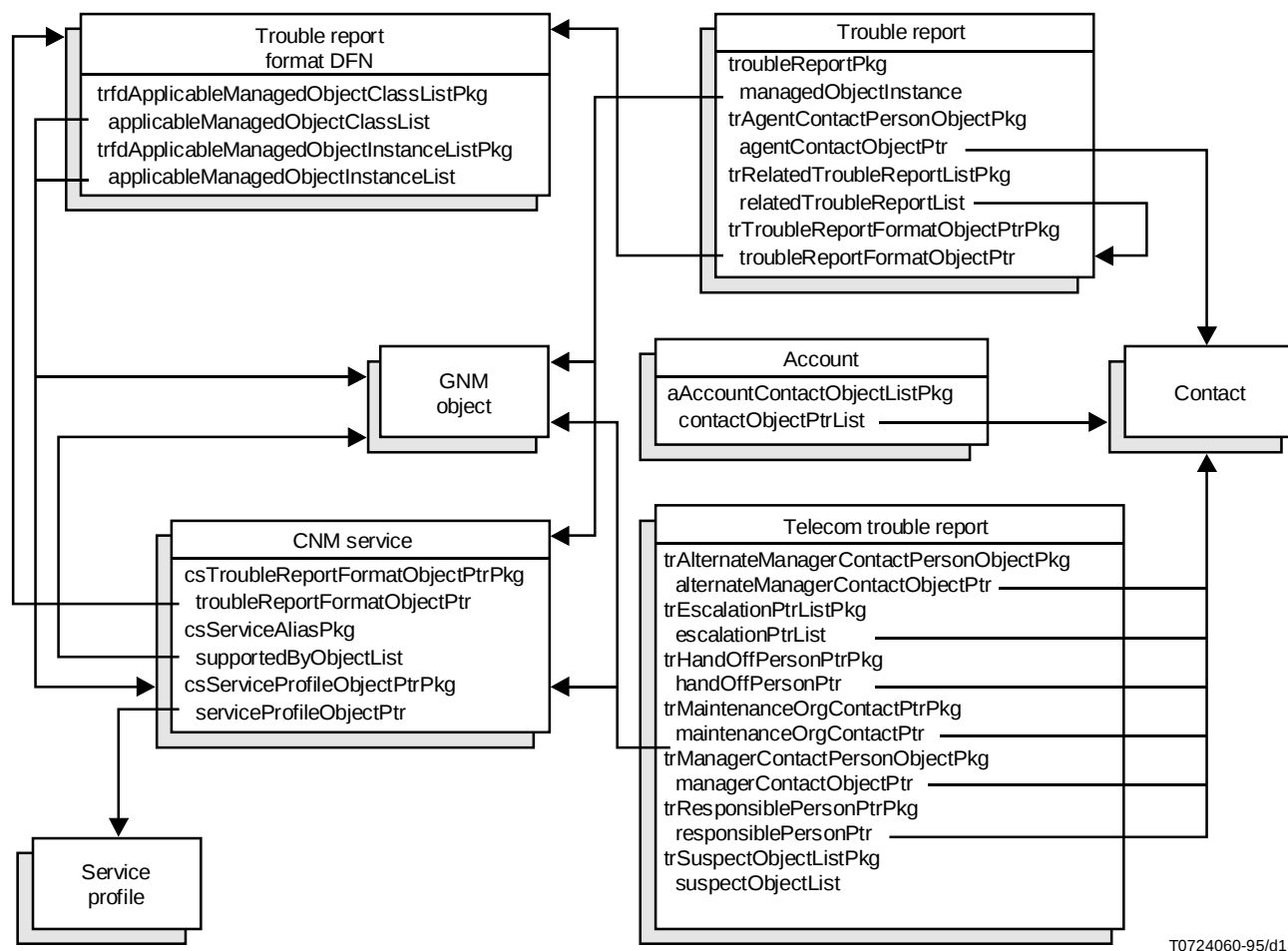


FIGURE V.1/X.790