3GPP TS 32.111-2 V11.1.0 (2012-12)

Technical Specification

**3rd Generation Partnership Project;
Technical Specification Group Services and System Aspects;
Telecommunication management;
Fault Management;
Part 2: Alarm Integration Reference Point (IRP):
Information Service (IS)
(Release 11)**

Keywords
UMTS, management, alarm

3GPP

Postal address

3GPP support office address
650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00   Fax: +33 4 93 65 47 16

Internet
http://www.3gpp.org

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x   the first digit:

1   presented to TSG for information;

2   presented to TSG for approval;

3   or greater indicates TSG approved document under change control.

y   the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z   the third digit is incremented when editorial only changes have been incorporated in the document.

Introduction

The present document is part of a TS-family covering the 3rd Generation Partnership Project: Technical Specification Group Services and System Aspects; Telecommunication management; as identified below:

32.111-1   "Fault Management; Part 1: 3G fault management requirements".

32.111-2   "Fault Management; Part 2: Alarm Integration Reference Point (IRP): Information Service (IS)".

32.111-3   "Fault Management; Part 3: Alarm Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)".

32.111-5   "Fault Management; Alarm Integration Reference Point (IRP): eXtensible Markup Language (XML) definitions".

32.111-7   "Fault Management; Alarm Integration Reference Point (IRP): SOAP Solution Set (SS)".

The present document is part of a set of TSs which describes the requirements and information model necessary for the Telecommunication Management (TM) of 3G systems. The TM principles and TM architecture are specified in 3GPP TS 32.101 [6] and 3GPP TS 32.102 [7].

A 3G system is composed of a multitude of Network Elements (NE) of various types and, typically, different vendors inter-operate in a co-ordinated manner in order to satisfy the network users' communication requirements.
The occurrence of failures in a NE may cause a deterioration of this NE's function and/or service quality and will, in severe cases, lead to the complete unavailability of the NE. In order to minimize the effects of such failures on the Quality of Service (QoS) as perceived by the network users it is necessary to:

-   detect failures in the network as soon as they occur and alert the operating personnel as fast as possible;

isolate the failures (autonomously or through operator intervention), i.e. switch off faulty units and, if applicable, limit the effect of the failure as much as possible by reconfiguration of the faulty NE/adjacent NEs;

-   if necessary, determine the cause of the failure using diagnosis and test routines; and,

-   repair/eliminate failures in due time through the application of maintenance procedures.

This aspect of the management environment is termed "Fault Management" (FM). The purpose of FM is to detect failures as soon as they occur and to limit their effects on the network QoS as far as possible.

The latter is achieved by bringing additional/redundant equipment into operation, reconfiguring existing equipment/NEs, or by repairing/eliminating the cause of the failure.

Fault Management (FM) encompasses all of the above functionalities except commissioning/decommissioning of NEs and potential operator triggered reconfiguration (these are a matter of Configuration Management).

FM also includes associated features in the Operations System (OS), such as the administration of alarm list, the presentation of operational state information of physical and logical devices/resources/functions, and the provision and analysis of the alarm and state history of the network.

1 Scope

The present document defines the Alarm Integration Reference Point (IRP) Information Service (IS), which addresses the alarm surveillance aspects of Fault Management (FM), applied to the N Interface.

The purpose of the AlarmIRP is to define an interface through which a "system" (typically a Network Element Manager or a Network Element) can communicate alarm information for its managed objects to one or several Manager Systems (typically Network Management Systems).

The Alarm IRP IS defines the semantics of alarms and the interactions visible across the reference point in a protocol neutral way. It defines the semantics of the operations and notifications visible in the IRP. It does not define the syntax or encoding of the operations, notifications and their parameters.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.

For a specific reference, subsequent revisions do not apply.

For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1]   3GPP TS 32.150: "Telecommunication management; Integration Reference Point (IRP) Concept and definitions".

[2]   ITU-T Recommendation X.733 (02/92): "Information technology - Open Systems Interconnection - Systems Management: Alarm reporting function".

[3]   ITU-T Recommendation X.721: "Information Technology - Open Systems Interconnection - Structure of management information: Definition of management information".

[4]   3GPP TS 32.401 "Telecommunication management; Performance Management (PM); Concept and requirements".

[5]   3GPP TS 32.302: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Information Service (IS)".

[6]   3GPP TS 32.101: "Telecommunication management; Principles and high level requirements".

[7]   3GPP TS 32.102: "Telecommunication management; Architecture".

[8]   Void.

[9]   3GPP TS 32.111-1: "Telecommunication management; Fault Management; Part 1: 3G fault management requirements".

[10]   .

[11]   ITU-T Recommendation M.3100 (07/95): "Generic network information model".

[12]   Void.

[13]   Void.

[14]   3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP) management; Information Service (IS)".

[15]   ITU-T Recommendation X.736: "Information technology - Open Systems Interconnection - Systems Management: Security alarm reporting function".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TS 32.111-1 [9] and the following apply:

active alarm: an alarm that has not been cleared (i.e. an alarm whose `perceivedSeverity` is not `Cleared`).

Event: occurrence that is of significance to network operators, the NEs under surveillance and Network Management applications. Events do not have state.

IRPAgent: See 3GPP TS 32.150 [1].

IRPManager: See 3GPP TS 32.150 [1].

IRP document version number string (IRPVersion): See 3GPP TS 32.312 [14].

Matching-Criteria-Attributes: which identifies a set of ITU-T Recommendation X.733 [2] defined attributes. Notifications carrying identical values for these attributes are considered to be carrying alarm information related to (a) the same network resource and (b) the same alarmed condition. The matching-criteria-attributes are: `objectInstance`, `eventType`, `probableCause` and `specificProblem`, if present.

Notification: which refers to the transport of events from `IRPAgent` to `IRPManager`.
In this IRP, notifications are used to carry alarm information from `IRPAgent` to `IRPManager`.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

DN	Distinguished Name
EM	Element Manager
FM	Fault Management
IOC	Information Object Class
IRP	Integration Reference Point
IS	Information Service
NE	Network Element
NM	Network Manager
OS	Operations System
QoS	Quality of Service
SS	Solution Set
SupportIOC	Support Information Object Class
TM	Telecommunication Management
UML	Unified Modelling Language

4 Basic aspects

4.1 Void

4.2 System Context

The general definition of the System Context for the present IRP is found in 3GPP TS 32.150 [1] subclause 4.7.
In addition, the set of related IRP(s) relevant to the present IRP is shown in the two diagrams below.
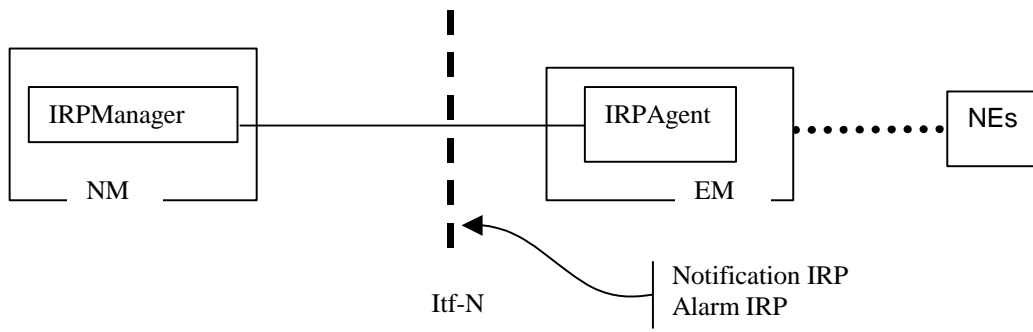
Figure 1: System Context A
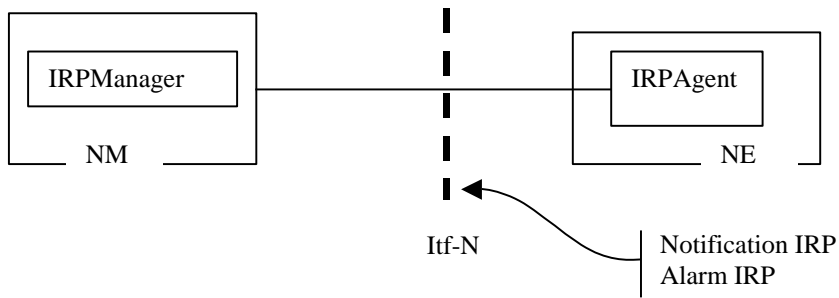
Figure 2: System Context B

5 Information Object Classes

5.1 Imported information entities and local labels

Label reference	Local label
32.302 [5], SupportIOC, NotificationIRP	NotificationIRP
32.302 [5], interface, notificationIRPNotification	NotificationIRPNotification
32.312 [14], SupportIOC, ManagedGenericIRP	ManagedGenericIRP

5.2 Class diagram

This clause introduces the set of classes (i.e. IOCs, SupportIOCs) that encapsulate information within the AlarmIRP. The intent is to identify the information required for the AlarmIRP implementation of its operations and notification emission. This clause provides the overview of all support object classes in UML. Subsequent clauses provide more detailed specification of various aspects of these support object classes.

5.2.1 Attributes and relationships
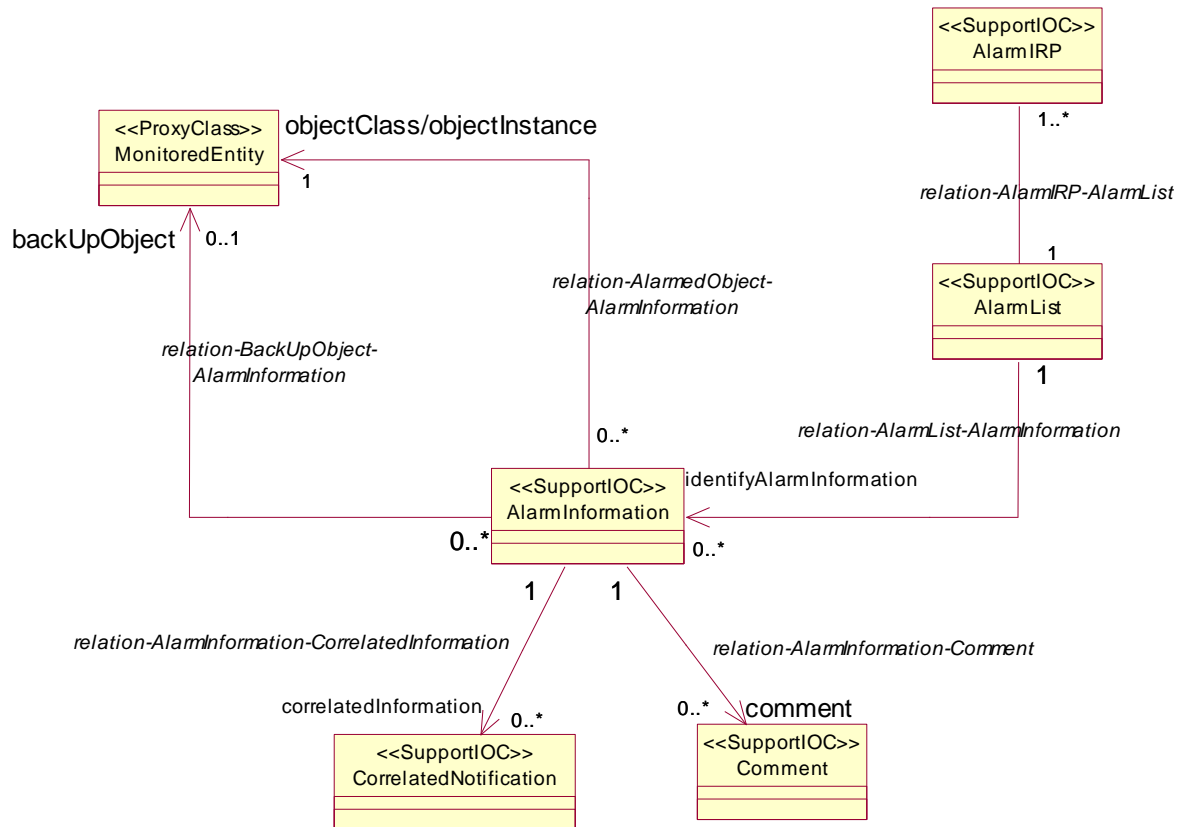
5.2.2 Inheritance
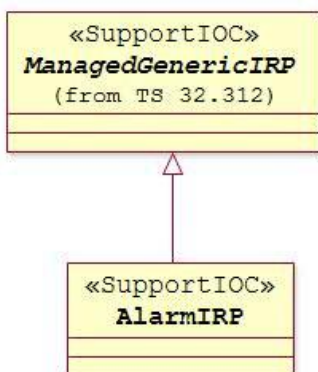
5.3 Information Object Class Definitions

5.3.1 `AlarmInformation`

5.3.1.1 Definition

`AlarmInformation` contains information about alarm condition of an alarmed `MonitoredEntity`.

One `AlarmIRP` is related to at most one `AlarmList`. The `IRPAgent` or its related AlarmIRP or the related `AlarmList` assigns an identifier, called `alarmId`, to each `AlarmInformation` in the `AlarmList`. An `alarmId` unambiguously identifies one `AlarmInformation` in the `AlarmList`.

5.3.1.2   Attribute

Attribute name	Support Qualifier
alarmId	M
notificationId (note 1)	M
alarmRaisedTime	M
alarmClearedTime	M
alarmChangedTime	O
eventType	M
probableCause	M
perceivedSeverity	M
rootCauseIndicator	O
specificProblem	O
backedUpStatus	O
trendIndication	O
thresholdInfo	O
stateChangeDefinition	O
monitoredAttributes	O
proposedRepairActions	O
additionalText	O
additionalInformation	O(see note 4)
ackTime	M
ackUserId	M
ackSystemId	O
ackState	M
clearUserId	O (see note 2)
clearSystemId	O (see note 2)
serviceUser	O (see note 3)
serviceProvider	O (see note 3)
securityAlarmDetector	O (see note 3)

NOTE 1:   This attribute may be "retired/removed" in Release 5 when Log IRP is introduced. Its removal implies that information carried in this attribute is no longer made accessible to IRPManager via the getAlarmList().

NOTE 2:   These attributes and qualifiers are applicable only if the IRPAgent supports clearAlarms() (they are absent if clearAlarms() is not supported).

NOTE 3:   These attributes must be supported if the IRPAgent emits `notifyNewAlarm` that carries security alarm information.

NOTE 4:   This attribute is optionally populated whenever vendor specific attributes are needed.
A specific condition for this optional population is when an alarm presented by the EM (e.g. EM user interface) has different values of perceived severity, and / or alarm type, compared with the values presented to the Itf-N.

5.3.1.3   State diagram

Alarms have states. The alarm state information is captured in `AlarmInformation` in `AlarmList`.

The solid circle icon represents the `Start State`. The double circle icon represents the `End State`. In this state, the alarm is `Cleared` and acknowledged. The AlarmInformation shall not be accessible via the IRP and is removed from the `AlarmList`.

Note the state diagram uses " X / Y ^ Z " to label the arc that indicates state transition. The meanings of X, Y and Z are:

-   X identifies the triggering event

-   Y identifies the action of `AlarmIRP` because of the triggering event

-   Z is the notification to be emitted by `AlarmIRP` because of the triggering event

Note that `acknowledgeAlarm^notifyAckStateChanged` and the `unacknowledgeAlarm^notifyAckStateChange` refer to cases when the request of the `IRPManager` is successful for the `AlarmInformation` concerned. They do not refer to the cases when the request is a failure since in the failure cases, no state transition would occur.

Note that, to reduce cluttering to the diagram, the `setComment^notifyComment` is not included in the figure. One transition should be applied from `unack&unclear` to itself. Similarly, another transition should be applied from `ack&unclear` to itself. Another one is from `unack&clear` to itself.

"PS" used in the state diagram stands for "perceived severity".

Figure A is used if it supports ^notifyChangedAlarm and Figure B is used if it does not support ^notifyChangedAlarm.
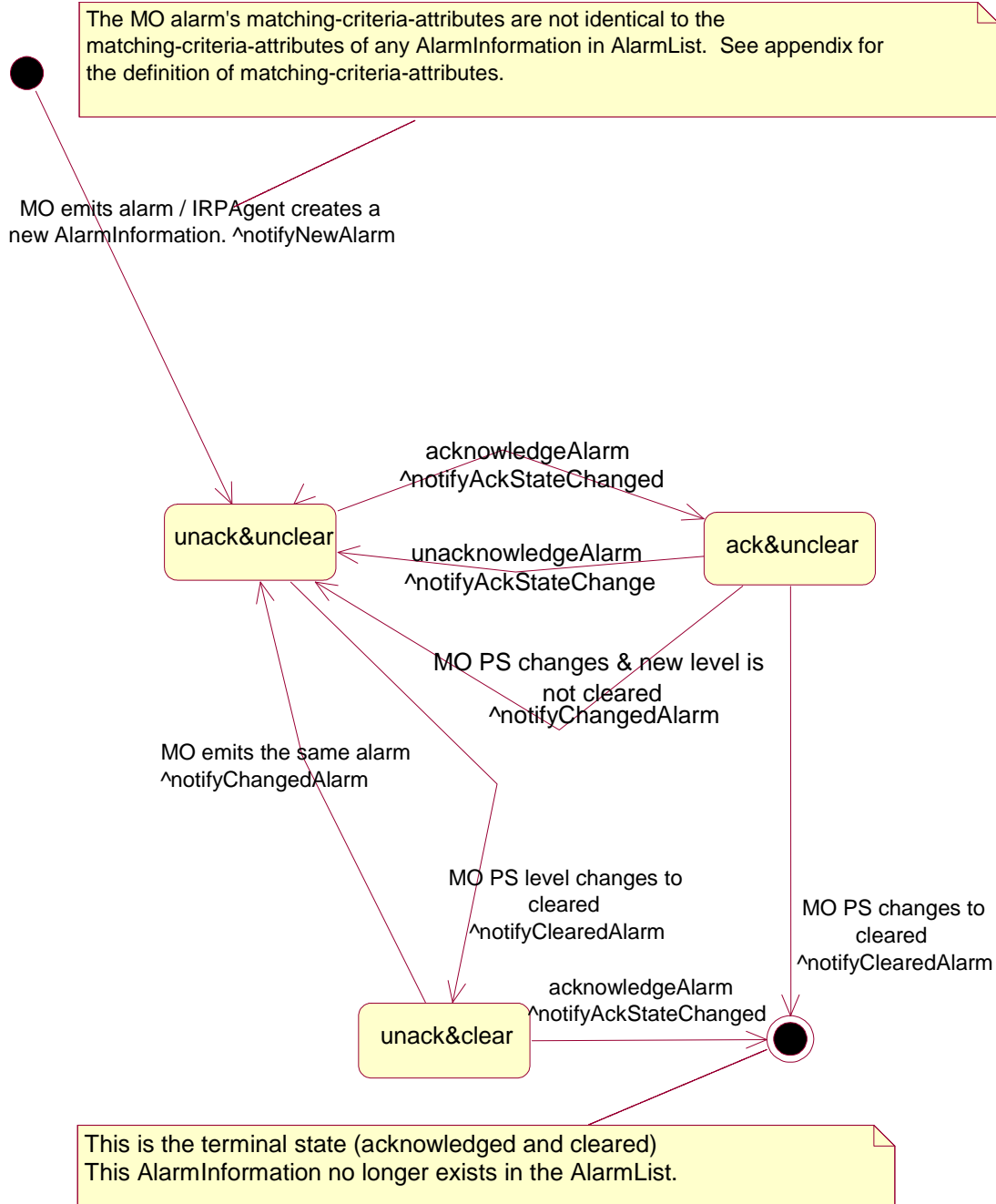

Figure A. ^notifyChangedAlarm supported

The MO alarm's matching-criteria-attributes are not identical to the matching-criteria-attributes of any AlarmInformation in AlarmList. See appendix for the definition of matching-criteria-attributes.

MO emits alarm / IRPAgent creates a new AlarmInformation. ^notifyNewAlarm

MO PS changes & new level is not cleared ^notifyClearedAlarm,

acknowledgeAlarm ^notifyAckStateChanged

unack&unclear

unacknowledgeAlarm ^notifyAckStateChange

ack&unclear

MO PS level changes to cleared ^notifyClearedAlarm

MO PS changes to cleared ^notifyClearedAlarm

acknowledgeAlarm / MO emits the same alarm ^notifyAckStateChanged

unack&clear

This is the terminal state (acknowledged and cleared) This AlarmInformation no longer exists in the AlarmList.

Figure B. ^notifyChangedAlarm not supported

5.3.2 AlarmList

5.3.2.1 Definition

AlarmIRP maintains an AlarmList that contains currently active alarms (i.e. AlarmInformation whose perceivedSeverity is not Cleared) and alarms that are Cleared but not yet acknowledged.

5.3.2.2 Attribute

There is no additional attribute defined for this class besides those inherited.

5.3.3 AlarmIRP

5.3.3.1 Definition

AlarmIRP is the representation of the alarm management capabilities specified by the present document. This class inherits from ManagedGenericIRP class specified in 3GPP TS 32.312 [14].

5.3.3.2 Attribute

There is no additional attribute defined for this class besides those inherited.

5.3.3.3 Notification Table

Name	Qualifier	Notes
notifyAlarmListRebuilt	M	See 6.8.4.
notifyPotentialFaultyAlarmList	O	See 6.11.1.

5.3.4 Comment

5.3.4.1 Definition

Comment contains commentary and associated information such as the time when the commentary is made.

5.3.4.2 Attribute

Attribute Name	Support Qualifier
commentTime	M
commentText	M
commentUserId	M
commentSystemId	O

5.3.5 CorrelatedNotification

5.3.5.1 Definition

It identifies one MonitoredEntity. For that MonitoredEntity identified, a set of notification identifiers is also identified. One or more CorrelatedNotification instances can be related to an AlarmInformation. In this case, the information of the AlarmInformation is said to be correlated to information carried in the notifications identified by the CorrelatedNotification instances. See further definition of correlated notification in ITU-T Recommendation X.733 [2], clause 8.1.2.9.

The notification identified by the CorrelatedNotification, as defined in ITU-T and used here, can carry all types of information and not restricted to carrying alarm information only (see TS 32.302 [5]). For example, a notification, identified by the CorrelatedNotification, can indicate a managed instance attribute value change. In this case, the information of the AlarmInformation is said to be correlated to the managed instance attribute value change event.

The meaning of correlation is dependent on the type of notification itself. See the comment column of the correlatedNotification input parameter for each type of notification, such as notifyNewAlarm.

Notification carries AlarmInformation. The AlarmInformation instances referred to by the correlatedNotification may or may not exist in the AlarmList. For example, the AlarmInformation carried by the identified notification may have been acknowledged and Cleared and therefore, no longer exist in the AlarmList.

5.3.5.2 Attribute

Attribute Name	Support Qualifier
source	M
notificationIdSet	M

5.3.6 MonitoredEntity

5.3.6.1 Definition

It represents classes that can have an alarmed state. The types of classes that can have alarmed state are:

a) All classes whose Notification Tables include alarm notifications.

b) VSE subclass of 3GPP defined classes and VSE defined classes that can have alarmed state.

The objectClass and objectInstance of this class identifies an instance of this class. The AlarmInformation uses this information in two places. In one place, the information is used to identify the instance that is in alarmed state. In another place, the information is used to identify an instance that can be used as the back up network resource for the instance that is in alarmed state.

5.3.6.2 Attribute

There is no attribute for this class.

5.4 Information relationships definition

5.4.1 relation-AlarmIRP-AlarmList (M)

5.4.1.1 Definition

This represents the relationship between `AlarmIRP` and `AlarmList`.

5.4.1.2 Role

There is no role defined for this relationship.

5.4.1.3 Constraint

There is no constraint for this relationship.

5.4.2 relation-AlarmList-AlarmInformation (M)

5.4.2.1 Definition

This represents the relationship between `AlarmList` and `AlarmInformation`.

5.4.2.2 Role

Name	Definition
identifyAlarmInformation	It represents a capability to obtain the information contained in AlarmInformation.

5.4.2.3 Constraint

Name	Definition
inv_ hasAlarmInformation1	No AlarmInformation playing the role of theAlarmInformation shall have its perceivedSeverity = "cleared" and its ackState = "acknowledged".
inv_ hasAlarmInformation2	The alarmId of all AlarmInformation instances playing the role of theAlarmInformation are distinct.

5.4.3 relation-AlarmInformation-Comment (M)

5.4.3.1 Definition

This represents the relationship between `AlarmInformation` and `Comment`.

5.4.3.2 Role

Name	Definition
comment	It represents a capability to obtain the information contained in Comment.

5.4.3.3 Constraint

There is no constraint.

5.4.4 relation-AlarmInformation-CorrelatedNotification (M)

5.4.4.1 Definition

This represents the relationship between `AlarmInformation` and `CorrelatedNotification`.

5.4.4.2 Role

Name	Definition
correlatedNotification	It represents a capability to obtain the information contained in CorrelatedNotification.

5.4.4.3 Constraint

There is no constraint.

5.4.5 relation-AlarmedObject-AlarmInformation (M)

5.4.5.1 Definition

This represents the relationship between `MonitoredEntity` and `AlarmInformation`.

5.4.5.2 Role

Name	Definition
objectClass/objectInstance	It represents the capability to obtain the identification, in terms of objectClass and objectInstance, of alarmed network resource.

5.4.5.3   Constraint

Name	Definition
inv_relation-AI-ME	All AlarmInformation involved in this relationship with the same MonitoredEntity shall have at least one different value in the following attributes: eventType, probableCause and specificProblem.

5.4.6   relation-backUpObject-AlarmInformation (O)

5.4.6.1   Definition

The relationship represents the relationship between `AlarmInformation` and the `backUpObject`.

5.4.6.2   Role

Name	Definition
backUpObject	It represents a capability to obtain the identification, in terms of objectClass and objectInstance, of the backUpObject.

5.4.6.3   Constraint

Name	Definition
inv_identifyBackUpObject	This relationship is present if and only if the AlarmInformation.backedUpStatus attribute is present and is indicating true.

5.5   Information attribute definition

5.5.1   Definition and legal values

Name	Definition	Legal Values
alarmId	It identifies one AlarmInformation in the AlarmList.	
notificationId	It identifies the notification that carries the AlarmInformation.	
alarmRaisedTime	It indicates the date and time when the alarm is first raised by the alarmed resource.	All values indicating valid time.
alarmChangedTime	It indicates the last date and time when the AlarmInformation is changed by the alarmed resource. Changes to AlarmInformation caused by invocations of the IRPManager would not change this date and time.	All values indicating valid time.
alarmClearedTime	It indicates the date and time when the alarm is Cleared.	All values indicating valid time.
eventType	It indicates the type of event. See Annex A for information on event type.	See Annex A.
probableCause	It qualifies alarm and provides further information than eventType. See Annex B for a complete listing.	See Annex B.
perceivedSeverity	It indicates the relative level of urgency for operator attention.	Critical, Major, Minor, Warning, Indeterminate, Cleared: see ITU-T Recommendation X.733 [2]. This IRP does not recommend the use of indeterminate.
specificProblem	It provides further qualification on the alarm than probableCause. This attribute value shall be single-value and of simple type such as integer or string. See definition in ITU-T Recommendation X.733 [2] clause 8.1.2.2.	Provided by vendor.
backedUpStatus	It indicates if an object (the MonitoredEntity) has a back up. See definition in ITU-T Recommendation X.733 [2] clause 8.1.2.4.	All values that carry the semantics of backedUpStatus defined by ITU-T X.733 [2] clause 8.1.2.4.
trendIndication	It indicates if some observed condition is getting better, worse, or not changing.	"Less severe", "no change", "more severe": see definition in ITU-T Recommendation X.733 [2] clause 8.1.2.6.
thresholdInfo	It indicates the crossed threshold information such as: • The identifier of the monitored attribute whose value has crossed a threshold, • The threshold settings, • The observed value that have crossed a threshold, etc. See definition in ITU-T Recommendation X.733 [2] clause 8.1.2.7. See also for information in TS 32.401 [4] subclause 5.6.	
stateChangeDefinition	It indicates MO attribute value changes. See definition in ITU-T Recommendation X.733 [2] clause 8.1.2.10.	
monitoredAttributes	It indicates MO attributes whose value changes are being monitored. See definition in ITU-T Recommendation X.733 [2] clause 8.1.2.11.	
proposedRepairActions	It indicates proposed repair actions. See definition in ITU-T Recommendation X.733 [2] clause 8.1.2.12.	

Name	Definition	Legal Values
additionalText	It carries semantics that is outside the scope of this IRP specification. It may provide the identity of the NE (e.g. RNC, Node-B) from which the alarm has been originated. It corresponds to the "user label" attribute of the object class representing the NE in the Generic Network Resource Model [10]. It can contain further information on the alarm.	N/A
additionalInformation	This attribute when present allows the inclusion of a set of vendor specific alarm information in the alarm. A specific condition for this optional population is when an alarm presented by the EM (e.g. EM user interface) has different values of perceived severity, and / or alarm type, compared with the values presented to the Itf-N. Any other uses of additional information on the alarm and its semantics is outside the scope of this IRP.	The additional information field is a list of one or more information parts. This specification allows the support of two such information parts to carry • vendor defined perceived severity • vendor defined alarm type using defined identification. Other vendor specific information parts are allowed by using vendor specific identifications.
ackTime	It identifies the time when the alarm has been acknowledged or unacknowledged the last time, i.e. it registers the time when ackState changes.	All values that indicate valid time that are later than that carried in alarmRaisedTime.
ackUserId	It identifies the last user who has changed the Acknowledgement State.	It can be used to identify the human operator such as "John Smith" or it can identify a group, such as "Team Six", or it can contain no information such as "".
ackSystemId	It identifies the system (EM or NM) that last changed the ackState of an alarm, i.e. acknowledged or unacknowledged the alarm.	It can be used to identify the system, such as "system 6" or it can contain no information such as "".
ackState	It identifies the Acknowledgement State of the alarm.	Acknowledged: the alarm has been acknowledged. Unacknowledged: the alarm has been unacknowledged or the alarm has never been acknowledged.
commentTime	It carries the time when the comment has been added to the alarm.	
commentText	It carries the textual comment.	
commentUserId	It carries the identification of the user who made the comment.	
commentSystemId	It carries the identification of the system (EM or NM) from which the comment is made. That system supports the user that made the comment.	
rootCauseIndicator	It indicates that this AlarmInformation is the root cause of the events captured by the notifications whose identifiers are in the related CorrelatedNotification instances.	"Yes", "No"
source	It identifies one MonitoredEntity.	All values that carry the semantics of DN.
notificationIdSet	It carries one or more notification identifiers.	
clearUserId	It carries the identity of the user who invokes the clearAlarms operation.	It can be used to identify the human operator such as "John Smith" or it can identify a group, such as "Team Six", or it can contain no information such as "".
clearSystemId	It carries the identity of the system in which the IRPManager runs. That IRPManager supports the user who invokes the clearAlarms().	It can be used to identify the system, such as "system 6" or it can contain no information such as "".

Name	Definition	Legal Values
serviceUser	It identifies the service-user whose request for service provided by the serviceProvider led to the generation of the security alarm.	This attribute may carry no information if the server user is not identifiable.
serviceProvider	It identifies the service-provider whose service is requested by the serviceUser and the service request provokes the generation of the security alarm.	
securityAlarmDetector	It carries the identity of the detector of the security alarm.	This attribute may carry no information if the security alarm detector is not identifiable.

5.5.2   Constraints

Name	Definition
inv_alarmChangedTime	Time indicated shall be later than that carried in alarmRaisedTime.
inv_alarmClearedTime	Time indicated shall be later than that carried in alarmRaisedTime.
inv_ackTime	Time indicated shall be later than that carried in alarmRaisedTime.
inv_notificationId	NotificationIds shall be chosen to be unique across all notifications of a particular Managed Object (representing the NE) throughout the time that alarm correlation is significant. The algorithm by which alarm correlation is accomplished is outside the scope of this IRP.

6 Interface Definition

6.1 Class diagram

<<SupportIOC>> AlarmIRP	<<use>> →	<<Notification>> AlarmIRPNotifications_1
		notifyNewAlarm() notifyAckStateChange() notifyClearedAlarm() notifyAlarmListRebuilt()

<<may use>>

<<Notification>> AlarmIRPNotification_2

notifyChangedAlarm()

<<may use>> <<may use>> <<may use>>

<<Notification>> AlarmIRPNotification_5	<<Notification>> AlarmIRPNotification_4	<<Notification>> AlarmIRPNotification_3
notifyCorrelatedNotificationChanged...	notifyPotentialFaultyAlarmLis...	notifyComments()
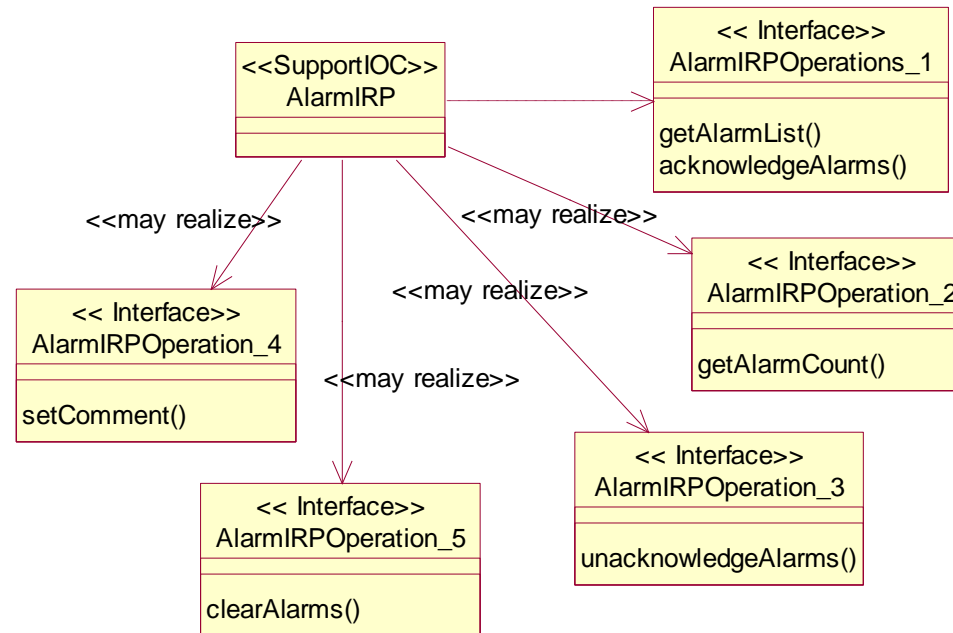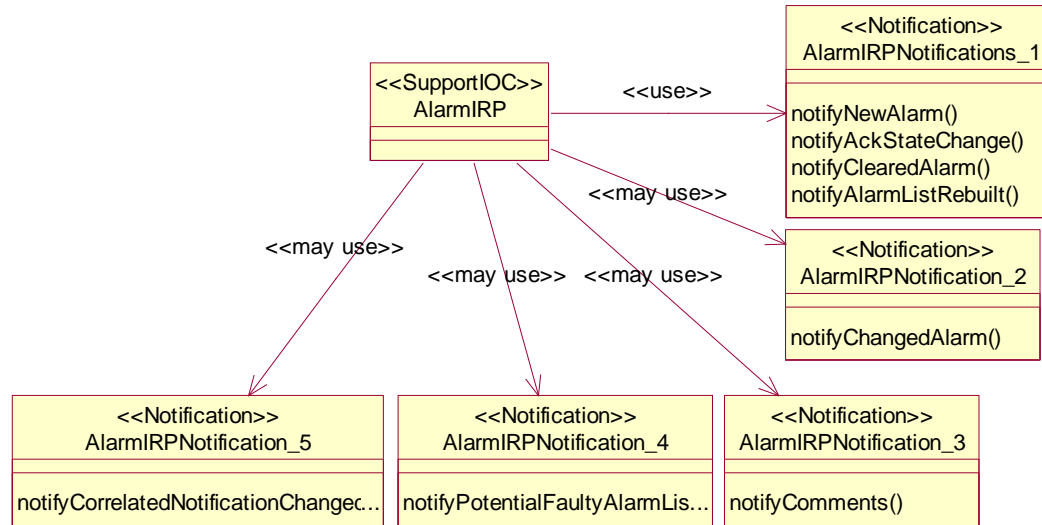
6.2   Generic rules

Rule 1: each operation with at least one input parameter supports a pre-condition valid_input_parameter which indicates that all input parameters shall be valid with regards to their information type. Additionally, each such operation supports an exception operation_failed_invalid_input_parameter which is raised when pre-condition valid_input_parameter is false. The exception has the same entry and exit state.

Rule 2: Each operation with at least one optional input parameter supports a set of pre-conditions supported_optional_input_parameter_xxx where "xxx" is the name of the optional input parameter and the pre-condition indicates that the operation supports the named optional input parameter. Additionally, each such operation supports an exception operation_failed_unsupported_optional_input_parameter_xxx which is raised when (a) the pre-condition supported_optional_input_parameter_xxx is false and (b) the named optional input parameter is carrying information. The exception has the same entry and exit state.

Rule 3: each operation shall support a generic exception operation_failed_internal_problem that is raised when an internal problem occurs and that the operation cannot be completed. The exception has the same entry and exit state.

6.3   Interface AlarmIRPOperations_1 (M)

6.3.1   acknowledgeAlarms (M)

6.3.1.1   Definition

The IRPManager invokes this operation to acknowledge one or more alarms.

The IRPManager may supply the identifier of the alarm and its perceivedSeverity. The reason for supplying the perceivedSeverity, in addition to the identifier of the alarm, is given in Annex E.

6.3.1.2   Input Parameters

Name	Qualifier	Information Type	Comment
alarmInformationAndSeverityReferenceList	M	List of AlarmInformation.alarmId and AlarmInformation.perceivedSeverity	It carries one or more identifiers identifying AlarmInformation instances in AlarmList, including optionally the perceivedSeverity of the AlarmInformation instance that is going to be acknowledged. alarmInformationAndSeverityReferenceList {alarmId – Mandatory; perceivedSeverity - Optional }
ackUserId	M	AlarmInformation.ackUserId	It identities the user acknowledging the alarm.
ackSystemId	O	AlarmInformation.ackSystemId	It identifies the processing system on which the subject IRPManager runs. It may be absent implying that IRPManager does not wish this information be kept in AlarmInformation in AlarmList.

6.3.1.3 Output Parameters

Name	Qualifier	Matching Information	Comment
badAlarm Information ReferenceList	M	List of pair of `AlarmInformation.alarmId`, ENUM (UnknownAlarmId, AcknowledgmentFailed, WrongPerceivedSeverity) and additional failure reason.	If `allAlarmsAcknowledged` is true, it contains no information. If `someAlarmAcknowledged` is true, then it contains identifications of `AlarmInformation` that are (a) present in input parameter `AlarmInformationReferenceList` but are absent in the `AlarmList` = UnknownAlarmId; or (b) present in input parameter `AlarmInformationReferenceList` and are present in the AlarmList but the Acknowledgement Information (see note below table) has not changed, in contrast to `IRPManager`'s request = AcknowledgmentFailed; or (c) present in input parameter `AlarmInformationReferenceList` and are present in the AlarmList but the `perceivedSeverity` to be acknowledged has changed and/or is different within the Alarm List = WrongPerceivedSeverity (applicable only if perceivedSeverity was provided).
status	M	ENUM (OperationSucceeded, OperationFailed, OperationPartiallySucceeded)	If someAlarmAcknowledged is true, status = OperationPartiallySuceeded. If allAlarmsAcknowledged is true, status = OperationSucceeded. If operation_failed is true, status = OperationFailed.

NOTE: Acknowledgement Information is defined as the information contained in `AlarmInformation.ackTime`, `AlarmInformation.ackUserId`, `AlarmInformaton.ackSystemId`, `AlarmInformation.ackState`.

6.3.1.4 Pre-condition

`atLeastOneValidId`.

Assertion Name	Definition
atLeastOneValidId	The AlarmInformationReferenceList contains at least one identifier that identifies one AlarmInformation in AlarmList and that this identified AlarmInformation shall have its ackState indicating "unacknowledged" and, if provided, an equal perceivedSeverity.

6.3.1.5 Post-condition

`someAlarmAcknowledged OR allAlarmsAcknowledged.`

Assertion Name	Definition
someAlarmAcknowledged	At least one but not all AlarmInformation identified in input parameter AlarmInformationReferenceList has been acknowledged. Acknowledgement of an AlarmInformation means that the ackState attribute has been set to "acknowledged", that ackUserId, ackSystemId attributes of this AlarmInformation have been set to the values provided as input parameter and that the time of acknowledgeAlarms operation has been registered in ackTime attribute.
allAlarmsAcknowledged	All AlarmInformation identified in input parameter have been acknowledged. Acknowledgement of an AlarmInformation means that the ackState attribute has been set to "acknowledged", that ackUserId, ackSystemId attributes of this AlarmInformation have been set to the values provided as input parameter and that the time of acknowledgeAlarms operation has been registered in ackTime attribute.

6.3.1.6   Exceptions

Name	Definition
operation_failed	**Condition:** Pre-condition is false or post-condition is false. **Returned Information:** The output parameter status. **Exit state:** Entry state.

6.3.2   getAlarmList (M)

6.3.2.1   Definition

The `IRPManager` invokes this operation to request the `AlarmIRP` to provide either the complete list of `AlarmInformation` instances in the `AlarmList` or only a part of this list (partial alarm alignment).

The parameters `baseObjectClass` and `baseObjectInstance` are used to identify the part of the alarm list to be returned. If they are absent, then the complete alarm list shall be provided (full alarm alignment). If they identify a particular class instance, then only a) the `AlarmInformation` instances related to this class instance and b) the `AlarmInformation` instances related to the subordinate class instances of this class instance shall be provided (partial alarm alignment). An instance-a is said to be subordinate to instance-b if the DN of the latter is part of the DN of the former.

There are two modes of operation. One mode is synchronous. In this mode, the list of `AlarmInformation instances` in `AlarmList` is returned synchronously with the operation. The other mode is asynchronous. In this mode, the list of `AlarmInformation instances` is returned via notifications. In asynchronous mode of operation, the only information returned synchronously is the status of the operation. A method allowing to abort an ongoing alarm alignment process shall be available in the asynchronous mode. The mode of operation to be used is determined by means outside the scope of specification. To use asynchronous mode, the `IRPManager` must have established a subscription with the `NotificationIRP` via the `subscribe` operation specified in 3GPP TS 32.302 [5].

6.3.2.2   Input Parameters

Name	Qualifier	Information Type	Comment
alarmAckState	O	ENUM (all alarms, all active alarms, all active and acknowledged alarms, all active and unacknowledged, all Cleared and unacknowledged alarms, all unacknowledged)	It carries a constraint. The `AlarmIRP` shall apply it on Alarm Information instances in Alarm List when constructing its output parameter Alarm Information List.
baseObjectClass	O, see note 1	This parameter is either absent or carries the object class of a certain class.	See how this attribute is used to support full alarm alignment and partial alarm alignment in 6.3.2.1. See note 2.
baseObjectInstance	O, see note 1	This parameter is either absent or carries the DN of a certain class instance.	See how this attribute is used to support full alarm alignment and partial alarm alignment in 6.3.2.1. See note 2.
filter	O	N/A	It carries a filter constraint. If the `filter` is present, the `AlarmIRP` shall apply it on `AlarmInformation` instances in `AlarmList` when constructing its output parameter `AlarmInformationList`. If the `filter` is not present, all of the `AlarmInformation` instances included by the scope are selected.

NOTE 1: If the notification `notifyAlarmListRebuilt` supports indicating that only a part of the alarm list has been rebuilt then the operation `getAlarmList` shall support partial alarm alignment.

NOTE 2: The legal values of the parameters `baseObjectClass` and `baseObjectInstance` are restricted to those carried by the parameters `baseObjectClass` and `baseObjectInstance` in the recent `notifyAlarmListRebuilt` notifications. The timeline for "recent" is vendor-specific.

6.3.2.3 Output Parameters

Name	Qualifier	Matching Information	Comment
alarmInformationList	M For the Qualifier of the parameters in each list entry see the following table	List of AlarmInformation.	It carries the requested `AlarmInformation` instances. Case when synchronous mode of operation is used: (a) The `AlarmIRP` shall apply the constraints expressed in alarmAckState and filter to AlarmInformation instances when constructing this output parameter. Case when asynchronous mode of operation is used (i.e. this output parameter is conveyed via notifications): (a) If the filter parameter is present, the IRPAgent shall apply the constraint when constructing this output parameter. Furthermore, if the alarmAckState constraint is present, the IRPAgent shall apply that constraint as well. The filter constraint, if any, that is currently active in the notification channel is not used for the construction of this output parameter. (b) If the filter parameter is absent, the IRPAgent shall apply the filter constraint currently active in the notification channel when constructing this output parameter. If the alarmAckState constraint is present, the IRPAgent shall apply that constraint as well.
status	M	ENUM (OperationSucceeded, OperationFailed)	If allAlarmInformationReturned is true, status = OperationSucceeded. If operation_failed is true, status = OperationFailed.

The following table lists the set of sub-elements of the alarmInformationList attribute, and alarmInformationList forms a list of such sets.

Name	Qualifier	Matching Information	Comment
notificationType	M	"notifyNewAlarm" or "notifyChangedAlarm" or "notifyClearedAlarm"	The parameter carries • notifyNewAlarm in case the alarm has not yet changed and has not yet been cleared. • notifyChangedAlarm in case the alarm has changed but has not yet been cleared. • notifyClearedAlarm in case the alarm has been cleared but not yet acknowledged.
alarmType	M	AlarmInformation.eventType	This parameter indicates "Communications Alarm", "Processing Error Alarm", "Environmental Alarm". "Quality Of Service Alarm" or "Equipment Alarm" for non-security-related alarms. It indicates "Integrity Violation", "Operational Violation", "Physical Violation", "Security Service or Mechanism Violation" or "Time Domain Violation" for security alarms.
objectClass, objectInstance	M	MonitoredEntity.objectClass where the MonitoredEntity is identified by the relation-AlarmedObject-AlarmInformation of the new AlarmInformation. MonitoredEntity.objectInstance where the MonitoredEntity is identified by the relation-AlarmedObject-AlarmInformation of the new AlarmInformation.	
notificationId	M	This carries the semantics of notification identifier.	
eventTime	O	AlarmInformation.alarmRaisedTime or AlarmInformation.alarmChangedTime or AlarmInformation.alarmClearedTime	The parameter carries the • alarmRaisedTime in case notificationType carries notifyNewAlarm • alarmChangedTime in case notificationType carries notifyChangedAlarm • alarmClearedTime in case notificationType carries notifyClearedAlarm The availability and accuracy of time carried by the time parameters in individual entries of the list (i.e. eventTime, alarmRaisedTime, alarmClearedTime and ackTime) shall be "best effort". Reason: An EMS is not required to persistently store these times or other alarm information (as in case of synchronization information may be provided by the NE), while also some NE's do not keep these times (and a later attempt to retrieve the alarm data from the NEs will not deliver these time data).
systemDN	C	See usage of this attribute in Notification header - see [5].	Presence dependent on solution set. See usage of this attribute in Notification header - see [5].
alarmId	M	AlarmInformation.alarmId	
alarmRaisedTime	M	AlarmInformation.alarmRaisedTime	The availability and accuracy of time carried by the time parameters in individual entries of the list (i.e. eventTime, alarmRaisedTime, alarmClearedTime and ackTime) shall be "best effort". Reason: An EMS is not required to persistently store these times or other alarm information (as in case of synchronization information may be provided by the NE), while also some NE's do not keep these times (and a later attempt to retrieve the alarm data from the NEs will not deliver these time data).
alarmChangedTime	O	AlarmInformation.alarmChangedTime	not applicable if the severity of related alarm was not changed The availability and accuracy of time carried by the time parameters in individual entries of the list (i.e. eventTime, alarmRaisedTime, alarmChangedTime, alarmClearedTime and ackTime) shall be "best effort". Reason: An EMS is not required to persistently store these times or other alarm information (as in case of synchronization information may be provided by the NE), while also some NE's do not keep these times (and a later attempt to retrieve the alarm data from the NEs will not deliver these time data).

alarmClearedTime	M	AlarmInformation.alarmClearedTime	not applicable if related alarm was not cleared The availability and accuracy of time carried by the time parameters in individual entries of the list (i.e. eventTime, alarmRaisedTime, alarmClearedTime and ackTime) shall be "best effort". Reason: An EMS is not required to persistently store these times or other alarm information (as in case of synchronization information may be provided by the NE), while also some NE's do not keep these times (and a later attempt to retrieve the alarm data from the NEs will not deliver these time data).
probableCause	M	AlarmInformation.probableCause	
perceivedSeverity	M	AlarmInformation.perceivedSeverity	
rootCauseIndicator	O	AlarmInformation.rootCauseIndicator	
specificProblem	O	AlarmInformation.specificProblem	
backedUpStatus	O	AlarmInformation.backedUpStatus	not applicable if related alarm is a security alarm
trendIndication	O	AlarmInformation.trendIndication	not applicable if related alarm is a security alarm
thresholdInfo	O	AlarmInformation.thresholdInfo	not applicable if related alarm is a security alarm
stateChangeDefinition	O	AlarmInformation.stateChange	not applicable if related alarm is a security alarm
monitoredAttributes	O	AlarmInformation.monitoredAttributes	not applicable if related alarm is a security alarm
proposedRepairActions	O	AlarmInformation.proposedRepairActions	not applicable if related alarm is a security alarm
additionalText	O	AlarmInformation.additionalText	
additionalInformation	O	AlarmInformation.additionalInformation	
ackTime	M	AlarmInformation.ackTime	not applicable if related alarm was not acknowledged nor unacknowledged The availability and accuracy of time carried by the time parameters in individual entries of the list (i.e. eventTime, alarmRaisedTime, alarmClearedTime and ackTime) shall be "best effort". Reason: An EMS is not required to persistently store these times or other alarm information (as in case of synchronization information may be provided by the NE), while also some NE's do not keep these times (and a later attempt to retrieve the alarm data from the NEs will not deliver these time data).
ackUserId	M	AlarmInformation.ackUserId	not applicable if related alarm was not acknowledged nor unacknowledged
ackSystemId	O	AlarmInformation.ackSystemId	not applicable if related alarm was not acknowledged nor unacknowledged
ackState	M	AlarmInformation.ackState	not applicable if related alarm was not acknowledged nor unacknowledged
clearUserId	O	AlarmInformation.clearUserId	not applicable if related alarm was not cleared
clearSystemId	O	AlarmInformation.clearSystemId	not applicable if related alarm was not cleared
backUpObject	O	MonitoredEntity.objectInstance where the MonitoredEntity is identified by relation-BackUpObject-AlarmInformation of the new AlarmInformation.	not applicable if related alarm is a security alarm
correlatedNotifications	O	The set of CorrelatedNotification related to this AlarmInformation.	
comments	M	The set of Comment instances involved in a relationship with this AlarmInformation.	not applicable if the related alarm has no appended comments
serviceUser	M	AlarmInformation.serviceUser	not applicable if related alarm is not a security alarm
serviceProvider	M	AlarmInformation.serviceProvider	not applicable if related alarm is not a security alarm
securityAlarmDetector	M	AlarmInformation.securityAlarmDetector	not applicable if related alarm is not a security alarm

6.3.2.4 Pre-condition

`baseObjectExists`

Assertion Name	Definition
baseObjectExists	If the parameters `baseObjectClass` and `baseObjectInstance` are provided the object identified by them has to exist. If they are not provided this pre-condition is not applicable.

6.3.2.5 Post-condition

`allAlarmInformationReturned.`

Assertion Name	Definition
allAlarmInformationReturned	All AlarmInformation that satisfy the constraints expressed in input parameters filter and alarmAckState and are present in the AlarmList at the moment of this operation invocation are returned. All AlarmInformation in AlarmList remains unchanged as the result of this operation.

6.3.2.6 Exceptions

Assertion Name	Definition
operation_failed	**Condition:** At least one input parameter is invalid or the pre-condition is false or the post-condition is not true. **Returned Information:** The output parameter status. **Exit state:** Entry state.
filter_complexity_limit	**Condition:** Operation not performed because the filter parameter was too complex. **Returned Information**: The output parameter status. **Exit state:** Entry state.

6.4 Interface AlarmIRPOperation_2 (O)

6.4.1 getAlarmCount (M)

6.4.1.1 Definition

An `IRPManager` wishes to know the amount of `AlarmInformation` kept in the AlarmList. The `IRPManager` requests the counts via this operation. Possible usage is for `IRPManager` to find out the number of `AlarmInformation` in `AlarmList` before invoking `getAlarmList` operation.

6.4.1.2 Input Parameters

Name	Qualifier	Information Type	Comment
filter	O	N/A	It carries a filter constraint. The operation shall apply it when counting the AlarmInformation instances in AlarmList. Case when synchronous mode of operation is used for getAlarmList: (a) If this parameter is present, the operation shall count the AlarmInformation instances which satisfy both (a) this filter constraint and (b) the condition set by input parameter alarmAckState. (b) If this parameter is absent, the operation shall count all AlarmInformation instances that satisfy the condition set by input parameter alarmAckState. Case when asynchronous mode of operation is used for getAlarmList: (a) If this parameter is present, the operation shall count all AlarmInformation instances that satisfy this filter constraint and the condition set by input parameter alarmAckState. (b) If this parameter is absent, the operation shall count AlarmInformation instances that satisfy (a) the filter constraint currently active in the notification channel established between the IRPManager and the IRPAgent that is equipped with NotificationIRP capabilities and (b) the condition set by input parameter alarmAckState.
alarmAckState	O	ENUM (all alarms, all active alarms, all active and acknowledged alarms, all active and unacknowledged, all cleared and unacknowledged alarms, all unacknowledged)	It carries a constraint. The operation shall apply it on AlarmInformation instances in AlarmList when counting.

6.4.1.3 Output Parameters

Name	Qualifier	Matching Information	Comment
criticalCount, majorCount, minorCount, warningCount, indeterminateCount, clearedCount	M	N/A	They carry the number of AlarmInformation in AlarmList that has the following properties. Case when synchronous mode of operation is used: (a) The operation shall apply the constraints expressed in alarmAckState and filter to AlarmInformation instances when counting. Case when asynchronous mode of operation is used (i.e. this output parameter is conveyed via notifications): (a) If the filter parameter is present, the operation shall apply the constraint when counting. Furthermore, if the alarmAckState constraint is present, the operation shall apply that constraint as well. The filter constraint, if any, that is currently active in the notification channel is not used for the counting. (b) If the filter parameter is absent, the operation shall apply the filter constraint currently active in the notification channel when counting. If the alarmAckState constraint is present, the operation shall apply that constraint as well.
status	M	ENUM (OperationSucceeded, OperationFailed)	If allAlarmInformationCounted is true, status = OperationSucceeded. If operation_failed is true, status = OperationFailed.

6.4.1.4 Pre-condition

There are no pre-conditions.

6.4.1.5 Post-condition

allAlarmInformationCounted.

Assertion Name	Definition
allAlarmInformationCounted	All AlarmInformation that satisfy the constraints expressed in input parameters filter and alarmAckState and are present in the AlarmList at the moment of this operation invocation are counted and the result returned. All AlarmInformation in AlarmList remains unchanged as the result of this operation.

6.4.1.6 Exceptions

Name	Definition
operation_failed	**Condition:** the pre-condition is false or the post-condition is true. **Returned Information:** The output parameter status. **Exit state:** Entry state.
filter_complexity_limit	**Condition:** Operation not performed because the filter parameter is too complex. **Returned Information**: The output parameter status. **Exit state:** Entry state.

6.5 Interface AlarmIRPOperation_3 (O)

6.5.1 unacknowledgeAlarms (M)

6.5.1.1 Definition

IRPManager invokes this operation to remove acknowledgement information kept in one or more AlarmInformation instances.

6.5.1.2 Input Parameters

Name	Qualifier	Information Type	Comment
alarmInformationReferenceList	M	List of AlarmInformation.alarmId	It carries one or more identifiers identifying AlarmInformation in AlarmList.
ackUserId	M	AlarmInformation.ackUserId	It identities the user that invokes this operation.
ackSystemId	O	AlarmInformation.ackSystemId	It identifies the processing system on which the subject IRPManager runs.

6.5.1.3 Output Parameters

Name	Qualifier	Matching Information	Comment
badAlarmInformationReferenceList	M	List of pair of AlarmInformation.alarmId and the failure reason.	If allAlarmsUnacknowledged is true, it contains no information. If someAlarmUnacknowledged is true, then it contains identifications of AlarmInformation that are (a) present in input parameter AlarmInformationReferenceList but are absent in the AlarmList; or (b) present in input parameter AlarmInformationReferenceList and are present in the AlarmList but the Acknowledgement Information (see note below table) has not changed, in contrast to IRPManager's request.
status	M	ENUM (OperationSucceeded, OperationFailed, OperationPartiallySucceeded)	If someAlarmUnacknowledged is true, status = OperationPartiallySucceeded. If allAlarmsUnacknowledged is true, status = OperationSucceeded. If operation_failed is true, status = OperationFailed.

NOTE: Acknowledgement Information is defined as the information contained in AlarmInformation.ackTime, AlarmInformation.ackUserId, AlarmInformaton.ackSystemId and AlarmInformation.ackState.

6.5.1.4   Pre-condition

`atLeastOneValidId AND validUserId&SystemId.`

Assertion Name	Definition
atLeastOneValidId	The AlarmInformationReferenceList contains at least one identifier that identifies one AlarmInformation in AlarmList and that this identified AlarmInformation shall have its ackState indicating "acknowledged".
validUserId&SystemId	The values of ackUserId and ackSystemId attributes of the AlarmInformation must be the same as the ones provided as input parameters. The AlarmInformation is identified by the input parameter AlarmInformationReferenceList.

6.5.1.5   Post-condition

`someAlarmUnacknowledged OR allAlarmsUnacknowledged.`

Assertion Name	Definition
someAlarmUnacknowledged	At least one but not all AlarmInformation identified in input parameter alarmListReferenceList has been unacknowledged. This means that the ackState attribute has been set to "unacknowledged", that ackTime, ackUserId, ackSystemId attributes of this AlarmInformation have been set to containing no information.
allAlarmsUnacknowledged	All AlarmInformation identified in input parameter have been unacknowledged. This means that the ackState attribute has been set to "unacknowledged", that ackTime, ackUserId, ackSystemId attributes of this AlarmInformation have been set to contain no information.

6.5.1.6   Exceptions

Name	Definition
operation_failed	**Condition:** Pre-condition is false or post-condition is false. **Returned Information:** The output parameter status. **Exit state:** Entry state.

6.6   Interface AlarmIRPOperation_4 (O)

6.6.1   setComment (M)

6.6.1.1   Definition

The `IRPManager` invokes this operation to record a comment in one or more `AlarmInformation` instances in AlarmList.

6.6.1.2 Input Parameters

Name	Qualifier	Information Type	Comment
alarmInformation ReferenceList	M	List of AlarmInformation.alarmId	It carries one or more identifiers identifying AlarmInformation instances in the AlarmList.
commentUserId	M	The Comment.commentUserId where Comment is involved in relation-AlarmInformation-Comment with an AlarmInformation.	
commentSystemId	O	The Comment.commentSystemId where Comment is involved in relation-AlarmInformation-Comment with an AlarmInformation.	
commentText	M	The comment.commentText where Comment is involved in relation-AlarmInformation-Comment with an AlarmInformation.	

6.6.1.3 Output Parameter

Name	Qualifier	Matching Information	Comment
badAlarm Information ReferenceList	M	List of pair of AlarmInformation.alarmId and the failure reason.	If allUpdated is true, it contains no information. If someUpdated is true, then it contains identifications of AlarmInformation that are not present in AlarmList or that they are present, but AlarmInformation.comments has not changed, in contrast to IRPManager's request.
Status	M	ENUM(Operation succeeded, Operation failed, Operation partially failed)	If allUpdated is true, then status = OperationSucceeded. If someUpdated is true, then status = OperationPartiallyFailed. If exception operationFailed is raised, then status = OperationFailed.

6.6.1.4 Pre-condition

atLeastOneValidId.

Assertion Name	Properties
atLeastOneValidId	The AlarmInformationReferenceList contains at least one identifier that identifies one AlarmInformation in Alarm List.

6.6.1.5 Post-condition

`allUpdated OR someUpdated.`

Assertion Name	Properties
allUpdated	The AlarmInformation.comment of all alarms identified by the input parameter AlarmInformationReferenceList has been updated. The input parameter commentText, commentUserId and commentSystemId are added to the AlarmInformation.comment. The time of the operation invocation is captured in the AlarmInformation.comment as well. To make it possible to add the new comment, the IRPAgent may remove one or more old comment previously held by AlarmInformation.comments.
someUpdated	The AlarmInformation.comment attribute of at least one but not all alarms identified by the input parameter AlarmInformationReferenceList has been updated. The input parameter commentText, commentUserId and commentSystemId are added to the AlarmInformation.comment. The time of the operation invocation is captured in the AlarmInformation.comment as well. To add a new Comment, it may be necessary to remove one or more old Comment instances being held. The commentTime of the removed Comment instances shall be older than that of the remaining Comment instances.

6.6.1.6 Exceptions

Name	Properties
operation_failed	**Condition:** the pre-condition is false or the post-condition is false. **Returned Information:** The output parameter status. **Exit state:** Entry state.

6.7 Interface AlarmIRPOperation_5 (O)

6.7.1 clearAlarms (M)

6.7.1.1 Definition

The `IRPManager` invokes this operation to clear one or more `AlarmInformation` instances in AlarmList. For example, this operation can be used to support the manual clearing of the ADMC (automatic detection and manual clearing, see also 3GPP TS 32.111-1 [9]) alarms.

6.7.1.2 Input Parameter

Name	Qualifier	Information Type	Comment
alarmInformation ReferenceList	M	List of AlarmInformation.alarmId	It carries one or more identifiers identifying AlarmInformation instances in the AlarmList.
clearUserId	M	AlarmInformation.clearUserId	It identities the user clearing the alarm.
clearSystemId	O	AlarmInformation.clearSystemId	It identifies the processing system on which the subject IRPManager runs. It may be absent implying that IRPManager does not wish this information be known to the IRPAgent.

6.7.1.3 Output Parameter

Name	Qualifier	Matching Information	Comment
badAlarmInformation ReferenceList	M	List of pair of AlarmInformation.alarmId and the failure reason.	If allCleared is true, it contains no information. If someCleared is true, then it contains identifications of AlarmInformation that are not present in AlarmList or that are present in AlarmList but remain unchanged, in contrast to IRPManager's request.
status	M	ENUM(OperationSucceeded, OperationFailed, OperationPartiallySucceeded)	If allCleared is true, then status = OperationSucceeded. If someCleared is true, then status = OperationPartiallySucceeded. If exception operationFailed is raised, then status = OperationFailed.

6.7.1.4 Pre-condition

atLeastOneValidId.

Assertion Name	Properties
atLeastOneValidId	The input parameter alarmInformationReferenceList contains at least one identifier that identifies one AlarmInformation in AlarmList.

6.7.1.5 Post-condition

allCleared OR someCleared.

Assertion Name	Properties
allCleared	The AlarmInformation.perceivedSeverity of all instances identified by the input parameter alarmInformationReferenceList are set to 'cleared'. The AlarmInformation.clearUserId and AlarmInformation.clearSystemId of all instances identified are set with values carried by input parameters clearUserId and clearSystemId respectively.
someCleared	It has the same properties as allCleared except that it is applicable to one or more but not all instances identified by the input parameter alarmInformationReferenceList.

6.7.1.6 Exceptions

Name	Properties
operation_failed	**Condition:** the pre-condition is false or the post-condition is false. **Returned Information:** The output parameter status. **Exit state:** Entry state.

6.8 Notification AlarmIRPNotifications_1 (M)

The present document does not specify methods for IRPManager to detect alarm loss. The use of `alarmId` to detect alarm loss is an arrangement made between `IRPAgent` and IRPManager. The use of such arrangement is outside the scope of the present document. For example, `IRPAgent` may use integer sequence (e.g. 1, 2, 3, 4, 5, …) as `alarmId instances` for its alarms. Based on this knowledge, IRPManager can detect alarm loss. This kind of arrangement may not be possible for all SS.

The present document does not specify how `IRPAgent` can determine if IRPManager has received alarms correctly. Not all SSs provide such capability.

The present document does not specify methods for `IRPManager` and `IRPAgent` to recover alarm loss. The only mechanism recommended to deal with alarm loss is the use of `getAlarmList` operation. The present document does not specify conditions under which `IRPManager` should invoke this operation.

The filter qualifiers in tables listing input parameters of notifications only refer to applying a filter constraint to that notification. In other words: The filter qualifiers Y(es)/N(o) specify if the input parameter can be used or not when constructing the input parameter `filter` of operations `subscribe` or `changeSubscriptionFilter` defined in 3GPP TS 32.302 [5].

6.8.1 notifyNewAlarm (M)

6.8.1.1 Definition

A new `AlarmInformation` has been added in the `AlarmList`. The subscribed `IRPManager` instances are notified of this fact if the added `AlarmInformation` satisfies the current filter constraint of their subscription.

There are two tables for Input Parameters. If alarmType parameter indicates "Communications Alarm", "Processing Error Alarm", "Environmental Alarm". "Quality Of Service Alarm" or "Equipment Alarm", the first table (see clause 6.8.1.2) shall be applicable for this notifyNewAlarm. If alarmType parameter indicates "Integrity Violation", "Operational Violation", "Physical Violation", "Security Service or Mechanism Violation" or "Time Domain Violation", the second table (see clause 6.8.1.3) shall be applicable.

6.8.1.2 Input Parameters

Parameter Name	Qualifier	Matching Information	Comment
objectClass	M,Y	MonitoredEntity.objectClass	Notification header - see [5]. It shall carry the MonitoredEntity class name. The MonitoredEntity is identified by the relation-AlarmedObject-AlarmInformation of the new AlarmInformation.
objectInstance	M,Y	MonitoredEntity.objectInstance	Notification header - see [5]. It shall carry the DN of the MonitoredEntity. The MonitoredEntity is identified by the relation-AlarmedObject-AlarmInformation of the new AlarmInformation.
notificationId	M,N	--	Notification header - see [5].
eventTime	M,Y	AlarmInformation.alarmRaisedTime	Notification header - see [5].
systemDN	C,Y	--	Notification header - see [5].
notificationType	M,Y	"notifyNewAlarm".	
probableCause	M,Y	AlarmInformation.probableCause	
perceivedSeverity	M,Y	AlarmInformation.perceivedSeverity	
rootCauseIndicator	O,N	It indicates that this `AlarmInformation` is the root cause of the events captured by the notifications whose identifiers are in the related `CorrelatedNotification` instances.	"Yes", "No"
alarmType	M,Y	AlarmInformation.eventType	The notification structure defined by this table is applicable if this parameter indicates "Communications Alarm", "Processing Error Alarm", "Environmental Alarm". "Quality Of Service Alarm" or "Equipment Alarm".
specificProblem	O,N	AlarmInformation.specificProblem	
correlatedNotifications	O,N	The set of CorrelatedNotification related to this AlarmInformation.	
backedUpStatus	O,N	AlarmInformation.backedUpStatus	
backUpObject	O,N	MonitoredEntity.objectInstance	It carries the DN of the back up object. The object is identified by relation-BackUpObject-AlarmInformation of the new AlarmInformation.
trendIndication	O,N	AlarmInformation.trendIndication	
thresholdInfo	O,N	AlarmInformation.thresholdInfo	
stateChangeDefinition	O,N	AlarmInformation.stateChange	
monitoredAttributes	O,N	AlarmInformation.monitoredAttributes	
proposedRepairActions	O,N	AlarmInformaton.proposedRepairActions	
additionalText	O,N	AlarmInformation.additionalText	
additionalInformation	O,N	AlarmInformation.additionalInformation	
alarmId	M,N	AlarmInformation.alarmId	

6.8.1.3   Input Parameters for notification related to security alarm

Parameter Name	Qualifier	Matching Information	Comment
objectClass	M,Y	MonitoredEntity.objectClass	See Table 6.8.1.2.
objectInstance	M,Y	MonitoredEntity.objectInstance	See Table 6.8.1.2.
notificationId	M,N	--	See Table 6.8.1.2.
eventTime	M,Y	AlarmInformation.alarmRaisedTime	See Table 6.8.1.2.
systemDN	C,Y	--	See Table 6.8.1.2.
notificationType	M,Y	"notifyNewAlarm".	
probableCause	M,Y	AlarmInformation.probableCause	
perceivedSeverity	M,Y	AlarmInformation.perceivedSeverity	
rootCauseIndicator	O,N	It indicates that this `AlarmInformation` is the root cause of the events captured by the notifications whose identifiers are in the related `CorrelatedNotification` instances.	"Yes", "No"
alarmType	M,Y	AlarmInformation.eventType	The notification structure of this table is applicable if this parameter indicates "Integrity Violation", "Operational Violation", "Physical Violation", "Security Service or Mechanism Violation", "Time Domain Violation".
correlatedNotifications	O,N	The set of CorrelatedNotification related to this AlarmInformation.	
additionalText	O,N	AlarmInformation.additionalText	
additionalInformation	O,N	AlarmInformation.additionalInformation	
serviceUser	M,N	AlarmInformation.serviceUser	This may contain no information if the identify of the service-user (requesting the service) is not known.
serviceProvider	M,N	AlarmInformation.serviceProvider	This shall always identify the service-provider receiving a service request, from serviceUser, that provokes the security alarm.
securityAlarmDetector	M,N	AlarmInformation.securityAlarmDetector	This may contain no information if the detector of the security alarm is the serviceProvider.
alarmId	M,N	AlarmInformation.alarmId	

6.8.1.4   Triggering Event

6.8.1.4.1   From-state

`noMatchedAlarm.`

Assertion Name	Definition
noMatchedAlarm	AlarmList does not contain an AlarmInformation that has the following properties: Its matching-criteria-attributes values are identical to that of the newly generated network alarm and it is involved in relation-AlarmObject-AlarmInformation with the same MonitoredEntity as the one identified by the newly generated network alarm.

6.8.1.4.2 To-state

`newAlarmInAlarmList.`

Assertion Name	Definition
newAlarmInAlarmList	AlarmList contains an AlarmInformation holding information conveyed by the newly generated network alarm. This AlarmInformation is involved in relation-AlarmObject-AlarmInformation with the same MonitoredEntity as the one identified by the newly generated network alarm. The following attributes of the AlarmInformation shall be populated with information in the newly generated alarm. alarmId, notificationId, alarmRaisedTime, eventType, probableCause, perceivedSeverity. The following attributes of the same AlarmInformation shall be populated with information in the newly generated alarm if the information is present (in the newly generated alarm) and if the attribute is supported: specificProblem, backedUpStatus, trendIndication, thresholdInfo, stateChangeDefinition, monitoredAttributes, proposedRepairActions, additionalText, additionalInformation.

6.8.2 notifyAckStateChanged (M)

6.8.2.1 Definition

The subscribed IRPManager instances are notified regarding changes in alarm Acknowledgement State. The `AlarmInformation` carried in the notification shall satisfy the current filter constraint of the subscription.

The notification shall contain all parameters that are filterable and are present in the original (related) `notifyNewAlarm` notification.

The IRPManager and the EM can acknowledge and unacknowledge alarms as defined by 3GPP TS 32.111-1 [9]. Specifically, the AlarmIRP itself can acknowledge alarms.

The capability that IRPAgent itself acknowledges alarms is optional. The trigger, of such capability, is vendor defined. For example, it runs once a day, once every 4 hours, or always. The algorithm for determining which cleared alarm should be acknowledged is vendor specific. For example: acknowledge alarm records that have been cleared more than 24 hours or acknowledge alarm records whose highest perceived severity level has been MINOR. When acknowledged, the alarm ackState changes and the AlarmIRP shall emit the corresponding notifyAckStateChanged.

6.8.2.2   Input Parameters

Parameter Name	Qualifier	Matching Information	Comment
objectClass	M,Y	MonitoredEntity.objectClass	See Table 6.8.1.2.
objectInstance	M,Y	MonitoredEntity.objectInstance	See Table 6.8.1.2.
notificationId	M,N	--	See Table 6.8.1.2.
eventTime	M,Y	AlarmInformation.ackTime	See Table 6.8.1.2.
systemDN	C,Y	--	See Table 6.8.1.2.
notificationType	M,Y	"notifyAckStateChanged"	
probableCause	M,Y	AlarmInformation.probableCause	
perceived Severity	M,Y	AlarmInformation.perceivedSeverity	
alarmType	M,Y	AlarmInformation.eventType	
alarmId	M,N	AlarmInformation.alarmId	
ackState	M,N	AlarmInformation.ackState	
ackUserId	M,N	AlarmInformation.ackUserId	If this AlarmInformation has been acknowledged by a human operator, than this parameter contains the operator identifier. If it has been acknowledged by a System (EM or NM), than this parameter contains the identifier of the System.
ackSystemId	O,N	AlarmInformation.ackSystemId	This parameter always contains the identifier of the System (EM or NM) where the acknowledgement request was originated.

6.8.2.3   Triggering Event

6.8.2.3.1   From-state

```
ackedByIRPManager OR ackedByIRPAgent AND alarmInformationExists.
```

Assertion Name	Definition
ackedByIRPManager	Reception of a acknowledgeAlarms operation and a subsequent operation success return.
ackedByIRPAgent	Reception of a local (non-standard) acknowlegeAlarms equivalent operation and a subsequent operation success return.
alarmInformationExists	The AlarmInformation exists in AlarmList.

6.8.2.3.2   To-state

```
alarmAckStateHasChanged.
```

Assertion Name	Definition
alarmAckStateHasChanged	The AlarmInformation.ackState of the AlarmInformation identified by from-state assertion alarmInformationExists have been updated. Specifically, the following attributes of the subject AlarmInformation are updated: -- notificationId, ackTime, ackUserId, ackState, ackSystemId.

6.8.3   notifyClearedAlarm (M)

6.8.3.1   Definition

IRPAgent notifies the subscribed `IRPManager` of alarm clearing if the subject `AlarmInformation` satisfies the optional filter constraint expressed in the `subscribe` operation.

The notification shall contain all parameters that are filterable and are present in the original (related) `notifyNewAlarm` notification.

6.8.3.2   Input Parameters

Parameter Name	Qualifier	Matching Information	Comment
objectClass	M,Y	MonitoredEntity.objectClass	See Table 6.8.1.2.
objectInstance	M,Y	MonitoredEntity.objectInstance	See Table 6.8.1.2.
notificationId	M,Y	--	See Table 6.8.1.2.
eventTime	M,Y	AlarmInformation.alarmClearedTime	See Table 6.8.1.2.
systemDN	C,Y	--	See Table 6.8.1.2.
notificationType	M,Y	"notifyClearedAlarm"	
probableCause	M,Y	AlarmInformation.probableCause	
perceivedSeverity	M,Y	AlarmInformation.perceivedSeverity	Its value shall indicate Cleared.
alarmType	M,Y	AlarmInformation.eventType	
correlated Notifications	O,N	The set of CorrelatedNotification related to this AlarmInformation.	It contains references to other AlarmInformation instances whose perceivedSeverity levels are Cleared as well. In this way, perceivedSeverity level of multiple AlarmInformation instances can be Cleared by one notification.
clearUserId	O,N	AlarmInformation.clearUserId	It is present if the AlarmInformation is cleared by the IRPManager using clearAlarms.
clearSystemId	O,N	AlarmInformation.clearSystemId	It is present if clearUserId is present and if AlarmInformation.clearSystemId contains information.
alarmId	M,N	AlarmInformation.alarmId	

6.8.3.3   Triggering Event

6.8.3.3.1   From-state

`alarmMatchedAndCleared OR clearedByIRPManager.`

Assertion Name	Definition
alarmMatchedAndCleared	The matching-criteria-attributes of the newly generated network alarm have values that are identical (matched) with ones in one AlarmInformation in AlarmList and the perceivedSeverity of the matched AlarmInformation is not Cleared AND The perceivedSeverity of the newly generated network alarm is cleared.
clearedByIRPManager	Reception of a valid clearAlarms operation that identifies the subject AlarmInformation instances. This triggering event shall occur regardless of the perceivedSeverity state of the identified AlarmInformation instances.

6.8.3.3.2   To-state

`alarmInformationCleared_1 OR alarmInformationCleared_2.`

Assertion Name	Definition
alarmInformationCleared_1	Case if From-state is alarmMatchedAndCleared: The following attributes of the subject `AlarmInformation` are updated: `notificationId, perceivedSeverity` (updated to `Cleared`), `alarmClearedTime`.
alarmInformationCleared_2	Case if From-state is clearedByIRPManager: The following attributes of the subject AlarmInformation are updated: notificationId, perceivedSeverity (updated to Cleared), alarmClearedTime, alarmClearedUserId, alarmClearedSystemId.

6.8.4   notifyAlarmListRebuilt (M)

6.8.4.1   Definition

The `IRPAgent` or its related `AlarmIRP` maintains an `AlarmList`. They can lose confidence in the integrity of its `AlarmList`. Under this condition, `IRPAgent` or its related `AlarmIRP` shall invoke `notifyAlarmListRebuilt` notification after the `AlarmList` has been rebuilt.

The `AlarmIRP` can also invoke `notifyAlarmListRebuilt` notification indicating that part of the `AlarmList` has been rebuilt. In this case, the notification carries the class instance indicating that the `AlarmList` only have been rebuilt for alarms concerning this class instance and its subordinate class instances. Furthermore, this notification indicates that there is no rebuilt going on for superior class instances of this class instance.

6.8.4.2   Input Parameters

Parameter Name	Qualifier	Matching Information	Comment
objectClass	M,Y	It identifies a) the class of the instance identified by systemDN or b) the class of MonitoredEntity.	Notification header - see [5]. If it identifies the class of the instance identified in systemDN, then all `AlarmInformation` instances in the `AlarmList` may have been rebuilt. If it identifies the class of `MonitoredEntity`, then some or all `AlarmInformation` instances in the `AlarmList` may have been rebuilt. See next parameter for the identification of the set of `AlarmInformation` that have been rebuilt.
objectInstance	M,Y	It identifies a) the instance identified by systemDN or b) an instance of MonitoredEntity.	Notification header - see [5]. If it identifies the instance identified by systemDN, then all `AlarmInformation` instances in the `AlarmList` may have been rebuilt. If it identifies an instance of `MonitoredENtity`, then the `AlarmList` only have been rebuilt for `AlarmInformation` of this instance and `AlarmInformation` of its subordinate instances.
notificationId	M,N	--	See Table 6.8.1.2.
eventTime	M,Y	--	Notification header - see [5]. It carries the time when the AlarmList is rebuilt.
systemDN	C,Y	--	See Table 6.8.1.2.
notificationType	M,Y	"notifyAlarmListRebuilt".	
reason	M,N	"Agent-NE communication error", "Agent restarts", "indeterminate". Other values can be added.	It carries the reason why the IRPAgent has rebuilt the AlarmList. This may carry different reasons than that carried by the immediate previous notifyPotentialFaultyAlarmList.
alarmListAlignmentRequirement	O(note),N	ENUM (alignmentRequired, alignmentNotRequired)	It carries an enumeration of "alignmentRequired" and "alignmentNotRequired". IRPAgent uses alignmentRequired to indicate that IRPAgent current AL is not identical to the one that could have been built using (a) IRPAgent AL information at the time it emits the immediate previous notifyPotentialFaultyAlarmList() and (b) the notifications (carrying alarm information) emitted after the previously identified notification and before the subject notification. Otherwise, the IRPAgent uses alignmentNotRequired. When this parameter is absent, it implies alignmentRequired.

NOTE: If IRPAgent supports notifyPotentialFaultyAlarmList() notification, it shall support this parameter. If IRPAgent does not support notifyPotentialFaultyAlarmList() notification, it shall not support this parameter.

6.8.4.3 Triggering Event

6.8.4.3.1 From-state

`alarmListRebuilt_0 OR alarmListRebuilt_1.`

Assertion Name	Definition
alarmListRebuilt_0	IRPAgent has cold-started, initialized, re-initialized or rebooted and it has initiated procedure to rebuild its AlarmList.
alarmListRebuilt_1	IRPAgent loses confidence in part or whole of its AlarmList. IRPAgent has initiated procedure to repair its AlarmList.

6.8.4.3.2 To-state

`alarmListRebuilt_2.`

Assertion Name	Definition
alarmListRebuilt_2	IRPAgent rebuilt the whole or part of AlarmList.

6.9 Notification AlarmIRPNotification_2 (O)

6.9.1 notifyChangedAlarm (M)

6.9.1.1 Definition

The subscribed `IRPManager` instances are notified regarding changes in `AlarmInformation` in `AlarmList`. This notification is only triggered by a change in `perceivedSeverity` attribute value (except to the value "Cleared"). The `AlarmInformation` carried in the notification shall satisfy the current filter constraint of the subscription.

The notification shall contain all parameters that are filterable and are present in the original (related) `notifyNewAlarm` notification.

6.9.1.2 Input Parameters

Parameter Name	Qualifier	Matching Information	Comment
objectClass	M,Y	MonitoredEntity.objectClass	See Table 6.8.1.2.
objectInstance	M,Y	MonitoredEntity.objectInstance	See Table 6.8.1.2.
notificationId	M,N	--	See Table 6.8.1.2.
eventTime	M,Y	AlarmInformation.alarmChangedTime	See Table 6.8.1.2.
systemDN	C,Y	--	See Table 6.8.1.2.
notificationType	M,Y	"notifyChangedAlarm"	
probableCause	M,Y	AlarmInformation.probableCause	
perceivedSeverity	M,Y	AlarmInformation.perceivedSeverity	
alarmType	M,Y	AlarmInformation.eventType	
alarmId	M,N	AlarmInformation.alarmId	

6.9.1.3 Triggering Event

6.9.1.3.1 From-state

alarmMatched AND alarmNotCleared AND alarmChanged.

Assertion Name	Definition
alarmMatched	The matching-criteria-attributes of the newly generated network alarm has values that are identical (matches) with ones in one AlarmInformation in AlarmList.
alarmNotCleared	The perceivedSeverity of the newly generated network alarm is not Cleared.
alarmChanged	The perceivedSeverity of the newly generated network alarm and of the matched AlarmInformation are different.

6.9.1.3.2 To-state

`informationUpdate.`

Assertion Name	Definition
informationUpdate	• The AlarmInformation identified in alarmMatched in from-state has been updated according to the following rules: perceivedSeverity is updated; • notificationId is updated; • alarmChangedTime is updated; • ackTime, ackUserId and ackSystemId are updated to contain no information; • ackState is updated to "unacknowledged";

6.10 Notification AlarmIRPNotification_3 (O)

6.10.1 notifyComments (M)

6.10.1.1 Definition

The subscribed `IRPManager` instances are notified regarding to the addition of a `Comment` instance to an `AlarmInformation` instance in the `AlarmList`. The `AlarmInformation` carried in the notification shall satisfy the current filter constraint of the subscription.

The notification shall contain all parameters that are filterable and are present in the original (related) `notifyNewAlarm` notification.

The IRPManager and the IRPAgent can add comments to instances of `AlarmInformation` as described in 3GPP TS 32.111-1 [9].

IRPAgent shall support this notification if it supports the operation `setComment`.

6.10.1.2 Input Parameters

Parameter Name	Qualifier	Matching Information	Comment
objectClass	M,Y	MonitoredEntity.objectClass	See Table 6.8.1.2.
objectInstance	M,Y	MonitoredEntity.objectInstance	See Table 6.8.1.2.
notificationId	M,N	--	See Table 6.8.1.2.
eventTime	M,Y	Comment.commentTime	Notification header - see [5]. It carries the time when the last Comment is added.
systemDN	C,Y	--	Notification header - see [5].
notificationType	M,Y	"notifyComments"	
alarmType	M,Y	AlarmInformation.eventType	
probableCause	M,Y	AlarmInformation.probableCause	
perceived Severity	M,Y	AlarmInformation.perceivedSeverity	
comments	M,N	The set of Comment instances involved in a relationship with this AlarmInformation.	
alarmId	M,N	AlarmInformation.alarmId	

6.10.1.3 Triggering Events

6.10.1.3.1 From-state

`commentedByIRPManager OR commentedByIRPAgent AND alarmInformationExists.`

Assertion Name	Definition
commentedByIRPManager	Reception of a setComment operation and a subsequent operation success return.
commentedByIRPAgent	Reception of a local (non-standard) setComment equivalent operation and a subsequent operation success return.
alarmInformationExists	The AlarmInformation is in AlarmLists.

6.10.1.3.2   To-state

`commentInserted`.

Assertion Name	Definition
commentInserted	One Comment has been created and it is involved in a relationship with the AlarmInformation identified by from-state assertion alarmInformationExists. The following attributes of the newly created Comment instance shall be populated: commentTime, commentText, commentUserId and commentSystemId.

6.11   Notification AlarmIRPNotification_4 (O)

6.11.1   notifyPotentialFaultyAlarmList (M)

6.11.1.1   Definition

The IRPAgent or its related AlarmIRP maintains an AlarmList. They can lose confidence in the integrity of its AlarmList. Under this condition, IRPAgent or its related AlarmIRP or the related AlarmList shall invoke notifyPotentialFaultyAlarmList. They then can begin to rebuild the faulty AlarmList, if found necessary. After the successful rebuilt or the discovery that rebuilt is not necessary, they shall invoke notifyAlarmListRebuilt notification.

This notification can identify a set of AlarmInformation that is potentially faulty or unreliable. This identification is done in the following way. If the MOI of an AlarmInformation is the same or is a subordinate to the MOI carried in the notification, then the AlarmInformation may be faulty or unreliable.

This notification can identify all the AlarmInformation instances of the AlarmList that are potentially faulty or unreliable. In this case, the notification shall carry a MOI identifying the IRPAgent.

The IRPManager behaviour, on reception of this notifyPotentialFaultyAlarmList notification, is not specified. The IRPManager behaviour is considered not essential for the specification of the interface itself. However, the following are recommended actions the IRPManager should take, in case it receives this notification.

1) The IRPManager should not perform any task requiring the integrity of the AlarmInformation identified as faulty or unreliable by the subject notification.

2) The IRPManager should not invoke operations that require integrity of the AlarmList such as getAlarmList., acknolwedgeAlarms operations.

6.11.1.2 Input Parameters

Parameter Name	Qualifier	Matching Information	Comment
objectClass	M,Y	It identifies a) the class of the instance identified by systemDN or b) the class of MonitoredEntity.	Notification header - see [5]. If it identifies the class of the instance identified in systemDN, then all `AlarmInformation` instances in the `AlarmList` may not be reliable. If it identifies the class of `MonitoredEntity`, then some or all `AlarmInformation` instances in the `AlarmList` may not be reliable. See next parameter for the identification of the set of `AlarmInformation` that may not be reliable.
objectInstance	M,Y	It identifies a) the instance identified by systemDN or b) an instance of MonitoredEntity.	Notification header - see [5]. If it identifies the instance identified by systemDN, then all `AlarmInformation` instances in the `AlarmList` may not be reliable. If it identifies an instance of `MonitoredENtity`, then `AlarmInformation` of this instance and `AlarmInformation` of its subordinate instances may not be reliable.
notificationId	M,N	--	Notification header - see [5].
eventTime	M,Y	--	Notification header - see [5]. It carries the time when the objectInstance has lost confidence of its AlarmList content.
systemDN	C,Y	--	See Table 6.8.1.2.
notificationType	M,Y	"notifyPotentialFaultyAlarmList".	
reason	M,N	"Agent-NE communication error", "Agent restarts", "indeterminate". Other values can be added.	It carries the reason why the IRPAgent has to rebuild its AlarmList.

6.11.1.3 Triggering Event

6.11.1.3.1 From-state

`faultyAlarmListDetected.`

Assertion Name	Definition
faultyAlarmListDetected	IRPAgent detects faults in part or whole of its AlarmList.

6.11.1.3.2 To-state

`faultyAlarmList`

Assertion Name	Definition
faultyAlarmList	IRPAgent initiates the AlarmList rebuild process.

6.12   Notification AlarmIRPNotification_5 (O)

6.12.1   notifyCorrelatedNotificationChanged (M)

6.12.1.1   Definition

The set of SupportIOC `CorrelatedNotification` instances has been created, updated or removed. The subscribed `IRPManager` instances are notified of this fact if the changes satisfy the current filter constraint of their subscription.

6.12.1.2   Input Parameters

Parameter Name	Qualifier	Matching Information	Comment
objectClass	M,Y	MonitoredEntity.objectClass	See Table 6.8.1.2.
objectInstance	M,Y	MonitoredEntity.objectInstance	See Table 6.8.1.2.
notificationId	M,N	--	See Table 6.8.1.2.
eventTime	M,Y	--	Notification header - see [5]. It carries the time when the CorrelatedNotification is added.
systemDN	C,Y	--	Notification header - see [5].
notificationType	M,Y	"notifyCorrelatedNotificationChanged"	
correlatedNotifications	M,N	The set of CorrelatedNotification related to this AlarmInformation.	
alarmId	M,N	AlarmInformation.alarmId	
rootCauseIndicator	O,N	AlarmInformation.rootCauseIndicator	

6.12.1.3   Triggering Events

6.12.1.3.1   From-state

`newAlarmCorrelationInfoIsAvailable AND alarmInformationExists.`

Assertion Name	Definition
newAlarmCorrelationInfoIsAvailable	New alarm correlation information is available but not yet conveyed to any IRPManager.
alarmInformationExists	The AlarmInformation is in AlarmList.

6.12.1.3.2 To-state

`alarmCorrelatedInfoUpdated.`

Assertion Name	Definition
alarmCorrelatedInfoUpdated	The set of SupportIOC `CorrelatedNotification` instances has been created, updated or removed.

Annex A (normative):
Event Types

This annex lists and explains event types used by the present document.

The table below lists the event types referred to in the present document.

Notification IRP: Information Service in 3GPP TS 32.302 [5] defines a parameter called notificationType that shall be present in all notification. The present document defines a parameter called alarmType that shall be present in all notifications carrying alarm information. Examples of the notificationType are "notification of new alarm", "notification of AlarmList rebuilt", "notification of alarm cleared", etc. Examples of the alarmType are the event types defined in table below.

The present document also defines an attribute of AlarmInformation called eventType. The mapping of this eventType (internal attribute and not visible to IRPManager) to notificationType or alarmType (both visible to IRPManager) is defined in relevant sections of the present document. The choice of using "eventType" is to keep the list of attributes of AlarmList unchanged (compared to Release 99). One can replace this eventType with two attributes, called notificationType and alarmType so that mapping of these two attributes to the externally visible parameters of the same name will be straight-forward.

It is noted that the mapping of the IS notificationType and alarmType to CORBA event_name or other fields are specified in the respective Solution Set.

Table A.1: Event Types

Event Types	Explanation
Communications Alarm	An alarm of this type is associated with the procedure and/or process required conveying information from one point to another (ITU-T Recommendation X.733 [2]).
Processing Error Alarm	An alarm of this type is associated with a software or processing fault (ITU-T Recommendation X.733 [2]).
Environmental Alarm	An alarm of this type is associated with a condition related to an enclosure in which the equipment resides (ITU-T Recommendation X.733 [2]).
Quality of Service Alarm	An alarm of this type is associated with degradation in the quality of a service (ITU-T Recommendation X.733 [2]).
Equipment Alarm	An alarm of this type is associated with an equipment fault (ITU-T Recommendation X.733 [2]).
Integrity Violation	An indication that information may have been illegally modified, inserted or deleted.
Operational Violation	An indication that the provision of the requested service was not possible due to the unavailability, malfunction or incorrect invocation of the service.
Physical Violation	An indication that a physical resource has been violated in a way that suggests a security attack.
Security Service or Mechanism Violation	An indication that a security attack has been detected by a security service or mechanism.
Time Domain Violation	An indication that an event has occurred at an unexpected or prohibited time.

Annex B (normative):
Probable Causes

This annex lists probable causes and their corresponding event types.

Sources of these probable causes are ITU-T Recommendation M.3100 [11], ITU-T Recommendation X.721 [3], ITU-T Recommendation X.733 [2], and ITU-T Recommendation X.736 [15]. In addition, probable causes for 2G and 3G wireless systems are listed.

Table B.1: Probable Causes from ITU-T Recommendation M.3100 [11]

M.3100 Probable cause	Event type
Indeterminate	Unknown
Alarm Indication Signal (AIS)	Communications
Broadcast Channel Failure	Communications
Call Setup Failure	Communications
Communications Receive Failure	Communications
Communications Transmit Failure	Communications
Connection Establishment Error	Communications
Degraded Signal	Communications
Demodulation Failure	Communications
Far End Receiver Failure (FERF)	Communications
Framing Error	Communications
Invalid Message Received	Communications
Local Node Transmission Error	Communications
Loss Of Frame (LOF)	Communications
Loss Of Pointer (LOP)	Communications
Loss Of Signal (LOS)	Communications
Modulation Failure	Communications
Payload Type Mismatch	Communications
Transmission Error	Communications
Remote Alarm Interface	Communications
Remote Node Transmission Error	Communications
Routing Failure	Communications
Excessive Bit Error Rate (EBER)	Communications
Path Trace Mismatch	Communications
Unavailable	Communications
Signal Label Mismatch	Communications
Loss Of Multi Frame	Communications
Antenna Failure	Equipment
Back Plane Failure	Equipment
Battery Charging Failure	Equipment
Data Set Problem	Equipment
Disk Failure	Equipment
Equipment Identifier Duplication	Equipment
External IF Device Problem	Equipment
Frequency Hopping Failure	Equipment
IO Device Error	Equipment
Line Card Problem	Equipment
Loss Of Redundancy	Equipment
Loss Of Synchronization	Equipment
Multiplexer Problem	Equipment
NE Identifier Duplication	Equipment
Power Problem	Equipment
Power Supply Failure	Equipment
Processor Problem	Equipment
Protection Path Failure	Equipment
Protecting Resource Failure	Equipment
Protection Mechanism Failure	Equipment
Real Time Clock Failure	Equipment
Receiver Failure	Equipment
Replaceable Unit Missing	Equipment
Replaceable Unit Type Mismatch	Equipment

M.3100 Probable cause	Event type
Signal Quality Evaluation Failure	Equipment
Synchronization Source Mismatch	Equipment
Terminal Problem	Equipment
Timing Problem	Equipment
Transceiver Failure	Equipment
Transmitter Failure	Equipment
Trunk Card Problem	Equipment
Replaceable Unit Problem	Equipment
Air Compressor Failure	Environmental
Air Conditioning Failure	Environmental
Air Dryer Failure	Environmental
Battery Discharging	Environmental
Battery Failure	Environmental
Commercial Power Failure	Environmental
Cooling Fan Failure	Environmental
Cooling System Failure	Environmental
Engine Failure	Environmental
Fire Detector Failure	Environmental
Fuse Failure	Environmental
Generator Failure	Environmental
Low Battery Threshold	Environmental
Pump Failure	Environmental
Rectifier Failure	Environmental
Rectifier High Voltage	Environmental
Rectifier Low F Voltage	Environmental
Ventilation System Failure	Environmental
Enclosure Door Open	Environmental
Explosive Gas	Environmental
External Equipment Failure	Environmental
External Point Failure	Environmental
Fire	Environmental
Flood	Environmental
High Humidity	Environmental
High Temperature	Environmental
High Wind	Environmental
Ice Build Up	Environmental
Intrusion Detection	Environmental
Low Fuel	Environmental
Low Humidity	Environmental
Low Cable Pressure	Environmental
Low Temperature	Environmental
Low Water	Environmental
Smoke	Environmental
Toxic Gas	Environmental
Application Subsystem Failure	Processing Error
Configuration Or Customisation Error	Processing Error
Database Inconsistency	Processing Error
File Error	Processing Error
Storage Capacity Problem	Processing Error
Memory Mismatch	Processing Error
Corrupt Data	Processing Error
Loss of Real Time	Processing Error
Out Of CPU Cycles	Processing Error
Out Of Memory	Processing Error
Reinitialized	Processing Error
Software Environment Problem	Processing Error
Software Error	Processing Error
Software Download Failure	Processing Error
Timeout Expired	Processing Error
Underlaying Resources Unavailable	Processing Error
Version Mismatch	Processing Error
Bandwidth Reduced	Quality of service
Congestion	Quality of service
Excessive Error Rate	Quality of service

M.3100 Probable cause	Event type
Excessive Response Time	Quality of service
Excessive Retransmission Rate	Quality of service
Reduced Logging Capability	Quality of service
System Resources Overload	Quality of service

Table B.2: Probable Causes from ITU-T Recommendation X.721 [3], X.733 [2], X.736 [15]

X.721/X.733/X.736 Probable Cause	Event type
Adapter Error	Equipment
Application Subsystem Failure	Processing error
Authentication Failure	Security Service or Mechanism Violation
Bandwidth Reduction	Quality of service
Breach of Confidentiality	Security Service or Mechanism Violation
Cable Tamper	Physical Violation
Call Establishment Error	Communications
Communication Protocol Error	Communications
Communication Subsystem Failure	Communications
Configuration or Customizing Error	Processing error
Congestion	Quality of service
Corrupt Data	Processing error
CPU Cycles Limit Exceeded	Processing error
Data Set or Modem Error	Equipment
Degraded Signal	Communications
Delayed Information	Time Domain Violation
Denial of Service	Operational Violation
DTE-DCE Interface Error	Communications
Duplicate Information	Integrity Violation
Enclosure Door Open	Environmental
Equipment Malfunction	Equipment
Excessive Vibration	Environmental
File Error	Processing error
Fire Detected	Environmental
Flood Detected	Environmental
Framing Error	Communications
Heating or Ventilation or Cooling System Problem	Environmental
Humidity Unacceptable	Environmental
Information Missing	Integrity Violation
Information Modification detected	Integrity Violation
Information out of Sequence	Integrity Violation
Input/Output Device Error	Equipment
Input Device Error	Equipment
Intrusion Detection	Physical Violation
Key Expired	Time Domain Violation
LAN Error	Communications
Leak Detection	Environmental
Local Node Transmission Error	Communications
Loss of Frame	Communications
Loss of Signal	Communications
Material Supply Exhausted	Environmental
Multiplexer Problem	Equipment
Non-Repudiation Failure	Security Service or Mechanism Violation
Out of Hours Activity	Time Domain Violation
Out of Memory	Processing error
Out of Service	Operational Violation
Output Device Error	Equipment
Performance Degraded	Quality of service
Power Problem	Equipment
Pressure Unacceptable	Environmental
Procedural Error	Operational Violation
Processor Problem	Equipment
Pump Failure	Environmental
Queue Size Exceeded	Quality of service
Receive Failure	Equipment

X.721/X.733/X.736 Probable Cause	Event type
Receiver Failure	Equipment
Remote Node Transmission Error	Communications
Resource at or Nearing Capacity	Quality of service
Response Time Excessive	Quality of service
Re-transmission Rate Excessive	Quality of service
Software Error	Processing error
Software Program Abnormally Terminated	Processing error
Software Program Error	Processing error
Storage Capacity Problem	Processing error
Temperature Unacceptable	Environmental
Threshold Crossed	Quality of service
Timing Problem	Equipment
Toxic Leak Detected	Environmental
Transmit Failure	Equipment
Transmitter Failure	Equipment
Unauthorised Access Attempt	Security Service or Mechanism Violation
Underlying Resource Unavailable	Processing error
Unexpected Information	Integrity Violation
Unspecified Reason	Operational Violation
Unspecified Reason	Physical Violation
Unspecified Reason	Security Service or Mechanism Violation
Version Mismatch	Processing error

Table B.3: Probable Causes for 2G & 3G Wireless Systems

2G & 3G Wireless Systems	Event Type
A-bis to BTS interface failure	Equipment
A-bis to TRX interface failure	Equipment
Antenna problem	Equipment
Battery breakdown	Equipment
Battery charging fault	Equipment
Clock synchronization problem	Equipment
Combiner problem	Equipment
Disk problem	Equipment
Equipment failure	Equipment
Excessive receiver temperature	Equipment
Excessive transmitter output power	Equipment
Excessive transmitter temperature	Equipment
Frequency hopping degraded	Equipment
Frequency hopping failure	Equipment
Frequency redefinition failed	Equipment
Line interface failure	Equipment
Link failure	Equipment
Loss of synchronization	Equipment
Lost redundancy	Equipment
Mains breakdown with battery back-up	Equipment
Mains breakdown without battery back-up	Equipment
Power supply failure	Equipment
Receiver antenna fault	Equipment
Receiver Failure	Equipment
Receiver multicoupler failure	Equipment
Reduced transmitter output power	Equipment
Signal quality evaluation fault	Equipment
Timeslot hardware failure	Equipment
Transceiver problem	Equipment
Transcoder problem	Equipment
Transcoder or rate adapter problem	Equipment
Transmitter antenna failure	Equipment
Transmitter antenna not adjusted	Equipment
Transmitter failure	Equipment
Transmitter low voltage or current	Equipment
Transmitter off frequency	Equipment
Database inconsistency	Processing error

2G & 3G Wireless Systems	Event Type
File system call unsuccessful	Processing error
Input parameter out of range	Processing error
Invalid parameter	Processing error
Invalid pointer	Processing error
Message not expected	Processing error
Message not initialized	Processing error
Message out of sequence	Processing error
System call unsuccessful	Processing error
Timeout expired	Processing error
Variable out of range	Processing error
Watch dog timer expired	Processing error
Cooling system failure	Environmental
External equipment failure	Environmental
External power supply failure	Environmental
External transmission device failure	Environmental
Fan failure	Environmental
High humidity	Environmental
High temperature	Environmental
Intrusion detected	Environmental
Low humidity	Environmental
Low temperature	Environmental
Smoke detected	Environmental
Excessive Error Rate	Quality of service
Reduced alarm reporting	Quality of service
Reduced event reporting	Quality of service
Reduced logging capability	Quality of service
System resources overload	Quality of service
Broadcast channel failure	Communications
Connection establishment error	Communications
Invalid message received	Communications
Invalid MSU received	Communications
LAPD link protocol failure	Communications
Local alarm indication	Communications
Remote alarm indication	Communications
Routing failure	Communications
SS7 protocol failure	Communications
Transmission error	Communications

Table B.4 identifies probable causes that are defined by more than one standard. This is for information only.

Table B.4: Duplicated Probable Causes

Duplicated Probable Cause	2G & 3G	X.721 X.733	X.736	M.3100	Event Type
Broadcast Channel Failure	X			X	Communications
Call Establishment Error (X.721/X.733) Call Setup Failure (M.3100)		X		X	Communications
Connection Establishment Error	X			X	Communications
Degraded Signal		X		X	Communications
Framing Error		X		X	Communications
Invalid Message Received	X			X	Communications
Local Node Transmission Error		X		X	Communications
Loss of Frame		X		X	Communications
Loss of Signal		X		X	Communications
Remote Node Transmission Error		X		X	Communications
Routing Failure	X			X	Communications
Antenna Failure (M.3100) Antenna Problem (2G & 3G)	X			X	Equipment
Battery Charging Failure (M.3100) Battery Charging Fault (2G & 3G)	X			X	Equipment
Disk Failure (M.3100) Disk Problem (2G & 3G)	X			X	Equipment
Equipment Failure (2G & 3G) Equipment Malfunction (X.721/X.733)	X	X			Equipment

Duplicated Probable Cause	2G & 3G	X.721 X.733	X.736	M.3100	Event Type
Frequency Hopping Failure	X			X	Equipment
IO Device Error (M.3100) Input/Output Device Error (X.721/X.733)		X		X	Equipment
Loss Of Redundancy (M.3100) Lost Redundancy (2G & 3G)	X			X	Equipment
Loss Of Synchronization	X			X	Equipment
Multiplexer Problem		X		X	Equipment
Power Problem		X		X	Equipment
Power Supply Failure	X			X	Equipment
Processor Problem		X		X	Equipment
Receiver Failure	X	X		X	Equipment
Signal Quality Evaluation Failure (M.3100) Signal Quality Evaluation Fault (2G & 3G)	X			X	Equipment
Timing Problem		X		X	Equipment
Transceiver Failure (M.3100) Transceiver Problem (2G & 3G)	X			X	Equipment
Transmitter Failure	X	X		X	Equipment
Cooling System Failure	X			X	Environmental
External Equipment Failure	X			X	Environmental
Enclosure Door Open		X		X	Environmental
Fan Failure (2G & 3G) Cooling Fan Failure (M.3100)	X			X	Environmental
Fire Detected (X.721/X.733) Fire (M.3100)		X		X	Environmental
Flood Detected (X.721/X.733) Flood (M.3100)		X		X	Environmental
High Humidity	X			X	Environmental
High Temperature	X			X	Environmental
Intrusion Detected (2G & 3G) Intrusion Detection (X.736/M.3100)	X		X	X	Environmental (2G & 3G); Physical Violation (X.736/M.3100)
Low Humidity	X			X	Environmental
Low Temperature	X			X	Environmental
Pump Failure		X		X	Environmental
Smoke Detected (2G & 3G) Smoke (M.3100)	X			X	Environmental
Application Subsystem Failure		X		X	Processing Error
Bandwidth Reduced Bandwidth Reduction (X.721/X.733)		X		X	Quality of Service
Configuration or Customization Error (M.3100) Configuration or Customizing Error (X.721/X.733)		X		X	Processing Error
Database Inconsistency	X			X	Processing Error
File Error		X		X	Processing Error
Storage Capacity Problem		X		X	Processing Error
Excessive Bit Error Rate (M.3100) Excessive Error Rate (2G & 3G) Excessive Error Rate	X			X	Communications (M.3100) Quality of Service (GSM 12.11/M.3100)
Corrupt Data		X		X	Processing Error
Out Of Memory		X		X	Processing Error
Software Error		X		X	Processing Error
Timeout Expired	X			X	Processing Error
Underlaying Resource Unavailable (M.3100) Underlying Resource Unavailable (X.721/X.733)		X		X	Processing Error
Version Mismatch		X		X	Processing Error
Congestion		X		X	Quality of Service
Reduced Logging Capability	X			X	Quality of Service
System Resources Overload	X			X	Quality of Service
Excessive Response Time (M.3100) Response Time Excessive (X.721/X.733)		X		X	Quality of Service
Excessive Retransmission Rate (M.3100) Re-Transmission Rate Excessive (X.721/X,733)		X		X	Quality of Service
Transmission Error	X			X	Communications

Annex C (informative):
Examples of using `notifyChangedAlarm`

This annex describes a number of valid and invalid interactions governing the case when IRPAgent is reporting a specific fault of a particular network resource whose alarm severity level changes from, e.g. "Critical" to "Minor" and then to "Cleared".

In the following examples:

> `ni`   is `notificationId`,
>
> `moc` is `managedObjectClass`,
>
> `moi` is `managedObjectInstance`,
>
> `et`   is `eventType`,
>
> `pc`   is `probableCause`,
>
> `sp`   is `specificProblem`,
>
> `ps`   is `perceivedSeverity` and
>
> `ai`   is `alarmId`.

EXAMPLE 1:   Valid sequence of a hypothetical case:

(1) `NotifyNewAlarm`

> `(ni=1, ai=X, moc=A, moi=B, et=C, pc=D, sp=E, ps=Critical)`

(2) `NotifyChangedAlarm`

> `(ni=2, ai=X, moc=A, moi=B, et=C, pc=D, sp=E, ps=Minor)`

(3) `NotifyClearedAlarm`

> `(ni=3, ai=X, moc=A, moi=B, et=C, pc=D, sp=E, ps=Cleared)`

EXAMPLE 2:   Valid sequence of a hypothetical case (assuming that the alarm with "ai=X" is acknowledged after either (1) or (2), but before (3)):

(1)   `NotifyNewAlarm`

> `(ni=1, ai=X, moc=A, moi=B, et=C, pc=D, sp=E, ps=Critical)`

(2)   `NotifyClearedAlarm`

> `(ni=2, ai=X, moc=A, moi=B, et=C, pc=D, sp=E, ps=Cleared)`

(3)   `NotifyNewAlarm`

> `(ni=3, ai=Y, moc=A, moi=B, et=C, pc=D, sp=E, ps=Minor)`

(4)   `NotifyClearedAlarm`

> `(ni=4, ai=Y, moc=A, moi=B, et=C, pc=D, sp=E, ps=Cleared)`

EXAMPLE 3:   Invalid sequence of a hypothetical case:

(1) `NotifyNewAlarm`

> `(ni=1, ai=X, moc=A, moi=B, et=C, pc=D, sp=E, ps=Critical)`

```
(2) NotifyChangedAlarm

    (ni=2, ai=Y, moc=A, moi=B, et=C, pc=D, sp=E, ps=Minor)

(3) NotifyClearedAlarm

    (ni=3, ai=Y, moc=A, moi=B, et=C, pc=D, sp=E, ps=Cleared)
```

Interaction (2) is illegal since it uses a different `ai` for the same alarm. It should use `ai=X` as in interaction (1).

EXAMPLE 4:   Invalid sequence of a hypothetical case:

```
(1)  NotifyNewAlarm

    (ni=1, ai=X, moc=A, moi=B, et=C, pc=D, sp=E, ps=Critical)

(2)  NotifyNewAlarm

    (ni=2, ai=X, moc=A, moi=B, et=C, pc=D, sp=E, ps=Minor)
```

Interaction (2) is illegal since it invokes notifyNewAlarm using same `ai` value. It should use `notifyChangedAlarm` with the same `ai` value.

Annex D (informative):
Examples of using `correlatedNotification`

This annex describes a number of examples of when the IRPAgent is indicating that several alarms are correlated.

EXAMPLE 1:   Alarms X and Y are correlated, but the root cause is unknown.

Information of `AlarmInformation` X has been captured in a notification whose identifier is "x".

`AlarmInformation` X holds a relation to a `CorrelatedNotification` instance which has the following attribute values

source="ABC"

notificationIdSet carries the identifier "y"

X.rootCauseIndicator="No"

Information of `AlarmInformation` Y has been captured in a notification whose identifier is "y".

Optionally, `AlarmInformation` Y may hold a relation to a `correlatedNotification` instance which has the following attribute values

source="DEF"

notificationIdSet carries the identifier "x"

Y.rootCauseIndicator="No"

EXAMPLE 2:   Alarms X and Y are correlated, where Alarm X is the root cause of Alarm Y.

Information of `AlarmInformation` X has been captured in a notification whose identifier is "x".

`AlarmInformation` X holds a relation to a `correlatedNotification` instance which has the following attribute values

source="ABC"

notificationIdSet carries the identifier "y"

X.rootCauseIndicator="Yes"

Information of `AlarmInformation` Y has been captured in a notification whose identifier is "y".

Optionally, `AlarmInformation` Y may hold a relation to a `correlatedNotification` instance which has the following attribute values

source="DEF"

notificationIdSet carries the identifier "x"

Y.rootCauseIndicator="No"

Annex E (informative):
`AcknowledgeAlarms` operation scenario

The `acknowledgeAlarms` operation may optionally include `perceivedSeverity` as input parameter.

The reason for using `perceivedSeverity` in the `acknowledgeAlarms` operation is to avoid an undesirable consequence. An example sequence of events is:

1. `IRPAgent AlarmList` has `alarmId=6` with `perceivedSeverity=minor`

2. `IRPManager` issues `getAlarmList`

3. `IRPAgent` updates `alarmId=6` with `perceivedSeverity=critical`

4. In case `IRPAgent` have not issued the `notifyChangedAlarm` in time or in the case `IRPManager` ignores the `notifyChangedAlarm` received, for examples…

5. `IRPManager` issues `acknowledgeAlarms` of `alarmId=6` with `perceivedSeverity=minor`

6. `IRPAgent` rejects acknowledgement, with reason `WrongPerceivedSeverity`

If the optional `perceivedSeverity` input parameter was not used in step 5, in step 6 the `IRPAgent` would have accepted the acknowledgement, with the undesirable consequences:

- `IRPManager` wrongly concludes that it had acknowledged alarm=6 with `perceivedSeverity=minor`.

- `IRPAgent` wrongly concludes that `alarmId=6` with `perceivedSeverity=critical` had been acknowledged.

- Other `IRPManagers` will see `alarmId=6` with `perceivedSeverity=critical` being acknowledged (and possibly taken care of) by an `IRPManager`.

Annex F (informative): Change history

Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
Sep 2006	SA_33	SP-060527	0057	--	Add missing Notification Table in Alarm IRP IS	6.8.0	6.9.0
Dec 2006	SA_34	SP-060722	0058	--	Add filter complexity limitation parameter	6.9.0	7.0.0
Mar 2007	SA_35	SP-070046	0059	--	Correct the references of IRPAgent and IRPManager	7.0.0	7.1.0
Mar 2007	--	--	--	--	Deleted reference to CMIP SS, discontinued from R7 onwards	7.0.0	7.1.0
Dec 2008	SA_42	SP-080846	0060	--	Spelling and naming corrections	7.1.0	8.0.0
Mar 2009	SA_43	SP-090207	0061	--	Include reference to SOAP Solution Set specification	8.0.0	8.1.0
Dec 2009	SA_46	--	--	--	Upgrade to Release 9	8.1.0	9.0.0
Mar 2010	SA_47	SP-100035	0062	--	Correct the description of the alarm list rebuilt handling capabilities. Align spec to follow recommendations of latest Repertoire and Templates.	9.0.0	9.1.0
Dec 2010	SA_50	SP-100833	0063	2	Add alarmChangedTime to the output parameters of getAlarmList operation.	9.1.0	10.0.0
Mar 2011	SA_51	SP-110095	0064	--	Correct the qualifier of clearUserId in AlarmInformation	10.0.0	10.1.0
May 2011	SA_52	SP-110285	0065	1	Improvements to description of examples	10.1.0	10.2.0
May 2011	SA_52	SP-110289	0066	1	Add indication for root cause of alarm	10.1.0	10.2.0
May 2011	SA_52	SP-110289	0067	1	Add notification for change of alarm correlation data	10.1.0	10.2.0
Sep 2011	SA_53	SP-110534	0068	--	Clarify usage of correlated notification	10.2.0	10.3.0
Sep 2011	SA_53	SP-110534	0069	--	Add absent rootCauseIndicators	10.2.0	10.3.0
Dec 2011	SA_54	SP-110707	0070	1	Add acknowledgeAlarms operation scenario	10.3.0	11.0.0
Dec 2012	SA_58	SP-120783	0072	-	CR 32.111-2 R11 Align usage of SupportIOC with repertoire and TS 32.152	11.0.0	11.1.0