International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

M.3703

(06/2010)

SERIES M: TELECOMMUNICATION MANAGEMENT, INCLUDING TMN AND NETWORK MAINTENANCE

Integrated services digital networks

Common management services – Alarm management – Protocol neutral requirements and analysis

Recommendation ITU-T M.3703

ITU-T M-SERIES RECOMMENDATIONS

TELECOMMUNICATION MANAGEMENT, INCLUDING TMN AND NETWORK MAINTENANCE

Introduction and general principles of maintenance and maintenance organization	M.10–M.299
International transmission systems	M.300–M.559
International telephone circuits	M.560–M.759
Common channel signalling systems	M.760–M.799
International telegraph systems and phototelegraph transmission	M.800–M.899
International leased group and supergroup links	M.900–M.999
International leased circuits	M.1000–M.1099
Mobile telecommunication systems and services	M.1100–M.1199
International public telephone network	M.1200–M.1299
International data transmission systems	M.1300–M.1399
Designations and information exchange	M.1400–M.1999
International transport network	M.2000–M.2999
Telecommunications management network	M.3000–M.3599
Integrated services digital networks	**M.3600–M.3999**
Common channel signalling systems	M.4000–M.4999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T M.3703

Common management services – Alarm management – Protocol neutral requirements and analysis

Summary

Recommendation ITU-T M.3703 provides the requirements and analysis for one of the common management services – alarm management. The functional requirements for the alarm management interface include the management functions for alarm forwarding and filtering, clearing of alarms, storage and retrieval of alarms in/from the agent, configuration of alarms, alarm acknowledgement and alarm notification failure. In the analysis part, the detailed information model supporting the above functions across the management interface is provided.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T M.3703	2010-06-29	2

Keywords

Agent, alarm management, analysis, fault management, interface, manager, requirements.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

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

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Introduction

A network is composed of a multitude of network elements (NE) of various types and, typically, different vendors, which interoperate in a coordinated 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 respective 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 quality of service (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. Degradation of service may be detected by monitoring error rates. Threshold mechanisms on counters and gauges are a method of detecting such trends and providing a warning to managers when the rate becomes high.

Alarms are a specific type of notification concerning detected faults or abnormal conditions. Managed object definers are encouraged to include in alarms information that will help understand the cause of the potentially abnormal situation, and other information related to side effects. An example of such diagnostic information is the current and past values of the configuration management state of the object.

A single incident may cause the generation of several notifications; it is important to be able to specify in a notification some correlation with other notifications. However, the mechanism, if any, for determining the relationship between notifications resulting from a single incident is for further study.

The functional areas specified in this Recommendation cover:

- notification of alarms (including alarm cease) and operational state changes;
- retrieval of current alarms;
- alarm filtering;
- management of alarm severity levels;
- retention of alarm and operational state data in the NEs and the operations system (OS).

Any (re)configuration activities exerted from the element manager (EM) as a consequence of faults is out of the scope of this Recommendation.

This Recommendation defines the requirements for fault management.

Recommendation ITU-T M.3703

Common management services – Alarm management – Protocol neutral requirements and analysis

1 Scope

This Recommendation defines an interface through which an agent (typically a network element or a network element manager) can communicate alarm information for its managed objects to one or several managers (typically network management systems).

This Recommendation 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 on the interface. It does not define the syntax or encoding of the operations, notifications and their parameters.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T M.3020] Recommendation ITU-T M.3020 (2009), *Management interface specification methodology.*

[ITU-T M.3100] Recommendation ITU-T M.3100 (1995), *Generic network information model.*

[ITU-T M.3160] Recommendation ITU-T M.3160 (2008), *Generic, protocol-neutral management information model.*

[ITU-T M.3702] Recommendation ITU-T M.3702 (2010), *Common management services – Notification management – Protocol neutral requirement and analysis.*

[ITU-T X.701] Recommendation ITU-T X.701 (1997) | ISO/IEC 10040:1998, *Information technology – Open Systems Interconnection – Systems management overview.*

[ITU-T X.721] Recommendation ITU-T X.721 (1992) | ISO/IEC 10165-2:1992, *Information technology – Open Systems Interconnection – Structure of management information: Definition of management information.*

[ITU-T X.733] Recommendation ITU-T X.733 (1992) | ISO/IEC 10164-4:1992, *Information technology – Open Systems Interconnection – Systems Management: Alarm reporting function.*

[ITU-T X.736] Recommendation ITU-T X.736 (1992) | ISO/IEC 10164-7:1992, *Information technology – Open Systems Interconnection – Systems Management: Security alarm reporting function.*

[ITU-T X.790] Recommendation ITU-T X.790 (1995), *Trouble management function for ITU-T applications.*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 alarm [ITU-T X.733].

3.1.2 agent [ITU-T M.3020].

3.1.3 event [ITU-T X.790].

3.1.4 error [ITU-T X.733].

3.1.5 fault [ITU-T X.733].

3.1.6 manager [ITU-T M.3020].

3.1.7 notification [ITU-T X.701].

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 active alarm: An alarm that has not been cleared and which is active until the fault that caused the alarm is corrected and a "clear alarm" is generated.

3.2.2 ADAC faults: Faults that are automatically detected and automatically cleared by the system when they occur and when they are repaired.

3.2.3 ADMC faults: Faults that are automatically detected by the system when they occur and manually cleared by the operator when they are repaired.

3.2.4 alarm notification: Notification used to inform the recipient about the occurrence of an alarm.

3.2.5 clear alarm: Notification used to inform the recipient about the cessation of an alarm and thus the underlying fault condition.

4 Abbreviations

This Recommendation uses the following abbreviations:

ADAC Automatically Detected and Automatically Cleared

ADMC Automatically Detected and Manually Cleared

ASAP Alarm Severity Assignment Profile

CO Conditional-Optional

DN Domain Name

EM Element Manager

FM Fault Management

FS Function Set

IOC Information Object Class

M Mandatory

MO Managed Object

MOC Managed Object Class

MOI Managed Object Instance

NE Network Element

NM Network Manager

O Optional

OS Operations System

QoS Quality of Service

5 Conventions

This Recommendation uses the conventions defined in [ITU-T M.3020] for requirements capture and analysis.

6 Requirements

6.1 Concepts and background

Any evaluation of the NEs' and the overall network health status requires the detection of faults in the network and, consequently, the notification of alarms to the OS (EM and/or NM). Depending on the nature of the fault, it may be combined with a change of the operational state of the logical and/or physical resource(s) affected by the fault. Detection and notification of these state changes is as essential as it is for the alarms. A list of active alarms in the network and operational state information as well as alarm/state history data may be required by the system operator for further analysis. Additionally, test procedures may be used in order to obtain more detailed information if necessary, or to verify an alarm, a state or the proper operation of NEs and their logical and physical resources.

This service uses the following other services and thus implicitly imports all the requirements defined therein:

- Notification, defined in [ITU-T M.3702].

6.2 Business-level requirements

Faults that may occur in the network can be grouped into one of the following categories:

– Hardware failures, i.e., the malfunction of a physical resource within a NE.

– Software problems, e.g., software bugs, database inconsistencies.

– Functional faults, i.e., a failure of a functional resource in a NE and no hardware component can be found responsible for the problem.

– Loss of some or all of the NE's specified capability due to overload situations.

– Communication failures between two NEs, or between NE and OS, or between two OSs.

6.2.1 Requirements

6.2.1.1 Alarm forwarding and filtering

REQ-FM-FUN-01 For each detected fault, the agent shall generate appropriate alarms (notifications of the fault), regardless of whether it is an ADAC or an ADMC fault. Each alarm should be uniquely identified. For each alarm, the agent shall supply the following information:

- the managed entity;
- the device/resource/file/functionality/smallest replaceable unit as follows:
  - for hardware faults, the smallest replaceable unit that is faulty;

  - for software faults, the affected software component, e.g., corrupted file(s) or databases or software code;
  - for functional faults, the affected functionality;
  - for faults caused by overload, information on the reason for the overload;
  - for all the above faults, wherever applicable, an indication of the physical and logical resources that are affected by the fault and a description of the loss of capability of the affected resource;
- the type of the fault (communication, environmental, equipment, processing error, QoS, security types, etc.);
- the severity of the fault (indeterminate, warning, minor, major, critical);
- the probable cause of the fault;
- the specific problem;
- the time at which the fault was detected;
- the nature of the fault, e.g., ADAC or ADMC; any other information that helps understanding the cause and the location of the abnormal situation (system/implementation specific).

REQ-FM-FUN-02 The manager shall be able to allow or suppress alarm reporting by setting the filtering on any combination of attributes. The following criteria shall minimally be supported for alarm notification filtering:

- the managed entity that generated the alarm, i.e., all alarm messages for that managed entity shall be suppressed;
- the device/resource/function to which the alarm relates;
- the severity of the alarm;
- the time at which the alarm was detected, i.e., the alarm time.

6.2.1.2 Clearing of alarms

REQ-FM-FUN-03 Each time an alarm is cleared, the agent shall generate an appropriate clear alarm event. A clear alarm is defined as an alarm. A clear alarm is identified as such through the use of perceived severity equal to CLEARED.

REQ-FM-FUN-04 The manager may explicitly request the clearing of one or more alarms. Once the alarm(s) has/have been cleared, the agent should reissue those alarms (as new alarms) in case the fault situation still persists.

6.2.1.3 Storage and retrieval of alarms in/from the agent

REQ-FM-FUN-05 The manager shall be able to retrieve alarm information optionally using filters (active and/or historic).

6.2.1.4 Configuration of ASAP

REQ-FM-FUN-06 It may be possible for the manager to create an ASAP on an agent to define the relationship between severity level and problem.

REQ-FM-FUN-07 It may be possible for the manager to modify the ASAP.

REQ-FM-FUN-08 It may be possible for the manager to delete the ASAP on an agent.

REQ-FM-FUN-09 It may be possible for the manager to request an agent to set or change the association between an ASAP and one or more specified managed entities.

REQ-FM-FUN-10 It may be possible for the manager to request an agent to remove the association between an ASAP and one or more of its associated managed entities.

REQ-FM-FUN-11 The manager may query the attribute information of an ASAP, which includes the ID of the ASAP, the list of the problem and the corresponding severity, and the list of managed entities that have been associated with this ASAP.

6.2.1.5 Alarm acknowledgement and management

REQ-FM-FUN-12 An agent may support alarm acknowledgement and unacknowledgement. Acknowledgement data shall include the current alarm state (active|cleared), the time of alarm acknowledgement and, optionally, the system (EM|NM) or the operator in charge of acknowledgement (the parameter operator name or, in case of auto-acknowledgement, a generic system name).

REQ-FM-FUN-13 An alarm acknowledgement means that an acknowledgement performed by the agent is notified to the manager and vice versa, thus the acknowledgement-related status of this alarm is the same across the whole management hierarchy.

REQ-FM-FUN-14 The agent may provide the ability to add a comment to an alarm. An agent may also have the capability to record more than one comment for each alarm. To make the same alarm look the same to all managers subscribing to the alarm, it will be possible to distribute the recorded comments.

REQ-FM-FUN-15 Acknowledgement state shall be a filterable criteria for alarms if acknowledgement is supported by the agent.

REQ-FM-FUN-16 Acknowledgement notifications shall be filtered with the same criteria applied to alarms.

6.2.1.6 Alarm notification failure

REQ-FM-FUN-17 The manager should be able to request an agent to synchronize alarm information following a failure in communication between the agent and the manager.

6.2.2 Actor roles

The capabilities described in this Recommendation are available and relevant to all agents and managers.

6.2.3 Telecommunication resources

The alarm management functionality is applicable to all types of telecommunication resources.

6.2.4 High-level use-case diagrams

The first overview use-case diagram in Figure 6-1 shows the overall interaction of the alarm interface.

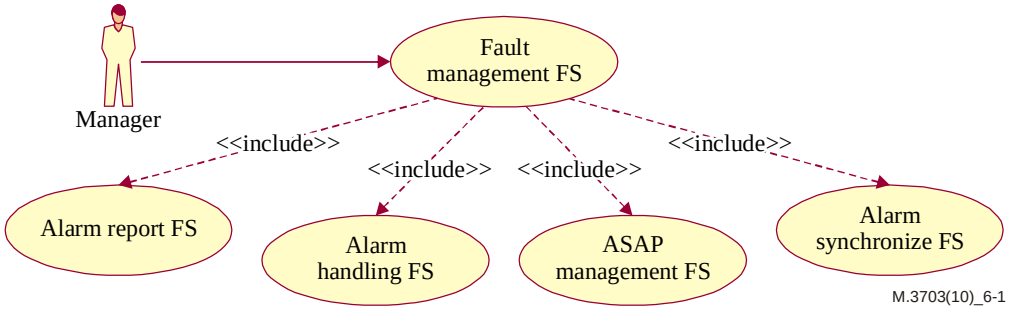

Figure 6-1 – Fault management function set

Figure 6-2 shows the functions involved in the alarm report function set.

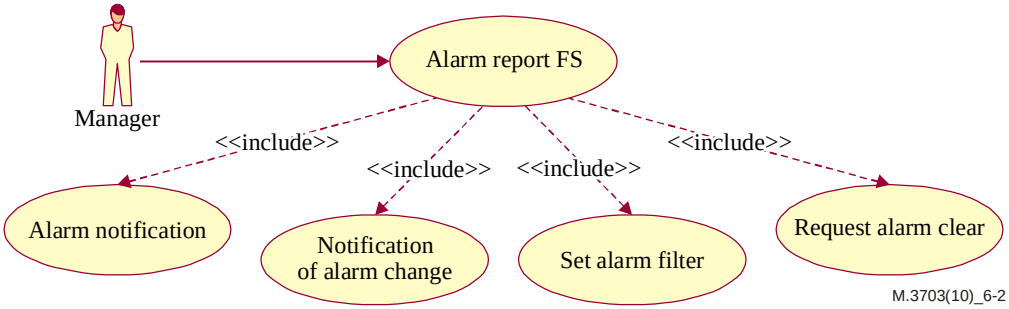

Figure 6-2 – Alarm report function set

Figure 6-3 shows the functions involved in the alarm handling function set.

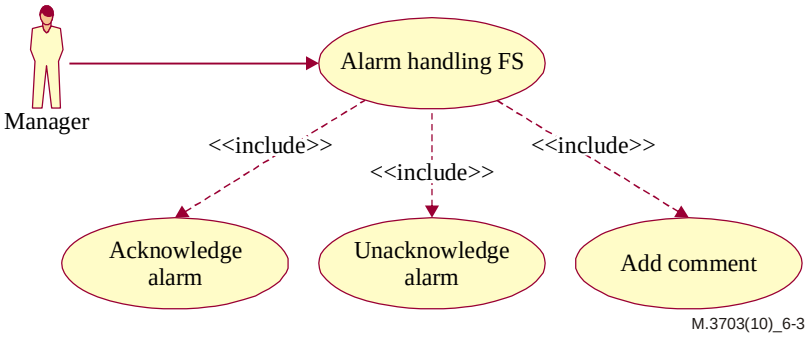

Figure 6-3 – Alarm handling function set

Figure 6-4 shows the functions involved in the ASAP management function set.

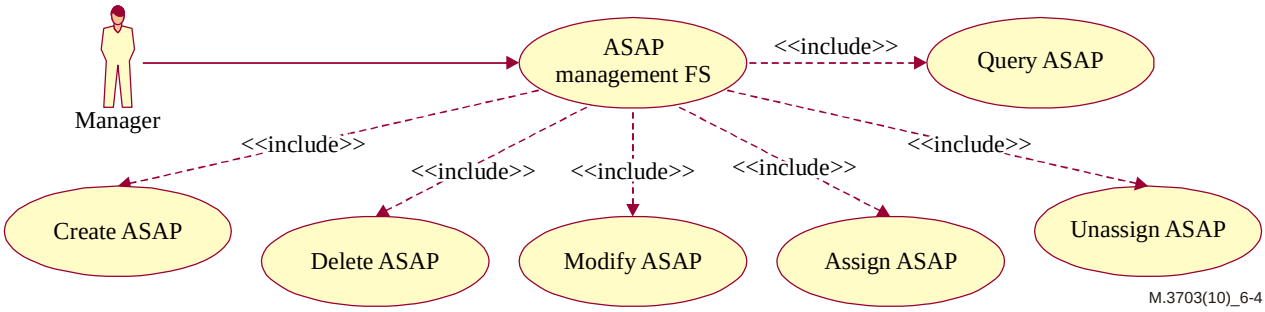

Figure 6-4 – ASAP management function set

Figure 6-5 shows the functions involved in the alarm synchronize function set.

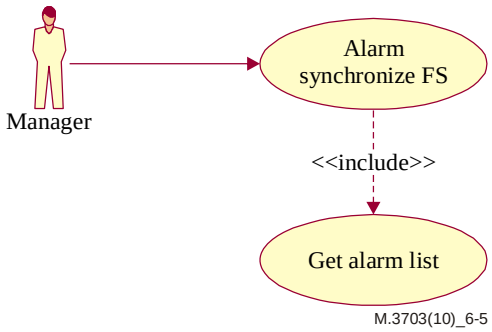

Figure 6-5 – Alarm synchronize function set

6.3 Specification-level requirements

6.3.1 Requirements

There are no specification-level requirements.

6.3.2 Actor roles

See clause 6.2.2.

6.3.3 Telecommunication resources

See clause 6.2.3.

6.3.4 Use cases

6.3.4.1 Use case: Set alarm filter

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	The manager sets a filter for alarm notification subscription.	
Actor and roles	The manager invokes the operation on the agent.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available.	
Preconditions	None.	
Begins when	The manager requests to set a filter to the agent.	
Step 1	The manager requests to set a filter to subscribe alarm information; the manager provides the criteria parameters (e.g., the managed entities, severity level, event type, acknowledgement state, etc.) in the request.	
Ends when	Requested information or an exception is returned to the manager, or the operation is cancelled by the manager.	
Exceptions	Invalid parameters.	
Post conditions	The filter is received by the agent.	
Traceability	REQ-FM-FUN-02, REQ-FM-FUN-15	

6.3.4.2 Use case: Alarm notification

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	An agent sends an alarm report notification of the relevant type to the manager.	
Actor and roles	The manager is a consumer of notifications from the agent.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available.	
Preconditions	A fault condition is detected.	
Begins when	The agent begins to send an alarm report notification to the manager.	
Step 1	The agent sends notification of fault and gives parameters (e.g., identifier of alarm, time, severity level, event type, etc.) in the notification.	
Ends when	The notification is emitted by the agent.	
Exceptions	None.	
Post conditions	The manager is informed of the fault condition in the agent.	
Traceability	REQ-FM-FUN-01, REQ-FM-FUN-03	

6.3.4.3 Use case: Notification of alarm change

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	An agent notifies the subscribed manager of the information change of an alarm.	
Actor and roles	The manager is a consumer of notifications from the agent.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available.	
Preconditions	The information (e.g., state, comment, etc.) of an alarm is changed.	
Begins when	The agent begins to send a notification to the subscribed managers.	
Step 1	When the agent notifies the managers of the change of information of an alarm, the agent will give some parameters (e.g., identifier of alarm, the new state/status value, the new comment, etc.) as part of the notification.	
Ends when	The notification is emitted by the agent.	
Exceptions	None.	
Post conditions	The manager is informed of the change of an alarm.	
Traceability	REQ-FM-FUN-09, REQ-FM-FUN-10	

6.3.4.4 Use case: Request alarm clear

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	The manager explicitly requests the clearing of one or more alarms.	
Actor and roles	The manager invokes the operation on the agent.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available.	
Preconditions	The alarm to be cleared exists.	
Begins when	The manager begins to request clearing of one or more alarms.	
Step 1	The manager sends a request to the agent to clear one or more alarms, the parameters given in the request can be identifier of alarm, severity level and other parameters.	
Ends when	Requested information or an exception is returned to the manager, or the operation is cancelled by the manager.	
Exceptions	Invalid parameters.	
Post conditions	The specified alarms are cleared and the agent sends a notification to the subscribed manager.	
Traceability	REQ-FM-FUN-04	

6.3.4.5 Use case: Get alarm list

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	The manager requests and receives a list of alarms from the agent.	
Actor and roles	The manager invokes operation on the agent.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available.	
Preconditions	The manager sends a request to the agent to get alarm list.	
Begins when	The manager requests active alarm list from an agent with a filtering criteria.	
Step 1	A list of alarms, starting at the oldest, according to the filtering criteria, is received by the manager.	
Ends when	Requested information or an exception is returned to the manager, or the operation is cancelled by the manager.	
Exceptions	Invalid criteria parameter.	
Post conditions	The complete list of alarms (active and/or historic) at the time of the initial request is received by the manager.	
Traceability	REQ-FM-FUN-05, REQ-FM-FUN-17	

6.3.4.6 Use case: Acknowledge alarm

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	The manager requests an alarm to be acknowledged.	
Actor and roles	The manager invokes operation on the agent.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available. The agent supports the acknowledged state alarm attribute.	
Preconditions	The specified alarm has not been acknowledged.	
Begins when	The manager requests an alarm to be marked as acknowledged.	
Step 1	The manager sends a request to the agent to set the acknowledged state of an alarm to TRUE.	
Ends when	Requested information or an exception is returned to the manager, or the operation is cancelled by the manager.	
Exceptions	– unknown alarm; – the alarm has been already acknowledged.	
Post conditions	The agent marks the alarm as acknowledged, and the agent sends a notification of the event to the subscribed managers.	
Traceability	REQ-FM-FUN-12, REQ-FM-FUN-13, REQ-FM-FUN-16	

6.3.4.7 Use case: Unacknowledge alarm

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	The manager requests that an acknowledged alarm be unacknowledged.	
Actor and roles	The manager invokes operation on the agent.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available. The agent supports the acknowledged state alarm attribute.	
Preconditions	The specified alarm has been acknowledged.	
Begins when	The manager requests an alarm be marked as not acknowledged.	
Step 1	The manager sends a request to the agent to set the acknowledged state of an alarm to FALSE.	
Ends when	Requested information or an exception is returned to the manager, or the operation is cancelled by the manager.	
Exceptions	– unknown alarm; – the alarm has not been acknowledged.	
Post conditions	The agent marks the alarm as not acknowledged, and the agent sends a notification of the event to the subscribed managers.	
Traceability	REQ-FM-FUN-12, REQ-FM-FUN-13, REQ-FM-FUN-16	

6.3.4.8 Use case: Create ASAP

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	A manager requests an agent to create an ASAP through the management interface.	
Actor and roles	The manager invokes operation on the agent.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available.	
Preconditions	The manager needs to assign the alarm severities for a set of problems, so that when an agent reports alarms, these pre-assigned severities can be referenced in the corresponding alarm notifications.	
Begins when	The manager sends a request to the agent to create an ASAP.	
Step 1	The manager requests to create an ASAP, and gives the parameters (problems and their corresponding severity) in the request.	
Ends when	Requested information or an exception is returned to the manager, or the operation is cancelled by the manager.	
Exceptions	– invalid parameter; – unknown managed entity.	
Post conditions	An ASAP is successfully created by the agent according to the request. The agent returns the identifier of the ASAP instance. The newly created ASAP will be associated with the managed entities, if specified in the request.	
Traceability	REQ-FM-FUN-06	

6.3.4.9 Use case: Delete ASAP

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	A manager requests an agent to delete an ASAP through the management interface.	
Actor and roles	The manager invokes operation on the agent.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available.	
Preconditions	The specified ASAP exists in the agent, and it is not associated with any managed entities.	
Begins when	The manager sends a request to the agent to delete an ASAP.	
Step 1	The manager sends a request to the agent to delete an ASAP. The request parameter is the identifier of the ASAP. The ASAP to be deleted should not be associated with any managed object; otherwise it cannot be deleted. If the deletion operation succeeds, the agent returns a success indication, and may send an object deletion notification to the manager.	

Use case stage	Evolution/Specification	<<Uses>> Related use
	If the operation fails, the agent will send back error information to the manager.	
Ends when	Requested information or an exception is returned to the manager, or the operation is cancelled by the manager.	
Exceptions	– unknown ASAP; – ASAP association not removed.	
Post conditions	The ASAP is successfully deleted by the agent according to the request.	
Traceability	REQ-FM-FUN-08	

6.3.4.10 Use case: Modify ASAP

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	A manager requests an agent to modify, add or delete table entries (problem and the corresponding alarm severity) of an ASAP.	
Actor and roles	The manager invokes operation on the agent.	
Assumptions	The communication between the manager and the agent is available.	
Telecom resources	All types of telecommunication resources.	
Preconditions	The manager needs to change the table entries of the alarm severity assignment of an ASAP.	
Begins when	The manager sends a request to an agent to modify an ASAP.	
Step 1	The manager sends a request to the agent to modify an ASAP. The request parameter is the new list of the problems and their corresponding severity to be modified. If the modification operation succeeds, the agent will send success information. If the operation fails, the agent will send back error information to the manager.	
Ends when	Requested information or an exception is returned to the manager, or the operation is cancelled by the manager.	
Exceptions	– unknown ASAP; – invalid parameter.	
Post conditions	An ASAP on the agent is successfully modified according to the request.	
Traceability	REQ-FM-FUN-07	

6.3.4.11 Use case: Assign ASAP

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	A manager requests an agent to set or change the association between an ASAP instance and one or more specified managed entities.	
Actor and roles	The manager invokes operation on the agent.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available.	
Preconditions	The specified ASAP exists in the agent.	
Begins when	The manager sends a request to the agent to change the association between an ASAP instance and one or more specified managed entities.	
Step 1	When an ASAP is created successfully, it will not take effect immediately until associated with managed entities. When a managed entity is about to report an alarm, it will first check in the associated ASAP whether the corresponding alarm severity is specified. If already specified, the corresponding severity is assigned to the alarm and then it is reported to the manager. Otherwise, the original severity is applied. In this use case, the manager sends a request to the agent to set or change the ASAP association. The request parameter is the ID or a list of IDs of the managed entities to be associated with the ASAP. If the operation succeeds, the agent will return success information, and the ASAP starts to take effect on the specified managed entity(s). If the operation fails, the agent will return error information.	
Ends when	Requested information or an exception is returned to the manager, or the operation is cancelled by the manager.	
Exceptions	– unknown managed entity; – unknown ASAP.	
Post conditions	The association between the ASAP and the specified managed entity(s) is successfully assigned by the agent. The ASAP then takes effect on the associated managed entity(s).	
Traceability	REQ-FM-FUN-09	

6.3.4.12 Use case: Unassign ASAP

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	A manager removes the association between an ASAP and some of its associated managed entity(s).	
Actor and roles	The manager invokes operation on the agent.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available.	

Use case stage	Evolution/Specification	<<Uses>> Related use
Preconditions	The specified ASAP exists in agent. The association between the ASAP and the specified managed entity(s) has been assigned. The manager does not want the specified managed entity(s) to refer to this ASAP.	
Begins when	The manager sends a request to the agent to remove the association between an ASAP and some of its associated managed entity(s).	
Step 1	When an ASAP is associated with a managed object, it starts to take effect. When the association between an ASAP and a managed element is no longer needed, it can be removed. If the manager wants to associate a managed object with another ASAP, the association with the previous ASAP must be removed first. In this use case, the manager sends a remove ASAP association request to the agent. The request parameter is: the ID or list of IDs of the managed entity(s) associated with the ASAP. If the operation succeeds, the agent will return success information, and the ASAP associated with specified managed object(s) will not take effect any longer. If the operation fails, the agent will return error information to the manager.	
Ends when	The requested information or an exception is returned to the manager, or the operation is cancelled by the manager.	
Exceptions	– unknown managed entity; – the association does not exist.	
Post conditions	The association between the ASAP and the specified managed object(s) is removed by the agent. The agent may send the related attribute value change notifications to the manager.	
Traceability	REQ-FM-FUN-10	

6.3.4.13 Use case: Query ASAP

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	A manager queries the information of an ASAP through the management interface.	
Actor and roles	The manager invokes operation on the agent.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available.	
Preconditions	The specified ASAP exists in the agent.	
Begins when	The manager sends a request to query the information of an ASAP.	
Step 1	The manager sends a request to the agent to query the attribute information of an ASAP, which includes the ID of the ASAP, the list of the problem and the corresponding severity, and the list of managed entities that have been associated with this ASAP.	

Use case stage	Evolution/Specification	<<Uses>> Related use
	If the operation succeeds, the agent will return the corresponding attribute values of the ASAP. If the operation fails, the agent will return error information.	
Ends when	The requested information or an exception is returned to the manager, or the operation is cancelled by the manager.	
Exceptions	None.	
Post conditions	The corresponding ASAP information is returned by the agent.	
Traceability	REQ-FM-FUN-11	

6.3.4.14 Use case: Add Comment

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	The manager requests that a comment be added to the attributes of an alarm.	
Actor and roles	The manager invokes operation on the agent.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available. The agent supports an operation of adding comment.	
Preconditions	The alarm exists.	
Begins when	The manager requests a comment be added to an alarm.	
Step 1	The manager sends a request to the agent to add a comment to an alarm, which includes the date and time of the annotation, the text of the comment, the userid, and the identifier of the manager.	
Ends when	The requested information or an exception is returned to the manager, or the operation is cancelled by the manager.	
Exceptions	– invalid alarm identifier; – invalid parameters.	
Post conditions	The agent adds the comment to the specified alarms and sends a notification of the event to subscribed managers.	
Traceability	REQ-FM-FUN-14	

7 Analysis

7.1 Concepts and background

The alarm management service makes use of the common management services shown in Figures 7-1 and 7-2.

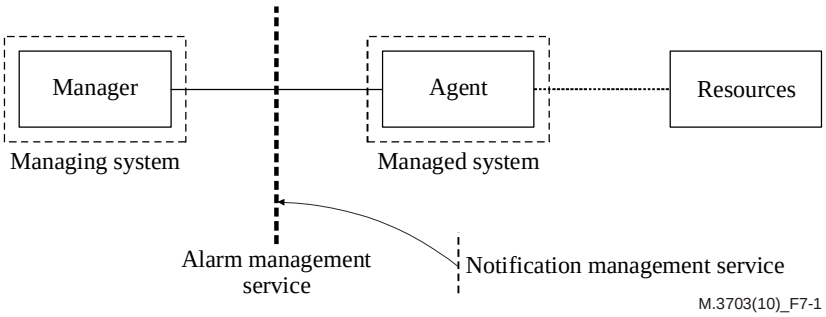

Figure 7-1 – System context A

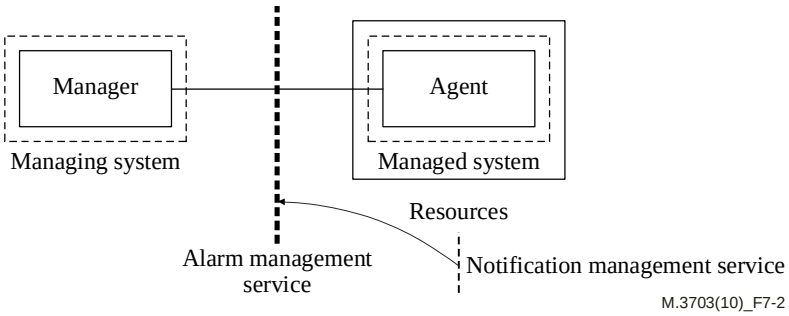

Figure 7-2 – System context B

7.2 Information object classes

7.2.1 Imported information entities and local label

Label reference	Local label
[ITU-T M.3702], information object class, NotificationIRP	NotificationIRP
[ITU-T M.3160], information object class, Top	Top

7.2.2 Class diagram

This clause introduces the set of information object classes (IOCs) that encapsulate information within the agent. The intent is to identify the information required for the alarm agent 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.

7.2.2.1 Attributes and relationships

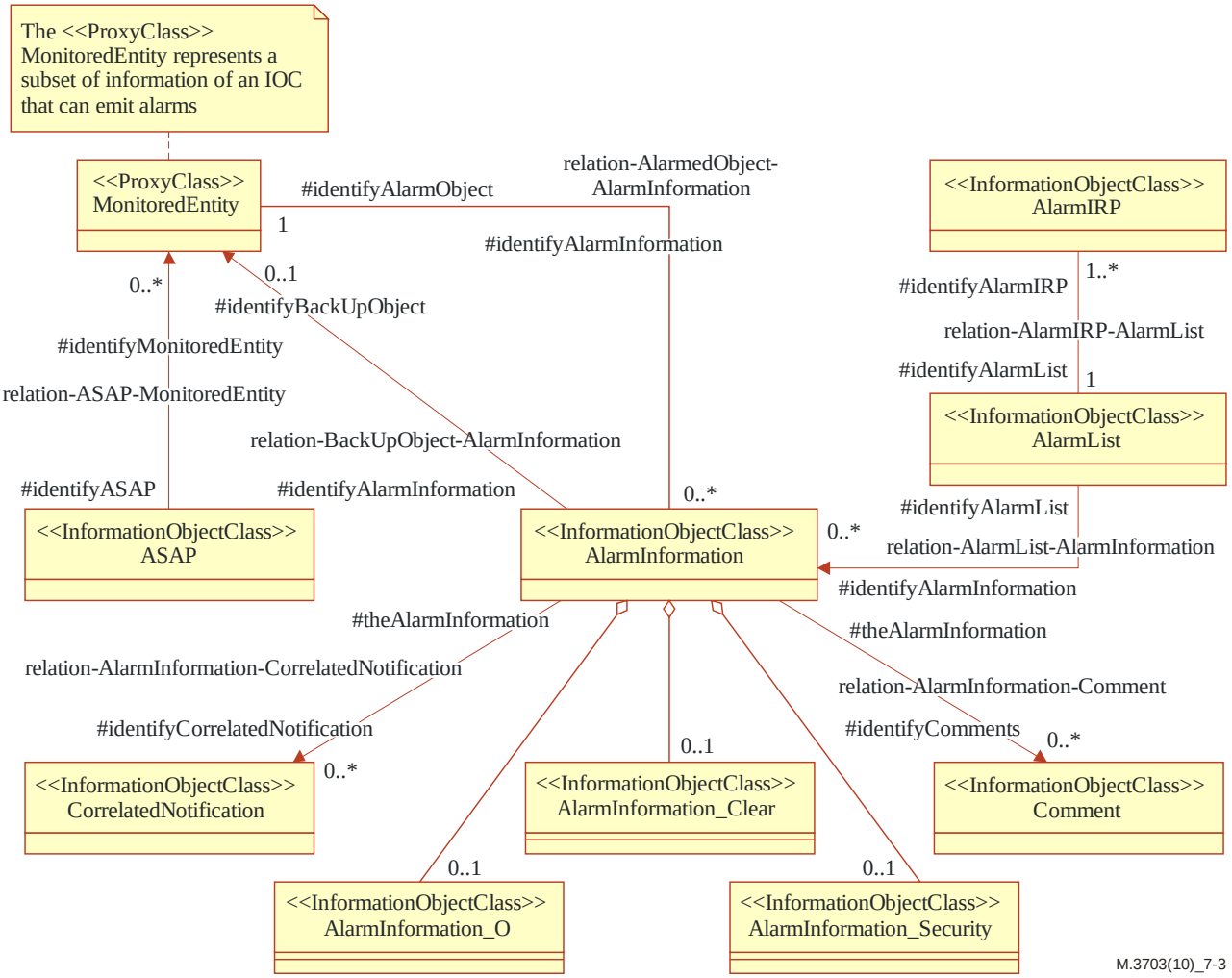

Figure 7-3 – Alarm management information object classes

7.2.2.2 Inheritance

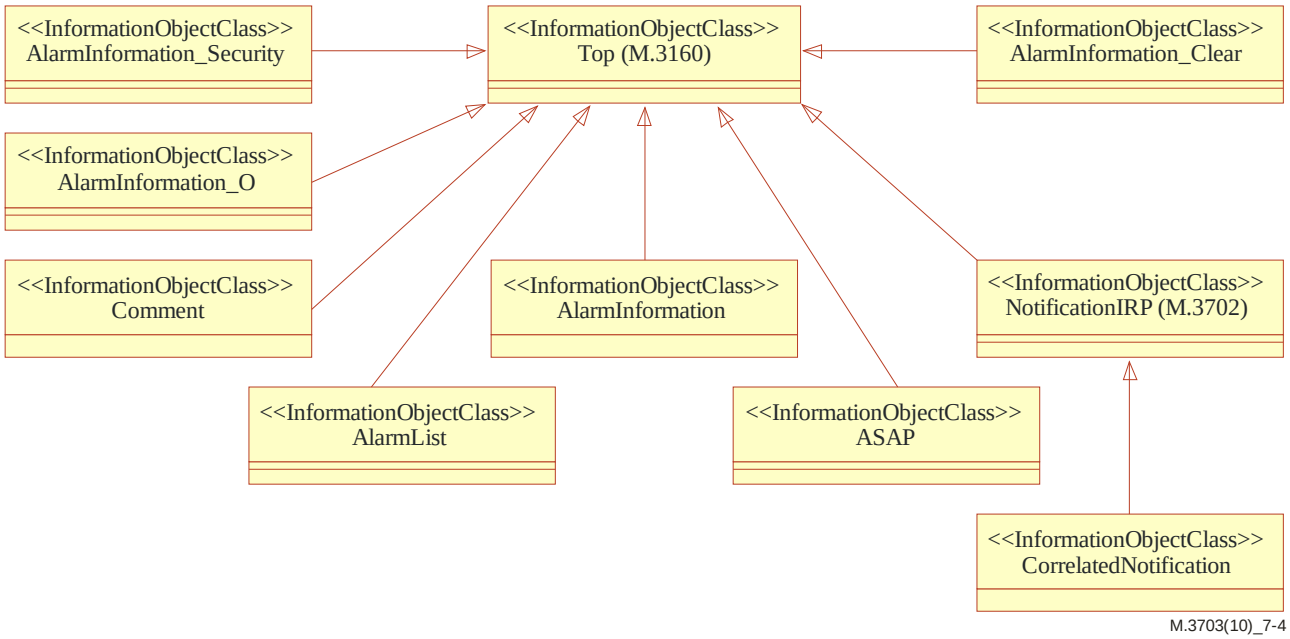

Figure 7-4 – Alarm management IOC inheritance

7.2.3 Information object class definitions

Class name	Qualifier	Requirement IDs
AlarmInformation	M	REQ-FM-FUN-01, REQ-FM-FUN-02, REQ-FM-FUN-03, REQ-FM-FUN-12
AlarmInformation_O	O	REQ-FM-FUN-01, REQ-FM-FUN-02, REQ-FM-FUN-03, REQ-FM-FUN-12
AlarmInformation_Clear	CO (see Note 1)	REQ-FM-FUN-04
AlarmInformation_Security	CO (see Note 2)	REQ-FM-FUN-01
AlarmList	M	REQ-FM-FUN-05, REQ-FM-FUN-17
Comment	M	REQ-FM-FUN-14
CorrelatedNotification	M	REQ-FM-FUN-01
ASAP	O	REQ-FM-FUN-06, REQ-FM-FUN-07, REQ-FM-FUN-08, REQ-FM-FUN-09, REQ-FM-FUN-10, REQ-FM-FUN-11
AlarmIRP	M	REQ-FM-FUN-01, REQ-FM-FUN-05, REQ-FM-FUN-17
NOTE 1 – These attributes and qualifiers are applicable only if the agent supports clearAlarms() (they are absent if clearAlarms() is not supported). NOTE 2 – These attributes must be supported if the agent emits `NewAlarmNotification` that carries security alarm information.		

7.2.3.1 AlarmInformation

7.2.3.1.1 Definition

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

One Agent is related to at most one AlarmList. The Agent 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.

7.2.3.1.2 Attributes

Attribute name	Support qualifier	Read qualifier	Write qualifier	Requirement IDs
alarmId	M	M	–	REQ-FM-FUN-01
alarmRaisedTime	M	M	–	REQ-FM-FUN-01
alarmClearedTime	M	M	M	REQ-FM-FUN-03
eventType	M	M	–	REQ-FM-FUN-01
probableCause	M	M	–	REQ-FM-FUN-01
perceivedSeverity	M	M	M	REQ-FM-FUN-01
ackTime	M	M	M	REQ-FM-FUN-12
ackUserId	M	M	M	REQ-FM-FUN-12
ackState	M	M	M	REQ-FM-FUN-12

7.2.3.1.3 State diagram

Alarms have states. The alarm state information is captured in AlarmInformation in perceivedSeverity and ackState.

The solid circle icon represents the Start State. The double circle icon represents the End State. In the End 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 Agent because of the triggering event;

– Z is the notification to be emitted by Agent because of the triggering event.

Note that acknowledgeAlarm^notifyAckStateChanged and the unacknowledgeAlarm^notifyAckStateChange refer to cases when the request of the Manager 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, 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 and another one from unack&clear to itself.

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

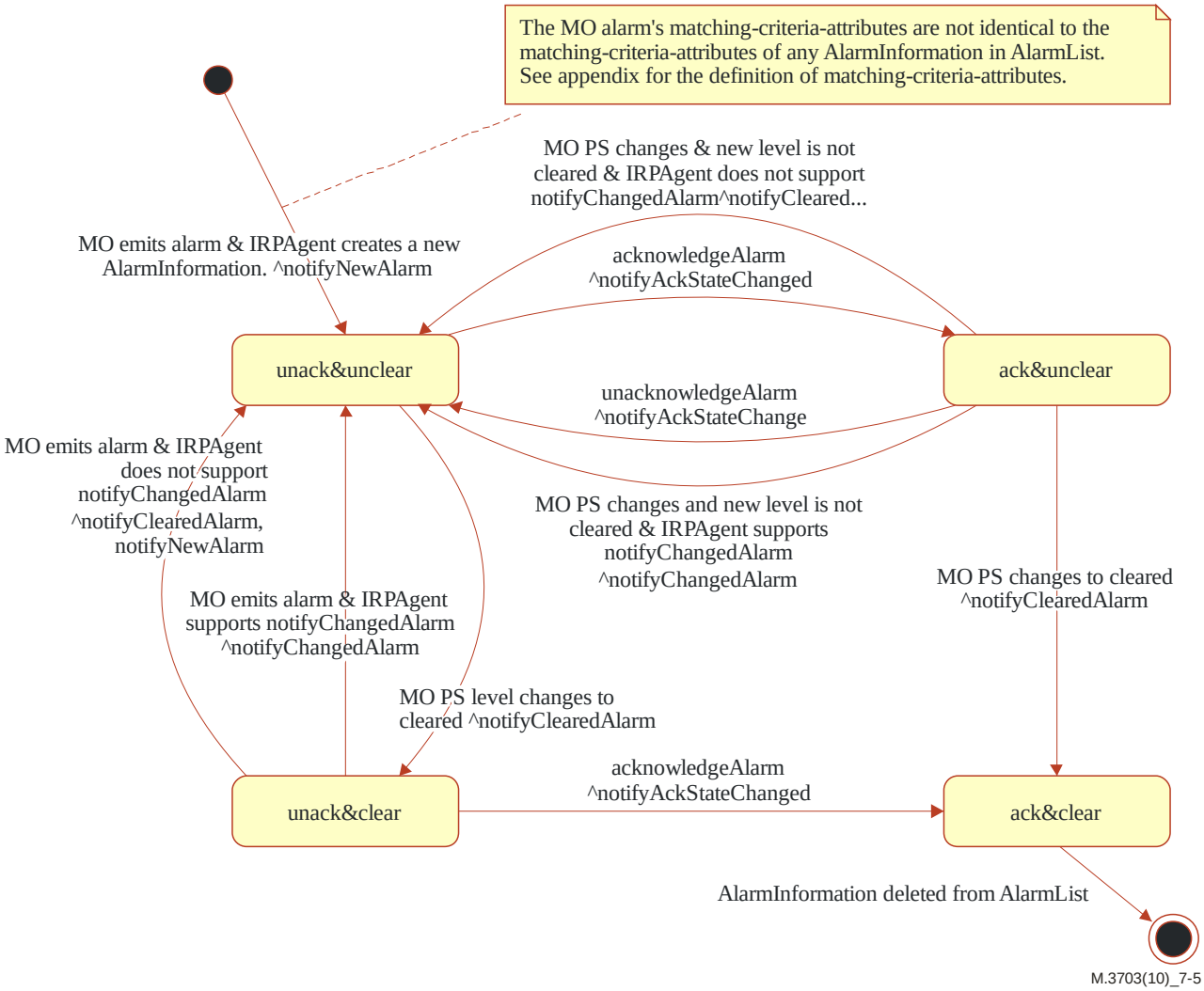

Figure 7-5 – Alarm state transfer

7.2.3.2 AlarmInformation_O

7.2.3.2.1 Definition

`AlarmInformation_O` contains optional information about the alarm condition of an alarmed `MonitoredEntity`.

7.2.3.2.2 Attributes

Attribute name	Support qualifier	Read qualifier	Write qualifier	Requirement IDs
alarmChangedTime	O	O	O	REQ-FM-FUN-03, REQ-FM-FUN-12
vendorSpecificAlarmType	O	O	–	REQ-FM-FUN-01
specificProblem	O	O	–	REQ-FM-FUN-01
backedUpStatus	O	O	–	REQ-FM-FUN-01
backedUpObject	O	O	–	REQ-FM-FUN-01
trendIndication	O	O	–	REQ-FM-FUN-01
stateChangedDefinition	O	O	–	REQ-FM-FUN-01, REQ-FM-FUN-03, REQ-FM-FUN-12
monitoredAttributes	O	O	–	REQ-FM-FUN-01, REQ-FM-FUN-02

Attribute name	Support qualifier	Read qualifier	Write qualifier	Requirement IDs
proposedRepairActions	O	O	–	REQ-FM-FUN-01
additionalText	O	O	–	REQ-FM-FUN-01
additionalInformation	O	O	–	REQ-FM-FUN-01
ackSystemId	O	O	O	REQ-FM-FUN-12

7.2.3.2.3 State diagram

The same as the state diagram for AlarmInformation.

7.2.3.3 AlarmInformation_Clear

7.2.3.3.1 Definition

`AlarmInformation_Clear` contains optional information. These attributes and qualifiers are applicable only if the agent supports clearAlarms() (they are absent if clearAlarms() is not supported).

7.2.3.3.2 Attributes

Attribute name	Support qualifier	Read qualifier	Write qualifier	Requirement IDs
clearUserId	O	M	M	REQ-FM-FUN-04
clearSystemId	O	O	O	REQ-FM-FUN-04

7.2.3.3.3 State diagram

The same as the state diagram for AlarmInformation.

7.2.3.4 AlarmInformation_Security

7.2.3.4.1 Definition

`AlarmInformation_Security` contains optional information. These attributes must be supported if the agent emits `NewAlarmNotification` that carries security alarm information.

7.2.3.4.2 Attributes

Attribute name	Support qualifier	Read qualifier	Write qualifier	Requirement IDs
serviceUser	O	O	–	REQ-FM-FUN-01
serviceProvider	O	O	–	REQ-FM-FUN-01
securityAlarmDetector	O	O	–	REQ-FM-FUN-01

7.2.3.4.3 State diagram

The same as the state diagram for AlarmInformation.

7.2.3.5 AlarmList

7.2.3.5.1 Definition

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

7.2.3.5.2 Attributes

Attribute name	Support qualifier	Read qualifier	Write qualifier	Requirement IDs
alarmListId	M	M	–	REQ-FM-FUN-05, REQ-FM-FUN-17
alarmList	M	M	M	REQ-FM-FUN-05, REQ-FM-FUN-17

7.2.3.5.3 State diagram

There is no state for this class.

7.2.3.6 ASAP

7.2.3.6.1 Definition

The manager needs to assign the alarm severities for a set of problems, so that when an agent reports alarms, these pre-assigned severities can be referenced in the corresponding alarm notifications. The manager can create an ASAP on an agent to define the relationship between severity level and problem.

7.2.3.6.2 Attributes

Attribute name	Support qualifier	Read qualifier	Write qualifier	Requirement IDs
aSAPId	M	M	–	REQ-FM-FUN-06
aSAPInfoList	M	M	M	REQ-FM-FUN-06

7.2.3.6.3 State diagram

There is no state for this class.

7.2.3.7 AlarmIRP

7.2.3.7.1 Definition

`AlarmIRP` is the representation of the alarm management capabilities specified by the present Recommendation. Through `AlarmIRP`, the manager can configure ASAP, retrieve AlarmList and acknowledge/unacknowledge Alarm, etc.

7.2.3.7.2 Attributes

Attribute name	Support qualifier	Read qualifier	Write qualifier	Requirement IDs
alarmIRPId	M	M	–	REQ-FM-FUN-01
alarmListId	M	M	–	REQ-FM-FUN-05, REQ-FM-FUN-17

7.2.3.7.3 State diagram

There is no state for this class.

7.2.3.8 Comment

7.2.3.8.1 Definition

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

7.2.3.8.2 Attributes

Attribute name	Support qualifier	Read qualifier	Write qualifier	Requirement IDs
commentTime	M	M	M	REQ-FM-FUN-14
commentText	M	M	M	REQ-FM-FUN-14
commentUserId	M	M	M	REQ-FM-FUN-14
commentSystemId	O	O	O	REQ-FM-FUN-14

7.2.3.8.3 State diagram

There is no state for this class.

7.2.3.9 CorrelatedNotification

7.2.3.9.1 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 clause 8.1.2.9 of [ITU-T X.733].

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 NewAlarmNotification.

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`.

7.2.3.9.2 Attributes

Attribute name	Support qualifier	Read qualifier	Write qualifier	Requirement IDs
source	M	M	–	REQ-FM-FUN-01
notificationIdSet	M	M	–	REQ-FM-FUN-01

7.2.3.9.3 State diagram

There is no state for this class.

7.2.4 Information relationship definitions

Relationship	Support qualifier	Requirement IDs
relation-AlarmIRP-AlarmList (M)	M	REQ-FM-FUN-05, REQ-FM-FUN-17
relation-AlarmList-AlarmInformation	M	REQ-FM-FUN-05, REQ-FM-FUN-07
relation-AlarmInformation-Comment	M	REQ-FM-FUN-14
relation-AlarmInformation-CorrelatedNotification	M	REQ-FM-FUN-01

Relationship	Support qualifier	Requirement IDs
relation-AlarmedObject-AlarmInformation	M	REQ-FM-FUN-01
relation-BackUpObject-AlarmInformation	O	REQ-FM-FUN-01
relation-ASAP-MonitoredEntity	M	REQ-FM-FUN-06

7.2.4.1 relation-AlarmIRP-AlarmList (M)

7.2.4.1.1 Definition

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

7.2.4.1.2 Role

Name	Definition
identifyAlarmIRP	It represents the capability to obtain the identities of one or more AlarmIRP.
identifyAlarmList	It represents the capability to obtain the identity of one AlarmList.

7.2.4.1.3 Constraint

There is no constraint for this relationship.

7.2.4.2 relation-AlarmList-AlarmInformation (M)

7.2.4.2.1 Definition

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

7.2.4.2.2 Role

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

7.2.4.2.3 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.

7.2.4.3 relation-AlarmInformation-Comment (M)

7.2.4.3.1 Definition

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

7.2.4.3.2 Role

Name	Definition
theAlarmInformation	It represents the AlarmInformation.
identifyComment	It represents a capability to obtain the information contained in Comment.

7.2.4.3.3 Constraint

There is no constraint.

7.2.4.4 relation-AlarmInformation-CorrelatedNotification (M)

7.2.4.4.1 Definition

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

7.2.4.4.2 Role

Name	Definition
theAlarmInformation	It represents the AlarmInformation.
identifyCorrelatedNotification	It represents a capability to obtain the information contained in CorrelatedNotification.

7.2.4.4.3 Constraint

There is no constraint.

7.2.4.5 relation-AlarmedObject-AlarmInformation (M)

7.2.4.5.1 Definition

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

7.2.4.5.2 Role

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

7.2.4.5.3 Constraint

There is no constraint.

7.2.4.6 relation-backUpObject-AlarmInformation (O)

7.2.4.6.1 Definition

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

7.2.4.6.2 Role

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

7.2.4.6.3 Constraint

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

7.2.4.7 relation-ASAP-MonitoredEntity (M)

7.2.4.7.1 Definition

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

7.2.4.7.2 Role

Name	Definition
identifyASAP	It represents the capability to obtain the identities of one ASAP.
identifyMonitoredEntity	It represents the capability to obtain the identify of one or more MonitoredEntity.

7.2.4.7.3 Constraint

There is no constraint for this relationship.

7.2.5 Information attribute definition

7.2.5.1 Definition and legal values

Name	Definition	Information type/Legal values
alarmId	It identifies one AlarmInformation in the AlarmList.	`AlarmIdType ::= INTEGER`
alarmRaisedTime	It indicates the date and time when the alarm is first raised by the alarmed resource.	`GeneralizedTime`
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 manager would not change this date and time.	`GeneralizedTime`
alarmClearedTime	It indicates the date and time when the alarm is cleared.	`GeneralizedTime`
eventType	It indicates the type of event. See Annex A for information on event type.	`EventType ::= ENUMERATED { communications_Alarm, processing_Error_Alarm, ...}` `--See Annex A for complete value set.`
vendorSpecificAlarmType	It indicates the vendor-specific alarm that identifies the NE alarm type or NE related alarm type. It is a vendor-specific expression of eventType.	`String`

Name	Definition	Information type/Legal values
probableCause	It qualifies alarm and provides further information than eventType. See Annex B for a complete listing.	`ProbableCauseType ::= ENUMERATED { alarm_Indication_Signal, broadcast_Channel_Failure,...}` `--See Annex B for complete value set.`
perceivedSeverity	It indicates the relative level of urgency for operator attention. This IRP does not recommend the use of indeterminate.	`PerceivedSeverityType ::= ENUMERATED {critical, major, minor, warning, indeterminate, cleared}`
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 clause 8.1.2.2 of [ITU-T X.733].	`String`
backedUpStatus	It indicates if an object (the MonitoredEntity) has a back up. See definition in clause 8.1.2.4 of [ITU-T X.733].	`BOOLEAN`
trendIndication	It indicates if some observed condition is getting better, worse, or is not changing.	`TrendIndicationType ::= ENUMERATED {less_Severe, no_Change, more_Severe}`
stateChangeDefinition	It indicates MO attribute value changes. See definition in clause 8.1.2.10 of [ITU-T X.733].	`String`
monitoredAttributes	It indicates MO attributes whose value changes are being monitored. See definition in clause 8.1.2.11 of [ITU-T X.733].	`SET OF AttributeName` `AttributeName ::= String`
proposedRepairActions	It indicates proposed repair actions. See definition in clause 8.1.2.12 of [ITU-T X.733].	`SET OF AttributeName`
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 [b-3GPP TS 32.622]. It can contain further information on the alarm.	`String`
additionalInformation	It contains information on the alarm. Its semantics is outside the scope of this IRP.	`String`
ackTime	It identifies the time when the alarm was acknowledged or unacknowledged the last time.	`GeneralizedTime`
ackUserId	It identifies the last user who changed the Acknowledgement State.	`String`

Name	Definition	Information type/Legal values
ackSystemId	It identifies the system (EM or NM) from which the alarm was acknowledged or unacknowledged the last time.	String
ackState	It identifies the Acknowledgement State of the alarm.	AckStateType::= ENUMERATED { acknowledged , unacknowledged } *-- acknowledged: the alarm has been acknowledged. -- unacknowledged: the alarm has been unacknowledged or the alarm has never been acknowledged.*
aSAPId	It identifies one ASAP in agent.	INTEGER
aSAPInfoList	It defines the relationship between severity level and problem.	ASAPInfoListType::= SEQUENCE { problem ProbableCauseType, severityLevel PerceivedSeverityType }
commentTime	It carries the time when the comment has been added to the alarm.	GeneralizedTime
commentText	It carries the textual comment.	String
commentUserId	It carries the identification of the user who made the comment.	String
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.	String
source	It identifies one MonitoredEntity.	String
notificationIdSet	It carries one or more notification identifiers.	SET OF AttributeName
clearUserId	It carries the identity of the user who invokes the clearAlarms operation.	String
clearSystemId	It carries the identity of the system in which the manager runs. That manager supports the user who invokes the clearAlarms().	String
serviceUser	It identifies the service-user whose request for service provided by the serviceProvider led to the generation of the security alarm.	String
serviceProvider	It identifies the service provider whose service is requested by the serviceUser and the service request provokes the generation of the security alarm.	String
securityAlarmDetector	It carries the identity of the detector of the security alarm.	String

7.2.5.2 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.

7.3 Interface definition

7.3.1 Class diagram

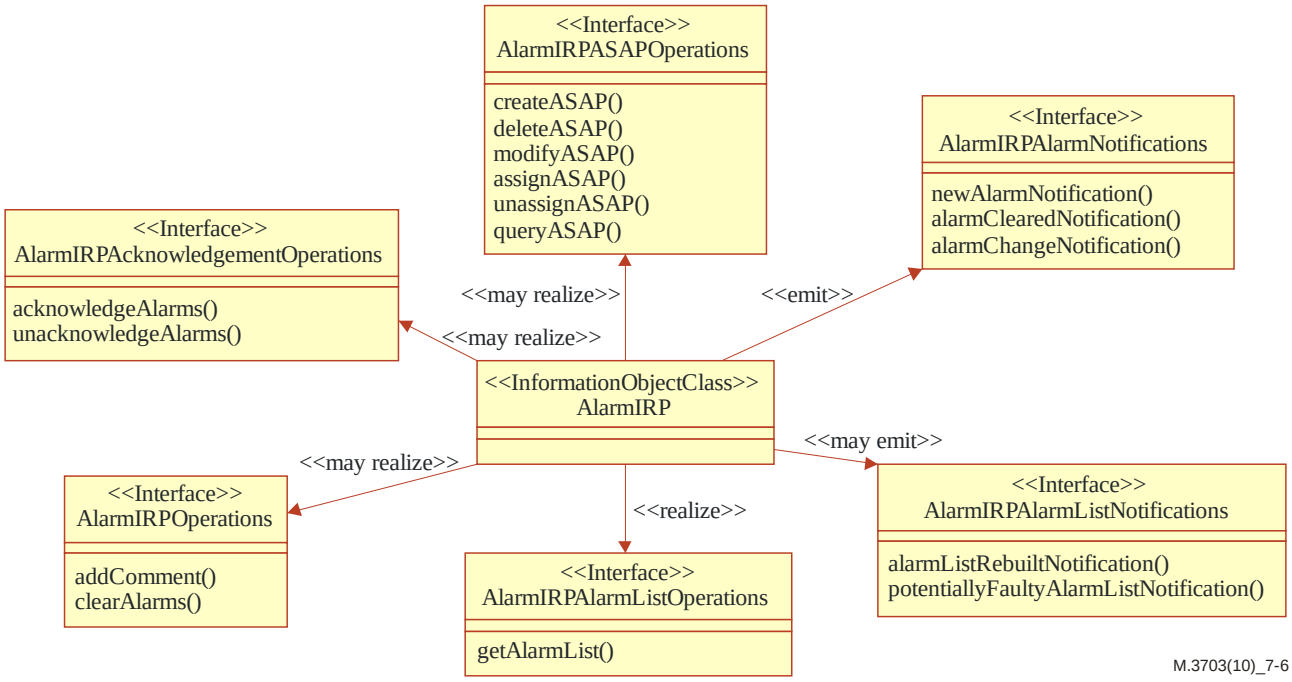

Figure 7-6 – Alarm management IRP class diagram

AlarmIRP must realize the operations defined by AlarmIRPAcknowledgementOperations and AlarmIRPAlarmListOperations, and it may realize the operations defined by AlarmIRPOperations. At the same time, AlarmIRP must have the capability to emit the notifications defined by the AlarmIRPAlarmNotifications, and it may have the capability to emit the notifications defined by the AlarmIRPAlarmListNotifications.

7.3.2 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 regard 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.

7.3.3 Interface AlarmIRPAcknowledgementOperations (O)

Operation name	Qualifier	Requirement IDs
acknowledgeAlarms	M	REQ-FM-FUN-12, REQ-FM-FUN-13, REQ-FM-FUN-15, REQ-FM-FUN-16
unacknowledgeAlarms	O	REQ-FM-FUN-12

7.3.3.1 acknowledgeAlarms (M)

7.3.3.1.1 Definition

The `Manager` invokes this operation to acknowledge one or more alarms.

7.3.3.1.2 Input parameters

Name	Qualifier	Information type/Legal values	Comment
alarmInformationAndSeverityReferenceList	M	`SET OF SEQUENCE { alarmId AlarmIdType, perceivedSeverity PerceivedSeverityType OPTIONAL }`	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. alarmInformationAndSeverity ReferenceList `{alarmId - Mandatory; perceivedSeverity - Optional }`
ackUserId	M	`String`	It identifies the user acknowledging the alarm.
ackSystemId	O	`String`	It identifies the processing system on which the subject `Manager` runs. It may be absent implying that `Manager` does not wish this information be kept in `AlarmInformation` in `AlarmList`.

7.3.3.1.3 Output parameters

Name	Qualifier	Matching information/ Information type/Legal values	Comment

badAlarmInformationReferenceList	M	`SET OF BadAlarmInfo`
		`BadAlarmInfo ::= SEQUENCE {`
		`alarmId AlarmIdType,`
		`reason ENUMERATED {`
		`unknownAlarmId,`
		`acknowledgmentFailed,`
		`wrongPerceivedSeverity`
		`}`
		`}`
			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) has not changed, in contrast to Manager'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 AlarmList = WrongPerceivedSeverity (applicable only if perceivedSeverity was provided).

status	M	`StatusType ::= ENUMERATED { 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`, `AlarmInformation.ackSystemId`, `AlarmInformation.ackState`.			

7.3.3.1.4 Pre-condition

`atLeastOneValidId`.

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

7.3.3.1.5 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.

7.3.3.1.6 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.

7.3.3.2 unacknowledgeAlarms (O)

7.3.3.2.1 Definition

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

7.3.3.2.2 Input parameters

Name	Qualifier	Information type/Legal values	Comment

correlatedIdSet	M	`SET OF CorrelatedIdInfo`
		`CorrelatedIdInfo ::= SEQUENCE {`
		`source String,`
		`alarmid AlarmIdType}`
			It carries one or more identifiers identifying `AlarmInformation` instances in `AlarmList`

ackUserId	M	`String`	It identities the user that invokes this operation.
ackSystemId	O	`String`	It identifies the processing system on which the subject Manager runs.

7.3.3.2.3 Output parameters

Name	Qualifier	Matching information/ Information type/Legal values	Comment
badAlarmInformation ReferenceList	M	`SET OF BadAlarmInfo`	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) has not changed, in contrast to manager's request.
status	M	`StatusType`	If someAlarmUnacknowledged is true, status = OperationPartiallySuceeded. 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`, `AlarmInformation.ackSystemId` and `AlarmInformation.ackState`.			

7.3.3.2.4 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 "acknowledged".

7.3.3.2.5 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.

7.3.3.2.6 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.

7.3.4 Interface AlarmIRPOperations (O)

Operation name	Qualifier	Requirement IDs
addComment	O	REQ-FM-FUN-14
clearAlarms	O	REQ-FM-FUN-04

7.3.4.1 addComment (O)

7.3.4.1.1 Definition

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

If this operation is supported, the AlarmChangeNotification notification must be supported by the IRP for issuing notifications of invocations of this interface.

7.3.4.1.2 Input parameters

Name	Qualifier	Information type/Legal values	Comment
alarmInformationReferenceList	M	`SET OF AlarmIdType`	It carries one or more identifiers identifying AlarmInformation instances in the AlarmList.
commentUserId	M	`String`	The Comment.commentUserId where Comment is involved in relation-AlarmInformation-Comment with an AlarmInformation.
commentSystemId	O	`String`	The Comment.commentSystemId where Comment is involved in relation-AlarmInformation-Comment with an AlarmInformation.
commentText	M	`String`	The comment.commentText where Comment is involved in relation-AlarmInformation-Comment with an AlarmInformation.

7.3.4.1.3 Output parameter

Name	Qualifier	Matching information/ Information type/Legal values	Comment
badAlarm Information ReferenceList	M	`SET OF BadAlarmInfo`	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 are present, but AlarmInformation.comments has not changed, in contrast to manager's request.
status	M	`StatusType`	If allUpdated is true, then status = OperationSucceeded. If someUpdated is true, then status = OperationPartiallyFailed. If exception operationFailed is raised, then status = OperationFailed.

7.3.4.1.4 Pre-condition

`atLeastOneValidId`.

Assertion name	Properties
atLeastOneValidId	The AlarmInformationReferenceList contains at least one identifier that identifies one AlarmInformation in AlarmList.

7.3.4.1.5 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 parameters 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 agent may remove one or more old comments 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 parameters 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.

7.3.4.1.6 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.

7.3.4.2 clearAlarms (O)

7.3.4.2.1 Definition

The `Manager` 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.

7.3.4.2.2 Input parameter

Name	Qualifier	Information type/Legal values	Comment
alarmInformation ReferenceList	M	`SET OF AlarmIdType`	It carries one or more identifiers identifying AlarmInformation instances in the AlarmList.
clearUserId	M	`String`	It identifies the user clearing the alarm.
clearSystemId	O	`String`	It identifies the processing system on which the subject the manager runs. It may be absent implying that the manager does not wish this information be known to the agent.

7.3.4.2.3 Output parameter

Name	Qualifier	Matching information/ Information type/Legal values	Comment
badAlarmInformation ReferenceList	M	`SET OF BadAlarmInfo`	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 manager's request.
status	M	`StatusType`	If allCleared is true, then status = OperationSucceeded. If someCleared is true, then status = OperationPartiallyFailed. If exception operationFailed is raised, then status = OperationFailed.

7.3.4.2.4 Pre-condition

`atLeastOneValidId`.

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

7.3.4.2.5 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.

7.3.4.2.6 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.

7.3.5 Interface AlarmIRPAlarmListOperations (M)

Operation name	Qualifier	Requirement IDs
getAlarmList	M	REQ-FM-FUN-05

7.3.5.1 getAlarmList (M)

7.3.5.1.1 Definition

The `Manager` invokes this operation in order to request the `Agent` to provide either the complete list of `AlarmInformation` instances in the `AlarmList`, including the IOC instances associated with the `AlarmInformation` instances (full alarm alignment), 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 certain MO, then only the `AlarmInformation` instances (and associated IOC instances) related to this MO and its subordinate MOs shall be provided (partial alarm alignment). If a `baseObjectClass` is specified but no `baseObjectInstance`, alarm information related to all MOs of the specified object type will be retrieved.

7.3.5.1.2 Input parameters

Name	Qualifier	Information type/Legal values	Comment
alarmStateFilter	O	`AlarmStateFlterType ::= ENUMERATED { 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 `Agent` shall apply it on AlarmInformation instances in AlarmList when constructing its output parameter AlarmInformationList.

Name	Qualifier	Information type/Legal values	Comment
baseObjectClass	O, (see Note 1)	`String`	If this parameter is absent, then all `AlarmInformation` instances in the `AlarmList` shall be returned. If the parameter carries the object class of a certain MO, then all `AlarmInformation` instances (and associated IOC instances) of the MO identified by the parameter `baseObjectInstance` and its subordinate MOs shall be returned. The `AlarmInformation` instances not related to the subject MO and its subordinate MOs shall not be returned (see Note 2).
baseObjectInstance	O, (see Note 1)	`Name`	If the `objectClass` parameter is absent, then this parameter shall be absent. If the `baseObjectClass` parameter carries the object class of a certain MO, then this parameter shall carry the DN of the related MO instance. The AlarmList has to be returned only for alarms concerning that MO and its subordinate MOs.
filter	O	`String`	It carries a filter constraint. The agent shall apply it on AlarmInformation instances in AlarmList when constructing its output parameter AlarmInformationList.
NOTE 1 – If the notification `AlarmListRebuiltNotification` 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 `alarmListRebuiltNotification` notifications. The timeline for "recent" is vendor-specific.			

7.3.5.1.3 Output parameters

Name	Qualifier	Matching information/ Information type/Legal values	Comment
alarmInformationList	M	`SET OF AlarmIdType`	It carries the requested AlarmInformation instances including the associated IOC instances in AlarmList. Case when synchronous mode of operation is used: a) The agent shall apply the constraints expressed in alarmStateFilter and filter to AlarmInformation instances when constructing this output parameter.

Name	Qualifier	Matching information/ Information type/Legal values	Comment
			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 agent shall apply the constraint when constructing this output parameter. Furthermore, if the alarmStateFilter constraint is present, the agent 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 agent shall apply the filter constraint currently active in the notification channel when constructing this output parameter. If the alarmAckState constraint is present, the agent shall apply that constraint as well.
status	M	`StatusType`	If allAlarmInformationReturned is true, status = OperationSucceeded. If operation_failed is true, status = OperationFailed.

7.3.5.1.4 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.

7.3.5.1.5 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.

7.3.5.1.6 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.

7.3.6 Interface AlarmIRPASAPOperations (O)

Notification name	Qualifier	Requirement IDs
createASAP	O	REQ-FM-FUN-06
deleteASAP	O	REQ-FM-FUN-08
modifyASAP	O	REQ-FM-FUN-07
assignASAP	O	REQ-FM-FUN-09
unassignASAP	O	REQ-FM-FUN-10
queryASAP	O	REQ-FM-FUN-11

7.3.6.1 createASAP (O)

7.3.6.1.1 Definition

`Manager` invokes this operation to create an ASAP.

7.3.6.1.2 Input parameters

Name	Qualifier	Information type/Legal values	Comment
aSAPInfoList	M	`ASAPInfoListType`	

7.3.6.1.3 Output parameters

Name	Qualifier	Matching information/ Information type/Legal values	Comment
aSAPId	M	`INTEGER`	This parameter specifies the identifier of the ASAP.
status	M	`StatusType`	

7.3.6.1.4 Pre-condition

None.

7.3.6.1.5 Post-condition

`aSAPCreated`.

Assertion name	Definition
aSAPCreated	An ASAP is successfully created based on the specified aSAPInfoList.

7.3.6.1.6 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.

7.3.6.2 deleteASAP (O)

7.3.6.2.1 Definition

`Manager` invokes this operation to delete an ASAP.

7.3.6.2.2 Input parameters

Name	Qualifier	Information type/Legal values	Comment
aSAPId	M	INTEGER	This parameter specifies the identifier of the ASAP.

7.3.6.2.3 Output parameters

Name	Qualifier	Matching information/ Information type/Legal values	Comment
status	M	StatusType	

7.3.6.2.4 Pre-condition

aSAPExists.

Assertion name	Definition
aSAPExists	The ASAP specified by the aSAPId parameter exists.

7.3.6.2.5 Post-condition

aSAPDeleted.

Assertion name	Definition
aSAPDeleted	The specified ASAP is successfully deleted.

7.3.6.2.6 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.

7.3.6.3 modifyASAP (O)

7.3.6.3.1 Definition

Manager invokes this operation to change the table entries of the alarm severity assignment of an ASAP.

7.3.6.3.2 Input parameters

Name	Qualifier	Information type/Legal values	Comment
aSAPId	M	INTEGER	This parameter specifies the identifier of the ASAP.
aSAPInfoList	M	ASAPInfoListType	The new list of the problems and their corresponding severity are to be modified.

7.3.6.3.3 Output parameters

Name	Qualifier	Matching information/ Information type/Legal values	Comment
status	M	`StatusType`	

7.3.6.3.4 Pre-condition

`aSAPExists`.

Assertion name	Definition
aSAPExists	The ASAP specified by the aSAPId parameter exists.

7.3.6.3.5 Post-condition

`aSAPModified`.

Assertion name	Definition
aSAPModified	The specified ASAP is successfully modified.

7.3.6.3.6 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.

7.3.6.4 assignASAP (O)

7.3.6.4.1 Definition

`Manager` invokes this operation to set or change the association between an ASAP instance and one or more specified monitored entities.

7.3.6.4.2 Input parameters

Name	Qualifier	Information type/Legal values	Comment
aSAPId	M	`INTEGER`	This parameter specifies the identifier of the ASAP.
monitoredEntityList	M	`SET OF Name`	One or more specified monitored entities' Id.

7.3.6.4.3 Output parameters

Name	Qualifier	Matching information/ Information type/Legal values	Comment
status	M	`StatusType`	

7.3.6.4.4 Pre-condition

`aSAPExists` AND `monitoredEntityExist`.

Assertion name	Definition
aSAPExists	The ASAP specified by the aSAPId parameter exists.
monitoredEntityExist	The MonitoredEntity specified by the monitoredEntityList parameter exist.

7.3.6.4.5 Post-condition

`aSAPAssigned`.

Assertion name	Definition
aSAPAssigned	The specified ASAP is successfully assigned to specified monitored entities.

7.3.6.4.6 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.

7.3.6.5 unassignASAP (O)

7.3.6.5.1 Definition

`Manager` invokes this operation to remove the association between an ASAP and some of its associated monitored entity(s).

7.3.6.5.2 Input parameters

Name	Qualifier	Information type/Legal values	Comment
aSAPId	M	`INTEGER`	This parameter specifies the identifier of the ASAP.
monitoredEntityList	M	`SET OF Name`	One or more specified monitored entities' Id.

7.3.6.5.3 Output parameters

Name	Qualifier	Matching information/ Information type/Legal values	Comment
status	M	`StatusType`	

7.3.6.5.4 Pre-condition

`aSAPExists` AND `associationExist`.

Assertion name	Definition
aSAPExists	The ASAP specified by the aSAPId parameter exists.
associationExist	The association between the specified ASAP and MonitoredEntity specified by the monitoredEntityList parameter exist.

7.3.6.5.5 Post-condition

`aSAPUnassigned`.

Assertion name	Definition
aSAPUnassigned	The relationship between specified ASAP and monitored entity is successfully removed.

7.3.6.5.6 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.

7.3.6.6 queryASAP (O)

7.3.6.6.1 Definition

`Manager` invokes this operation to query the information of an ASAP.

7.3.6.6.2 Input Parameters

Name	Qualifier	Information type/Legal values	Comment
aSAPId	M	`INTEGER`	This parameter specifies the identifier of the ASAP.

7.3.6.6.3 Output parameters

Name	Qualifier	Matching information/ Information type/Legal values	Comment
aSAPInfoList	M	`ASAPInfoListType`	The relationship between severity level and problem defined by specified ASAP.
monitoredEntityList	M	`SET OF Name`	The monitored entities using specified ASAP.
status	M	`StatusType`	

7.3.6.6.4 Pre-condition

`aSAPExists`.

Assertion name	Definition
aSAPExists	The ASAP specified by the aSAPId parameter exists.

7.3.6.6.5 Post-condition

None.

7.3.6.6.6 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.

7.3.7 Interface AlarmIRPAlarmNotifications (M)

Notification name	Qualifier	Requirement IDs
newAlarmNotification	M	REQ-FM-FUN-01
alarmClearedNotification	M	REQ-FM-FUN-03
alarmChangeNotification	M	REQ-FM-FUN-13, REQ-FM-FUN-14

This Recommendation does not specify how the agent can determine if the manager has received alarms correctly.

This Recommendation does not specify methods for the manager and the agent to recover alarm loss. The only mechanism recommended to deal with alarm loss is the use of getAlarmList operation. This Recommendation does not specify conditions under which the manager should invoke this operation.

7.3.7.1 newAlarmNotification (M)

7.3.7.1.1 Definition

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

7.3.7.1.2 Input parameters

Parameter name	Qualifier	Matching information/ Information type/Legal values	Comment
probableCause	M	`ProbableCauseType`	
perceivedSeverity	M	`PerceivedSeverityType`	
eventType	M	`EventType`	
vendorSpecificAlarmType	O	`String`	
specificProblem	O	`String`	
correlatedNotifications	O	`SET OF AttributeName`	It contains references to AlarmInformation instances whose perceivedSeverity levels are Cleared. In this way, alarms may replace older alarms.
backedUpStatus	O	`BOOLEAN`	
backUpObject	O	`String`	It carries the DN of the backup object.
trendIndication	O	`TrendIndicationType`	
stateChangeDefinition	O	`String`	

Parameter name	Qualifier	Matching information/ Information type/Legal values	Comment
monitoredAttributes	O	SET OF AttributeName AttributeName ::= String	
proposedRepairActions	O	SET OF AttributeName	
additionalText	O	String	
additionalInformation	O	String	
serviceUser	CO (see Note 1)	String	
serviceProvider	CO (see Note 1)	String	
securityAlarmDetector	CO (see Note 1)	String	
alarmId	M	AlarmIdType	
NOTE 1 – These attributes must be supported if the agent emits NewAlarmNotification that carries security alarm information. NOTE 2 – All the common attributes defined in the Notification Header as described in clause 7.3.5 of [ITU-T M.3702] will also be included, when this notification is instantiated. The value of the notificationType is "newAlarmNotification".			

7.3.7.1.3 Triggering event

7.3.7.1.3.1 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.

7.3.7.1.3.2 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, 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, stateChangedDefinition, monitoredAttributes, proposedRepairActions, additionalText, additionalInformation.

7.3.7.2 alarmClearedNotification (M)

7.3.7.2.1 Definition

`Agent` notifies the subscribed `Manager` 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) `NewAlarmNotification` notification.

7.3.7.2.2 Input parameters

Parameter name	Qualifier	Matching information/ Information type/Legal values	Comment
probableCause	M	`ProbableCauseType`	
perceivedSeverity	M	`PerceivedSeverityType`	Its value shall indicate Cleared.
eventType	M	`EventType`	
correlatedNotifications	O	`SET OF AttributeName` `AttributeName ::= String`	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	`String`	It is present if the AlarmInformation is cleared by the Manager using clearAlarms.
clearSystemId	O	`String`	It is present if clearUserId is present and if AlarmInformation.clearSystemId contains information.
alarmId	M	`AlarmIdType`	

NOTE – All the common attributes defined in the Notification Header as described in clause 7.3.5 of [ITU-T M.3702] will also be included, when this notification is instantiated. The value of the notificationType is "notifyClearedAlarm".

7.3.7.2.3 Triggering event

7.3.7.2.3.1 From-state

`alarmMatchedAndCleared` OR `clearedByManager`.

Assertion name	Definition
alarmMatchedAndCleared	The matching-criteria-attributes of the newly generated network alarm have values that are identical (matched) with the 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.
clearedByManager	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.

7.3.7.2.3.2 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 clearedByManager: The following attributes of the subject AlarmInformation are updated: notificationId, perceivedSeverity (updated to Cleared), alarmClearedTime, alarmClearedUserId, alarmClearedSystemId.

7.3.7.3 alarmChangeNotification (M)

7.3.7.3.1 Definition

The subscribed `Manager` 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"), in a change to any of the acknowledgement status `ackState`, or a change in the `comment` attribute. 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) `NewAlarmNotification` notification.

7.3.7.3.2 Input parameters

Parameter name	Qualifier	Matching information/ Information type/Legal values	Comment
probableCause	M	`ProbableCauseType`	
perceivedSeverity	CO (see Note 1)	`PerceivedSeverityType`	
eventType	M	`EventType`	
alarmId	M	`AlarmIdType`	
ackState	CO (see Note 2)	`AckStateType`	
ackUserId	M	`String`	If this AlarmInformation has been acknowledged by a human operator, then this parameter contains the operator identifier. If it has been acknowledged by a System (EM or NM), then this parameter contains the identifier of the system.
ackSystemId	M	`String`	This parameter always contains the identifier of the system (EM or NM) where the acknowledgement request was originated.

Parameter name	Qualifier	Matching information/ Information type/Legal values	Comment
comments	O	SET OF AttributeName	The set of Comment instances involved in a relationship with this AlarmInformation. Required if triggering event is alarmCommentChanged
NOTE 1 – Required if triggering event is alarmSeverityChanged. NOTE 2 – Required if triggering event is alarmAckStateHasChanged. NOTE 3 – All the common attributes defined in the Notification Header as described in clause 7.3.5 of [ITU-T M.3702] will also be included, when this notification is instantiated. The value of the notificationType is "notifyChangedAlarm".			

7.3.7.3.3 Triggering event

7.3.7.3.3.1 From-state

alarmMatched AND ((alarmNotCleared AND alarmSeverityChanged) OR alarmAckStateHasChanged OR alarmCommentChanged).

Assertion name	Definition
alarmMatched	The matching-criteria-attributes of the newly generated network alarm have values that are identical (matches) with the ones in one AlarmInformation in AlarmList.
alarmNotCleared	The perceivedSeverity of the newly generated network alarm is not Cleared.
alarmSeverityChanged	The perceivedSeverity of the newly generated network alarm and of the matched AlarmInformation are different.
alarmAckStateHasChanged	The AlarmInformation.ackState of the AlarmInformation identified by from-state assertion alarmInformationExists has been updated. Specifically, the following attributes of the subject AlarmInformation are updated. notificationId, ackTime, ackUserId, ackState, ackSystemId.
alarmCommentChanged	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.

7.3.8 Interface AlarmIRPAlarmListNotifications (O)

Notification name	Qualifier	Requirement IDs
alarmListRebuiltNotification	O	REQ-FM-FUN-17
potentiallyFaultyAlarmListNotification	O	REQ-FM-FUN-17

7.3.8.1 alarmListRebuiltNotification (O)

7.3.8.1.1 Definition

The Agent or its related AlarmIRP maintains an AlarmList. It can lose confidence in the integrity of its AlarmList. Under this condition, Agent or its related AlarmIRP or the related AlarmList shall invoke AlarmListRebuiltNotification notification after the AlarmList has been rebuilt.

The Agent can also invoke AlarmListRebuiltNotification notification indicating that part of the AlarmList has been rebuilt. In this case, the notification carries the managed object (MO) instance

indicating that the `AlarmList` only has been rebuilt for alarms concerning this MO and its subordinate MOs. Furthermore, this notification indicates that there is no rebuilding going on for superior MOs of this MO.

7.3.8.1.2 Input parameters

Parameter name	Qualifier	Matching information/ Information type/Legal values	Comment
reason	M	`ENUMERATED { agent-NE_Communication_Error, agent_Restarts, indeterminate}`	It carries the reason why the agent has rebuilt the AlarmList. This may carry different reasons than that carried by the immediate previous notifyPotentialFaultyAlarmList.
alarmListAlignment Requirement	CO (see Note 1)	`ENUMERATED { alignmentRequired, alignmentNotRequired}`	It carries an enumeration of "alignmentRequired" and "alignmentNotRequired". The agent uses alignmentRequired to indicate that Agent current AL is not identical to the one that could have been built using a) Agent 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 agent uses alignmentNotRequired. When this parameter is absent, it implies alignmentRequired.
NOTE 1 – If the agent supports notifyPotentialFaultyAlarmList() notification, it shall support this parameter. If the agent does not support notifyPotentialFaultyAlarmList() notification, it shall not support this parameter. NOTE 2 – All the common attributes defined in the notification header as described in clause 7.3.5 of [ITU-T M.3702] will also be included, when this notification is instantiated. The value of the notificationType is "notifyAlarmListRebuilt".			

7.3.8.1.3 Triggering event

7.3.8.1.3.1 From-state

`alarmListRebuilt_0` OR `alarmListRebuilt_1`.

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

7.3.8.1.3.2 To-state

`alarmListRebuilt_2`.

Assertion name	Definition
alarmListRebuilt_2	The agent rebuilt the whole or part of AlarmList.

7.3.8.2 potentiallyFaultyAlarmListNotification (O)

7.3.8.2.1 Definition

The agent or its related AlarmIRP maintains an AlarmList. It can lose confidence in the integrity of its AlarmList. Under this condition, the agent or its related AlarmIRP or the related AlarmList shall invoke PotentiallyFaultyAlarmListNotification. They then can begin to rebuild the faulty AlarmList, if necessary. After the successful rebuilding or the discovery that rebuilt is not necessary, they shall invoke AlarmListRebuiltNotification 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 agent.

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

1) The manager should not perform any task requiring the integrity of the AlarmInformation identified as faulty or unreliable by the subject notification.
2) The manager should not invoke operations that require integrity of the AlarmList, such as getAlarmList and acknowledgeAlarms operations.

7.3.8.2.2 Input parameters

Parameter name	Qualifier	Matching information/Information type/Legal values	Comment
reason	M	`ENUMERATED {agent-NE_Communication_Error, agent_Restarts, indeterminate}`	It carries the reason why the agent has to rebuild its AlarmList.

NOTE – All the common attributes defined in the notification header as described in clause 7.3.5 of [ITU-T M.3702] will also be included when this notification is instantiated. The value of the notificationType is "notifyPotentialFaultyAlarmList".

7.3.8.2.3 Triggering event

7.3.8.2.3.1 From-state

`faultyAlarmListDetected`.

Assertion name	Definition
faultyAlarmListDetected	The agent detects faults in part or the whole of its AlarmList.

7.3.8.2.3.2 To-state

`faultyAlarmList`

Assertion name	Definition
faultyAlarmList	The agent initiates the AlarmList rebuild process.

Annex A

Event Types

(This annex forms an integral part of this Recommendation)

This annex lists and explains event types used by this Recommendation.

Event type is defined in [ITU-T X.733]. Table A.1 lists some of the event types referred to in this Recommendation.

Notification IRP: Information Service in [b-3GPP TS 32.302] defines a parameter called `notificationType` that shall be present in all notifications. This Recommendation defines a parameter called `alarmType` that shall be present in all notifications carrying alarm information. Examples of `notificationType` are "notification of new alarm", "notification of AlarmList rebuilt", "notification of alarm cleared", etc. Examples of `alarmType` are the event types defined in Table A.1.

This Recommendation also defines an attribute of `AlarmInformation` called `eventType`. The mapping of this `eventType` (internal attribute and not visible to `Manager`) to `notificationType` or `alarmType` (both visible to `Manager`) is defined in the relevant clauses of this Recommendation. The choice of using "`eventType`" is to keep the list of attributes of `AlarmList` unchanged (compared to 3GPP's 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 `AlarmInformation.eventType` can capture more information than the ITU-T defined event types [ITU-T X.733]. One example is "notification of alarm list rebuilt".

Table A.1 – Event types

Event types	Explanation
Communications	An alarm of this type is associated with the procedure and/or process required conveying information from one point to another ([ITU-T X.733]).
Processing error	An alarm of this type is associated with a software or processing fault ([ITU-T X.733]).
Environmental	An alarm of this type is associated with a condition related to an enclosure in which the equipment resides ([ITU-T X.733]).
Quality of service	An alarm of this type is associated with degradation in the quality of a service ([ITU-T X.733]).
Equipment	An alarm of this type is associated with an equipment fault ([ITU-T X.733]).
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.
Unknown	Event type that cannot be supported by the above definitions.

Annex B

Probable Causes

(This annex forms an integral part of this Recommendation)

This annex lists probable causes and their corresponding event types.

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

Table B.1 – [ITU-T M.3100] probable causes

ITU-T 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

Table B.1 – [ITU-T M.3100] probable causes

ITU-T M.3100 probable cause	Event type
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
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

Table B.1 – [ITU-T M.3100] probable causes

ITU-T M.3100 probable cause	Event type
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

Table B.1 – [ITU-T M.3100] probable causes

ITU-T M.3100 probable cause	Event type
Software download failure	Processing error
Timeout expired	Processing error
Underlying resources unavailable	Processing error
Version mismatch	Processing error
Bandwidth reduced	Quality of service
Congestion	Quality of service
Excessive error rate	Quality of service
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 – [ITU-T X.721], [ITU-T X.733] and [ITU-T X.736] probable causes

ITU-T X.721/ITU-T X.733/ITU-T 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

Table B.2 – [ITU-T X.721], [ITU-T X.733] and [ITU-T X.736] probable causes

ITU-T X.721/ITU-T X.733/ITU-T X.736 probable cause	Event type
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
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

Table B.2 – [ITU-T X.721], [ITU-T X.733] and [ITU-T X.736] probable causes

ITU-T X.721/ITU-T X.733/ITU-T X.736 probable cause	Event type
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
Unauthorized 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 identifies probable causes that are defined by more than one standard. This is for information only.

Table B.3 – Duplicated probable causes

Duplicated probable cause	2G and 3G	ITU-T X.721 ITU-T X.733	ITU-T X.736	ITU-T M.3100	Event type
Broadcast channel failure	X			X	Communications
Call establishment failure (ITU-T X.721/ITU-T X.733) call setup failure (ITU-T 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 (ITU-T M.3100) Antenna problem (2G and 3G)	X			X	Equipment

Table B.3 – Duplicated probable causes

Duplicated probable cause	2G and 3G	ITU-T X.721 ITU-T X.733	ITU-T X.736	ITU-T M.3100	Event type
Battery charging failure (ITU-T M.3100) Battery charging fault (2G and 3G)	X			X	Equipment
Disk failure (ITU-T M.3100) Disk problem (2G and 3G)	X			X	Equipment
Equipment failure (2G and 3G) equipment malfunction (ITU-T X.721/ITU-T X.733)	X	X			Equipment
Frequency hopping failure	X			X	Equipment
IO device error (ITU-T M.3100) Input/output device error (ITU-T X.721/ITU-T X.733)		X		X	Equipment
Loss of redundancy (ITU-T M.3100) Lost redundancy (2G and 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 (ITU-T M.3100) Signal quality evaluation fault (2G and 3G)	X			X	Equipment
Timing problem		X		X	Equipment
Transceiver failure (ITU-T M.3100) Transceiver problem (2G and 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 and 3G) cooling fan failure (ITU-T M.3100)	X			X	Environmental
Fire detected (ITU-T X.721/ITU-T X.733) fire (ITU-T M.3100)		X		X	Environmental

Table B.3 – Duplicated probable causes

Duplicated probable cause	2G and 3G	ITU-T X.721 ITU-T X.733	ITU-T X.736	ITU-T M.3100	Event type
Flood detected (ITU-T X.721/ITU-T X.733) flood (ITU-T M.3100)		X		X	Environmental
High humidity	X			X	Environmental
High temperature	X			X	Environmental
Intrusion detected (2G and 3G) intrusion detection (ITU-T X.736/ITU-T M.3100)	X		X	X	Environmental (2G and 3G); Physical violation (ITU-T X.736/ ITU-T M.3100)
Low humidity	X			X	Environmental
Low temperature	X			X	Environmental
Pump failure		X		X	Environmental
Smoke detected (2G and 3G) smoke (ITU-T M.3100)	X			X	Environmental
Application subsystem failure		X		X	Processing error
Bandwidth reduced Bandwidth reduction (ITU-T X.721/ITU-T X.733)		X		X	Quality of service
Configuration or customization error (ITU-T M.3100) Configuration or customizing error (ITU-T X.721/ ITU-T 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 (ITU-T M.3100) Excessive error rate (2G and 3G) Excessive error rate	X			X	Communications (ITU-T M.3100) Quality of service (GSM 12.11/ ITU-T 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
Underlying resource unavailable (ITU-T M.3100) Underlying resource unavailable (ITU-T X.721/ITU-T X.733)		X		X	Processing error
Version mismatch		X		X	Processing error

Table B.3 – Duplicated probable causes

Duplicated probable cause	2G and 3G	ITU-T X.721 ITU-T X.733	ITU-T X.736	ITU-T M.3100	Event type
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 (ITU-T M.3100) Response time excessive (ITU-T X.721/ITU-T X.733)		X		X	Quality of service
Excessive retransmission rate (ITU-T M.3100) Re-transmission rate excessive (ITU-T X.721/ITU-T X.733)		X		X	Quality of service

Appendix I

Examples of using notifyChangedAlarm

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

This appendix describes a number of valid and invalid interactions governing the case when the agent reports 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 1 to support the 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 2 to support the 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)

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

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

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

Example 3: Invalid sequence 1 to support the 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 2 to support the 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.

Appendix II

Background information about fault management

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

This appendix contains some background information about fault detection, alarm acknowledgement, alarm clearance and fault recovery. This information is out of the scope of the interface between agent and manager, but useful for understanding the background of fault management.

II.1 Fault detection

When a fault occurs within a network, the affected network entities shall be able to detect them immediately and generate an alarm. The network entities accomplish this task using autonomous self-check circuits/procedures, including, in the case of NEs, the observation of measurements, counters and thresholds. The threshold measurements may be predefined by the manufacturer and executed autonomously in the NE, or they may be based on performance measurements administered by the EM, see [b-ITU-T M.3704]. The fault detection mechanism as defined above shall cover both active and standby components of the network entities.

In order to ease the fault localization and repair, the faulty network entity should generate, for each single fault, one single alarm, also in the case where a single fault causes a degradation of the operational capabilities of more than one physical or logical resource within the network entity. An example of this is a hardware fault, which affects not only a physical resource but also degrades the logical resource(s) that this hardware supports. In this case the network entity should generate one single alarm for the faulty resource (i.e., the resource which needs to be repaired) and a number of events related to state management for all the physical/logical resources affected by the fault, including the faulty one itself. In case a network entity is not able to recognize that a single fault manifests itself in different ways, the single fault is detected as multiple faults and originates multiple alarms. In this case however, when the fault is repaired the network entity should be able to detect the repair of all the multiple faults and clear the related multiple alarms. When a fault occurs on the connection media between two NEs or between a NE and an OS, and affects the communication capability between such NE/OS, each affected NE/OS shall detect the fault and generate its own associated communication alarm toward the managing OS. In this case, it is the responsibility of the OS to correlate alarms received from different NEs/OSs and localize the fault in the best possible way.

The majority of the faults should have well-defined conditions for the declaration of their presence or absence, i.e., fault occurrence and fault clearing conditions. Any such incident shall be referred to in this Recommendation as an ADAC fault. The network entities should be able to recognize when a previously detected ADAC fault is no longer present, i.e., the clearing of the fault, using similar techniques as they use to detect the occurrence of the fault. When an ADAC fault is detected, the appropriate alarm shall be generated by the faulty network entities.

For some faults, no clearing condition exists. For the purpose of this Recommendation, these faults shall be referred to as ADMC faults. An example of this is when the network entity has to restart a software process due to some inconsistencies, and normal operation can be resumed afterwards. In this case, although the inconsistencies are cleared, the cause of the problem is not yet corrected. Manual intervention by the system operator shall always be necessary to clear ADMC faults since these, by definition, cannot be cleared by the network entity itself. When an ADMC fault is detected, the appropriate alarm shall be generated by the faulty network entities.

For faults which do not result in standing conditions there is no need for any short-term action, neither from the system operator nor from the network entity itself, since the fault condition lasted

for a short period of time only and then disappeared. An example of this is when a NE detects the crossing of some observed threshold, and in the next sampling interval, the observed value stays within its limits. Such faults also shall be generated by the faulty network entities.

A fault condition is uniquely identified by the combination of the managed object instance experiencing the fault and either the structured probable cause or the probable cause and the specific problem.

An alarm is uniquely identified by the unique fault condition parameters and the time of fault detection.

When an alarm is generated a corresponding active alarm is added to the active alarm list by the agent. The agent shall be able to provide such a list of active alarms to the manager when requested.

II.2 Alarm acknowledgement

The acknowledgement of an alarm is a maintenance function that aids the operator in his day-to-day management activity of his network. An alarm is acknowledged by the operator to indicate he has started the activity to resolve this specific problem. In general a human operator performs the acknowledgement, however a management system (NM or EM) may automatically acknowledge an alarm as well.

The alarm acknowledgement function requires that:

- all involved OSs have the same information about the alarms to be managed (including the current responsibility for alarm handling);
- all involved OSs have the capability to send and to receive acknowledgement messages associated to previous alarm reports.

II.3 Clearing of alarms

The alarms originated in consequence of faults need to be cleared. To clear an alarm it is generally necessary to repair the corresponding fault. The procedures to repair faults are implementation dependent and are therefore out of the scope of this Recommendation, however, in general:

- the equipment faults are repaired by replacing the faulty units with working ones;
- the software faults are repaired by means of partial or global system initializations, by means of software patches or by means of updated software loads;
- the communication faults are repaired by replacing the faulty transmission equipment or, in case of excessive noise, by removing the cause of the noise;
- the QoS faults are repaired either by removing the causes that degraded the QoS or by improving the capability of the system to react against the causes that could result in a degradation of the QoS;
- solving the environmental problem repairs the environment faults (high temperature, high humidity, etc.).

It is also possible that an ADAC fault is spontaneously repaired, without the intervention of the operator (e.g., a threshold crossed fault). In this case, the agent behaves as for the ADAC faults repaired by the operator.

In principle, the agent uses the same mechanisms to detect that a fault has been repaired, as for the detection of the occurrence of the fault. However, for ADMC faults, manual intervention by the operator is always necessary to clear the fault. Practically, various methods exist for the system to detect that a fault has been repaired and clear alarms and the faults that triggered them. For example:

- The system operator implicitly requests the agent to clear a fault, e.g., by initializing a new device that replaces a faulty one. Once the new device has been successfully put into service, the agent shall clear the fault(s). Consequently, the agent shall clear all related alarms.
- The system operator explicitly requests the clearing of one or more alarms. Once the alarm(s) has/have been cleared, the agent should reissue those alarms (as new alarms) in case the fault situation still persists.
- The agent detects the exchange of a faulty device by a new one and initializes it autonomously. Once the new device has been successfully put into service, the agent shall clear the fault(s). Consequently, the agent shall clear all related alarms.
- The agent detects that a previously reported threshold crossed alarm is no longer valid. It shall then clear the corresponding active alarm and the associated fault, without requiring any operator intervention. The details for the administration of thresholds and the exact condition for the agent to clear a threshold crossed alarm are implementation specific and depend on the definition of the threshold measurement, see also clause II.1.
- By definition, ADMC faults/alarms cannot be cleared by the agent autonomously. Therefore, system operator functions shall be available to request the clearing of ADMC alarms/faults in the agent. Once an ADMC alarm/fault has been cleared, the agent shall clear the associated ADAC fault/alarm.

Details of these mechanisms are system/implementation specific.

II.4 Fault recovery

After a fault has been detected and the replaceable faulty units/components have been identified, some management functions are necessary in order to perform system recovery and/or restoration, either automatically by the agent, or manually by the operator.

The fault recovery functions are used in various phases of fault management (FM):

1) Once a fault has been detected, the NE may be able to evaluate the effect of the fault on the telecommunication services and autonomously take recovery actions in order to minimize service degradation or disruption.
2) Once the faulty unit(s) has (have) been replaced or repaired, it shall be possible from the EM to put the previously faulty unit(s) back into service so that normal operation is restored. This transition should be done in such a way that the currently provided telecommunication services are not, or only minimally, disturbed.
3) At any time the NE may be able to perform recovery actions if requested by the operator. The operator may have several reasons to require such actions; e.g., he has deduced a faulty condition by analysing and correlating alarm reports, or he wants to verify that the NE is capable of performing the recovery actions (proactive maintenance).

The recovery actions that the NE performs (autonomously or on demand) in case of faults depend on the nature and severity of the faults, on the hardware and software capabilities of the NE and on the current configuration of the NE.

Faults are distinguished in two categories-software faults and hardware faults. In the case of software faults, depending on the severity of the fault, the recovery actions may be system initializations (at different levels), activation of a backup software load, activation of a fallback software load, download of a software unit, etc. In the case of hardware faults, the recovery actions depend on the existence and type of redundant (i.e., back-up) resources. Redundancy of some resources may be provided in the NE in order to achieve fault tolerance and to improve system availability. Data and configuration errors are treated similarly to software errors.

If the faulty resource has no redundancy, the recovery actions should be:

a) Isolate and remove from service the faulty resource so that it cannot disturb other working resources.

b) Remove from service the physical and functional resources (if any) which are dependent on the faulty one. This prevents the propagation of the fault effects to other fault-free resources.

c) Generate and forward appropriate notifications to inform the OS about all the changes performed.

If the faulty resource has redundancy, the NE should perform actions a) and c) above and, in addition, the recovery sequence that is specific to that type of redundancy. Several types of redundancy exist (e.g., hot standby, cold standby, duplex, symmetric/asymmetric, N plus one or N plus K, etc.), and for each one, there is a specific sequence of actions to be performed in case of failure. This Recommendation specifies the fault management aspects of the redundancies, but it does not define the specific recovery sequences of the redundancy types.

In the case of a failure of a resource providing service, the recovery sequence should start immediately upon detection of the failure. Before or during the changeover, a temporary and limited loss of service shall be acceptable. In the case of a management-initiated recovery command, the NE should perform the changeover without degradation of the telecommunication services.

The detailed definition of the management of the redundancies is out of the scope of this Recommendation.

Bibliography

[b-ITU-T M.3704] Recommendation ITU-T M.3704 (2010), *Common management service – Performance management – Protocol neutral requirements and analysis.*

[b-3GPP TS 32.101] 3GPP TS 32.101 V9.1.0 (2010), *Telecommunication management; Principles and high level requirements.*

[b-3GPP TS 32.102] 3GPP TS 32.102 V9.0.0 (2009), *Telecommunication management; Architecture.*

[b-3GPP TS 32.111-1] 3GPP TS 32.111-1 V9.0.0 (2009), *Telecommunication management; Fault Management; Part 1: 3G fault management requirements.*

[b-3GPP TS 32.111-2] 3GPP TS 32.111-2 V9.1.0 (2010), *Telecommunication management; Fault Management; Part 2: Alarm Integration Reference Point (IRP); Information Service (IS).*

[b-3GPP TS 32.150] 3GPP TS 32.150 V9.1.0 (2010), *Telecommunication management; Integration Reference Point (IRP) Concept and definitions.*

[b-3GPP TS 32.302] 3GPP TS 32.302 V9.0.0 (2009), *Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP; Information Service (IS).*

[b-3GPP TS 32.312] 3GPP TS 32.312 V9.0.0 (2009), *Telecommunication management; Generic Integration Reference Point (IRP) management; Information Service (IS).*

[b-3GPP TS 32.401] 3GPP TS 32.401 V9.0.0 (2009), *Telecommunication management; Performance Management (PM); Concept and requirements.*

[b-3GPP TS 32.622] 3GPP TS 32.622 V9.1.0 (2010), *Telecommunication management; Configuration Management (CM); Generic network resources Integration Reference Point (IRP): Network Resource Model (NRM).*

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	**Telecommunication management, including TMN and network maintenance**
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2011