3GPP TS 32.111-6 V16.0.0 (2020-07)

Technical Specification

3rd Generation Partnership Project;

Technical Specification Group Services and System Aspects;

Telecommunication management;

Fault management;

Part 6: Alarm Integration Reference Point (IRP):

Solution Set (SS) definitions

(Release 16)

* *

The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP..  
The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented.  
This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification.  
Specifications and reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices.

Keywords

UMTS, Log Management, Alarm Management, CORBA, XML, SOAP

***3GPP***

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis

Valbonne - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

***Copyright Notification***

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

UMTS™ is a Trade Mark of ETSI registered for the benefit of its members

3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  
LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners

GSM® and the GSM logo are registered and owned by the GSM Association

Contents

Foreword [5](#__RefHeading___Toc399143441)

Introduction [5](#__RefHeading___Toc399143442)

1 Scope [6](#__RefHeading___Toc399143443)

2 References [6](#__RefHeading___Toc399143444)

3 Definitions and abbreviations [7](#__RefHeading___Toc399143445)

3.1 Definitions [7](#__RefHeading___Toc399143446)

3.2 Abbreviations [7](#__RefHeading___Toc399143447)

4 Solution Set Definitions [7](#__RefHeading___Toc399143448)

Annex A (normative): CORBA Solution Set [9](#__RefHeading___Toc399143449)

A.0 General [9](#__RefHeading___Toc399143450)

A.1 Architectural Features [9](#__RefHeading___Toc399143451)

A.1.0 Introduction [9](#__RefHeading___Toc399143452)

A.1.1 Syntax for Distinguished Names [9](#__RefHeading___Toc399143453)

A.1.2 Notification Services [9](#__RefHeading___Toc399143454)

A.1.3 Push and Pull Style [9](#__RefHeading___Toc399143455)

A.1.4 Support multiple notifications in one push operation [9](#__RefHeading___Toc399143456)

A.1.5 AlarmIRPNotifications Interface [9](#__RefHeading___Toc399143457)

A.1.5.1 Method push (M) [10](#__RefHeading___Toc399143458)

A.1.6 Filter [10](#__RefHeading___Toc399143459)

A.2 Mapping [11](#__RefHeading___Toc399143460)

A.2.1 Operation and Notification mapping [11](#__RefHeading___Toc399143461)

A.2.2 Operation parameter mapping [12](#__RefHeading___Toc399143462)

A.2.3 Notification parameter mapping [18](#__RefHeading___Toc399143463)

A.3 Solution Set definitions [26](#__RefHeading___Toc399143464)

A.3.1 IDL definition structure [26](#__RefHeading___Toc399143465)

A.3.2 IDL specification "AlarmIRPConstDefs.idl" [26](#__RefHeading___Toc399143466)

A.3.3 IDL specification “AlarmIRPSystem.idl” [34](#__RefHeading___Toc399143467)

A.3.4 IDL specification "AlarmIRPNotifications.idl" [36](#__RefHeading___Toc399143468)

Annex B (normative): XML Definitions [41](#__RefHeading___Toc399143469)

B.0 General [41](#__RefHeading___Toc399143470)

B.1 Architectural Features [41](#__RefHeading___Toc399143471)

B.1.0 Introduction [41](#__RefHeading___Toc399143472)

B.1.1 Syntax for Distinguished Names [41](#__RefHeading___Toc399143473)

B.1.2 Notification Services [41](#__RefHeading___Toc399143474)

B.1.3 IOC Definitions [41](#__RefHeading___Toc399143475)

B.2 Mapping [41](#__RefHeading___Toc399143476)

B.3 Solution Set definitions [41](#__RefHeading___Toc399143477)

B.3.1 XML definition structure [41](#__RefHeading___Toc399143478)

B.3.2 Graphical Representation [42](#__RefHeading___Toc399143479)

B.3.3 XML Schema “alarmIRPNotif.xsd” [47](#__RefHeading___Toc399143480)

B.3.4 XML Schema “alarmIRPIOCs.xsd” [50](#__RefHeading___Toc399143481)

Annex C (normative): SOAP Solution Set [57](#__RefHeading___Toc399143482)

C.0 General [57](#__RefHeading___Toc399143483)

C.1 Architectural features [57](#__RefHeading___Toc399143484)

C.1.0 Introduction [57](#__RefHeading___Toc399143485)

C.1.1 Syntax for Distinguished Names [57](#__RefHeading___Toc399143486)

C.1.2 Notification Services [57](#__RefHeading___Toc399143487)

C.1.3 Supported W3C specifications [57](#__RefHeading___Toc399143488)

C.1.4 Prefixes and namespaces [57](#__RefHeading___Toc399143489)

C.2 Mapping [58](#__RefHeading___Toc399143490)

C.2.1 Operation and notification mapping [58](#__RefHeading___Toc399143491)

C.2.2 Operation parameter mapping [59](#__RefHeading___Toc399143492)

C.2.2.1 Operation acknowledgeAlarms [59](#__RefHeading___Toc399143493)

C.2.2.1.1 Input parameters [59](#__RefHeading___Toc399143494)

C.2.2.1.2 Output parameters [59](#__RefHeading___Toc399143495)

C.2.2.1.3 Fault definition [59](#__RefHeading___Toc399143496)

C.2.2.2 Operation getAlarmList [59](#__RefHeading___Toc399143497)

C.2.2.2.1 Input parameters [59](#__RefHeading___Toc399143498)

C.2.2.2.2 Output parameters [60](#__RefHeading___Toc399143499)

C.2.2.2.3 Fault definition [60](#__RefHeading___Toc399143500)

C.2.2.3 Operation getAlarmCount [60](#__RefHeading___Toc399143501)

C.2.2.3.1 Input parameters [60](#__RefHeading___Toc399143502)

C.2.2.3.2 Output parameters [60](#__RefHeading___Toc399143503)

C.2.2.3.3 Fault definition [61](#__RefHeading___Toc399143504)

C.2.2.4 Operation unacknowledgeAlarms [61](#__RefHeading___Toc399143505)

C.2.2.4.1 Input parameters [61](#__RefHeading___Toc399143506)

C.2.2.4.2 Output parameters [61](#__RefHeading___Toc399143507)

C.2.2.4.3 Fault definition [61](#__RefHeading___Toc399143508)

C.2.2.5 Operation setComment [62](#__RefHeading___Toc399143509)

C.2.2.5.1 Input parameters [62](#__RefHeading___Toc399143510)

C.2.2.5.2 Output parameters [62](#__RefHeading___Toc399143511)

C.2.2.5.3 Fault definition [62](#__RefHeading___Toc399143512)

C.2.2.6 Operation clearAlarms [62](#__RefHeading___Toc399143513)

C.2.2.6.1 Input parameters [62](#__RefHeading___Toc399143514)

C.2.2.6.2 Output parameters [63](#__RefHeading___Toc399143515)

C.2.2.6.3 Fault definition [63](#__RefHeading___Toc399143516)

C.3 Solution Set definitions [64](#__RefHeading___Toc399143517)

C.3.1 WSDL definition structure [64](#__RefHeading___Toc399143518)

C.3.2 Graphical Representation [64](#__RefHeading___Toc399143519)

C.3.3 WSDL specification “AlarmIRPSystem.wsdl” [65](#__RefHeading___Toc399143520)

Annex D (informative): Change history [73](#__RefHeading___Toc399143521)

# Foreword

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

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

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

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

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

# Introduction

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

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

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

**32.111-6 "Fault Management; Alarm Integration Reference Point (IRP); Solution Set (SS) definitions".**

The present document is part of a TS-family, which describes the requirements and information model necessary for Telecommunications Management (TM). The TM principles and TM architecture are specified in 3GPP TS 32.101 [1] and 3GPP TS 32.102 [2].

A communications system is composed of a multitude of Network Elements (NE) of various types and, typically, different vendors, which inter-operate in a coordinated manner in order to satisfy the network users' communication requirements.

The occurrence of faults in an NE may cause deterioration or loss of this NE's function. Fault Management is the functional area, which allows the operator to detect the occurrence of faults in the network in real-time. Configuration Management and Performance Management are two more functional areas, which require the operator to be alerted to certain conditions in the network.

A standard general-purpose mechanism for the management of logs containing selected notifications from the network is required to provide an ability to perform historical analysis on faults and conditions, which occurred in the network. The TS 32.33x-series, constituting the Notification Log IRP, sets forth such a mechanism - and Annex B of the present document contains the XML definition related to Alarm IRP notifications.

# 1 Scope

The present document contains the Solution Sets for the IRP whose semantics is specified in Alarm IRP: Information Service (TS 32.111-2 [4]).

These Solution Set specifications are related to 3GPP TS 32.111-2 V14. 0.X [4].

# 2 References

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

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

- For a specific reference, subsequent revisions do not apply.

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

[1] 3GPP TS 32.101: "Telecommunication management; Principles and high level requirements".

[2] 3GPP TS 32.102: "Telecommunication management; Architecture".

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

[4] 3GPP TS 32.111-2: "Telecommunication management; Fault Management; Part 2: Alarm Integration Reference Point: Information Service (IS)".

[5] 3GPP TS 32.150: "Telecommunication management; Integration Reference Point (IRP) Concept and definitions".

[6] 3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects".

[7] 3GPP TS 32.302: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Information Service (IS)".

[8] 3GPP TS 32.306: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Solution Set (SS) definitions".

[9] 3GPP TS 32.311: "Telecommunication management; Generic Integration Reference Point (IRP) management; Requirements".

[10] 3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP) management; Information Service (IS)".

[11] 3GPP TS 32.316: "Telecommunication management; Generic Integration Reference Point (IRP) management: Solution Set (SS) definitions".

[12] 3GPP TS 32.331: "Telecommunication management; Notification Log (NL) Integration Reference Point (IRP): Requirements".

[13] 3GPP TS 32.336: "Telecommunication management; Notification Log (NL) Integration Reference Point (IRP): Solution Set (SS) definitions".

[14] OMG TC Document telecom/98-11-01: "OMG Notification Service". <http://www.omg.org/technology/documents/>

[15] OMG CORBA Services: "Common Object Services Specification, Update: November 22, 1996" (Clause 4 contains the Event Service specification). <http://www.omg.org/technology/documents/>

[16] W3C SOAP 1.1 specification (<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>)

[17] W3C SOAP 1.2 specification (<http://www.w3.org/TR/soap12-part1/>)

[18] W3C WSDL 1.1 specification (<http://www.w3.org/TR/2001/NOTE-wsdl-20010315>)

[19] W3C XPath 1.0 specification (<http://www.w3.org/TR/1999/REC-xpath-19991116>)

# 3 Definitions and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 32.101 [1], 3GPP TS 32.102 [2], 3GPP TS 32.111-1 [3], 3GPP TS 32.111-2 [4], 3GPP TS 32.150 [5] and 3GPP TS 32.331 [12] apply.

**IRP document version number string**

The IRP document version number (sometimes called “IRP version” or “version number”) string is used to identify the present document. The definition of “IRP document version number string” in 3GPP TS 32.311 [9] provides the rule to derive such a string.

This string is used for the return value of get\_alarm\_irp\_versions(). It is used as return value of get\_notification\_categories() if the Notification IRP supports the emission of notifications defined by this Alarm IRP version. It is also used in the domain\_name attribute of a structured event carrying alarm information defined by this Alarm IRP version.

## 3.2 Abbreviations

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

CORBA Common Object Request Broker Architecture

IDL Interface Definition Language

IRP Integration Reference Point

IS Information Service

MOC Managed Object Class

MOI Managed Object Instance

NE Network Element

NL Notification Log

OMG Object Management Group

SS Solution Set

TMN Telecommunications Management Network

UML Unified Modelling Language

WSDL Web Service Description Language

XML eXtensible Markup Language

# 4 Solution Set Definitions

This specification defines the following 3GPP Alarm IRP Solution Set Definitions:

Annex A provides the CORBA Solution Set.

Annex B provides the XML Definitions.

Annex C provides the SOAP Solution Set.

Annex A (normative):  
CORBA Solution Set

# A.0 General

This annex contains the CORBA Solution Set for the IRP whose semantics is specified in Alarm IRP: Information Service (TS 32.111-2 [4]).

# A.1 Architectural Features

## A.1.0 Introduction

The overall architectural feature of Alarm IRP is specified in 3G TS 32.111-2 [4]. This clause specifies features that are specific to the CORBA SS.

## A.1.1 Syntax for Distinguished Names

The syntax of a Distinguished Name is defined in 3GPP TS 32.300 [6].

## A.1.2 Notification Services

In implementations of CORBA SS, IRPAgent conveys Alarm Information to IRPManager via OMG Notification Service (OMG Notification Service [14]).

OMG Event Service [15] provides event routing and distribution capabilities. OMG Notification Service provides, in addition to Event Service, event filtering and Quality Of Service (QOS) as well.

A necessary and sufficient sub set of OMG Notification Services shall be used to support AlarmIRPNotifications notifications as specified in 3G TS 32.111-2 [4].

## A.1.3 Push and Pull Style

OMG Notification Service defines two styles of interaction. One is called push style. In this style, IRPAgent pushes notifications to IRPManager as soon as they are available. The other is called pull style. In this style, IRPAgent keeps the notifications till IRPManager requests for them.

This CORBA SS specifies that support of Push style is Mandatory (M) and that support of Pull style is Optional (O).

## A.1.4 Support multiple notifications in one push operation

For efficiency reasons, IRPAgent may send multiple notifications using one single push operation. To pack multiple notifications into one push operation, IRPAgent may wait and not invoke the push operation as soon as notifications are available. To avoid IRPAgent to wait for an extended period of time that is objectionable to IRPManager, IRPAgent shall implement an IRPAgent wide timer configurable by administrator. On expiration of this timer, IRPAgent shall invoke push if there is at least one notification to be conveyed to IRPManager. This timer is re-started after each push invocation.

## A.1.5 AlarmIRPNotifications Interface

OMG CORBA Notification push operation is used to realise the notification of AlarmIRPNotifications. All the notifications in this interface are implemented using this push\_structured\_event method.

### A.1.5.1 Method push (M)

module CosNotifyComm {

…

Interface SequencePushConsumer : NotifyPublish {

void push\_structured\_events(

in CosNotification::EventBatch notifications)

raises( CosEventComm::Disconnected);

…

}; // SequencePushConsumer

…

}; // CosNotifyComm

NOTE 1: The push\_structured\_events method takes an input parameter of type EventBatch as defined in the OMG CosNotification module (OMG Notification Service [14]). This data type is the same as a sequence of Structured Events. Upon invocation, this parameter will contain a sequence of Structured Events being delivered to IRPManager by IRPAgent to which it is connected.

NOTE 2: The maximum number of events that will be transmitted within a single invocation of this operation is controlled by IRPAgent wide configuration parameter.

NOTE 3: The amount of time the supplier (IRPAgent) of a sequence of Structured Events will accumulate individual events into the sequence before invoking this operation is controlled by IRPAgent wide configuration parameter as well.

NOTE 4: IRPAgent may push EventBatch with only one Structured Event.

## A.1.6 Filter

IRPAgent shall optionally support alarm filtering based on IRPManager’s supplied alarm filter constraints (e.g. as parameter in subscribe() of 3G TS 32.302 [7]. Alarm filtering can be applied in the following cases:

- It is applicable to alarms emitted by IRPAgent via AlarmIRPNotifications. IRPManager supplies alarm filter constraint via the subscribe method. This filter is effective during the period of subscription.

- It is applicable to alarms returned by IRPAgent via the out parameter of get\_alarm\_list method. IRPManager supplies alarm filter constraint via the get\_alarm\_list method. This filter is effective only for this method invocation.

- It is applicable to the calculation of alarm counts returned by IRPAgent via the out parameters of get\_alarm\_count method. IRPManager supplies alarm filter constraint via the get\_alarm\_count method. This filter is effective only for this method invocation.

This SS shall use of filter constraint grammar specified by reference OMG Notification Service [14]. The name of the grammar is called "EXTENDED\_TCL". See clause 2.4, Default Filter Constraint Language in OMG Notification Service [14]. This SS shall use this grammar only.

# A.2 Mapping

## A.2.1 Operation and Notification mapping

Alarm IRP: IS 3G TS 32.111-2 [4] defines semantics of operation and notification visible across the Alarm IRP. Table A.2.1.1 indicates mapping of these operations and notifications to their equivalents defined in this SS.

Table A.2.1.1: Mapping from IS Notification/Operation to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation/ notification 3G TS 32.111-2 [4] | SS Method | Qualifier |
| acknowledgeAlarms | acknowledge\_alarms | M |
| unacknowledgeAlarms | unacknowledge\_alarms | O |
| getAlarmList | get\_alarm\_list | M |
| getIRPVersion (note) | get\_alarm\_irp\_versions | M |
| getAlarmCount | get\_alarm\_count | O |
| setComment | comment\_alarms | O |
| clearAlarms | clear\_alarms | O |
| getOperationProfile (note) | get\_alarm\_irp\_operations\_profile | O |
| getNotificationProfile (note) | get\_alarm\_irp\_notification\_profile | O |
| notifyNewAlarm | push\_structured\_event  Note that OMG Notification Service OMG Notification Service [14] defines this method.  See clause A.3.1 | M |
| notifyClearedAlarm | push\_structured\_event  See clause A.3.1 | M |
| notifyChangedAlarm | push\_structured\_event  See clause A.3.1 | M |
| notifyChangedAlarmGeneral | push\_structured\_event  See clause A.3.1 | O |
| notifyAckStateChanged | push\_structured\_event  See clause A.3.1 | M |
| notifyAlarmListRebuilt | push\_structured\_event  See clause A.3.1 | M |
| notifyComments | push\_structured\_event  See clause A.3.1 | O |
| notifyPotentialFaultyAlarmList | push\_structured\_event  See clause A.3.1 | O |
| notifyCorrelatedNotificationChanged | push\_structured\_event  See clause A.3.1 | O |
| NOTE: This operation is of ManagedGenericIRP IOC specified in [10]. The AlarmIRP IOC of [4] inherits from it. | | |

## A.2.2 Operation parameter mapping

Reference 3G TS 32.111-2 [4] defines semantics of parameters carried in operations across the Alarm IRP. The following set of tables indicates the mapping of these parameters, as per operation, to their equivalents defined in this SS.

Table A.2.2.1: Mapping from IS acknowledgeAlarms parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| alarmInformationAndSeverityReferenceList | AlarmIRPConstDefs::AlarmInformationIdAndSevSeq alarm\_information\_id\_and\_sev\_list  Note: perceivedSeverity is optional  { alarmId - Mandatory;  perceivedSeverity - Optional  } | M |
| ackUserId | string ack\_user\_id | M |
| ackSystemId | ManagedGenericIRPConstDefs::StringOpt ack\_system\_id | O |
| badAlarmInformationReferenceList | AlarmIRPConstDefs::BadAcknowledgeAlarmInfoSeq bad\_ack\_alarm\_info\_list | M |
| status | ManagedGenericIRPConstDefs::Signal  Exceptions:  AcknowledgeAlarms,  ManagedGenericIRPSystem::ParameterNotSupported,  ManagedGenericIRPSystem::InvalidParameter | M |

Table A.2.2.2: Mapping from IS unacknowledgeAlarms parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| alarmInformationReferenceList | AlarmIRPConstDefs::AlarmInformationIdSeq alarm\_information\_id\_list | M |
| ackUserId | string ack\_user\_id | M |
| ackSystemId | ManagedGenericIRPConstDefs::StringOpt ack\_system\_id | O |
| badAlarmInformationReferenceList | AlarmIRPConstDefs:: BadAlarmInformationIdSeq bad\_alarm\_information\_id\_list | M |
| status | ManagedGenericIRPConstDefs::Signal  Exceptions:  UnacknowledgeAlarms,  ManagedGenericIRPSystem::OperationNotSupported,  ManagedGenericIRPSystem::ParameterNotSupported,  ManagedGenericIRPSystem::InvalidParameter | M |

Table A.2.2.3: Mapping from IS getAlarmList parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| alarmAckState, filter | ManagedGenericIRPConstDefs::StringOpt filter | O |
| baseObjectClass, baseObjectInstance | AlarmIRPConstDefs::DNOpt base\_object | O  (Note 1) |
| alarmInformation List | Return value of type AlarmIRPConstDefs::AlarmInformationSeq (Note 2) | M |
| status | Exceptions:  GetAlarmList,  FilterComplexityLimit,  ManagedGenericIRPSystem::ParameterNotSupported,  ManagedGenericIRPSystem::InvalidParameter | M |
| Note 1: If notification notifyAlarmListRebuilt supports indicating that only a part of the alarm list has been rebuilt then this parameter shall be supported.  Note 2: Each Structured Event of AlarmIRPConstDefs::AlarmInformationSeq shall have the fields defined in Table A.2.2.4 or Table A.2.2.5. | | |

Table A.2.2.4: Definition of a Structured Event of AlarmIRPConstDefs::AlarmInformationSeq for alarms not related to security

| IS Parameters | OMG CORBA Structured Event attribute | Qualifier | Comment |
| --- | --- | --- | --- |
| There is no corresponding IS attribute. | domain\_name |  |  |
| notificationType | type\_name | M | The parameter carries   * - NOTIFY\_FM\_NEW\_ALARM of interface NotificationType of module AlarmIRPConstDefs in case the alarm has not yet changed and has not yet been cleared. * - NOTIFY\_FM\_CHANGED\_ALARM of interface NotificationType of module AlarmIRPConstDefs in case the alarm has changed but has not yet been cleared. * - NOTIFY\_FM\_CHANGED\_ALARM\_GENERAL of interface NotificationType of module AlarmIRPConstDefs in case the alarm has changed but has not yet been cleared. * - NOTIFY\_FM\_CLEARED\_ALARM of interface NotificationType of module AlarmIRPConstDefs in case the alarm has been cleared but not yet acknowledged. |
| alarmType | See Table A.2.3.1 | M | See Table A.2.3.1 |
| There is no corresponding SS attribute. | variable Header |  |  |
| objectClass, objectInstance | See Table A.2.3.1 | M | See Table A.2.3.1 |
| notificationId | See Table A.2.3.1 | M | See Table A.2.3.1 |
| eventTime | See Table A.2.3.1 | O | Value of NV pair is a IRPTime of module ManagedGenericIRPConstDefs.  The parameter carries the   * alarmRaisedTime in case notificationType carries NOTIFY\_FM\_NEW\_ALARM * alarmChangedTime in case notificationType carries NOTIFY\_FM\_CHANGED\_ALARM * alarmChangedTime in case notificationType carries NOTIFY\_FM\_CHANGED\_ALARM\_GENERAL * alarmClearedTime in case notificationType carries NOTIFY\_FM\_CLEARED\_ALARM |
| systemDN | See Table A.2.3.1 | M | See Table A.2.3.1 |
| alarmId | See Table A.2.3.1 | M | See Table A.2.3.1 |
| alarmRaisedTime | One NV pair of filterable\_ body\_fields | M | Name of NV pair is the ALARM\_RAISED\_TIME of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a IRPTime of module ManagedGenericIRPConstDefs. |
| alarmChangedTime | One NV pair of filterable\_ body\_fields | O | Name of NV pair is the ALARM\_CHANGED\_TIME of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a IRPTime of module ManagedGenericIRPConstDefs. |
| alarmClearedTime | One NV pair of filterable\_ body\_fields | M | present if related alarm was cleared; not present if related alarm was not cleared  Name of NV pair is the ALARM\_CLEARED\_TIME of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a IRPTime of module ManagedGenericIRPConstDefs. |
| probableCause | See Table A.2.3.1 | M | See Table A.2.3.1 |
| perceivedSeverity | See Table A.2.3.1 | M | See Table A.2.3.1 |
| rootCauseIndicator | See Table A.2.3.1 | O | See Table A.2.3.1 |
| specificProblem | See Table A.2.3.1 | O | See Table A.2.3.1 |
| backedUpStatus | See Table A.2.3.1 | O | See Table A.2.3.1 |
| trendIndication | See Table A.2.3.1 | O | See Table A.2.3.1 |
| thresholdInfo | See Table A.2.3.1 | O | See Table A.2.3.1 |
| stateChangeDefinition | See Table A.2.3.1 | O | See Table A.2.3.1 |
| monitoredAttributes | See Table A.2.3.1 | O | See Table A.2.3.1 |
| proposedRepairActions | See Table A.2.3.1 | O | See Table A.2.3.1 |
| additionalText | See Table A.2.3.1 | O | See Table A.2.3.1 |
| additionalInformation | See Table A.2.3.1 | O | See Table A.2.3.1 |
| ackTime | See Table A.2.3.3 | M | present if related alarm was acknowledged or unacknowledged; not present if related alarm was not acknowledged or unacknowledged  See Table A.2.3.3 |
| ackUserId | See Table A.2.3.3 | M | present if related alarm was acknowledged or unacknowledged; not present if related alarm was not acknowledged or unacknowledged  See Table A.2.3.3 |
| ackSystemId | See Table A.2.3.3 | O | optionally present if related alarm was acknowledged or unacknowledged; not present if related alarm was not acknowledged or unacknowledged  See Table A.2.3.3 |
| ackState | See Table A.2.3.3 | M | present if related alarm was acknowledged or unacknowledged; not present if related alarm was not acknowledged or unacknowledged  See Table A.2.3.3 |
| clearUserId | See Table A.2.3.4 | O | optionally present if related alarm was cleared; not present if related alarm was not cleared  See Table A.2.3.4 |
| clearSystemId | See Table A.2.3.4 | O | optionally present if related alarm was cleared; not present if related alarm was not cleared  See Table A.2.3.4 |
| backUpObject | See Table A.2.3.1 | O | See Table A.2.3.1 |
| correlatedNotifications | See Table A.2.3.1 | O | See Table A.2.3.1 |
| comments | See Table A.2.3.7 | M | present if related alarm was commented; not present if related alarm was not commented  See Table A.2.3.7 |
| There is no corresponding IS attribute. | remaining\_ body |  |  |

Table A.2.2.5: Definition of a Structured Event of AlarmIRPConstDefs::AlarmInformationSeq for alarms related to security

| IS Parameters | OMG CORBA Structured Event attribute | Qualifier | Comment |
| --- | --- | --- | --- |
| There is no corresponding IS attribute. | domain\_name |  |  |
| notificationType | type\_name | M | See Table A.2.3.1  The parameter carries   * NOTIFY\_FM\_NEW\_ALARM of interface NotificationType of module AlarmIRPConstDefs in case the alarm has not yet changed and has not yet been cleared. * NOTIFY\_FM\_CHANGED\_ALARM of interface NotificationType of module AlarmIRPConstDefs in case the alarm has changed but has not yet been cleared. * NOTIFY\_FM\_CHANGED\_ALARM\_GENERAL of interface NotificationType of module AlarmIRPConstDefs in case the alarm has changed but has not yet been cleared. * NOTIFY\_FM\_CLEARED\_ALARM of interface NotificationType of module AlarmIRPConstDefs in case the alarm has been cleared but not yet acknowledged. |
| alarmType | See Table A.2.3.1 | M | See Table A.2.3.1 |
| There is no corresponding SS attribute. | variable Header |  |  |
| objectClass, objectInstance | See Table A.2.3.1 | M | See Table A.2.3.1 |
| notificationId | See Table A.2.3.1 | M | See Table A.2.3.1 |
| eventTime | See Table A.2.3.1 | O | See Table A.2.3.1  The parameter carries the   * alarmRaisedTime in case notificationType carries NOTIFY\_FM\_NEW\_ALARM * alarmChangedTime in case notificationType carries NOTIFY\_FM\_CHANGED\_ALARM * alarmChangedTime in case notificationType carries NOTIFY\_FM\_CHANGED\_ALARM\_GENERAL * alarmClearedTime in case notificationType carries NOTIFY\_FM\_CLEARED\_ALARM |
| systemDN | See Table A.2.3.1 | M | See Table A.2.3.1 |
| alarmId | See Table A.2.3.1 | M | See Table A.2.3.1 |
| alarmRaisedTime | One NV pair of filterable\_ body\_fields | M | Name of NV pair is the ALARM\_RAISED\_TIME of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a IRPTime of module ManagedGenericIRPConstDefs. |
| alarmChangedTime | One NV pair of filterable\_ body\_fields | O | Name of NV pair is the ALARM\_CHANGED\_TIME of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a IRPTime of module ManagedGenericIRPConstDefs. |
| alarmClearedTime | One NV pair of filterable\_ body\_fields | M | present if related alarm was cleared; not present if related alarm was not cleared  Name of NV pair is the ALARM\_CLEARED\_TIME of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a IRPTime of module ManagedGenericIRPConstDefs. |
| probableCause | See Table A.2.3.1 | M | See Table A.2.3.1 |
| perceivedSeverity | See Table A.2.3.1 | M | See Table A.2.3.1 |
| rootCauseIndicator | See Table A.2.3.1 | O | See Table A.2.3.1 |
| specificProblem | See Table A.2.3.1 | O | See Table A.2.3.1 |
| additionalText | See Table A.2.3.1 | O | See Table A.2.3.1 |
| additionalInformation | See Table A.2.3.1 | O | See Table A.2.3.1 |
| ackTime | See Table A.2.3.3 | M | present if related alarm was acknowledged or unacknowledged; not present if related alarm was not acknowledged or unacknowledged  See Table A.2.3.3 |
| ackUserId | See Table A.2.3.3 | M | present if related alarm was acknowledged or unacknowledged; not present if related alarm was not acknowledged or unacknowledged  See Table A.2.3.3 |
| ackSystemId | See Table A.2.3.3 | O | optionally present if related alarm was acknowledged or unacknowledged; not present if related alarm was not acknowledged or unacknowledged  See Table A.2.3.3 |
| ackState | See Table A.2.3.3 | M | present if related alarm was acknowledged or unacknowledged; not present if related alarm was not acknowledged or unacknowledged  See Table A.2.3.3 |
| clearUserId | See Table A.2.3.4 | O | optionally present if related alarm was cleared; not present if related alarm was not cleared  See Table A.2.3.4 |
| clearSystemId | See Table A.2.3.4 | O | optionally present if related alarm was cleared; not present if related alarm was not cleared  See Table A.2.3.4 |
| correlatedNotifications | See Table A.2.3.1 | O | See Table A.2.3.1 |
| comments | See Table A.2.3.7 | M | present if related alarm was commented; not present if related alarm was not commented  See Table A.2.3.7 |
| serviceUser | See Table A.2.3.2 | M | See Table A.2.3.2 |
| serviceProvider | See Table A.2.3.2 | M | See Table A.2.3.2 |
| securityAlarmDetector | See Table A.2.3.2 | M | See Table A.2.3.2 |
| There is no corresponding IS attribute. | remaining\_ body |  |  |

Table A.2.2.6: Mapping from IS getAlarmCount parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| alarmAckState, filter | ManagedGenericIRPConstDefs::StringOpt filter | O |
| criticalCount, majorCount, minorCount, warningCount, indeterminateCount, clearedCount | long critical\_count, long major\_count, long minor\_count, long warning\_count, long indeterminate\_count, long cleared\_count | M |
| status | Exceptions:  GetAlarmCount,  FilterComplexityLimit,  ManagedGenericIRPSystem::OperationNotSupported,  ManagedGenericIRPSystem::ParameterNotSupported,  ManagedGenericIRPSystem::InvalidParameter | M |

Table A.2.2.7: Mapping from IS getIRPVersion parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| versionNumberSet | Return value of type ManagedGenericIRPConstDefs::VersionNumberSet | M |
| status | Exceptions:  GetAlarmIRPVersions | M |

Table A.2.2.8: Mapping from IS setComment parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| alarmInformationReferenceList | AlarmIRPConstDefs::AlarmInformationIdSeq alarm\_information\_id\_list | M |
| commentUserId | string comment\_user\_id | M |
| commentSystemId | ManagedGenericIRPConstDefs::StringOpt comment\_system\_id | O |
| commentText | string comment\_text | M |
| badAlarmInformationReferenceList | AlarmIRPConstDefs::BadAlarmInformationIdSeq bad\_alarm\_information\_id\_list | M |
| status | ManagedGenericIRPConstDefs::Signal  Exceptions:  CommentAlarms,  ManagedGenericIRPSystem::OperationNotSupported,  ManagedGenericIRPSystem::ParameterNotSupported,  ManagedGenericIRPSystem::InvalidParameter | M |

Table A.2.2.9: Mapping from IS getOperationProfile parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| irpVersion | ManagedGenericIRPConstDefs::VersionNumber alarm\_irp\_version | M |
| operationNameProfile, operationParameterProfile | Return value of type ManagedGenericIRPConstDefs::MethodList | M |
| status | Exceptions:  GetAlarmIRPOperationsProfile,  ManagedGenericIRPSystem::OperationNotSupported,  ManagedGenericIRPSystem::InvalidParameter | M |

Table A.2.2.10: Mapping from IS getNotificationProfile parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| irpVersion | ManagedGenericIRPConstDefs::VersionNumber alarm\_irp\_version | M |
| notificationNameProfile, notificationParameterProfile | Return value of type ManagedGenericIRPConstDefs::MethodList | M |
| status | Exceptions:  GetAlarmIRPNotificationProfile,  ManagedGenericIRPSystem::OperationNotSupported,  ManagedGenericIRPSystem::InvalidParameter | M |

Table A.2.2.11: Mapping from IS clearAlarms parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Method parameter | Qualifier |
| alarmInformationReferenceList | AlarmIRPConstDefs::AlarmInformationIdSeq alarm\_information\_id\_list | M |
| clearUserId | string clear\_user\_id | M |
| clearSystemId | string clear\_system\_id | O |
| badAlarmInformationReferenceList | AlarmIRPConstDefs:: BadAlarmInformationIdSeq bad\_alarm\_information\_id\_list | M |
| status | ManagedGenericIRPConstDefs::Signal  Exceptions:  ClearAlarms,  ManagedGenericIRPSystem::OperationNotSupported,  ManagedGenericIRPSystem::ParameterNotSupported,  ManagedGenericIRPSystem::InvalidParameter | M |

## A.2.3 Notification parameter mapping

Reference 3G TS 32.111-2 [4] defines semantics of parameters carried in notifications. The following tables indicate the mapping of these parameters to their OMG CORBA Structured Event (defined in OMG Notification Service [14]) equivalents. The composition of OMG Structured Event, as defined in the OMG Notification Service [14], is:

Header

Fixed Header

domain\_name

type\_name

event\_name

Variable Header

Body

filterable\_body\_fields

remaining\_body

The following tables list all OMG Structured Event attributes in the second column. The first column identifies the Alarm IRP: IS [4] defined notification parameters.

Table A.2.3.1: Mapping for notifyNewAlarm (to carry non-security-related alarms)

| IS Parameters | OMG CORBA Structured Event attribute | Qualifier | Comment |
| --- | --- | --- | --- |
| There is no corresponding SS attribute. | domain\_name |  | It carries the IRP document version number string. See sub-clause 3.1.  It indicates the syntax and semantics of the Structured Event as defined by this specification. |
| notificationType | type\_name | M | This is the NOTIFY\_FM\_NEW\_ALARM of interface NotificationType of module AlarmIRPConstDefs. |
| alarmType | event\_name | M | It identifies one of the following:  communications alarm,  processing error alarm,  environmental alarm,  quality of service alarm and  equipment alarm.  It is a string defined by interface AlarmType of module AlarmIRPConstDefs. |
| There is no corresponding SS attribute. | variable Header |  |  |
| objectClass, objectInstance | One NV pair of filterable\_body\_fields | M | NV stands for name-value pair. Order arrangement of NV pairs is not significant. The name of NV-pair is always encoded in string.  Name of NV pair is the MANAGED\_OBJECT\_INSTANCE of interface AttributeNameValue of module NotificationIRPConstDefs.  Value of NV pair is a string. |
| notificationId | One NV pair of remaining\_body | M | Name of NV pair is the NOTIFICATION\_ID of interface AttributeNameValue of module NotificationIRPConstDefs.  Value of NV pair is a long. |
| eventTime | One NV pair of filterable\_body\_fields | M | Name of NV pair is the EVENT\_TIME of interface AttributeNameValue of module NotificationIRPConstDefs.  Value of NV pair is a IRPTime of module ManagedGenericIRPConstDefs. |
| systemDN | One NV pair of filterable\_body\_fields | M | Name of NV pair is the SYSTEM\_DN of interface AttributeNameValue of module NotificationIRPConstDefs.  Value of NV pair is a string. |
| probableCause | One NV pair of filterable\_body\_fields | M | Name of NV pair is the PROBABLE\_CAUSE of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a short defined by interface ProbableCause of module AlarmIRPConstDefs. |
| perceivedSeverity | One NV pair of filterable\_body\_fields | M | Name of NV pair is the PERCEIVED\_SEVERITY of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a short defined by interface PerceivedSeverity of module AlarmIRPConstDefs. |
| rootCauseIndicator | One NV pair of filterable\_body\_fields | O | Name of NV pair is the ROOT\_CAUSE\_INDICATOR of interface AttributeNameValue of module AlarmIRPConstDefs.    Value of NV pair is a boolean defined by interface RootCauseIndicator of module AlarmIRPConstDefs. |
| specificProblem | One NV pair of remaining\_body | O | Name of NV pair is the SPECIFIC\_PROBLEM of interface AttributeNameValue of module AlarmIRPConstDefs.    Value of NV pair is a string. |
| correlatedNotifications | One NV pair of remaining\_body | O | Name of NV pair is the CORRELATED\_NOTIFICATIONS of interface AttributeNameValue.  Value of NV pair is a CorrelatedNotificationSet of module AlarmIRPConstDefs. |
| backedUpStatus | One NV pair of remaining\_body | O | Name of NV pair is the BACKED\_UP\_STATUS of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a boolean BackedUpStatus of module AlarmIRPConstDefs. |
| backUpObject | One NV pair of remaining\_body | O | Name of NV pair is the BACK\_UP\_OBJECT of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a string carrying of DN of the back-up object. See 3G TS 32.300 [6] for the DN string representation. |
| trendIndication | One NV pair of remaining\_body | O | Name of NV pair is the TREND\_INDICATION of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is an enum TrendIndication of module AlarmIRPConstDefs. |
| thresholdInfo | One NV pair of remaining\_body | O | Name of NV pair is the THRESHOLD\_INFO of interface ParameterNameValue of module AlarmIRPConstDefs.  Value of NV pair is a ThresholdInfo of module AlarmIRPConstDefs. |
| stateChangeDefinition | One NV pair of remaining\_body | O | Name of NV pair is the STATE\_CHANGE\_DEFINITION of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is an AttributeChangeSet of module AlarmIRPConstDefs. |
| monitoredAttributes | One NV pair of remaining\_body | O | Name of NV pair is the MONITORED\_ATTRIBUTES of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is an AttributeSet of module AlarmIRPConstDefs. |
| proposedRepairActions | One NV pair of remaining\_body | O | Name of NV pair is the PROPOSED\_REPAIR\_ACTIONS of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a string. |
| additionalText | One NV pair of remaining\_body | O | Name of NV pair is the ADDITIONAL\_TEXT of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a string. |
| additionalInformation | One or more NV pairs of remaining\_body | O | 3GPP defines two Additional Information (AI) NV pairs for carrying vendor-specific (VS) perceived severity and alarm type.  The names of these two NV pairs are:   * AI\_VS\_PERCEIVED\_SEVERITY * AI\_VS\_ALARM\_TYPE   of interface AdditionalInformation of module AlarmIRPConstDefs.  The values of these two NV pairs are vendor defined.  Other AI NV pairs are permitted to capture specific information.  To be easily identified as vendor specific, the name value of these NV pairs should take the form of "ai\_vs\_<non-empty string>".  The value of this NV pair is vendor defined. |
| alarmId | One NV pair of remaining\_body | M | Name of NV pair is the ALARM\_ID of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a string.  If the string is a zero-length string or if this NV pair is absent, the default semantics is that alarmId is a concatenation of managedObjectInstance, eventType, probableCause and specificProblem, if present, of this Structured Event. Since probableCause is encoded as a short, it shall be converted into string before concatenation. The resultant string shall not contain spaces. |

Table A.2.3.2: Mapping for notifyNewAlarm (to carry security alarm)

| IS Parameters | OMG CORBA Structured Event attribute | Qualifier | Comment |
| --- | --- | --- | --- |
| There is no corresponding SS attribute. | domain\_name |  | It carries the IRP document version number string. See sub-clause 3.1.  It indicates the syntax and semantics of the Structured Event as defined by this specification. |
| notificationType | type\_name | M | This is the NOTIFY\_FM\_NEW\_ALARM of interface NotificationType of module AlarmIRPConstDefs. |
| alarmType | event\_name | M | It identifies one of the following:  Integrity violation, operational violation, physical violation, security violation and time domain violation.  It is a string defined by interface AlarmType of module AlarmIRPConstDefs. |
| There is no corresponding SS attribute. | variable Header |  |  |
| objectClass, objectInstance | One NV pair of filterable\_body\_fields | M | NV stands for name-value pair. Order arrangement of NV pairs is not significant. The name of NV-pair is always encoded in string.  Name of NV pair is the MANAGED\_OBJECT\_INSTANCE of interface AttributeNameValue of module NotificationIRPConstDefs.  Value of NV pair is a string. |
| notificationId | One NV pair of remaining\_body | M | Name of NV pair is the NOTIFICATION\_ID of interface AttributeNameValue of module NotificationIRPConstDefs.  Value of NV pair is a long. |
| eventTime | One NV pair of filterable\_body\_fields | M | Name of NV pair is the EVENT\_TIME of interface AttributeNameValue of module NotificationIRPConstDefs.  Value of NV pair is a IRPTime of module ManagedGenericIRPConstDefs. |
| systemDN | One NV pair of filterable\_body\_fields | M | Name of NV pair is the SYSTEM\_DN of interface AttributeNameValue of module NotificationIRPConstDefs.  Value of NV pair is a string. |
| probableCause | One NV pair of filterable\_body\_fields | M | Name of NV pair is the PROBABLE\_CAUSE of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a short defined by interface ProbableCause of module AlarmIRPConstDefs. |
| perceivedSeverity | One NV pair of filterable\_body\_fields | M | Name of NV pair is the PERCEIVED\_SEVERITY of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a short defined by interface PerceivedSeverity of module AlarmIRPConstDefs. |
| rootCauseIndicator | One NV pair of filterable\_body\_fields | O | Name of NV pair is the ROOT\_CAUSE\_INDICATOR of interface AttributeNameValue.  Value of NV pair is a boolean defined by interface RootCauseIndicator of module AlarmIRPConstDefs. |
| correlatedNotifications | One NV pair of remaining\_body | O | Name of NV pair is the CORRELATED\_NOTIFICATIONS of interface AttributeNameValue.  Value of NV pair is a CorrelatedNotificationSet of module AlarmIRPConstDefs. |
| additionalText | One NV pair of remaining\_body | O | Name of NV pair is the ADDITIONAL\_TEXT of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a string. |
| additionalInformation | One or more NV pairs of remaining\_body | O | 3GPP defines two Additional Information (AI) NV pairs for carrying vendor-specific (VS) perceived severity and alarm type.  The names of these two NV pairs are:   * AI\_VS\_PERCEIVED\_SEVERITY * AI\_VS\_ALARM\_TYPE   of interface AdditionalInformation of module AlarmIRPConstDefs.  The values of these two NV pairs are vendor defined.  Other AI NV pairs are permitted to capture specific information.  To be easily identified as vendor specific, the name value of these NV pairs should take the form of "ai\_vs\_<non-empty string>".  The value of this NV pair is vendor defined. |
| alarmId | One NV pair of remaining\_body | M | Name of NV pair is the ALARM\_ID of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a string.  If the string is a zero-length string or if this NV pair is absent, the default semantics is that alarmId is a concatenation of managedObjectInstance, eventType, probableCause and specificProblem, if present, of this Structured Event. Since probableCause is encoded as a short, it shall be converted into string before concatenation. The resultant string shall not contain spaces. |
| serviceUser | One NV pair of remaining\_body | M | Name of NV pair is the SERVICE\_USER of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a string. |
| serviceProvider | One NV pair of remaining\_body | M | Name of NV pair is the SERVICE\_PROVIDER of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a string. |
| securityAlarmDetector | One NV pair of remaining\_body | M | Name of NV pair is the SECURITY\_ALARM\_DETECTOR of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a string. |

Table A.2.3.3: Mapping for notifyAckStateChanged

| IS Parameters | OMG CORBA Structured Event attribute | Qualifier | Comment |
| --- | --- | --- | --- |
| There is no corresponding IS attribute. | domain\_name |  | See that of notifyNewAlarm. |
| notificationType | type\_name | M | This is the NOTIFY\_FM\_ACK\_STATE\_CHANGED of interface NotificationType of module AlarmIRPConstDefs. |
| alarmType | event\_name | M | See that of notifyNewAlarm. |
| There is no corresponding IS attribute. | variable Header |  |  |
| objectClass, objectInstance | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| notificationId | One NV pair of remaining\_body | M | See that of notifyNewAlarm. |
| eventTime | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| systemDN | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| probableCause | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| perceivedSeverity | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| alarmId | One NV pair of remaining\_body | M | See that of notifyNewAlarm. |
| ackUserId | One NV pair of remaining\_body | M | Name of NV pair is the ACK\_USER\_ID of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a string. |
| ackSystemId | One NV pair of remaining\_body | O | Name of NV pair is the ACK\_SYSTEM\_ID of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a string. |
| ackState | One NV pair of remaining\_body | M | Name of NV pair is the ACK\_STATE of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a short defined by interface AckState of module AlarmIRPConstDefs. |

Table A.2.3.4: Mapping for notifyClearedAlarm

|  |  |  |  |
| --- | --- | --- | --- |
| IS Parameters | OMG CORBA Structured Event attribute | Qualifier | Comment |
| There is no corresponding IS attribute. | domain\_name |  | See that of notifyNewAlarm. |
| notificationType | type\_name | M | This is the NOTIFY\_FM\_CLEARED\_ALARM of interface NotificationType of module AlarmIRPConstDefs. |
| alarmType | event\_name | M | See that of notifyNewAlarm. |
| There is no corresponding IS attribute. | variable Header |  |  |
| objectClass, objectInstance | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| notificationId | One NV pair of remaining\_body | M | See that of notifyNewAlarm. |
| eventTime | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| systemDN | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| probableCause | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| perceivedSeverity | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| correlatedNotifications | -- | -- | See Note. |
| alarmId | One NV pair of remaining\_body | M | See that of notifyNewAlarm. |
| clearUserId | One NV pair of remaining\_body | O | Name of NV pair is the CLEAR\_USER\_ID of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a string. |
| clearSystemId | One NV pair of remaining\_body | O | Name of NV pair is the CLEAR\_SYSTEM\_ID of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a string. |
| NOTE: In the CORBA Solution Set the correlatedNotifications is not used. In the CORBA Solution Set, one notifyClearedAlarm notification can only clear a single alarmInformation. | | | |

Table A.2.3.5: Mapping for notifyAlarmListRebuilt

|  |  |  |  |
| --- | --- | --- | --- |
| IS Parameters | OMG CORBA Structured Event attribute | Qualifier | Comment |
| There is no corresponding IS attribute. | domain\_name |  | See that of notifyNewAlarm. |
| notificationType | type\_name | M | This is the NOTIFY\_FM\_ALARM\_LIST\_REBUILT of interface NotificationType of module AlarmIRPConstDefs. |
| There is no corresponding IS attribute. | event\_name | M | Carry an empty string. |
| There is no corresponding IS attribute. | variable Header |  |  |
| objectClass, objectInstance | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| notificationId | One NV pair of remaining\_body | M | See that of notifyNewAlarm. |
| eventTime | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| systemDN | One NV pair of filterable\_body\_fields | O | See that of notifyNewAlarm. |
| reason | One NV pair of remaining\_body | M | Name of NV pair is the REASON of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a string. |
| alarmListAlignmentRequirement | One NV pair of remaining\_body | O | Name of NV pair is the ALARM\_LIST\_ALIGNMENT\_REQUIREMENT of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is an enum AlarmListAlignmentRequirement of module AlarmIRPConstDefs. |

Table A.2.3.6: Mapping for notifyChangedAlarm

|  |  |  |  |
| --- | --- | --- | --- |
| IS Parameters | OMG CORBA Structured Event attribute | Qualifier | Comment |
| There is no corresponding IS attribute. | domain\_name |  | See that of notifyNewAlarm. |
| notificationType | type\_name | M | This is the NOTIFY\_FM\_CHANGED\_ALARM of interface NotificationType of module AlarmIRPConstDefs. |
| alarmType | event\_name | M | See that of notifyNewAlarm. |
| There is no corresponding IS attribute. | variable Header |  |  |
| objectClass, objectInstance | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| notificationId | One NV pair of remaining\_body | M | See that of notifyNewAlarm. |
| eventTime | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| systemDN | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| probableCause | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| perceived Severity | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| alarmId | One NV pair of remaining\_body | M | See that of notifyNewAlarm. |

Table A.2.3.7: Mapping for notifyComments

|  |  |  |  |
| --- | --- | --- | --- |
| IS Parameters | OMG CORBA Structured Event attribute | Qualifier | Comment |
| There is no corresponding IS attribute. | domain\_name |  | See that of notifyNewAlarm. |
| notificationType | type\_name | M | This is the NOTIFY\_FM\_COMMENT\_ADDED of interface NotificationType of module AlarmIRPConstDefs. |
| alarmType | event\_name | M | See that of notifyNewAlarm. |
| There is no corresponding IS attribute. | variable Header |  |  |
| objectClass, objectInstance | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| notificationId | One NV pair of remaining\_body | M | See that of notifyNewAlarm. |
| eventTime | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| systemDN | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| probableCause | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| perceivedSeverity | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| alarmId | One NV pair of remaining\_body | M | See that of notifyNewAlarm. |
| comments | One NV pair of remaining\_body | M | Name of NV pair is the COMMENTS of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a CommentSet of module AlarmIRPConstDefs. |

Table A.2.3.8: Mapping for notifyPotentialFaultyAlarmList

| IS Parameters | OMG CORBA Structured Event attribute | Qualifier | Comment |
| --- | --- | --- | --- |
| There is no corresponding IS attribute. | domain\_name |  | See that of notifyNewAlarm. |
| notificationType | type\_name | M | This is the NOTIFY\_FM\_POTENTIAL\_FAULTY\_ALARM\_LIST of interface NotificationType of module AlarmIRPConstDefs. |
| There is no corresponding IS attribute. | event\_name | M | It contains a NULL string. |
| There is no corresponding IS attribute. | variable Header |  |  |
| objectClass, objectInstance | One NV pair of filterable\_body\_fields | M | See notifyNewAlarm.  See sub-clause “Definition” of this notification in [4] for the description of the usage of this field to indicate if part or all AlarmList is potentially faulty. |
| notificationId | One NV pair of remaining\_body | M |  |
| eventTime | One NV pair of filterable\_body\_fields | M | See notifyNewAlarm. |
| systemDN | One NV pair of filterable\_body\_fields | M | See notifyNewAlarm. |
| reason | One NV pair of remaining\_body | M | Name of NV pair is the REASON of interface AttributeNameValue of module AlarmIRPConstDefs.  Value of NV pair is a string. |

Table A.2.3.9: Mapping for notifyCorrelatedNotificationChanged

|  |  |  |  |
| --- | --- | --- | --- |
| IS Parameters | OMG CORBA Structured Event attribute | Qualifier | Comment |
| There is no corresponding IS attribute. | domain\_name |  | See that of notifyNewAlarm. |
| notificationType | type\_name | M | This is the NOTIFY\_FM\_CORRELATED\_NOTIFICATION\_CHANGED of interface NotificationType of module AlarmIRPConstDefs. |
| alarmType | event\_name | M | See that of notifyNewAlarm. |
| There is no corresponding IS attribute. | variable Header |  |  |
| objectClass, objectInstance | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| notificationId | One NV pair of remaining\_body | M | See that of notifyNewAlarm. |
| eventTime | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| systemDN | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| correlatedNotifications | One NV pair of remaining\_body | O | Name of NV pair is the CORRELATED\_NOTIFICATIONS of interface AttributeNameValue.  Value of NV pair is a CorrelatedNotificationSet of module AlarmIRPConstDefs. |
| alarmId | One NV pair of remaining\_body | M | See that of notifyNewAlarm. |
| rootCauseIndicator | One NV pair of remaining\_body | O | See that of notifyNewAlarm. |

Table A.2.3.10: Mapping for notifyChangedAlarmGeneral (to carry non-security-related alarms)

| IS Parameters | OMG CORBA Structured Event attribute | Qualifier | Comment |
| --- | --- | --- | --- |
| There is no corresponding SS attribute. | domain\_name |  | See that of notifyNewAlarm. |
| notificationType | type\_name | M | This is the NOTIFY\_FM\_ CHANGED\_ALARM\_GENERAL of interface NotificationType of module AlarmIRPConstDefs. |
| alarmType | event\_name | M | It identifies one of the following:  communications alarm,  processing error alarm,  environmental alarm,  quality of service alarm and  equipment alarm.  It is a string defined by interface AlarmType of module AlarmIRPConstDefs. |
| There is no corresponding SS attribute. | variable Header |  |  |
| objectClass, objectInstance | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| notificationId | One NV pair of remaining\_body | M | See that of notifyNewAlarm. |
| eventTime | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| systemDN | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| probableCause | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| perceivedSeverity | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| rootCauseIndicator | One NV pair of filterable\_body\_fields | O | See that of notifyNewAlarm. |
| specificProblem | One NV pair of remaining\_body | O | See that of notifyNewAlarm. |
| correlatedNotifications | One NV pair of remaining\_body | O | See that of notifyNewAlarm. |
| backedUpStatus | One NV pair of remaining\_body | O | See that of notifyNewAlarm. |
| backUpObject | One NV pair of remaining\_body | O | See that of notifyNewAlarm. |
| trendIndication | One NV pair of remaining\_body | O | See that of notifyNewAlarm. |
| thresholdInfo | One NV pair of remaining\_body | O | See that of notifyNewAlarm. |
| stateChangeDefinition | One NV pair of remaining\_body | O | See that of notifyNewAlarm. |
| monitoredAttributes | One NV pair of remaining\_body | O | See that of notifyNewAlarm. |
| proposedRepairActions | One NV pair of remaining\_body | O | See that of notifyNewAlarm. |
| additionalText | One NV pair of remaining\_body | O | See that of notifyNewAlarm. |
| additionalInformation | One or more NV pairs of remaining\_body | O | See that of notifyNewAlarm. |
| alarmId | One NV pair of remaining\_body | M | See that of notifyNewAlarm. |
| changedAlarmAttributes | One or more NV pairs of remaining\_body | M | The changed alarm attributes (name/value pairs) (with old values). Name of NV pair is the CHANGED\_ALARM\_ATTRIBUTES of interface AttributeNameValue of module AlarmIRPConstDefs. |

Table A.2.3.11: Mapping for notifyChangedAlarmGeneral (to carry security alarm)

| IS Parameters | OMG CORBA Structured Event attribute | Qualifier | Comment |
| --- | --- | --- | --- |
| There is no corresponding SS attribute. | domain\_name |  | It carries the IRP document version number string. See sub-clause 3.1.  It indicates the syntax and semantics of the Structured Event as defined by this specification. |
| notificationType | type\_name | M | This is the NOTIFY\_FM\_ CHANGED\_ALARM\_GENERAL of interface NotificationType of module AlarmIRPConstDefs. |
| alarmType | event\_name | M | It identifies one of the following:  Integrity violation, operational violation, physical violation, security violation and time domain violation.  It is a string defined by interface AlarmType of module AlarmIRPConstDefs. |
| There is no corresponding SS attribute. | variable Header |  |  |
| objectClass, objectInstance | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| notificationId | One NV pair of remaining\_body | M | See that of notifyNewAlarm. |
| eventTime | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| systemDN | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| probableCause | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| perceivedSeverity | One NV pair of filterable\_body\_fields | M | See that of notifyNewAlarm. |
| rootCauseIndicator | One NV pair of filterable\_body\_fields | O | See that of notifyNewAlarm. |
| correlatedNotifications | One NV pair of remaining\_body | O | See that of notifyNewAlarm. |
| additionalText | One NV pair of remaining\_body | O | See that of notifyNewAlarm. |
| additionalInformation | One or more NV pairs of remaining\_body | O | See that of notifyNewAlarm. |
| alarmId | One NV pair of remaining\_body | M | See that of notifyNewAlarm. |
| serviceUser | One NV pair of remaining\_body | M | See that of notifyNewAlarm. |
| serviceProvider | One NV pair of remaining\_body | M | See that of notifyNewAlarm. |
| securityAlarmDetector | One NV pair of remaining\_body | M | See that of notifyNewAlarm. |
| changedAlarmAttributes | One or more NV pairs of remaining\_body | M | The changed alarm attributes (name/value pairs) (with old values). Name of NV pair is the CHANGED\_ALARM\_ATTRIBUTES of interface AttributeNameValue of module AlarmIRPConstDefs. |

# A.3 Solution Set definitions

## A.3.1 IDL definition structure

Clause A.3.2 defines the constants and types used by the Alarm IRP.

Clause A.3.3 defines the operations which are performed by the Alarm IRP agent.

Clause A.3.4 defines the notifications which are emitted by the Alarm IRP agent.

## A.3.2 IDL specification "AlarmIRPConstDefs.idl"

//File: AlarmIRPConstDefs.idl

#ifndef \_ALARM\_IRP\_CONST\_DEFS\_IDL\_

#define \_ALARM\_IRP\_CONST\_DEFS\_IDL\_

#include <CosNotification.idl>

#include <ManagedGenericIRPConstDefs.idl>

// This statement must appear after all include statements

#pragma prefix "3gppsa5.org"

/\* ## Module: AlarmIRPConstDefs

This module contains commonly used definitions for Alarm IRP

================================================================

\*/

module AlarmIRPConstDefs

{

/\*

The format of Distinguished Name (DN) is specified in 3GPP TS 32.300

"Name Conventions for Managed Objects".

\*/

typedef string DN;

/\* DNOpt is an optional type.

If the discriminator is true the value is present.

Otherwise the value is null.

\*/

union DNOpt switch (boolean)

{

case TRUE: DN value;

};

/\*

This block identifies the alarm types specified for this IRP version.

These types carry the same semantics as the TMN ITU-T defined event

types of the same name.

Their encodings for this version of Alarm IRP are defined here. Other IRP

documents, or other versions of Alarm IRP, shall identify their own

alarm types for their use. They shall define their encodings

as well. Values defined here are unique among themselves.

\*/

interface AlarmType

{

const string COMMUNICATIONS\_ALARM = "x1";

const string PROCESSING\_ERROR\_ALARM = "x2";

const string ENVIRONMENTAL\_ALARM = "x3";

const string QUALITY\_OF\_SERVICE\_ALARM = "x4";

const string EQUIPMENT\_ALARM = "x5";

const string INTEGRITY\_VIOLATION = "x6";

const string OPERATIONAL\_VIOLATION = "x7";

const string PHYSICAL\_VIOLATION = "x8";

const string SECURITY\_SERVICE\_OR\_MECHANISM\_VIOLATION = "x9";

const string TIME\_DOMAIN\_VIOLATION = "x10";

};

/\*

This block identifies the notification types defined by this

Alarm IRP version.

\*/

interface NotificationType

{

const string NOTIFY\_FM\_NEW\_ALARM = "x1";

const string NOTIFY\_FM\_CHANGED\_ALARM = "x2";

const string NOTIFY\_FM\_ACK\_STATE\_CHANGED = "x3";

const string NOTIFY\_FM\_COMMENT\_ADDED = "x4";

const string NOTIFY\_FM\_CLEARED\_ALARM = "x5";

const string NOTIFY\_FM\_ALARM\_LIST\_REBUILT = "x6";

const string NOTIFY\_FM\_POTENTIAL\_FAULTY\_ALARM\_LIST = "x7";

const string NOTIFY\_FM\_CORRELATED\_NOTIFICATION\_CHANGED = "x8";

const string NOTIFY\_FM\_CHANGED\_ALARM\_GENERAL = “x9";

};

/\*

This block identifies the levels of severity.

\*/

interface PerceivedSeverity

{

const short INDETERMINATE = 1;

const short CRITICAL = 2;

const short MAJOR = 3;

const short MINOR = 4;

const short WARNING = 5;

const short CLEARED = 6;

};

/\*

This block identifies the probable cause of a reported alarm.

\*/

interface ProbableCause

{

/\*

Probable causes originating from M.3100.

Values below correspond to M.3100 values.

\*/

const short INDETERMINATE = 0;

const short ALARM\_INDICATION\_SIGNAL = 1;

const short CALL\_SETUP\_FAILURE = 2;

const short DEGRADED\_SIGNAL = 3;

const short FAR\_END\_RECEIVER\_FAILURE = 4;

const short FRAMING\_ERROR = 5;

const short LOSS\_OF\_FRAME = 6;

const short LOSS\_OF\_POINTER = 7;

const short LOSS\_OF\_SIGNAL = 8;

const short PAYLOAD\_TYPE\_MISMATCH = 9;

// Values 10 correspond to a duplicated probable cause

const short REMOTE\_ALARM\_INTERFACE = 11;

const short EXCESSIVE\_BIT\_ERROR\_RATE = 12;

const short PATH\_TRACE\_MISMATCH = 13;

const short UNAVAILABLE = 14;

const short SIGNAL\_LABEL\_MISMATCH = 15;

const short LOSS\_OF\_MULTI\_FRAME = 16;

const short COMMUNICATIONS\_RECEIVE\_FAILURE = 17;

const short COMMUNICATIONS\_TRANSMIT\_FAILURE = 18;

const short MODULATION\_FAILURE = 19;

const short DEMODULATION\_FAILURE = 20;

// Values 21-26 correspond to duplicated probable causes

// Values 27-50 are reserved for M.3100 potential future extensions

const short BACK\_PLANE\_FAILURE = 51;

const short DATA\_SET\_PROBLEM = 52;

const short EQUIPMENT\_IDENTIFIER\_DUPLICATION = 53;

const short EXTERNAL\_IF\_DEVICE\_PROBLEM = 54;

const short LINE\_CARD\_PROBLEM = 55;

const short MULTIPLEXER\_PROBLEM = 56;

const short NE\_IDENTIFIER\_DUPLICATION = 57;

const short POWER\_PROBLEM = 58;

const short PROCESSOR\_PROBLEM = 59;

const short PROTECTION\_PATH\_FAILURE = 60;

const short RECEIVER\_FAILURE = 61;

const short REPLACEABLE\_UNIT\_MISSING = 62;

const short REPLACEABLE\_UNIT\_TYPE\_MISMATCH = 63;

const short SYNCHRONIZATION\_SOURCE\_MISMATCH = 64;

const short TERMINAL\_PROBLEM = 65;

const short TIMING\_PROBLEM = 66;

const short TRANSMITTER\_FAILURE = 67;

const short TRUNK\_CARD\_PROBLEM = 68;

const short REPLACEABLE\_UNIT\_PROBLEM = 69;

const short REAL\_TIME\_CLOCK\_FAILURE = 70;

// Values 71-80 correspond to duplicated probable causes

const short PROTECTION\_MECHANISM\_FAILURE = 81;

const short PROTECTING\_RESOURCE\_FAILURE = 82;

// Values 83-100 are reserved for M.3100 potential future extensions

const short AIR\_COMPRESSOR\_FAILURE = 101;

const short AIR\_CONDITIONING\_FAILURE = 102;

const short AIR\_DRYER\_FAILURE = 103;

const short BATTERY\_DISCHARGING = 104;

const short BATTERY\_FAILURE = 105;

const short COMMERCIAL\_POWER\_FAILURE = 106;

const short COOLING\_FAN\_FAILURE = 107;

const short ENGINE\_FAILURE = 108;

const short FIRE\_DETECTOR\_FAILURE = 109;

const short FUSE\_FAILURE = 110;

const short GENERATOR\_FAILURE = 111;

const short LOW\_BATTERY\_THRESHOLD = 112;

const short PUMP\_FAILURE = 113;

const short RECTIFIER\_FAILURE = 114;

const short RECTIFIER\_HIGH\_VOLTAGE = 115;

const short RECTIFIER\_LOW\_F\_VOLTAGE = 116;

const short VENTILATION\_SYSTEM\_FAILURE = 117;

const short ENCLOSURE\_DOOR\_OPEN = 118;

const short EXPLOSIVE\_GAS = 119;

const short FIRE = 120;

const short FLOOD = 121;

const short HIGH\_HUMIDITY = 122;

const short HIGH\_TEMPERATURE = 123;

const short HIGH\_WIND = 124;

const short ICE\_BUILD\_UP = 125;

const short INTRUSION\_DETECTION = 126;

const short LOW\_FUEL = 127;

const short LOW\_HUMIDITY = 128;

const short LOW\_CABLE\_PRESSURE = 129;

const short LOW\_TEMPERATURE = 130;

const short LOW\_WATER = 131;

const short SMOKE = 132;

const short TOXIC\_GAS = 133;

// Values 134-135 correspond to duplicated probable causes

const short EXTERNAL\_POINT\_FAILURE = 136;

// Values 137-150 are reserved for potential M.3100 future extensions

const short STORAGE\_CAPACITY\_PROBLEM = 151;

const short MEMORY\_MISMATCH = 152;

const short CORRUPT\_DATA = 153;

const short OUT\_OF\_CPU\_CYCLES = 154;

const short SOFTWARE\_ENVIRONMENT\_PROBLEM = 155;

const short SOFTWARE\_DOWNLOAD\_FAILURE = 156;

const short LOSS\_OF\_REAL\_TIME = 157;

const short REINITIALIZED = 158;

// Values 159-167 correspond to duplicated probable causes

// Values 168-200 are reserved for potential M.3100 future extensions

// Values 201-202 correspond to duplicated probable causes

const short EXCESSIVE\_ERROR\_RATE = 203;

// Values 204-207 correspond to duplicated probable causes

// Values 208-300 are reserved for potential M.3100 future extensions

/\*

Probable causes originating from X.721.

Values below correspond to X.721 values with an offset of 300.

\*/

const short ADAPTER\_ERROR = 301;

const short APPLICATION\_SUBSYSTEM\_FAILURE = 302;

const short BANDWIDTH\_REDUCED = 303;

// Value 304 corresponds to a duplicated probable cause

const short COMMUNICATIONS\_PROTOCOL\_ERROR = 305;

const short COMMUNICATIONS\_SUBSYSTEM\_FAILURE = 306;

const short CONFIGURATION\_OR\_CUSTOMIZATION\_ERROR = 307;

const short CONGESTION = 308;

// Value 309 corresponds to a duplicated probable cause

const short CPU\_CYCLES\_LIMIT\_EXCEEDED = 310;

const short DATA\_SET\_OR\_MODEM\_ERROR = 311;

// Value 312 corresponds to a duplicated probable cause

const short DTE\_DCE\_INTERFACE\_ERROR = 313;

// Value 314 corresponds to a duplicated probable cause

const short EQUIPMENT\_MALFUNCTION = 315;

const short EXCESSIVE\_VIBRATION = 316;

const short FILE\_ERROR = 317;

// Values 318-320 correspond to duplicated probable causes

const short HEATING\_OR\_VENTILATION\_OR\_COOLING\_SYSTEM\_PROBLEM = 321;

const short HUMIDITY\_UNACCEPTABLE = 322;

const short INPUT\_OUTPUT\_DEVICE\_ERROR = 323;

const short INPUT\_DEVICE\_ERROR = 324;

const short LAN\_ERROR = 325;

const short LEAK\_DETECTED = 326;

const short LOCAL\_NODE\_TRANSMISSION\_ERROR = 327;

// Values 328-329 correspond to duplicated probable causes

const short MATERIAL\_SUPPLY\_EXHAUSTED = 330;

// Value 331 corresponds to a duplicated probable cause

const short OUT\_OF\_MEMORY = 332;

const short OUTPUT\_DEVICE\_ERROR = 333;

const short PERFORMANCE\_DEGRADED = 334;

// Value 335 corresponds to a duplicated probable cause

const short PRESSURE\_UNACCEPTABLE = 336;

// Values 337-338 correspond to duplicated probable causes

const short QUEUE\_SIZE\_EXCEEDED = 339;

const short RECEIVE\_FAILURE = 340;

// Value 341 corresponds to a duplicated probable cause

const short REMOTE\_NODE\_TRANSMISSION\_ERROR = 342;

const short RESOURCE\_AT\_OR\_NEARING\_CAPACITY = 343;

const short RESPONSE\_TIME\_EXCESSIVE = 344;

const short RETRANSMISSION\_RATE\_EXCESSIVE = 345;

const short SOFTWARE\_ERROR = 346;

const short SOFTWARE\_PROGRAM\_ABNORMALLY\_TERMINATED = 347;

const short SOFTWARE\_PROGRAM\_ERROR = 348;

// Value 349 corresponds to a duplicated probable cause

const short TEMPERATURE\_UNACCEPTABLE = 350;

const short THRESHOLD\_CROSSED = 351;

// Value 352 corresponds to a duplicated probable cause

const short TOXIC\_LEAK\_DETECTED = 353;

const short TRANSMIT\_FAILURE = 354;

// Value 355 corresponds to a duplicated probable cause

const short UNDERLYING\_RESOURCE\_UNAVAILABLE = 356;

const short VERSION\_MISMATCH = 357;

// Values 358-500 are reserved for potential X.721 future extensions

/\*

Probable causes for 2G & 3G wireless systems.

\*/

const short A\_BIS\_TO\_BTS\_INTERFACE\_FAILURE = 501;

const short A\_BIS\_TO\_TRX\_INTERFACE\_FAILURE = 502;

const short ANTENNA\_PROBLEM = 503;

const short BATTERY\_BREAKDOWN = 504;

const short BATTERY\_CHARGING\_FAULT = 505;

const short CLOCK\_SYNCHRONIZATION\_PROBLEM = 506;

const short COMBINER\_PROBLEM = 507;

const short DISK\_PROBLEM = 508;

// Value 509 corresponds to a duplicated probable cause

const short EXCESSIVE\_RECEIVER\_TEMPERATURE = 510;

const short EXCESSIVE\_TRANSMITTER\_OUTPUT\_POWER = 511;

const short EXCESSIVE\_TRANSMITTER\_TEMPERATURE = 512;

const short FREQUENCY\_HOPPING\_DEGRADED = 513;

const short FREQUENCY\_HOPPING\_FAILURE = 514;

const short FREQUENCY\_REDEFINITION\_FAILED = 515;

const short LINE\_INTERFACE\_FAILURE = 516;

const short LINK\_FAILURE = 517;

const short LOSS\_OF\_SYNCHRONIZATION = 518;

const short LOST\_REDUNDANCY = 519;

const short MAINS\_BREAKDOWN\_WITH\_BATTERY\_BACKUP = 520;

const short MAINS\_BREAKDOWN\_WITHOUT\_BATTERY\_BACKUP = 521;

const short POWER\_SUPPLY\_FAILURE = 522;

const short RECEIVER\_ANTENNA\_FAULT = 523;

// Value 524 corresponds to a duplicated probable cause

const short RECEIVER\_MULTICOUPLER\_FAILURE = 525;

const short REDUCED\_TRANSMITTER\_OUTPUT\_POWER = 526;

const short SIGNAL\_QUALITY\_EVALUATION\_FAULT = 527;

const short TIMESLOT\_HARDWARE\_FAILURE = 528;

const short TRANSCEIVER\_PROBLEM = 529;

const short TRANSCODER\_PROBLEM = 530;

const short TRANSCODER\_OR\_RATE\_ADAPTER\_PROBLEM = 531;

const short TRANSMITTER\_ANTENNA\_FAILURE = 532;

const short TRANSMITTER\_ANTENNA\_NOT\_ADJUSTED = 533;

// Value 534 corresponds to a duplicated probable cause

const short TRANSMITTER\_LOW\_VOLTAGE\_OR\_CURRENT = 535;

const short TRANSMITTER\_OFF\_FREQUENCY = 536;

const short DATABASE\_INCONSISTENCY = 537;

const short FILE\_SYSTEM\_CALL\_UNSUCCESSFUL = 538;

const short INPUT\_PARAMETER\_OUT\_OF\_RANGE = 539;

const short INVALID\_PARAMETER = 540;

const short INVALID\_POINTER = 541;

const short MESSAGE\_NOT\_EXPECTED = 542;

const short MESSAGE\_NOT\_INITIALIZED = 543;

const short MESSAGE\_OUT\_OF\_SEQUENCE = 544;

const short SYSTEM\_CALL\_UNSUCCESSFUL = 545;

const short TIMEOUT\_EXPIRED = 546;

const short VARIABLE\_OUT\_OF\_RANGE = 547;

const short WATCH\_DOG\_TIMER\_EXPIRED = 548;

const short COOLING\_SYSTEM\_FAILURE = 549;

const short EXTERNAL\_EQUIPMENT\_FAILURE = 550;

const short EXTERNAL\_POWER\_SUPPLY\_FAILURE = 551;

const short EXTERNAL\_TRANSMISSION\_DEVICE\_FAILURE = 552;

// Values 553-560 correspond to duplicated probable causes

const short REDUCED\_ALARM\_REPORTING = 561;

const short REDUCED\_EVENT\_REPORTING = 562;

const short RECUCED\_LOGGING\_CAPABILITY = 563;

const short SYSTEM\_RESOURCES\_OVERLOAD = 564;

const short BROADCAST\_CHANNEL\_FAILURE = 565;

const short CONNECTION\_ESTABLISHMENT\_ERROR = 566;

const short INVALID\_MESSAGE\_RECEIVED = 567;

const short INVALID\_MSU\_RECEIVED = 568;

const short LAPD\_LINK\_PROTOCOL\_FAILURE = 569;

const short LOCAL\_ALARM\_INDICATION = 570;

const short REMOTE\_ALARM\_INDICATION = 571;

const short ROUTING\_FAILURE = 572;

const short SS7\_PROTOCOL\_FAILURE = 573;

const short TRANSMISSION\_ERROR = 574;

// Value 575 corresponds to a duplicated probable cause

// Values 576-700 are reserved for potential future extensions

// for 2G & 3G wireless systems

/\*

Probable causes originating from M.3100 security alarm causes.

Values below correspond to M.3100 values with an offset of 700.

\*/

const short AUTHENTICATION\_FAILURE = 701;

const short BREACH\_OF\_CONFIDENTIALITY = 702;

const short CABLE\_TAMPER = 703;

const short DELAYED\_INFORMATION = 704;

const short DENIAL\_OF\_SERVICE = 705;

const short DUPLICATE\_INFORMATION = 706;

const short INFORMATION\_MISSING = 707;

const short INFORMATION\_MODIFICATION\_DETECTED = 708;

const short INFORMATION\_OUT\_OF\_SEQUENCE = 709;

// Value 710 corresponds to a duplicated probable cause

const short KEY\_EXPIRED = 711;

const short NON\_REPUDIATION\_FAILURE = 712;

const short OUT\_OF\_HOURS\_ACTIVITY = 713;

const short OUT\_OF\_SERVICE = 714;

const short PROCEDURAL\_ERROR = 715;

const short UNAUTHORISED\_ACCESS\_ATTEMPT = 716;

const short UNEXPECTED\_INFORMATION = 717;

const short UNSPECIFIED\_REASON = 718;

// Values 719-800 are reserved for potential M.3100 future extensions

};

/\*

It indicates that this AlarmInformation is the root cause of the events captured by the notifications whose identifiers are in the related CorrelatedNotification instances.

\*/

typedef boolean RootCauseIndicator;

/\*

This block identifies the acknowledgement state of a reported alarm.

\*/

interface AckState

{

const short ACKNOWLEDGED = 1;

const short UNACKNOWLEDGED = 2;

};

/\*

This block identifies attributes which are included as part of the Alarm IRP

These attribute values should not clash with those defined for the attributes

of notification header (see IDL of Notification IRP).

\*/

interface AttributeNameValue

{

const string ALARM\_ID = "f";

const string PROBABLE\_CAUSE = "g";

const string PERCEIVED\_SEVERITY = "h";

const string SPECIFIC\_PROBLEM = "i";

const string ADDITIONAL\_TEXT = "j";

const string ACK\_TIME = "k";

const string ACK\_USER\_ID = "l";

const string ACK\_SYSTEM\_ID = "m";

const string ACK\_STATE = "n";

const string COMMENTS = "o";

const string BACKED\_UP\_STATUS = "p";

const string BACK\_UP\_OBJECT = "q";

const string THRESHOLD\_INFO = "r";

const string TREND\_INDICATION = "s";

const string STATE\_CHANGE\_DEFINITION = "t";

const string MONITORED\_ATTRIBUTES = "u";

const string PROPOSED\_REPAIR\_ACTIONS = "v";

const string CORRELATED\_NOTIFICATIONS = "w";

const string REASON = "x";

const string CLEAR\_USER\_ID = "y";

const string CLEAR\_SYSTEM\_ID = "z";

const string ALARM\_LIST\_ALIGNMENT\_REQUIREMENT = "ff";

const string SERVICE\_USER = "gg";

const string SERVICE\_PROVIDER = "hh";

const string SECURITY\_ALARM\_DETECTOR = "ii";

const string ALARM\_RAISED\_TIME = "kk";

const string ALARM\_CLEARED\_TIME = "ll";

const string ALARM\_CHANGED\_TIME = "mm";

const string ROOT\_CAUSE\_INDICATOR = "nn";

const string CHANGED\_ALARM\_ATTRIBUTES = "pp";

};

/\*

constants for use in populating the additional information

name field Names

\*/

interface AdditionalInformation

{

const string AI\_VS\_PERCEIVED\_SEVERITY = "ai\_ps";

const string AI\_VS\_ALARM\_TYPE = "ai\_at";

};

/\*

Defines the content of a Comment

\*/

struct Comment

{

ManagedGenericIRPConstDefs::IRPTime comment\_time;

string comment\_text;

string user\_id;

string system\_id;

};

/\*

Defines a set of comments which are placed in the COMMENTS attribute

of a structured event.

\*/

typedef sequence <Comment> CommentSet;

/\*

It indicates if an object has a back up.

True implies backed up. False implies not backed up.

\*/

typedef boolean BackedUpStatus;

/\*

It indicates if the threshold crossed was in the up or down direction.

\*/

enum ThresholdIndication {UP, DOWN};

/\*

It indicates if the AlarmList alignment is required.

\*/

enum AlarmListAlignmentRequirement {REQUIRED, NOTREQUIRED};

/\* FloatOpt is an optional type.

If the discriminator is true the value is present.

Otherwise the value is null.

\*/

union FloatOpt switch (boolean)

{

case TRUE: float value;

};

/\* ThresholdLevelInd describes multi-level

threshold crossings.

Up is the only permitted choice for a counter.

If indication is "up", low value is optional.

@member indication: indicates up or down direction

of crossing.

@member low: the low observed value.

@member high: the high observed value.

\*/

struct ThresholdLevelInd

{

ThresholdIndication indication;

FloatOpt low;

float high;

};

/\* ThresholdLevelIndOpt is an optional type.

If the discriminator is true the value is present.

Otherwise, the value is null.

\*/

union ThresholdLevelIndOpt switch (boolean)

{

case TRUE: ThresholdLevelInd value;

};

/\* ThresholdInfo indicates some gauge or counter

attribute passed a set threshold.

@member attribute\_id: identifies the attribute that

crossed the threshold.

@member observed\_value: attributes that are of type

integer will be converted to floats.

@member threshold\_level: This parameter is for

multi-level thresholds. Optional.

@member arm\_time: May contain empty string.

\*/

struct ThresholdInfo

{

string attribute\_id;

float observed\_value;

ThresholdLevelIndOpt threshold\_level;

string arm\_time;

};

/\*

It indicates if some observed condition is getting better, worse,

or not changing.

\*/

enum TrendIndication {LESS\_SEVERE, NO\_CHANGE, MORE\_SEVERE};

/\*

It is used to report a changed attribute value.

\*/

struct AttributeValueChange

{

string attribute\_name;

any old\_value; // type depends on attribute

any new\_value; // type depends on attribute

};

typedef sequence <AttributeValueChange> AttributeChangeSet;

/\*

It is used to report a changed alarm attribute value.

\*/

struct AlarmAttributeValueChange

{

string attribute\_name;

any old\_value; // type depends on attribute

};

typedef sequence <AlarmAttributeValueChange> AlarmAttributeChangeSet;

/\*

It is used to report an attribute and its value.

\*/

struct AttributeValue

{

string attribute\_name;

any value; // type depends on the attribute

};

typedef sequence <AttributeValue> AttributeSet;

typedef sequence <long> NotifIdSet;

/\*

This holds identifiers of notifications that are correlated.

\*/

struct CorrelatedNotification

{

DN source; // Contains DN of MO that emitted the set of notifications

// DN string format in compliance with Name Convention for

// Managed Object.

// This may be a zero-length string. In this case, the MO

// is identified by the value of the MOI attribute

// of the Structured Event, i.e., the notification.

NotifIdSet notif\_id\_set; // Set of related notification ids

};

/\*

Correlated Notification sets are sets of Correlated Notification

structures.

\*/

typedef sequence <CorrelatedNotification> CorrelatedNotificationSet;

/\*

Define the structure of Alarm ID and Perceived Severity used within the

alarm acknowledgment operation. Note: perceived\_severity is an optional

parameter. If this value is present, it must have one of the defined values

of Interface PerceivedSeverity.

\*/

struct AlarmInformationIdAndSev

{

string alarm\_id;

ManagedGenericIRPConstDefs::ShortOpt perceived\_severity;

};

/\*

Define set of the above structure of Alarm ID and Perceived Severity.

\*/

typedef sequence <AlarmInformationIdAndSev> AlarmInformationIdAndSevSeq;

/\*

It indicates the reason for an alarm acknowledgement to have failed:

- The specified Alarm Information is absent from the Alarm List

- The Perceived Severity to be acknowledged has changed and/or is different

within the Alarm List

- The acknowledgement failed for some other reason

\*/

enum AcknowledgeFailureCategories

{

UNKNOWN\_ALARM\_ID,

WRONG\_PERCEIVED\_SEVERITY,

ACKNOWLEDGMENT\_FAILED

};

/\*

Define the structure returned when an operation fails for a set of alarm ids.

A reason is provided in order to indicate why the operation failed.

\*/

struct BadAlarmInformationId

{

string alarm\_id;

string reason;

};

/\*

Define the structure returned when the acknowledge operation fails for a set

of alarm ids.

A failure category and a reason are provided in order to indicate why the

operation failed.

\*/

struct BadAcknowledgeAlarmInfo

{

string alarm\_id;

AcknowledgeFailureCategories failure\_category;

string reason;

};

typedef sequence <BadAlarmInformationId> BadAlarmInformationIdSeq;

typedef sequence <BadAcknowledgeAlarmInfo> BadAcknowledgeAlarmInfoSeq;

typedef sequence <string> AlarmInformationIdSeq;

typedef CosNotification::EventBatch AlarmInformationSeq;

/\*

Define the cause values of notifyPotentialFaultyAlarmList and

notifyAlarmListRebuilt, which have been defined in 32111-2.

\*/

const string AGENT\_NE\_COMMUNICATION\_ERROR = "Agent-NE communication error";

const string AGENT\_RESTARTS = "Agent restarts";

const string INDETERMINATE = "Indeterminate";

};

#endif // \_ALARM\_IRP\_CONST\_DEFS\_IDL\_

## A.3.3 IDL specification “AlarmIRPSystem.idl”

//File: AlarmIRPSystem.idl

#ifndef \_ALARM\_IRP\_SYSTEM\_IDL\_

#define \_ALARM\_IRP\_SYSTEM\_IDL\_

#include <AlarmIRPConstDefs.idl>

#include <ManagedGenericIRPSystem.idl>

// This statement must appear after all include statements

#pragma prefix "3gppsa5.org"

/\* ## Module: AlarmIRPSystem

This module contains the specification of all operations of Alarm IRP Agent.

================================================================

\*/

module AlarmIRPSystem

{

/\*

System fails to complete the operation. System can provide reason

to qualify the exception. The semantics carried in reason

is outside the scope of this IRP.

\*/

exception GetAlarmIRPVersions { string reason; };

exception GetAlarmIRPOperationsProfile { string reason; };

exception GetAlarmIRPNotificationProfile { string reason; };

exception AcknowledgeAlarms { string reason; };

exception UnacknowledgeAlarms { string reason; };

exception CommentAlarms { string reason; };

exception ClearAlarms { string reason; };

exception GetAlarmList { string reason; };

exception GetAlarmCount { string reason; };

exception NextAlarmInformations { string reason; };

exception FilterComplexityLimit { string reason; };

/\*

The AlarmInformationIterator is used to iterate through a snapshot of

Alarm Informations taken from the Alarm List when IRPManager invokes

get\_alarm\_list. IRPManager uses it to pace the return of Alarm

Informations.

IRPAgent controls the life-cycle of the iterator. However, a destroy

operation is provided to handle the case where IRPManager wants to stop

the iteration procedure before reaching the last iteration.

\*/

interface AlarmInformationIterator

{

/\*

This method returns between 1 and "how\_many" Alarm Informations. The

IRPAgent may return less than "how\_many" items even if there are more

items to return. "how\_many" must be non-zero. Return TRUE if there may

be more Alarm Information to return. Return FALSE if there are no more

Alarm Information to be returned.

If FALSE is returned, the IRPAgent will automatically destroy the

iterator.

\*/

boolean next\_alarm\_informations (

in unsigned short how\_many,

out AlarmIRPConstDefs::AlarmInformationSeq alarm\_informations

)

raises (NextAlarmInformations, ManagedGenericIRPSystem::InvalidParameter);

/\*

This method destroys the iterator.

\*/

void destroy();

};

interface AlarmIRP

{

/\*

Return the list of all supported Alarm IRP versions.

Implementations are to provide a return value consisting of one or more

IRPVersions.

Each IRPVersion is defined by the rule in the clause titled

"IRP document version number string"

\*/

ManagedGenericIRPConstDefs::VersionNumberSet get\_alarm\_irp\_versions (

)

raises (GetAlarmIRPVersions);

/\*

Return the list of all supported operations and their supported

parameters for a specific Alarm IRP version.

\*/

ManagedGenericIRPConstDefs::MethodList get\_alarm\_irp\_operations\_profile (

in ManagedGenericIRPConstDefs::VersionNumber alarm\_irp\_version

)

raises (GetAlarmIRPOperationsProfile,

ManagedGenericIRPSystem::OperationNotSupported,

ManagedGenericIRPSystem::InvalidParameter);

/\*

Return the list of all supported notifications and their supported

parameters for a specific Alarm IRP version.

\*/

ManagedGenericIRPConstDefs::MethodList get\_alarm\_irp\_notification\_profile

(

in ManagedGenericIRPConstDefs::VersionNumber alarm\_irp\_version

)

raises (GetAlarmIRPNotificationProfile,

ManagedGenericIRPSystem::OperationNotSupported,

ManagedGenericIRPSystem::InvalidParameter);

/\*

Request to acknowledge one or more alarms.

\*/

ManagedGenericIRPConstDefs::Signal acknowledge\_alarms (

in AlarmIRPConstDefs::AlarmInformationIdAndSevSeq

alarm\_information\_id\_and\_sev\_list,

in string ack\_user\_id,

in ManagedGenericIRPConstDefs::StringOpt ack\_system\_id,

out AlarmIRPConstDefs::BadAcknowledgeAlarmInfoSeq

bad\_ack\_alarm\_info\_list

)

raises (AcknowledgeAlarms, ManagedGenericIRPSystem::ParameterNotSupported,

ManagedGenericIRPSystem::InvalidParameter);

/\*

Request to remove acknowledgement information of one or more alarms.

\*/

ManagedGenericIRPConstDefs::Signal unacknowledge\_alarms (

in AlarmIRPConstDefs::AlarmInformationIdSeq alarm\_information\_id\_list,

in string ack\_user\_id,

in ManagedGenericIRPConstDefs::StringOpt ack\_system\_id,

out AlarmIRPConstDefs::BadAlarmInformationIdSeq

bad\_alarm\_information\_id\_list

)

raises (UnacknowledgeAlarms,

ManagedGenericIRPSystem::OperationNotSupported,

ManagedGenericIRPSystem::ParameterNotSupported,

ManagedGenericIRPSystem::InvalidParameter);

/\*

Make comment to one or more alarms.

\*/

ManagedGenericIRPConstDefs::Signal comment\_alarms (

in AlarmIRPConstDefs::AlarmInformationIdSeq alarm\_information\_id\_list,

in string comment\_user\_id,

in ManagedGenericIRPConstDefs::StringOpt comment\_system\_id,

in string comment\_text,

out AlarmIRPConstDefs::BadAlarmInformationIdSeq

bad\_alarm\_information\_id\_list

)

raises (CommentAlarms, ManagedGenericIRPSystem::OperationNotSupported,

ManagedGenericIRPSystem::ParameterNotSupported,

ManagedGenericIRPSystem::InvalidParameter);

/\*

Request to clear one or more alarms.

\*/

ManagedGenericIRPConstDefs::Signal clear\_alarms (

in AlarmIRPConstDefs::AlarmInformationIdSeq alarm\_information\_id\_list,

in string clear\_user\_id,

in ManagedGenericIRPConstDefs::StringOpt clear\_system\_id,

out AlarmIRPConstDefs::BadAlarmInformationIdSeq

bad\_alarm\_information\_id\_list

)

raises (ClearAlarms, ManagedGenericIRPSystem::OperationNotSupported,

ManagedGenericIRPSystem::ParameterNotSupported,

ManagedGenericIRPSystem::InvalidParameter);

/\*

This method returns Alarm Informations.

If flag is TRUE, all returned Alarm Informations shall be

in AlarmInformationSeq that contains 0 or more Alarm Informations.

Output parameter iter shall be useless.

If flag is FALSE, no Alarm Informations shall be in AlarmInformationSeq.

IRPAgent needs to use iter to retrieve them.

\*/

AlarmIRPConstDefs::AlarmInformationSeq get\_alarm\_list (

in ManagedGenericIRPConstDefs::StringOpt filter,

in AlarmIRPConstDefs::DNOpt base\_object,

out boolean flag,

out AlarmInformationIterator iter

)

raises (GetAlarmList, FilterComplexityLimit, ManagedGenericIRPSystem::ParameterNotSupported,

ManagedGenericIRPSystem::InvalidParameter);

/\*

This method returns the count of Alarm Informations.

\*/

void get\_alarm\_count (

in ManagedGenericIRPConstDefs::StringOpt filter,

out unsigned long critical\_count,

out unsigned long major\_count,

out unsigned long minor\_count,

out unsigned long warning\_count,

out unsigned long indeterminate\_count,

out unsigned long cleared\_count

)

raises (GetAlarmCount, FilterComplexityLimit, ManagedGenericIRPSystem::OperationNotSupported,

ManagedGenericIRPSystem::ParameterNotSupported,

ManagedGenericIRPSystem::InvalidParameter);

};

};

#endif // \_ALARM\_IRP\_SYSTEM\_IDL\_

## A.3.4 IDL specification "AlarmIRPNotifications.idl"

//File: AlarmIRPNotifications.idl

#ifndef \_ALARM\_IRP\_NOTIFICATIONS\_IDL\_

#define \_ALARM\_IRP\_NOTIFICATIONS\_IDL\_

#include <AlarmIRPConstDefs.idl>

#include <NotificationIRPNotifications.idl>

// This statement must appear after all include statements

#pragma prefix "3gppsa5.org"

/\* ## Module: AlarmIRPNotifications

This module contains notifications for Alarm IRP

================================================================

\*/

module AlarmIRPNotifications

{

interface NotifyNewAlarm: NotificationIRPNotifications::Notify

{

const string EVENT\_TYPE = "notifyNewAlarm";

/\*\*

\* This constant defines the name of the probableCause property.

\* The data type for the value of this property

\* is short.

\*/

const string PROBABLE\_CAUSE =

AlarmIRPConstDefs::AttributeNameValue::PROBABLE\_CAUSE;

/\*\*

\* This constant defines the name of the

\* perceivedSeverity property.

\* The data type for the value of this property

\* is short.

\*/

const string PERCEIVED\_SEVERITY =

AlarmIRPConstDefs::AttributeNameValue::PERCEIVED\_SEVERITY;

/\*\*

\* This constant defines the name of the specificProblem

\* property. The data type for the value of this property

\* is string.

\*/

const string SPECIFIC\_PROBLEM =

AlarmIRPConstDefs::AttributeNameValue::SPECIFIC\_PROBLEM;

/\*\*

\* The two constants defines the name of the additionalInformation

\* property (the vendor-specific (VS) perceived severity and alarm type).

\* The data type for the value of this property

\* is string.

\*/

const string AI\_VS\_PERCEIVED\_SEVERITY =

AlarmIRPConstDefs::AdditionalInformation::AI\_VS\_PERCEIVED\_SEVERITY;

const string AI\_VS\_ALARM\_TYPE =

AlarmIRPConstDefs::AdditionalInformation::AI\_VS\_ALARM\_TYPE;

/\*\*

\* This constant defines the name of the

\* rootCauseIndicator property.

\* The data type for the value of this property

\* is AlarmIRPConstDefs::RootCauseIndicator.

\*/

const string ROOT\_CAUSE\_INDICATOR =

AlarmIRPConstDefs::AttributeNameValue::ROOT\_CAUSE\_INDICATOR;

/\*\*

\* This constant defines the name of the

\* correlatedNotifications property.

\* The data type for the value of this property

\* is AlarmIRPConstDefs::CorrelatedNotificationSet.

\*/

const string CORRELATED\_NOTIFICATIONS =

AlarmIRPConstDefs::AttributeNameValue::

CORRELATED\_NOTIFICATIONS;

/\*\*

\* This constant defines the name of the

\* backedUpStatus property.

\* The data type for the value of this property

\* is AlarmIRPConstDefs::BackedUpStatus.

\*/

const string BACKED\_UP\_STATUS =

AlarmIRPConstDefs::AttributeNameValue::BACKED\_UP\_STATUS;

/\*\*

\* This constant defines the name of the backUpObject property.

\* The data type for the value of this property

\* is a string carrying of DN of the back-up object.

\*/

const string BACK\_UP\_OBJECT =

AlarmIRPConstDefs::AttributeNameValue::BACK\_UP\_OBJECT;

/\*\*

\* This constant defines the name of the

\* trendIndication property.

\* The data type for the value of this property

\* is AlarmIRPConstDefs::TrendIndication.

\*/

const string TREND\_INDICATION =

AlarmIRPConstDefs::AttributeNameValue::TREND\_INDICATION;

/\*\*

\* This constant defines the name of the thresholdInfo property.

\* The data type for the value of this property

\* is AlarmIRPConstDefs::ThresholdInfo.

\*/

const string THRESHOLD\_INFO =

AlarmIRPConstDefs::AttributeNameValue::THRESHOLD\_INFO;

/\*\*

\* This constant defines the name of the

\* stateChangeDefinition property.

\* The data type for the value of this property

\* is AlarmIRPConstDefs::AttributeChangeSet.

\*/

const string STATE\_CHANGE\_DEFINITION =

AlarmIRPConstDefs::AttributeNameValue::STATE\_CHANGE\_DEFINITION;

/\*\*

\* This constant defines the name of the

\* monitoredAttributes property.

\* The data type for the value of this property

\* is AlarmIRPConstDefs::AttributeSet.

\*/

const string MONITORED\_ATTRIBUTES =

AlarmIRPConstDefs::AttributeNameValue::MONITORED\_ATTRIBUTES;

/\*\*

\* This constant defines the name of the

\* proposedRepairActions property.

\* The data type for the value of this property

\* is string.

\*/

const string PROPOSED\_REPAIR\_ACTIONS =

AlarmIRPConstDefs::AttributeNameValue::PROPOSED\_REPAIR\_ACTIONS;

/\*\*

\* This constant defines the name of the additionalText

\* property.

\* The data type for the value of this property

\* is string.

\*/

const string ADDITIONAL\_TEXT =

AlarmIRPConstDefs::AttributeNameValue::ADDITIONAL\_TEXT;

/\*\*

\* This constant defines the name of the alarmId property.

\* The data type for the value of this property

\* is string. If the string is a zero-length string or if

\* this NV pair is absent, the default semantics is that

\* alarmId is a concatenation of

\* managedObjectInstance, eventType, probableCause and

\* specificProblem, if present, of this Structured Event.

\* Since probableCause is encoded as a short, it shall be

\* converted into string before concatenation.

\* The resultant string shall not contain spaces.

\*/

const string ALARM\_ID =

AlarmIRPConstDefs::AttributeNameValue::ALARM\_ID;

/\*\*

\* This constant defines the name of the serviceUser property.

\* The data type for the value of this property

\* is string.

\*/

const string SERVICE\_USER =

AlarmIRPConstDefs::AttributeNameValue::SERVICE\_USER;

/\*\*

\* This constant defines the name of the serviceProvider

\* property.

\* The data type for the value of this property

\* is string.

\*/

const string SERVICE\_PROVIDER =

AlarmIRPConstDefs::AttributeNameValue::SERVICE\_PROVIDER;

/\*\*

\* This constant defines the name of the

\* securityAlarmDetector property.

\* The data type for the value of this property

\* is string.

\*/

const string SECURITY\_ALARM\_DETECTOR =

AlarmIRPConstDefs::AttributeNameValue::SECURITY\_ALARM\_DETECTOR;

};

interface NotifyAckStateChanged:

NotificationIRPNotifications::Notify

{

const string EVENT\_TYPE = "notifyAckStateChanged";

const string PROBABLE\_CAUSE =

AlarmIRPConstDefs::AttributeNameValue::PROBABLE\_CAUSE;

const string PERCEIVED\_SEVERITY =

AlarmIRPConstDefs::AttributeNameValue::PERCEIVED\_SEVERITY;

const string ALARM\_ID =

AlarmIRPConstDefs::AttributeNameValue::ALARM\_ID;

/\*\*

\* This constant defines the name of the ackTime property.

\* The data type for the value of this property

\* is ManagedGenericIRPConstDefs::IRPTime.

\*/

const string ACK\_TIME =

AlarmIRPConstDefs::AttributeNameValue:: ACK\_TIME;

const string ACK\_USER\_ID =

AlarmIRPConstDefs::AttributeNameValue::ACK\_USER\_ID;

const string ACK\_SYSTEM\_ID =

AlarmIRPConstDefs::AttributeNameValue::ACK\_SYSTEM\_ID;

const string ACK\_STATE =

AlarmIRPConstDefs::AttributeNameValue::ACK\_STATE;

};

interface NotifyClearedAlarm: NotificationIRPNotifications::Notify

{

const string EVENT\_TYPE = "notifyClearedAlarm";

const string PROBABLE\_CAUSE =

AlarmIRPConstDefs::AttributeNameValue::PROBABLE\_CAUSE;

const string PERCEIVED\_SEVERITY =

AlarmIRPConstDefs::AttributeNameValue::PERCEIVED\_SEVERITY;

const string ALARM\_ID =

AlarmIRPConstDefs::AttributeNameValue::ALARM\_ID;

const string CLEAR\_USER\_ID =

AlarmIRPConstDefs::AttributeNameValue::CLEAR\_USER\_ID;

const string CLEAR\_SYSTEM\_ID =

AlarmIRPConstDefs::AttributeNameValue::CLEAR\_SYSTEM\_ID;

};

interface NotifyAlarmListRebuilt:

NotificationIRPNotifications::Notify

{

const string EVENT\_TYPE = "notifyAlarmListRebuilt";

const string REASON =

AlarmIRPConstDefs::AttributeNameValue::REASON;

const string ALARM\_LIST\_ALIGNMENT\_REQUIREMENT =

AlarmIRPConstDefs::AttributeNameValue::

ALARM\_LIST\_ALIGNMENT\_REQUIREMENT;

};

interface NotifyChangedAlarm: NotificationIRPNotifications::Notify

{

const string EVENT\_TYPE = "notifyChangedAlarm";

const string PROBABLE\_CAUSE =

AlarmIRPConstDefs::AttributeNameValue::PROBABLE\_CAUSE;

const string PERCEIVED\_SEVERITY =

AlarmIRPConstDefs::AttributeNameValue::PERCEIVED\_SEVERITY;

const string ALARM\_ID =

AlarmIRPConstDefs::AttributeNameValue::ALARM\_ID;

};

interface NotifyComments: NotificationIRPNotifications::Notify

{

const string EVENT\_TYPE = "notifyComments";

const string PROBABLE\_CAUSE =

AlarmIRPConstDefs::AttributeNameValue::PROBABLE\_CAUSE;

const string PERCEIVED\_SEVERITY =

AlarmIRPConstDefs::AttributeNameValue::PERCEIVED\_SEVERITY;

const string ALARM\_ID =

AlarmIRPConstDefs::AttributeNameValue::ALARM\_ID;

/\*\*

\* This constant defines the name of the comments property.

\* The data type for the value of this property

\* is AlarmIRPConstDefs::CommentSet.

\*/

const string COMMENTS =

AlarmIRPConstDefs::AttributeNameValue::COMMENTS;

};

interface NotifyPotentialFaultyAlarmList:

NotificationIRPNotifications::Notify

{

const string EVENT\_TYPE = "notifyPotentialFaultyAlarmList";

/\*\*

\* This constant defines the name of the reason property.

\* The data type for the value of this property

\* is string.

\*/

const string REASON =

AlarmIRPConstDefs::AttributeNameValue::REASON;

};

interface NotifyCorrelatedNotificationChanged:

NotificationIRPNotifications::Notify

{

const string EVENT\_TYPE = "notifyCorrelatedNotificationChanged";

const string CORRELATED\_NOTIFICATIONS =

AlarmIRPConstDefs::AttributeNameValue::

CORRELATED\_NOTIFICATIONS;

const string ALARM\_ID =

AlarmIRPConstDefs::AttributeNameValue::ALARM\_ID;

const string ROOT\_CAUSE\_INDICATOR =

AlarmIRPConstDefs::AttributeNameValue::ROOT\_CAUSE\_INDICATOR;

};

interface NotifyChangedAlarmGeneral: NotificationIRPNotifications::Notify

{

const string EVENT\_TYPE = "notifyChangedAlarmGeneral";

/\*\*

\* This constant defines the name of the probableCause property.

\* The data type for the value of this property

\* is short.

\*/

const string PROBABLE\_CAUSE =

AlarmIRPConstDefs::AttributeNameValue::PROBABLE\_CAUSE;

/\*\*

\* This constant defines the name of the

\* perceivedSeverity property.

\* The data type for the value of this property

\* is short.

\*/

const string PERCEIVED\_SEVERITY =

AlarmIRPConstDefs::AttributeNameValue::PERCEIVED\_SEVERITY;

/\*\*

\* This constant defines the name of the specificProblem

\* property. The data type for the value of this property

\* is string.

\*/

const string SPECIFIC\_PROBLEM =

AlarmIRPConstDefs::AttributeNameValue::SPECIFIC\_PROBLEM;

/\*\*

\* The two constants defines the name of the additionalInformation

\* property (the vendor-specific (VS) perceived severity and alarm type).

\* The data type for the value of this property

\* is string.

\*/

const string AI\_VS\_PERCEIVED\_SEVERITY =

AlarmIRPConstDefs::AdditionalInformation::AI\_VS\_PERCEIVED\_SEVERITY;

const string AI\_VS\_ALARM\_TYPE =

AlarmIRPConstDefs::AdditionalInformation::AI\_VS\_ALARM\_TYPE;

/\*\*

\* This constant defines the name of the

\* rootCauseIndicator property.

\* The data type for the value of this property

\* is AlarmIRPConstDefs::RootCauseIndicator.

\*/

const string ROOT\_CAUSE\_INDICATOR =

AlarmIRPConstDefs::AttributeNameValue::ROOT\_CAUSE\_INDICATOR;

/\*\*

\* This constant defines the name of the

\* correlatedNotifications property.

\* The data type for the value of this property

\* is AlarmIRPConstDefs::CorrelatedNotificationSet.

\*/

const string CORRELATED\_NOTIFICATIONS =

AlarmIRPConstDefs::AttributeNameValue::

CORRELATED\_NOTIFICATIONS;

/\*\*

\* This constant defines the name of the

\* backedUpStatus property.

\* The data type for the value of this property

\* is AlarmIRPConstDefs::BackedUpStatus.

\*/

const string BACKED\_UP\_STATUS =

AlarmIRPConstDefs::AttributeNameValue::BACKED\_UP\_STATUS;

/\*\*

\* This constant defines the name of the backUpObject property.

\* The data type for the value of this property

\* is a string carrying of DN of the back-up object.

\*/

const string BACK\_UP\_OBJECT =

AlarmIRPConstDefs::AttributeNameValue::BACK\_UP\_OBJECT;

/\*\*

\* This constant defines the name of the

\* trendIndication property.

\* The data type for the value of this property

\* is AlarmIRPConstDefs::TrendIndication.

\*/

const string TREND\_INDICATION =

AlarmIRPConstDefs::AttributeNameValue::TREND\_INDICATION;

/\*\*

\* This constant defines the name of the thresholdInfo property.

\* The data type for the value of this property

\* is AlarmIRPConstDefs::ThresholdInfo.

\*/

const string THRESHOLD\_INFO =

AlarmIRPConstDefs::AttributeNameValue::THRESHOLD\_INFO;

/\*\*

\* This constant defines the name of the

\* stateChangeDefinition property.

\* The data type for the value of this property

\* is AlarmIRPConstDefs::AttributeChangeSet.

\*/

const string STATE\_CHANGE\_DEFINITION =

AlarmIRPConstDefs::AttributeNameValue::STATE\_CHANGE\_DEFINITION;

/\*\*

\* This constant defines the name of the

\* monitoredAttributes property.

\* The data type for the value of this property

\* is AlarmIRPConstDefs::AttributeSet.

\*/

const string MONITORED\_ATTRIBUTES =

AlarmIRPConstDefs::AttributeNameValue::MONITORED\_ATTRIBUTES;

/\*\*

\* This constant defines the name of the

\* proposedRepairActions property.

\* The data type for the value of this property

\* is string.

\*/

const string PROPOSED\_REPAIR\_ACTIONS =

AlarmIRPConstDefs::AttributeNameValue::PROPOSED\_REPAIR\_ACTIONS;

/\*\*

\* This constant defines the name of the additionalText

\* property.

\* The data type for the value of this property

\* is string.

\*/

const string ADDITIONAL\_TEXT =

AlarmIRPConstDefs::AttributeNameValue::ADDITIONAL\_TEXT;

/\*\*

\* This constant defines the name of the alarmId property.

\* The data type for the value of this property

\* is string. If the string is a zero-length string or if

\* this NV pair is absent, the default semantics is that

\* alarmId is a concatenation of

\* managedObjectInstance, eventType, probableCause and

\* specificProblem, if present, of this Structured Event.

\* Since probableCause is encoded as a short, it shall be

\* converted into string before concatenation.

\* The resultant string shall not contain spaces.

\*/

const string ALARM\_ID =

AlarmIRPConstDefs::AttributeNameValue::ALARM\_ID;

/\*\*

\* This constant defines the name of the serviceUser property.

\* The data type for the value of this property

\* is string.

\*/

const string SERVICE\_USER =

AlarmIRPConstDefs::AttributeNameValue::SERVICE\_USER;

/\*\*

\* This constant defines the name of the serviceProvider

\* property.

\* The data type for the value of this property

\* is string.

\*/

const string SERVICE\_PROVIDER =

AlarmIRPConstDefs::AttributeNameValue::SERVICE\_PROVIDER;

/\*\*

\* This constant defines the name of the

\* securityAlarmDetector property.

\* The data type for the value of this property

\* is string.

\*/

const string SECURITY\_ALARM\_DETECTOR =

AlarmIRPConstDefs::AttributeNameValue::SECURITY\_ALARM\_DETECTOR;

};

/\*\*

\* This constant defines the name of the changedAlarmAttributes

\* property.

\* The data type for the value of this property

\* is a list of sequence <AttributeName, OldAttributeValue> which contains

\* the changed alarm attributes (name/value pairs) (with old values).

\*/

const string CHANGED\_ALARM\_ATTRIBUTES =

AlarmIRPConstDefs::AttributeNameValue::CHANGED\_ALARM\_ATTRIBUTES;

};

#endif // \_ALARM\_IRP\_NOTIFICATIONS\_IDL\_

Annex B (normative):  
XML Definitions

# B.0 General

This annex contains the XML Definitions for the Alarm Integration Reference Point (Alarm IRP) as it applies to Itf-N, in accordance with Alarm IRP IS definitions [4] as well as Notification Log IRP XML Definitions [13].

Apart from being used for the Notification Log, the XML definitions for Alarm IRP notifications are also used by the Alarm IRP SOAP SS.

# B.1 Architectural Features

## B.1.0 Introduction

The overall architectural feature of Alarm IRP is specified in 3G TS 32.111-2 [4]. This clause specifies features that are specific to the XML definitions.

## B.1.1 Syntax for Distinguished Names

The syntax of a Distinguished Name is defined in 3GPP TS 32.300 [6].

## B.1.2 Notification Services

This annex defines the XML syntax of Alarm IRP notifications that is to be used for the Alarm IRP SOAP Solution Set and in conjunction with Notification Log IRP XML Definitions for Notification Log IRP XML Data File and the NL IRP XML Notification Format [13].

## B.1.3 IOC Definitions

This annex defines the XML syntax for the IOC definitions of the Alarm IRP IS [4], which are used by the XML definitions for the Alarm IRP notifications and the Alarm IRP IS operations.

# B.2 Mapping

Not present in the current version of this specification.

# B.3 Solution Set definitions

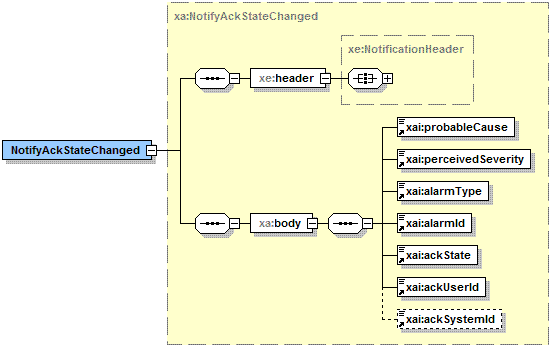
## B.3.1 XML definition structure

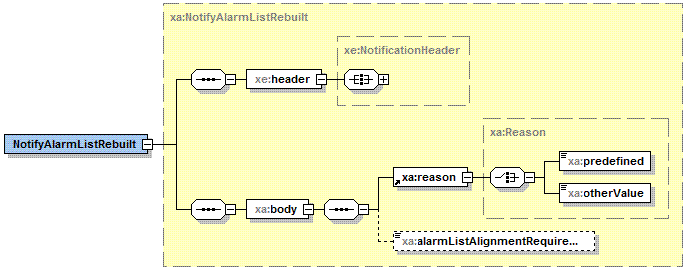
Clause B.3.2 provides a graphical representation of the XML elements.

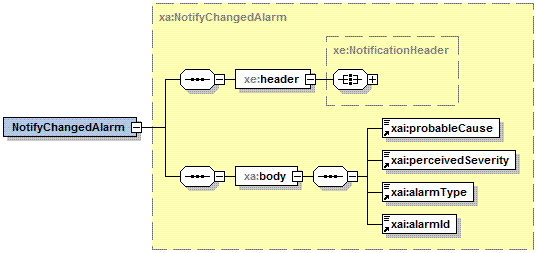
Clause B.3.3 provides XML definitions of Alarm IRP notifications as defined in [4]. These definitions are to be used for the Alarm IRP SOAP Solution Set and in conjunction with Notification Log IRP XML Definitions for Notification Log IRP XML Data File and the NL IRP XML Notification Format [13], as well as considerations for NL IRP XML File Name Conventions defined therein.

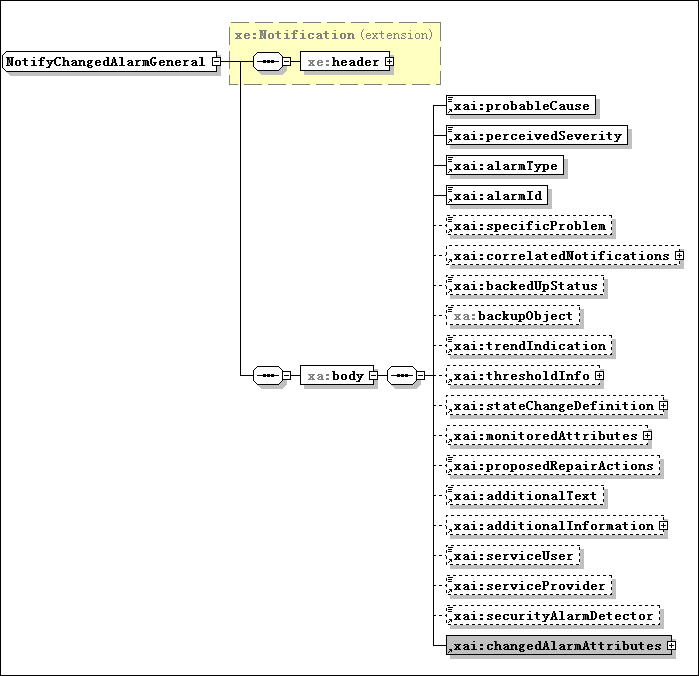
Clause B.3.4 provides XML definitions of Alarm IOC as defined in [4].

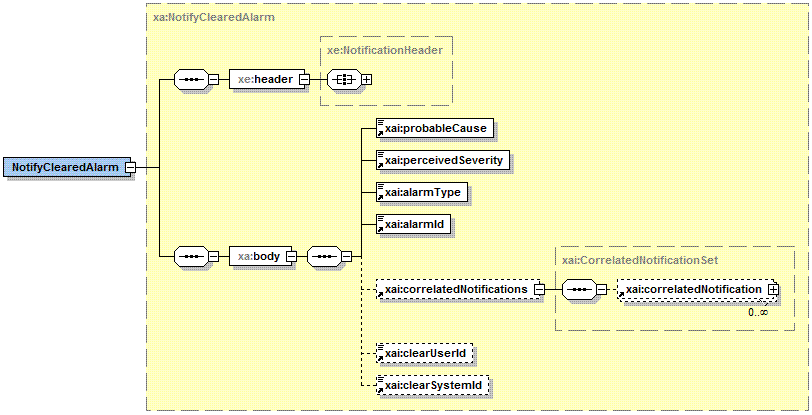
## B.3.2 Graphical Representation

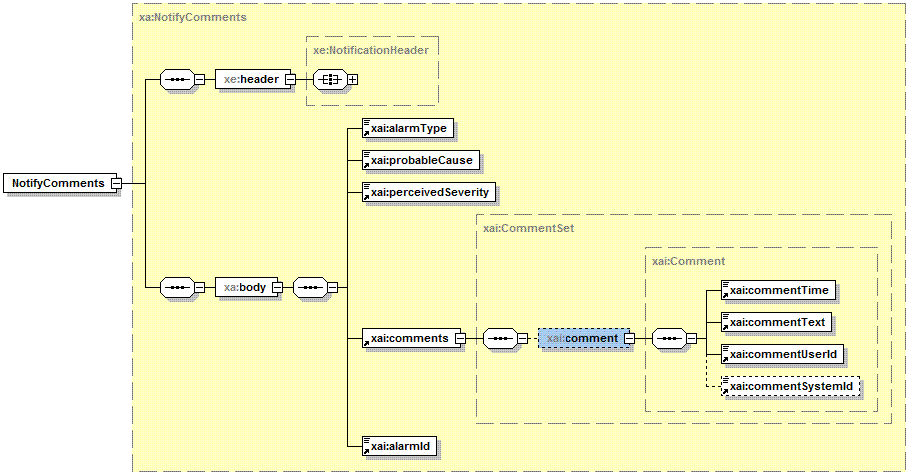


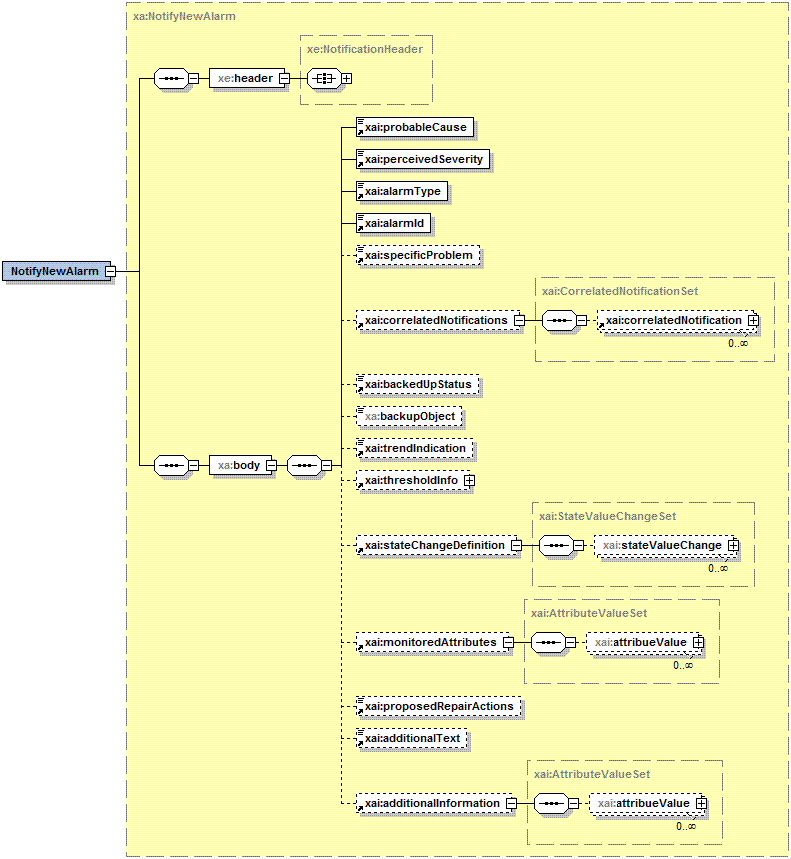


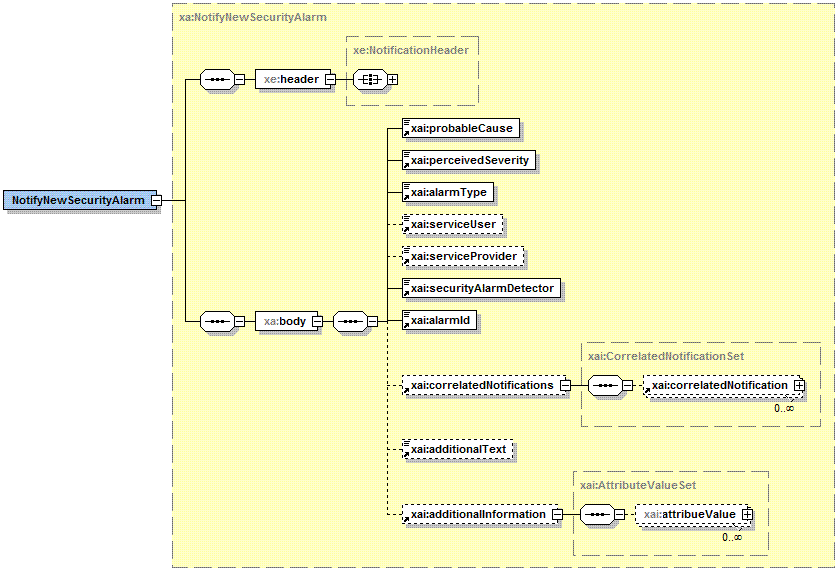


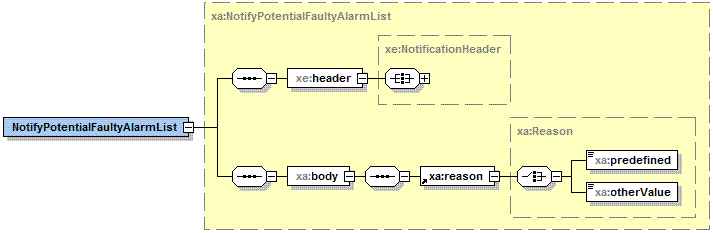


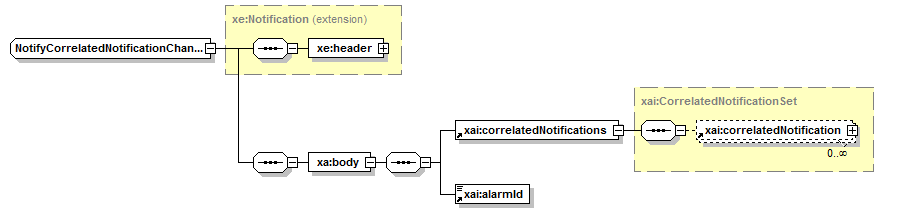












## B.3.3 XML Schema “alarmIRPNotif.xsd”

<?xml version="1.0" encoding="UTF-8"?>

<!--

3GPP TS 32.111-6 Alarm IRP Notifications XML Schema

alarmIRPNotif.xsd

-->

<schema xmlns:xa="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#alarmIRPNotif" xmlns:xai="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#alarmIRPIOCs" xmlns:xe="http://www.3gpp.org/ftp/specs/archive/32\_series/32.306#notification" xmlns:xn="http://www.3gpp.org/ftp/specs/archive/32\_series/32.626#genericNrm" xmlns:sm="http://www.3gpp.org/ftp/specs/archive/32\_series/32.676#stateManagementIRP" xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#alarmIRPNotif" elementFormDefault="qualified" attributeFormDefault="unqualified">

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#alarmIRPIOCs"/>

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.306#notification"/>

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.626#genericNrm"/>

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.676#stateManagementIRP"/>

<simpleType name="AlarmListAlignmentRequirement">

<restriction base="string">

<enumeration value="Alignment Required"/>

<enumeration value="Alignment Not Required"/>

</restriction>

</simpleType>

<simpleType name="AlarmListRebuiltReason">

<restriction base="string">

<enumeration value="Agent NE Communications Error"/>

<enumeration value="Agent Restarts"/>

<enumeration value="Indeterminate"/>

</restriction>

</simpleType>

<complexType name="Reason">

<choice>

<element name="predefined" type="xa:AlarmListRebuiltReason"/>

<element name="otherValue" type="string"/>

</choice>

</complexType>

<element name="reason" type="xa:Reason"/>

<complexType name="NotifyAckStateChanged">

<complexContent>

<extension base="xe:Notification">

<sequence>

<element name="body">

<complexType>

<sequence>

<element ref="xai:probableCause"/>

<element ref="xai:perceivedSeverity"/>

<element ref="xai:alarmType"/>

<element ref="xai:alarmId"/>

<element ref="xai:ackState"/>

<element ref="xai:ackUserId"/>

<element ref="xai:ackSystemId" minOccurs="0"/>

</sequence>

</complexType>

</element>

</sequence>

</extension>

</complexContent>

</complexType>

<complexType name="NotifyAlarmListRebuilt">

<complexContent>

<extension base="xe:Notification">

<sequence>

<element name="body">

<complexType>

<sequence>

<element ref="xa:reason"/>

<element name="alarmListAlignmentRequirement" type="xa:AlarmListAlignmentRequirement" minOccurs="0"/>

</sequence>

</complexType>

</element>

</sequence>

</extension>

</complexContent>

</complexType>

<complexType name="NotifyChangedAlarm">

<complexContent>

<extension base="xe:Notification">

<sequence>

<element name="body">

<complexType>

<sequence>

<element ref="xai:probableCause"/>

<element ref="xai:perceivedSeverity"/>

<element ref="xai:alarmType"/>

<element ref="xai:alarmId"/>

</sequence>

</complexType>

</element>

</sequence>

</extension>

</complexContent>

</complexType>

<complexType name="NotifyChangedAlarmGeneral">

<complexContent>

<extension base="xe:Notification">

<sequence>

<element name="body">

<complexType>

<sequence>

<element ref="xai:probableCause"/>

<element ref="xai:perceivedSeverity"/>

<element ref="xai:alarmType"/>

<element ref="xai:alarmId"/>

<element ref="xai:specificProblem" minOccurs="0"/>

<element ref="xai:correlatedNotifications" minOccurs="0"/>

<element ref="xai:backedUpStatus" minOccurs="0"/>

<element name="backupObject" type="xn:dn" minOccurs="0"/>

<element ref="xai:trendIndication" minOccurs="0"/>

<element ref="xai:thresholdInfo" minOccurs="0"/>

<element ref="xai:stateChangeDefinition" minOccurs="0"/>

<element ref="xai:monitoredAttributes" minOccurs="0"/>

<element ref="xai:proposedRepairActions" minOccurs="0"/>

<element ref="xai:additionalText" minOccurs="0"/>

<element ref="xai:additionalInformation" minOccurs="0"/>

<element ref="xai:serviceUser" minOccurs="0"/>

<element ref="xai:serviceProvider" minOccurs="0"/>

<element ref="xai:securityAlarmDetector" minOccurs="0"/>

<element ref="xai:changedAlarmAttributes"/>

</sequence>

</complexType>

</element>

</sequence>

</extension>

</complexContent>

</complexType>

<complexType name="NotifyClearedAlarm">

<complexContent>

<extension base="xe:Notification">

<sequence>

<element name="body">

<complexType>

<sequence>

<element ref="xai:probableCause"/>

<element ref="xai:perceivedSeverity"/>

<element ref="xai:alarmType"/>

<element ref="xai:alarmId"/>

<element ref="xai:correlatedNotifications" minOccurs="0"/>

<element ref="xai:clearUserId" minOccurs="0"/>

<element ref="xai:clearSystemId" minOccurs="0"/>

</sequence>

</complexType>

</element>

</sequence>

</extension>

</complexContent>

</complexType>

<complexType name="NotifyComments">

<complexContent>

<extension base="xe:Notification">

<sequence>

<element name="body">

<complexType>

<sequence>

<element ref="xai:alarmType"/>

<element ref="xai:probableCause"/>

<element ref="xai:perceivedSeverity"/>

<element ref="xai:comments"/>

<element ref="xai:alarmId"/>

</sequence>

</complexType>

</element>

</sequence>

</extension>

</complexContent>

</complexType>

<complexType name="NotifyNewAlarm">

<complexContent>

<extension base="xe:Notification">

<sequence>

<element name="body">

<complexType>

<sequence>

<element ref="xai:probableCause"/>

<element ref="xai:perceivedSeverity"/>

<element ref="xai:alarmType"/>

<element ref="xai:alarmId"/>

<element ref="xai:specificProblem" minOccurs="0"/>

<element ref="xai:correlatedNotifications" minOccurs="0"/>

<element ref="xai:backedUpStatus" minOccurs="0"/>

<element name="backupObject" type="xn:dn" minOccurs="0"/>

<element ref="xai:trendIndication" minOccurs="0"/>

<element ref="xai:thresholdInfo" minOccurs="0"/>

<element ref="xai:stateChangeDefinition" minOccurs="0"/>

<element ref="xai:monitoredAttributes" minOccurs="0"/>

<element ref="xai:proposedRepairActions" minOccurs="0"/>

<element ref="xai:additionalText" minOccurs="0"/>

<element ref="xai:additionalInformation" minOccurs="0"/>

</sequence>

</complexType>

</element>

</sequence>

</extension>

</complexContent>

</complexType>

<complexType name="NotifyNewSecurityAlarm">

<complexContent>

<extension base="xe:Notification">

<sequence>

<element name="body">

<complexType>

<sequence>

<element ref="xai:probableCause"/>

<element ref="xai:perceivedSeverity"/>

<element ref="xai:alarmType"/>

<element ref="xai:serviceUser"/>

<element ref="xai:serviceProvider"/>

<element ref="xai:securityAlarmDetector"/>

<element ref="xai:alarmId"/>

<element ref="xai:correlatedNotifications" minOccurs="0"/>

<element ref="xai:additionalText" minOccurs="0"/>

<element ref="xai:additionalInformation" minOccurs="0"/>

</sequence>

</complexType>

</element>

</sequence>

</extension>

</complexContent>

</complexType>

<complexType name="NotifyPotentialFaultyAlarmList">

<complexContent>

<extension base="xe:Notification">

<sequence>

<element name="body">

<complexType>

<sequence>

<element ref="xa:reason"/>

</sequence>

</complexType>

</element>

</sequence>

</extension>

</complexContent>

</complexType>

<!-- Notification definitions -->

<complexType name="NotifyCorrelatedNotificationChanged">

<complexContent>

<extension base="xe:Notification">

<sequence>

<element name="body">

<complexType>

<sequence>

<element ref="xai:correlatedNotifications"/>

<element ref="xai:alarmId"/>

<element ref="xai:rootCauseIndicator"/>

</sequence>

</complexType>

</element>

</sequence>

</extension>

</complexContent>

</complexType>

<element name="NotifyAckStateChanged" type="xa:NotifyAckStateChanged"/>

<element name="NotifyAlarmListRebuilt" type="xa:NotifyAlarmListRebuilt"/>

<element name="NotifyChangedAlarm" type="xa:NotifyChangedAlarm"/>

<element name="NotifyChangedAlarmGeneral" type="xa:NotifyChangedAlarmGeneral"/>

<element name="NotifyClearedAlarm" type="xa:NotifyClearedAlarm"/>

<element name="NotifyComments" type="xa:NotifyComments"/>

<element name="NotifyNewAlarm" type="xa:NotifyNewAlarm"/>

<element name="NotifyNewSecurityAlarm" type="xa:NotifyNewSecurityAlarm"/>

<element name="NotifyPotentialFaultyAlarmList" type="xa:NotifyPotentialFaultyAlarmList"/>

<element name="NotifyCorrelatedNotificationChanged" type="xa:NotifyCorrelatedNotificationChanged"/>

</schema>

## B.3.4 XML Schema “alarmIRPIOCs.xsd”

<?xml version="1.0" encoding="UTF-8"?>

<!--

3GPP TS 32.111-6 Alarm IRP IOC XML Schema

alarmIRPIOCs.xsd

-->

<schema xmlns:xai="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#alarmIRPIOCs" xmlns:xe="http://www.3gpp.org/ftp/specs/archive/32\_series/32.306#notification" xmlns:xn="http://www.3gpp.org/ftp/specs/archive/32\_series/32.626#genericNrm" xmlns:sm="http://www.3gpp.org/ftp/specs/archive/32\_series/32.676#stateManagementIRP" xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#alarmIRPIOCs" elementFormDefault="qualified" attributeFormDefault="unqualified">

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.306#notification"/>

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.626#genericNrm"/>

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.676#stateManagementIRP"/>

<!-- Type definitions -->

<simpleType name="AckState">

<restriction base="string">

<enumeration value="Acknowledged"/>

<enumeration value="Unacknowledged"/>

</restriction>

</simpleType>

<complexType name="CorrelatedNotification">

<sequence>

<element name="source" type="xn:dn"/>

<element name="notificationIdSet">

<complexType>

<sequence>

<element name="notificationId" type="xe:NotificationId" minOccurs="0" maxOccurs="unbounded"/>

</sequence>

</complexType>

</element>

</sequence>

</complexType>

<complexType name="CorrelatedNotificationSet">

<sequence>

<element ref="xai:correlatedNotification" minOccurs="0" maxOccurs="unbounded"/>

</sequence>

</complexType>

<simpleType name="EventType">

<restriction base="string">

<enumeration value="Communications Alarm"/>

<enumeration value="Processing Error Alarm"/>

<enumeration value="Environmental Alarm"/>

<enumeration value="Quality Of Service Alarm"/>

<enumeration value="Equipment Alarm"/>

<enumeration value="Integrity Violation"/>

<enumeration value="Operational Violation"/>

<enumeration value="Physical Violation"/>

<enumeration value="Security Service Or Mechanism Violation"/>

<enumeration value="Time Domain Violation"/>

</restriction>

</simpleType>

<simpleType name="PerceivedSeverity">

<restriction base="string">

<enumeration value="Critical"/>

<enumeration value="Major"/>

<enumeration value="Minor"/>

<enumeration value="Warning"/>

<enumeration value="Indeterminate"/>

<enumeration value="Cleared"/>

</restriction>

</simpleType>

<simpleType name="ProbableCause">

<restriction base="string">

<enumeration value="Indeterminate"/>

<enumeration value="Alarm Indication Signal"/>

<enumeration value="Call Setup Failure"/>

<enumeration value="Degraded Signal"/>

<enumeration value="Far End Receiver Failure"/>

<enumeration value="Framing Error"/>

<enumeration value="Loss Of Frame"/>

<enumeration value="Loss Of Pointer"/>

<enumeration value="Loss Of Signal"/>

<enumeration value="Payload Type Mismatch"/>

<enumeration value="Transmission Error"/>

<enumeration value="Remote Alarm Interface"/>

<enumeration value="Excessive Bit Error Rate"/>

<enumeration value="Path Trace Mismatch"/>

<enumeration value="Unavailable"/>

<enumeration value="Signal Label Mismatch"/>

<enumeration value="Loss Of Multi Frame"/>

<enumeration value="Communications Receive Failure"/>

<enumeration value="Communications Transmit Failure"/>

<enumeration value="Modulation Failure"/>

<enumeration value="Demodulation Failure"/>

<enumeration value="Back Plane Failure"/>

<enumeration value="Data Set Problem"/>

<enumeration value="Equipment Identifier Duplication"/>

<enumeration value="External Device Problem"/>

<enumeration value="Line Card Problem"/>

<enumeration value="Multiplexer Problem"/>

<enumeration value="NE Identifier Duplication"/>

<enumeration value="Power Problem"/>

<enumeration value="Processor Problem"/>

<enumeration value="Protection Path Failure"/>

<enumeration value="Receiver Failure"/>

<enumeration value="Replaceable Unit Missing"/>

<enumeration value="Replaceable Unit Type Mismatch"/>

<enumeration value="Synchronization Source Mismatch"/>

<enumeration value="Terminal Problem"/>

<enumeration value="Timing Problem"/>

<enumeration value="Transmitter Failure"/>

<enumeration value="Trunk Card Problem"/>

<enumeration value="Replaceable Unit Problem"/>

<enumeration value="Real Time Clock Failure"/>

<enumeration value="Protection Mechanism Failure"/>

<enumeration value="Protecting Resource Failure"/>

<enumeration value="Air Compressor Failure"/>

<enumeration value="Air Conditioning Failure"/>

<enumeration value="Air Dryer Failure"/>

<enumeration value="Battery Discharging"/>

<enumeration value="Battery Failure"/>

<enumeration value="Commercial Power Failure"/>

<enumeration value="Cooling Fan Failure"/>

<enumeration value="Engine Failure"/>

<enumeration value="Fire Detector Failure"/>

<enumeration value="Fuse Failure"/>

<enumeration value="Generator Failure"/>

<enumeration value="Low Battery Threshold"/>

<enumeration value="Pump Failure"/>

<enumeration value="Rectifier Failure"/>

<enumeration value="Rectifier High Voltage"/>

<enumeration value="Rectifier Low F Voltage"/>

<enumeration value="Ventilation System Failure"/>

<enumeration value="Enclosure Door Open"/>

<enumeration value="Explosive Gas"/>

<enumeration value="Fire"/>

<enumeration value="Flood"/>

<enumeration value="High Humidity"/>

<enumeration value="High Temperature"/>

<enumeration value="High Wind"/>

<enumeration value="Ice Build Up"/>

<enumeration value="Intrusion Detection"/>

<enumeration value="Low Fuel"/>

<enumeration value="Low Humidity"/>

<enumeration value="Low Cable Pressure"/>

<enumeration value="Low Temperature"/>

<enumeration value="Low Water"/>

<enumeration value="Smoke"/>

<enumeration value="Toxic Gas"/>

<enumeration value="External Point Failure"/>

<enumeration value="Storage Capacity Problem"/>

<enumeration value="Memory Mismatch"/>

<enumeration value="Corrupt Data"/>

<enumeration value="Out Of CPU Cycles"/>

<enumeration value="Software Environment Problem"/>

<enumeration value="Software Download Failure"/>

<enumeration value="Loss Of Real Time"/>

<enumeration value="Reinitialized"/>

<enumeration value="Excessive Error Rate"/>

<enumeration value="Adapter Error"/>

<enumeration value="Application Subsystem Failure"/>

<enumeration value="Bandwidth Reduced"/>

<enumeration value="Communications Protocol Error"/>

<enumeration value="Communications Subsystem Failure"/>

<enumeration value="Configuration Or Customizing Error"/>

<enumeration value="Congestion"/>

<enumeration value="CPU Cycles Limit Exceeded"/>

<enumeration value="Data Set Or Modem Error"/>

<enumeration value="DTE DCE Interface Error"/>

<enumeration value="Equipment Malfunction"/>

<enumeration value="Excessive Vibration"/>

<enumeration value="File Error"/>

<enumeration value="Heating Or Ventilation Or Cooling System Problem"/>

<enumeration value="Humidity Unacceptable"/>

<enumeration value="Input Output Device Error"/>

<enumeration value="Input Device Error"/>

<enumeration value="LAN Error"/>

<enumeration value="Leak Detected"/>

<enumeration value="Local Node Transmission Error"/>

<enumeration value="Material Supply Exhausted"/>

<enumeration value="Out Of Memory"/>

<enumeration value="Output Device Error"/>

<enumeration value="Performance Degraded"/>

<enumeration value="Pressure Unacceptable"/>

<enumeration value="Queue Size Exceeded"/>

<enumeration value="Receive Failure"/>

<enumeration value="Remote Node Transmission Error"/>

<enumeration value="Resource At Or Nearing Capacity"/>

<enumeration value="Response Time Excessive"/>

<enumeration value="Retransmission Rate Excessive"/>

<enumeration value="Software Error"/>

<enumeration value="Software Program Abnormally Terminated"/>

<enumeration value="Software Program Error"/>

<enumeration value="Temperature Unacceptable"/>

<enumeration value="Threshold Crossed"/>

<enumeration value="Toxic Leak Detected"/>

<enumeration value="Transmit Failure"/>

<enumeration value="Underlying Resource Unavailable"/>

<enumeration value="Version Mismatch"/>

<enumeration value="A BIS To BTS Interface Failure"/>

<enumeration value="A BIS To TRX Interface Failure"/>

<enumeration value="Antenna Problem"/>

<enumeration value="Battery Breakdown"/>

<enumeration value="Battery Charging Fault"/>

<enumeration value="Clock Synchronization Problem"/>

<enumeration value="Combiner Problem"/>

<enumeration value="Disk Problem"/>

<enumeration value="Excessive Receiver Temperature"/>

<enumeration value="Excessive Transmitter Output Power"/>

<enumeration value="Excessive Transmitter Temperature"/>

<enumeration value="Frequency Hopping Degraded"/>

<enumeration value="Frequency Hopping Failure"/>

<enumeration value="Frequency Redefinition Failed"/>

<enumeration value="Line Interface Failure"/>

<enumeration value="Link Failure"/>

<enumeration value="Loss Of Synchronization"/>

<enumeration value="Lost Redundancy"/>

<enumeration value="Mains Breakdown With Battery Backup"/>

<enumeration value="Mains Breakdown Without Battery Backup"/>

<enumeration value="Power Supply Failure"/>

<enumeration value="Receiver Antenna Fault"/>

<enumeration value="Receiver Multicoupler Failure"/>

<enumeration value="Reduced Transmitter Output Power"/>

<enumeration value="Signal Quality Evaluation Fault"/>

<enumeration value="Timeslot Hardware Failure"/>

<enumeration value="Transceiver Problem"/>

<enumeration value="Transcoder Problem"/>

<enumeration value="Transcoder Or Rate Adapter Problem"/>

<enumeration value="Transmitter Antenna Failure"/>

<enumeration value="Transmitter Antenna Not Adjusted"/>

<enumeration value="Transmitter Low Voltage Or Current"/>

<enumeration value="Transmitter Off Frequency"/>

<enumeration value="Database Inconsistency"/>

<enumeration value="File System Call Unsuccessful"/>

<enumeration value="Input Parameter Out Of Range"/>

<enumeration value="Invalid Parameter"/>

<enumeration value="Invalid Pointer"/>

<enumeration value="Message Not Expected"/>

<enumeration value="Message Not Initialised"/>

<enumeration value="Message Out Of Sequence"/>

<enumeration value="System Call Unsuccessful"/>

<enumeration value="Timeout Expired"/>

<enumeration value="Variable Out Of Range"/>

<enumeration value="Watch Dog Timer Expired"/>

<enumeration value="Cooling System Failure"/>

<enumeration value="External Equipment Failure"/>

<enumeration value="External Power Supply Failure"/>

<enumeration value="External Transmission Device Failure"/>

<enumeration value="Reduced Alarm Reporting"/>

<enumeration value="Reduced Event Reporting"/>

<enumeration value="Recuced Logging Capability"/>

<enumeration value="System Resources Overload"/>

<enumeration value="Broadcast Channel Failure"/>

<enumeration value="Call Establishment Error"/>

<enumeration value="Invalid Message Received"/>

<enumeration value="Invalid MSU Received"/>

<enumeration value="LAPD Link Protocol Failure"/>

<enumeration value="Local Alarm Indication"/>

<enumeration value="Remote Alarm Indication"/>

<enumeration value="Routing Failure"/>

<enumeration value="SS7 Protocol Failure"/>

<enumeration value="Authentication Failure"/>

<enumeration value="Breach Of Confidentiality"/>

<enumeration value="Cable Tamper"/>

<enumeration value="Delayed Information"/>

<enumeration value="Denial Of Service"/>

<enumeration value="Duplicate Information"/>

<enumeration value="Information Missing"/>

<enumeration value="Information Modification Detected"/>

<enumeration value="Information Out Of Sequence"/>

<enumeration value="Key Expired"/>

<enumeration value="Non Repudiation Failure"/>

<enumeration value="Out Of Hours Activity"/>

<enumeration value="Out Of Service"/>

<enumeration value="Procedural Error"/>

<enumeration value="Unauthorised Access Attempt"/>

<enumeration value="Unexpected Information"/>

<enumeration value="Unspecified Reason"/>

</restriction>

</simpleType>

<simpleType name="ThresholdIndicator">

<restriction base="string">

<enumeration value="Up"/>

<enumeration value="Down"/>

</restriction>

</simpleType>

<simpleType name="TrendIndicator">

<restriction base="string">

<enumeration value="Less Severe"/>

<enumeration value="No Change"/>

<enumeration value="More Severe"/>

</restriction>

</simpleType>

<simpleType name="AttributeType">

<restriction base="string">

<enumeration value="string"/>

<enumeration value="integer"/>

<enumeration value="unsignedInt"/>

<enumeration value="boolean"/>

<enumeration value="dateTime"/>

<enumeration value="base64Binary"/>

</restriction>

</simpleType>

<complexType name="AttributeValue">

<sequence>

<element name="attributeName" type="string"/>

<element name="attributeValue" type="anySimpleType"/>

<element name="attributeType" type="xai:AttributeType" minOccurs="0"/>

</sequence>

</complexType>

<complexType name="AttributeValueSet">

<sequence>

<element name="attribueValue" type="xai:AttributeValue" minOccurs="0" maxOccurs="unbounded"/>

</sequence>

</complexType>

<complexType name="AttributeValueChange">

<sequence>

<element name="attributeName" type="string"/>

<element name="oldValue" type="anySimpleType"/>

<element name="newValue" type="anySimpleType"/>

</sequence>

</complexType>

<complexType name="AttributeValueChangeSet">

<sequence>

<element name="attributeValueChange" type="xai:AttributeValueChange" minOccurs="0" maxOccurs="unbounded"/>

</sequence>

</complexType>

<complexType name="AlarmAttributeValueChange">

<sequence>

<element name="attributeName" type="string"/>

<element name="oldValue" type="anySimpleType"/>

</sequence>

</complexType>

<complexType name="ChangedAlarmAttributes">

<sequence>

<element name="attributeValueChange" type="xai:AlarmAttributeValueChange" minOccurs="0" maxOccurs="unbounded"/>

</sequence>

</complexType>

<complexType name="StateValueChange">

<sequence>

<element name="attributeName" type="xai:StateAttributeNames"/>

<element name="oldValue" type="xai:StateAttributeType" minOccurs="0"/>

<element name="newValue" type="xai:StateAttributeType"/>

</sequence>

</complexType>

<simpleType name="StateAttributeNames">

<restriction base="string">

<enumeration value="Operational State"/>

<enumeration value="Usage State"/>

<enumeration value="Administrative State"/>

<enumeration value="Alarm Status"/>

<enumeration value="Procedural Status"/>

<enumeration value="Availability Status"/>

<enumeration value="Control Status"/>

<enumeration value="Standby Status"/>

<enumeration value="Unknown Status"/>

</restriction>

</simpleType>

<complexType name="StateAttributeType">

<choice>

<element name="operationalState" type="sm:operationalStateType"/>

<element name="usageState" type="sm:usageStateType"/>

<element name="administrativeState" type="sm:administrativeStateType"/>

<element name="alarmStatus" type="sm:alarmStatusType"/>

<element name="proceduralStatus" type="sm:proceduralStatusType"/>

<element name="availabilityStatus" type="sm:availabilityStatusType"/>

<element name="controlStatus" type="sm:controlStatusType"/>

<element name="standbyStatus" type="sm:standbyStatusType"/>

<element name="unknownStatus" type="sm:unknownStatusType"/>

</choice>

</complexType>

<complexType name="StateValueChangeSet">

<sequence>

<element name="stateValueChange" type="xai:StateValueChange" minOccurs="0" maxOccurs="unbounded"/>

</sequence>

</complexType>

<complexType name="Comment">

<sequence>

<element ref="xai:commentTime"/>

<element ref="xai:commentText"/>

<element ref="xai:commentUserId"/>

<element ref="xai:commentSystemId" minOccurs="0"/>

</sequence>

</complexType>

<complexType name="CommentSet">

<sequence>

<element name="comment" type="xai:Comment" minOccurs="0"/>

</sequence>

</complexType>

<simpleType name="NotificationType">

<restriction base="string">

<enumeration value="notifyNewAlarm"/>

<enumeration value="notifyChangedAlarm"/>

<enumeration value="notifyChangedAlarmGeneral"/>

<enumeration value="notifyClearedAlarm"/>

</restriction>

</simpleType>

<complexType name="NotificationIdSet">

<sequence>

<element name="notificationId" type="xe:NotificationId" minOccurs="0" maxOccurs="unbounded"/>

</sequence>

</complexType>

<complexType name="ThresholdInfo">

<sequence>

<element name="attributeId" type="string"/>

<element name="observedValue" type="float"/>

<element name="thresholdLevel" type="xai:ThresholdLevel" minOccurs="0"/>

<element name="armTime" type="dateTime"/>

</sequence>

</complexType>

<complexType name="ThresholdLevel">

<sequence>

<element name="indication" type="xai:ThresholdIndicator"/>

<element name="low" type="float" minOccurs="0"/>

<element name="high" type="float"/>

</sequence>

</complexType>

<!-- Attributes of the AlarmInformation IOC -->

<element name="alarmId" type="string"/>

<element name="alarmRaisedTime" type="dateTime"/>

<element name="alarmClearedTime" type="dateTime"/>

<element name="alarmChangedTime" type="dateTime"/>

<element name="eventType" type="xai:EventType"/>

<element name="alarmType" substitutionGroup="xai:eventType"/>

<element name="probableCause" type="xai:ProbableCause"/>

<element name="perceivedSeverity" type="xai:PerceivedSeverity"/>

<element name="specificProblem" type="string"/>

<element name="backedUpStatus" type="boolean"/>

<element name="trendIndication" type="xai:TrendIndicator"/>

<element name="thresholdInfo" type="xai:ThresholdInfo"/>

<element name="stateChangeDefinition" type="xai:StateValueChangeSet"/>

<element name="monitoredAttributes" type="xai:AttributeValueSet"/>

<element name="proposedRepairActions" type="string"/>

<element name="additionalText" type="string"/>

<element name="additionalInformation" type="xai:AttributeValueSet"/>

<element name="ackTime" type="dateTime"/>

<element name="ackUserId" type="string"/>

<element name="ackSystemId" type="string"/>

<element name="ackState" type="xai:AckState"/>

<element name="clearUserId" type="string"/>

<element name="clearSystemId" type="string"/>

<element name="serviceUser" type="string"/>

<element name="serviceProvider" type="string"/>

<element name="securityAlarmDetector" type="string"/>

<element name="rootCauseIndicator" type="boolean"/>

<!-- AlarmList entity for alarms not related to security -->

<complexType name="NonSecurityAlarm">

<sequence>

<element name="notificationType" type="xai:NotificationType"/>

<element ref="xai:alarmType"/>

<element ref="xai:objectClass"/>

<element ref="xai:objectInstance"/>

<element name="notificationId" type="xe:NotificationId" minOccurs="0"/>

<element name="eventTime" type="dateTime" minOccurs="0"/>

<element ref="xn:systemDN" minOccurs="0"/>

<element ref="xai:alarmId"/>

<element ref="xai:alarmRaisedTime"/>

<element ref="xai:alarmClearedTime"/>

<element ref="xai:alarmChangedTime" minOccurs="0"/>

<element ref="xai:probableCause"/>

<element ref="xai:perceivedSeverity"/>

<element ref="xai:rootCauseIndicator" minOccurs="0"/>

<element ref="xai:specificProblem" minOccurs="0"/>

<element ref="xai:backedUpStatus" minOccurs="0"/>

<element ref="xai:trendIndication" minOccurs="0"/>

<element ref="xai:thresholdInfo" minOccurs="0"/>

<element ref="xai:stateChangeDefinition" minOccurs="0"/>

<element ref="xai:monitoredAttributes" minOccurs="0"/>

<element ref="xai:proposedRepairActions" minOccurs="0"/>

<element ref="xai:additionalText" minOccurs="0"/>

<element ref="xai:additionalInformation" minOccurs="0"/>

<element ref="xai:ackTime"/>

<element ref="xai:ackUserId"/>

<element ref="xai:ackSystemId" minOccurs="0"/>

<element ref="xai:ackState"/>

<element ref="xai:clearUserId" minOccurs="0"/>

<element ref="xai:clearSystemId" minOccurs="0"/>

<element name="backUpObject" type="xn:dn" minOccurs="0"/>

<element ref="xai:correlatedNotifications" minOccurs="0"/>

<element ref="xai:comments"/>

</sequence>

</complexType>

<!-- AlarmList entity for alarms related to security -->

<complexType name="SecurityAlarm">

<sequence>

<element name="notificationType" type="xai:NotificationType"/>

<element ref="xai:alarmType"/>

<element ref="xai:objectClass"/>

<element ref="xai:objectInstance"/>

<element name="notificationId" type="xe:NotificationId" minOccurs="0"/>

<element name="eventTime" type="dateTime" minOccurs='0'/>

<element ref="xn:systemDN" minOccurs="0"/>

<element ref="xai:alarmId"/>

<element ref="xai:alarmRaisedTime"/>

<element ref="xai:alarmClearedTime"/>

<element ref="xai:alarmChangedTime" minOccurs="0"/>

<element ref="xai:probableCause"/>

<element ref="xai:perceivedSeverity"/>

<element ref="xai:rootCauseIndicator" minOccurs="0"/>

<element ref="xai:specificProblem" minOccurs="0"/>

<element ref="xai:additionalText" minOccurs="0"/>

<element ref="xai:additionalInformation" minOccurs="0"/>

<element ref="xai:ackTime"/>

<element ref="xai:ackUserId"/>

<element ref="xai:ackSystemId" minOccurs="0"/>

<element ref="xai:ackState"/>

<element ref="xai:clearUserId" minOccurs="0"/>

<element ref="xai:clearSystemId" minOccurs="0"/>

<element ref="xai:correlatedNotifications" minOccurs="0"/>

<element ref="xai:comments"/>

<element ref="xai:serviceUser"/>

<element ref="xai:serviceProvider"/>

<element ref="xai:securityAlarmDetector"/>

</sequence>

</complexType>

<!-- Attributes of the Comment IOC -->

<element name="commentTime" type="dateTime"/>

<element name="commentText" type="string"/>

<element name="commentUserId" type="string"/>

<element name="commentSystemId" type="string"/>

<!-- A list of comment IOC instances -->

<element name="comments" type="xai:CommentSet"/>

<!-- Attributes of the CorrelatedNotification IOC -->

<element name="source" type="xn:dn"/>

<element name="notificationIdSet" type="xai:NotificationIdSet"/>

<!-- A correlatedNotification IOC instance -->

<element name="correlatedNotification" type="xai:CorrelatedNotification"/>

<!-- A set of correlatedNotification IOC instance -->

<element name="correlatedNotifications" type="xai:CorrelatedNotificationSet"/>

<!-- A set of changedAlarmAttributes instance for notifyChangedAlarmGeneral-->

<element name="changedAlarmAttributes" type="xai:ChangedAlarmAttributes"/>

<!-- MonitoredEntity IOC attributes; inherits from Top in order to obtain NRM IOC properties -->

<element name="objectClass" type="string"/>

<element name="objectInstance" type="xn:dn"/>

</schema>

Annex C (normative):  
SOAP Solution Set

# C.0 General

This annex specifies the SOAP Solution Set for the IRP whose semantics are specified in Alarm IRP: Information Service (3GPP TS 32.111-2 [4]).

# C.1 Architectural features

## C.1.0 Introduction

The overall architectural feature of the Alarm IRP is specified in 3GPP TS 32.111-2 [4]. This clause specifies features that are specific to the SOAP solution set.

## C.1.1 Syntax for Distinguished Names

The syntax of a Distinguished Name is defined in 3GPP TS 32.300 [6].

## C.1.2 Notification Services

The Alarm IRP SOAP SS uses the Notification IRP SOAP SS of 3GPP TS 32.306 [8]. The IRPAgent shall support the push interface model, which means that the IRPAgent sends alarm notifications to the IRPManager as soon as new events occur. The IRPManager does not need to check ("pull") for events.

Relevant definitions are imported from the Alarm IRP XML definitions in Annex B.

## C.1.3 Supported W3C specifications

The SOAP 1.1 specification [16] and WSDL 1.1 specification [18] are supported.

The SOAP 1.2 specification [17] is supported optionally.

This specification uses "document" style in WSDL file.

This specification uses "literal" encoding style in WSDL file.

The filter language used in the SS is the XPath Language (see W3C XPath 1.0 specification [19]). IRPAgents may throw a FilterComplexityLimit fault when a given filter is too complex.

## C.1.4 Prefixes and namespaces

This specification uses a number of namespace prefixes throughout that are listed in Table C.1.4.

Table C.1.4: Prefixes and Namespaces used in this specification

|  |  |
| --- | --- |
| **PREFIX** | **NAMESPACE** |
| (no prefix) | http://schemas.xmlsoap.org/wsdl/ |
| soap | http://schemas.xmlsoap.org/wsdl/soap/ |
| alarmIRPSystem | http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#AlarmIRPSystem |
| alarmIRPData | http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#AlarmIRPData |
| xai | http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#alarmIRPIOCs |
| xn | http://www.3gpp.org/ftp/specs/archive/32\_series/32.626#genericNrm |
| genericIRPSystem | http://www.3gpp.org/ftp/specs/archive/32\_series/32.316#GenericIRPSystem |
| ntfIRPNtfSystem | http://www.3gpp.org/ftp/specs/archive/32\_series/32.306#NotificationIRPNtfSystem |

# C.2 Mapping

## C.2.1 Operation and notification mapping

The Alarm IRP IS (3GPP TS 32.111-2 [4]) defines the operations and their semantics.

Table C.2.1 maps the operations defined in the Alarm IRP IS to their equivalent types, messages, port type operation, and binding operation in this Solution Set (SS).

Table C.2.1 also maps the notifications of the Alarm IRP IS, as well as inherited operations.

Table C.2.1 also qualifies if an operation is Mandatory (M) or Optional (O).

Table C.2.1: Mapping from IS Operation to SS Equivalents

|  |  |  |  |
| --- | --- | --- | --- |
| IS Operation in 3GPP TS 32.111-2 [4] | SS: Operation of WSDL port type and WSDL binding | SS: Port of AlarmIRPService | Qualifier |
| acknowledgeAlarms | acknowledgeAlarms (note 1) | AlarmIRPPort | M |
| getAlarmList | getAlarmList (note 1) | AlarmIRPPort | M |
| getAlarmCount | getAlarmCount (note 1) | AlarmIRPPort | O |
| unacknowledgeAlarms | unacknowledgeAlarms (note 1) | AlarmIRPPort | O |
| setComment | setComment (note 1) | AlarmIRPPort | O |
| clearAlarms | clearAlarms (note 1) | AlarmIRPPort | O |
| notifyNewAlarm | notify (note 2) | NotificationIRPNtfPort | M |
| notifyAckStateChanged | notify (note 2) | NotificationIRPNtfPort | M |
| notifyClearedAlarm | notify (note 2) | NotificationIRPNtfPort | M |
| notifyAlarmListRebuilt | notify (note 2) | NotificationIRPNtfPort | M |
| notifyChangedAlarm | notify (note 2) | NotificationIRPNtfPort | O |
| notifyChangedAlarmGeneral | notify (note 2) | NotificationIRPNtfPort | O |
| notifyComments | notify (note 2) | NotificationIRPNtfPort | O |
| notifyPotentialFaultyAlarmList | notify (note 2) | NotificationIRPNtfPort | O |
| getIRPVersion (note 3) | See TS 32.316 [11] | GenericIRPPort | M |
| getOperationProfile (note 3) | See TS 32.316 [11] | GenericIRPPort | O |
| getNotificationProfile (note 3) | See TS 32.316 [11] | GenericIRPPort | O |
| NOTE 1: The operation is under the port type alarmIRPSystem:AlarmIRPPortType and under the binding alarmIRPSystem:AlarmIRPBinding. | | | |
| NOTE 2: The IS equivalent maps to an XML definition specified in Annex B, and this being an input parameter to the operation notify under the port type ntfIRPNtfSystem:NotificationIRPNtf and under the binding ntfIRPNtfSystem:NotificationIRPNtf of 3GPP TS 32.306 [8]. This binding is linked to a port of the AlarmIRPService as indicated in the table above. | | | |
| NOTE 3: The IS operation is inherited from the ManagedGenericIRP IOC specified in 3GPP TS 32.312 [10].  This inheritance is by the AlarmIRP IOC of 3GPP TS 32.111-2 [4] inheriting from the ManagedGenericIRP IOC. The corresponding binding is linked to a port of the AlarmIRPService as indicated in the table above. | | | |

## C.2.2 Operation parameter mapping

The Alarm IRP IS (3GPP TS 32.111-2 [4]) defines semantics of parameters carried in the operations. The tables below show the mapping of these parameters, as per operation, to their equivalents defined in this SS.

### C.2.2.1 Operation acknowledgeAlarms

#### C.2.2.1.1 Input parameters

Table C.2.2.1.1: Mapping from IS acknowledgeAlarms input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS WSDL type sub-element  used in corresponding input message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| alarmInformationAndSeverityReferenceList | alarmInformationAndSeverityReferenceList | M |
| ackUserId | ackUserId | M |
| ackSystemId | ackSystemId | O |

#### C.2.2.1.2 Output parameters

Table C.2.2.1.2: Mapping from IS acknowledgeAlarms output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS WSDL type sub-element  used in corresponding output message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| badAlarmInformationReferenceList | badAlarmInformationReferenceList | M |
| status | status | M |

#### C.2.2.1.3 Fault definition

Table C.2.2.1.3: Mapping from IS acknowledgeAlarms exceptions to SS equivalents

|  |  |  |
| --- | --- | --- |
| Assertion name | SS WSDL type enumeration value  used in corresponding fault message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| operation\_failed | OperationFailed | M |

### C.2.2.2 Operation getAlarmList

#### C.2.2.2.1 Input parameters

Table C.2.2.2.1: Mapping from IS getAlarmList input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS WSDL type sub-element  used in corresponding input message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| alarmAckState | alarmAckState | O |
| baseObjectClass | baseObjectClass | O |
| baseObjectInstance | baseObjectInstance | O |
| filter | filter | O |

#### C.2.2.2.2 Output parameters

Table C.2.2.2.2: Mapping from IS getAlarmList output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS WSDL type sub-element  used in corresponding output message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| alarmInformationList | alarmInformationList | M |
| status | status | M |

#### C.2.2.2.3 Fault definition

Table C.2.2.2.3: Mapping from IS getAlarmList exceptions to SS equivalents

|  |  |  |
| --- | --- | --- |
| Assertion name | SS WSDL type enumeration value  used in corresponding fault message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| operation\_failed | OperationFailed | M |
| filter\_complexity\_limit | FilterComplexityLimit | M |

### C.2.2.3 Operation getAlarmCount

#### C.2.2.3.1 Input parameters

Table C.2.2.3.1: Mapping from IS getAlarmCount input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS WSDL type sub-element  used in corresponding input message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| filter | filter | O |
| alarmAckState | alarmAckState | O |

#### C.2.2.3.2 Output parameters

Table C.2.2.3.2: Mapping from IS getAlarmCount output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS WSDL type sub-element  used in corresponding output message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| criticalCount | criticalCount | M |
| majorCount | majorCount | M |
| minorCount | minorCount | M |
| warningCount | warningCount | M |
| indeterminateCount | indeterminateCount | M |
| clearedCount | clearedCount | M |
| status | status | M |

#### C.2.2.3.3 Fault definition

Table C.2.2.3.3: Mapping from IS getAlarmCount exceptions to SS equivalents

|  |  |  |
| --- | --- | --- |
| Assertion name | SS WSDL type enumeration value  used in corresponding fault message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| operation\_failed | OperationFailed | M |
| filter\_complexity\_limit | FilterComplexityLimit | M |

### C.2.2.4 Operation unacknowledgeAlarms

#### C.2.2.4.1 Input parameters

Table C.2.2.4.1: Mapping from IS unacknowledgeAlarms input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS WSDL type sub-element  used in corresponding input message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| alarmInformationReferenceList | alarmInformationReferenceList | M |
| ackUserId | ackUserId | M |
| ackSystemId | ackSystemId | O |

#### C.2.2.4.2 Output parameters

Table C.2.2.4.2: Mapping from IS unacknowledgeAlarms output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS WSDL type sub-element  used in corresponding output message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| badAlarmInformationReferenceList | badAlarmInformationReferenceList | M |
| status | status | M |

#### C.2.2.4.3 Fault definition

Table C.2.2.4.3: Mapping from IS unacknowledgeAlarms exceptions to SS equivalents

|  |  |  |
| --- | --- | --- |
| Assertion name | SS WSDL type enumeration value  used in corresponding fault message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| operation\_failed | OperationFailed | M |

### C.2.2.5 Operation setComment

#### C.2.2.5.1 Input parameters

Table C.2.2.5.1: Mapping from IS setComment input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS WSDL type sub-element  used in corresponding input message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| alarmInformationReferenceList | alarmInformationReferenceList | M |
| commentUserId | commentUserId | M |
| commentSystemId | commentSystemId | O |
| commentText | commentText | M |

#### C.2.2.5.2 Output parameters

Table C.2.2.5.2: Mapping from IS setComment output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS WSDL type sub-element  used in corresponding output message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| badAlarmInformationReferenceList | badAlarmInformationReferenceList | M |
| status | status | M |

#### C.2.2.5.3 Fault definition

Table C.2.2.5.3: Mapping from IS setComment exceptions to SS equivalents

|  |  |  |
| --- | --- | --- |
| Assertion name | SS WSDL type enumeration value  used in corresponding fault message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| operation\_failed | OperationFailed | M |

### C.2.2.6 Operation clearAlarms

#### C.2.2.6.1 Input parameters

Table C.2.2.6.1: Mapping from IS clearAlarms input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS WSDL type sub-element  used in corresponding input message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| alarmInformationReferenceList | alarmInformationReferenceList | M |
| clearUserId | clearUserId | M |
| clearSystemId | clearSystemId | O |

#### C.2.2.6.2 Output parameters

Table C.2.2.6.2: Mapping from IS clearAlarms output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS WSDL type sub-element  used in corresponding output message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| badAlarmInformationReferenceList | badAlarmInformationReferenceList | M |
| status | status | M |

#### C.2.2.6.3 Fault definition

Table C.2.2.6.3: Mapping from IS clearAlarms exceptions to SS equivalents

|  |  |  |
| --- | --- | --- |
| Assertion name | SS WSDL type enumeration value  used in corresponding fault message  under corresponding port type operation  as indicated in Table C.2.1 | Qualifier |
| operation\_failed | OperationFailed | M |

# C.3 Solution Set definitions

## C.3.1 WSDL definition structure

Clause C.3.2 provides a graphical representation of the Alarm IRP service.

Clause C.3.3 defines the services which are supported the Alarm IRP agent.

## C.3.2 Graphical Representation

The WSDL structure is depicted in Figure C.3.2 below, depicting port type, binding and service. The port type contains port type operations, which again contains input, output and fault messages. The binding contains binding operations, which have the same name as the port type operations. The binding connects to a port inside the service.



Figure C.3.2: Alarm IRP SOAP Solution Set WSDL structure

## C.3.3 WSDL specification “AlarmIRPSystem.wsdl”

<?xml version="1.0" encoding="UTF-8"?>

<!--

3GPP TS 32.111-6 Alarm IRP SOAP Solution Set

-->

<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:alarmIRPSystem="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#AlarmIRPSystem" xmlns:alarmIRPData="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#AlarmIRPData" xmlns:xai="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#alarmIRPIOCs" xmlns:xn="http://www.3gpp.org/ftp/specs/archive/32\_series/32.626#genericNrm" xmlns:genericIRPSystem="http://www.3gpp.org/ftp/specs/archive/32\_series/32.316#GenericIRPSystem" xmlns:ntfIRPNtfSystem="http://www.3gpp.org/ftp/specs/archive/32\_series/32.306#NotificationIRPNtfSystem" targetNamespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#AlarmIRPSystem">

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.306#NotificationIRPNtfSystem"/>

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.316#GenericIRPSystem"/>

<types>

<schema targetNamespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#AlarmIRPData" xmlns="http://www.w3.org/2001/XMLSchema">

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#alarmIRPIOCs"/>

<import namespace="http://www.3gpp.org/ftp/specs/archive/32\_series/32.626#genericNrm"/>

<!-- The following types are defined for the Alarm IRP operations -->

<simpleType name="AlarmAckState">

<restriction base="string">

<enumeration value="AllAlarms"/>

<enumeration value="AllActiveAlarms"/>

<enumeration value="AllActiveAndAcknowledgedAlarms"/>

<enumeration value="AllActiveAndUnacknowledgedAlarms"/>

<enumeration value="AllClearedAndUnacknowledgedAlarms"/>

<enumeration value="AllUnacknowledgedAlarms"/>

</restriction>

</simpleType>

<complexType name="AlarmIdAndSeverityElement">

<sequence>

<element ref="xai:alarmId"/>

<element ref="xai:perceivedSeverity" minOccurs="0"/>

</sequence>

</complexType>

<complexType name="AlarmIdAndSeverityList">

<sequence minOccurs="1" maxOccurs="unbounded">

<element name="listElement" type="alarmIRPData:AlarmIdAndSeverityElement"/>

</sequence>

</complexType>

<complexType name="AlarmIdList">

<sequence minOccurs="1" maxOccurs="unbounded">

<element ref="xai:alarmId"/>

</sequence>

</complexType>

<complexType name="AlarmInformationList">

<choice minOccurs="0" maxOccurs="unbounded">

<element name="nonSecurityAlarm" type="xai:NonSecurityAlarm"/>

<element name="securityAlarm" type="xai:SecurityAlarm"/>

</choice>

</complexType>

<complexType name="BadAcknowledgeAlarmInfoRefElement">

<sequence>

<element ref="xai:alarmId"/>

<element name="failureReason" type="alarmIRPData:FailureReasonBadAck"/>

<element name="additionalFailureReason" type="string"/>

</sequence>

</complexType>

<complexType name="BadAcknowledgeAlarmInfoRefList">

<sequence minOccurs="0" maxOccurs="unbounded">

<element name="listElement" type="alarmIRPData:BadAcknowledgeAlarmInfoRefElement"/>

</sequence>

</complexType>

<complexType name="BadClearAlarmInfoRefElement">

<sequence>

<element ref="xai:alarmId"/>

<element name="failureReason" type="string"/>

</sequence>

</complexType>

<complexType name="BadClearAlarmInfoRefList">

<sequence minOccurs="0" maxOccurs="unbounded">

<element name="listElement" type="alarmIRPData:BadClearAlarmInfoRefElement"/>

</sequence>

</complexType>

<complexType name="BadCommentAlarmInfoRefElement">

<sequence>

<element ref="xai:alarmId"/>

<element name="failureReason" type="string"/>

</sequence>

</complexType>

<complexType name="BadCommentAlarmInfoRefList">

<sequence minOccurs="0" maxOccurs="unbounded">

<element name="listElement" type="alarmIRPData:BadCommentAlarmInfoRefElement"/>

</sequence>

</complexType>

<complexType name="BadUnacknowledgeAlarmInfoRefElement">

<sequence>

<element ref="xai:alarmId"/>

<element name="failureReason" type="string"/>

</sequence>

</complexType>

<complexType name="BadUnacknowledgeAlarmInfoRefList">

<sequence minOccurs="0" maxOccurs="unbounded">

<element name="listElement" type="alarmIRPData:BadUnacknowledgeAlarmInfoRefElement"/>

</sequence>

</complexType>

<simpleType name="FailureReasonBadAck">

<restriction base="string">

<enumeration value="UnknownAlarmId"/>

<enumeration value="AcknowledgmentFailed"/>

<enumeration value="WrongPerceivedSeverity"/>

</restriction>

</simpleType>

<simpleType name="OperationStatusTwo">

<restriction base="string">

<enumeration value="OperationSucceeded"/>

<enumeration value="OperationFailed"/>

</restriction>

</simpleType>

<simpleType name="OperationStatusThree">

<restriction base="string">

<enumeration value="OperationSucceeded"/>

<enumeration value="OperationFailed"/>

<enumeration value="OperationPartiallySucceeded"/>

</restriction>

</simpleType>

<!-- acknowledgeAlarms Request-->

<element name="acknowledgeAlarms">

<complexType>

<sequence>

<element name="alarmInformationAndSeverityReferenceList" type="alarmIRPData:AlarmIdAndSeverityList"/>

<element ref="xai:ackUserId"/>

<element ref="xai:ackSystemId" minOccurs="0"/>

</sequence>

</complexType>

</element>

<!-- acknowledgeAlarms Response -->

<element name="acknowledgeAlarmsResponse">

<complexType>

<sequence>

<element name="badAlarmInformationReferenceList" type="alarmIRPData:BadAcknowledgeAlarmInfoRefList"/>

<element name="status" type="alarmIRPData:OperationStatusThree"/>

</sequence>

</complexType>

</element>

<!-- acknowledgeAlarms Fault -->

<element name="acknowledgeAlarmsFault">

<simpleType>

<restriction base="string">

<enumeration value="OperationFailed"/>

</restriction>

</simpleType>

</element>

<!-- getAlarmList Request -->

<element name="getAlarmList">

<complexType>

<sequence>

<element name="alarmAckState" type="alarmIRPData:AlarmAckState" minOccurs="0"/>

<element name="baseObjectClass" type="string" minOccurs="0"/>

<element name="baseObjectInstance" type="xn:dn" minOccurs="0"/>

<element name="filter" type="string" minOccurs="0"/>

</sequence>

</complexType>

</element>

<!-- getAlarmList Response -->

<element name="getAlarmListResponse">

<complexType>

<sequence>

<element name="alarmInformationList" type="alarmIRPData:AlarmInformationList"/>

<element name="status" type="alarmIRPData:OperationStatusTwo"/>

</sequence>

</complexType>

</element>

<!-- getAlarmList Fault -->

<element name="getAlarmListFault">

<simpleType>

<restriction base="string">

<enumeration value="OperationFailed"/>

<enumeration value="FilterComplexityLimit"/>

</restriction>

</simpleType>

</element>

<!-- getAlarmCount Request -->

<element name="getAlarmCount">

<complexType>

<sequence>

<element name="filter" type="string" minOccurs="0"/>

<element name="alarmAckState" type="alarmIRPData:AlarmAckState" minOccurs="0"/>

</sequence>

</complexType>

</element>

<!-- getAlarmCount Response -->

<element name="getAlarmCountResponse">

<complexType>

<sequence>

<element name="criticalCount" type="nonNegativeInteger"/>

<element name="majorCount" type="nonNegativeInteger"/>

<element name="minorCount" type="nonNegativeInteger"/>

<element name="warningCount" type="nonNegativeInteger"/>

<element name="indeterminateCount" type="nonNegativeInteger"/>

<element name="clearedCount" type="nonNegativeInteger"/>

<element name="status" type="alarmIRPData:OperationStatusTwo"/>

</sequence>

</complexType>

</element>

<!-- getAlarmCount Fault -->

<element name="getAlarmCountFault">

<simpleType>

<restriction base="string">

<enumeration value="OperationFailed"/>

<enumeration value="FilterComplexityLimit"/>

</restriction>

</simpleType>

</element>

<!-- unacknowledgeAlarms Request -->

<element name="unacknowledgeAlarms">

<complexType>

<sequence>

<element name="alarmInformationReferenceList" type="alarmIRPData:AlarmIdList"/>

<element ref="xai:ackUserId"/>

<element ref="xai:ackSystemId" minOccurs="0"/>

</sequence>

</complexType>

</element>

<!-- unacknowledgeAlarms Response -->

<element name="unacknowledgeAlarmsResponse">

<complexType>

<sequence>

<element name="badAlarmInformationReferenceList" type="alarmIRPData:BadUnacknowledgeAlarmInfoRefList"/>

<element name="status" type="alarmIRPData:OperationStatusThree"/>

</sequence>

</complexType>

</element>

<!-- unacknowledgeAlarms Fault -->

<element name="unacknowledgeAlarmsFault">

<simpleType>

<restriction base="string">

<enumeration value="OperationFailed"/>

</restriction>

</simpleType>

</element>

<!-- setComment Request -->

<element name="setComment">

<complexType>

<sequence>

<element name="alarmInformationReferenceList" type="alarmIRPData:AlarmIdList"/>

<element ref="xai:commentUserId"/>

<element ref="xai:commentSystemId" minOccurs="0"/>

<element ref="xai:commentText"/>

</sequence>

</complexType>

</element>

<!-- setComment Response -->

<element name="setCommentResponse">

<complexType>

<sequence>

<element name="badAlarmInformationReferenceList" type="alarmIRPData:BadCommentAlarmInfoRefList"/>

<element name="status" type="alarmIRPData:OperationStatusThree"/>

</sequence>

</complexType>

</element>

<!-- setComment Fault -->

<element name="setCommentFault">

<simpleType>

<restriction base="string">

<enumeration value="OperationFailed"/>

</restriction>

</simpleType>

</element>

<!-- clearAlarms Request-->

<element name="clearAlarms">

<complexType>

<sequence>

<element name="alarmInformationReferenceList" type="alarmIRPData:AlarmIdList"/>

<element ref="xai:clearUserId"/>

<element ref="xai:clearSystemId" minOccurs="0"/>

</sequence>

</complexType>

</element>

<!-- clearAlarms Response-->

<element name="clearAlarmsResponse">

<complexType>

<sequence>

<element name="badAlarmInformationReferenceList" type="alarmIRPData:BadClearAlarmInfoRefList"/>

<element name="status" type="alarmIRPData:OperationStatusThree"/>

</sequence>

</complexType>

</element>

<!-- clearAlarms Fault-->

<element name="clearAlarmsFault">

<simpleType>

<restriction base="string">

<enumeration value="OperationFailed"/>

</restriction>

</simpleType>

</element>

</schema>

</types>

<message name="acknowledgeAlarms">

<part name="parameter" element="alarmIRPData:acknowledgeAlarms"/>

</message>

<message name="acknowledgeAlarmsResponse">

<part name="parameter" element="alarmIRPData:acknowledgeAlarmsResponse"/>

</message>

<message name="acknowledgeAlarmsFault">

<part name="parameter" element="alarmIRPData:acknowledgeAlarmsFault"/>

</message>

<message name="getAlarmList">

<part name="parameter" element="alarmIRPData:getAlarmList"/>

</message>

<message name="getAlarmListResponse">

<part name="parameter" element="alarmIRPData:getAlarmListResponse"/>

</message>

<message name="getAlarmListFault">

<part name="parameter" element="alarmIRPData:getAlarmListFault"/>

</message>

<message name="getAlarmCount">

<part name="parameter" element="alarmIRPData:getAlarmCount"/>

</message>

<message name="getAlarmCountResponse">

<part name="parameter" element="alarmIRPData:getAlarmCountResponse"/>

</message>

<message name="getAlarmCountFault">

<part name="parameter" element="alarmIRPData:getAlarmCountFault"/>

</message>

<message name="unacknowledgeAlarms">

<part name="parameter" element="alarmIRPData:unacknowledgeAlarms"/>

</message>

<message name="unacknowledgeAlarmsResponse">

<part name="parameter" element="alarmIRPData:unacknowledgeAlarmsResponse"/>

</message>

<message name="unacknowledgeAlarmsFault">

<part name="parameter" element="alarmIRPData:unacknowledgeAlarmsFault"/>

</message>

<message name="setComment">

<part name="parameter" element="alarmIRPData:setComment"/>

</message>

<message name="setCommentResponse">

<part name="parameter" element="alarmIRPData:setCommentResponse"/>

</message>

<message name="setCommentFault">

<part name="parameter" element="alarmIRPData:setCommentFault"/>

</message>

<message name="clearAlarms">

<part name="parameter" element="alarmIRPData:clearAlarms"/>

</message>

<message name="clearAlarmsResponse">

<part name="parameter" element="alarmIRPData:clearAlarmsResponse"/>

</message>

<message name="clearAlarmsFault">

<part name="parameter" element="alarmIRPData:clearAlarmsFault"/>

</message>

<portType name="AlarmIRPPortType">

<operation name="acknowledgeAlarms">

<input message="alarmIRPSystem:acknowledgeAlarms"/>

<output message="alarmIRPSystem:acknowledgeAlarmsResponse"/>

<fault name="acknowledgeAlarmsFault" message="alarmIRPSystem:acknowledgeAlarmsFault"/>

</operation>

<operation name="getAlarmList">

<input message="alarmIRPSystem:getAlarmList"/>

<output message="alarmIRPSystem:getAlarmListResponse"/>

<fault name="getAlarmListFault" message="alarmIRPSystem:getAlarmListFault"/>

</operation>

<operation name="getAlarmCount">

<input message="alarmIRPSystem:getAlarmCount"/>

<output message="alarmIRPSystem:getAlarmCountResponse"/>

<fault name="getAlarmCountFault" message="alarmIRPSystem:getAlarmCountFault"/>

</operation>

<operation name="unacknowledgeAlarms">

<input message="alarmIRPSystem:unacknowledgeAlarms"/>

<output message="alarmIRPSystem:unacknowledgeAlarmsResponse"/>

<fault name="unacknowledgeAlarmsFault" message="alarmIRPSystem:unacknowledgeAlarmsFault"/>

</operation>

<operation name="setComment">

<input message="alarmIRPSystem:setComment"/>

<output message="alarmIRPSystem:setCommentResponse"/>

<fault name="setCommentFault" message="alarmIRPSystem:setCommentFault"/>

</operation>

<operation name="clearAlarms">

<input message="alarmIRPSystem:clearAlarms"/>

<output message="alarmIRPSystem:clearAlarmsResponse"/>

<fault name="clearAlarmsFault" message="alarmIRPSystem:clearAlarmsFault"/>

</operation>

</portType>

<binding name="AlarmIRPBinding" type="alarmIRPSystem:AlarmIRPPortType">

<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>

<operation name="acknowledgeAlarms">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#acknowledgeAlarms" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="acknowledgeAlarmsFault">

<soap:fault name="acknowledgeAlarmsFault" use="literal"/>

</fault>

</operation>

<operation name="getAlarmList">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#getAlarmList" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="getAlarmListFault">

<soap:fault name="getAlarmListFault" use="literal"/>

</fault>

</operation>

<operation name="getAlarmCount">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#getAlarmCount" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="getAlarmCountFault">

<soap:fault name="getAlarmCountFault" use="literal"/>

</fault>

</operation>

<operation name="unacknowledgeAlarms">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#unacknowledgeAlarms" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="unacknowledgeAlarmsFault">

<soap:fault name="unacknowledgeAlarmsFault" use="literal"/>

</fault>

</operation>

<operation name="setComment">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#setComment" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="setCommentFault">

<soap:fault name="setCommentFault" use="literal"/>

</fault>

</operation>

<operation name="clearAlarms">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#clearAlarms" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="clearAlarmsFault">

<soap:fault name="clearAlarmsFault" use="literal"/>

</fault>

</operation>

</binding>

<service name="AlarmIRPService">

<port name="AlarmIRPPort" binding="alarmIRPSystem:AlarmIRPBinding">

<soap:address location="http://www.3gpp.org/ftp/specs/archive/32\_series/32.111-6#AlarmIRP"/>

</port>

<port name="GenericIRPPort" binding="genericIRPSystem:GenericIRPBinding">

<soap:address location="http://www.3gpp.org/ftp/specs/archive/32\_series/32.316#GenericIRP"/>

</port>

<port name="NotificationIRPNtfPort" binding="ntfIRPNtfSystem:NotificationIRPNtf">

<soap:address location="http://www.3gpp.org/ftp/specs/archive/32\_series/32.306#NotificationIRPNtf"/>

</port>

</service>

</definitions>

Annex D (informative):  
Change history

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | | |
| **Date** | **TSG #** | **TSG Doc.** | **CR** | **Rev** | **Subject/Comment** | **Cat** | **Old** | **New** |
| 05-2010 | SA-48 | SP-100284 | -- | -- | Presentation to SA for information and approval | -- | --- | 1.0.0 |
| 06-2010 | SA-48 | -- | -- | -- | Publication | -- | 1.0.0 | 10.0.0 |
| 09-2010 | SA-49 | SP-100489 | 001 | -- | Correct the qualifier of serviceUser and serviceProvider of NotifyNewSecurityAlarm | F | 10.0.0 | 10.1.0 |
| 12-2010 | SA-50 | SP-100833 | 002 | 2 | Add alarmChangedTime parameter to the definition of a Structured Event of AlarmInformationSeq - Align with 32.111-2 IS | C | 10.1.0 | 10.2.0 |
| 06-2011 | SA-52 | SP-110289 | 003 | 1 | Add indication for root cause of alarm | B | 10.2.0 | 10.3.0 |
| 06-2011 | SA-52 | SP-110289 | 004 | 1 | Add notification for change of alarm correlation data | B | 10.2.0 | 10.3.0 |
| 09-2011 | SA-53 | SP-110534 | 005 | 1 | Correct definition of rootCauseIndicator of IDL specification | F | 10.3.0 | 10.4.0 |
| 09-2012 | SA-57 | - | - | - | Automatic upgrade from previous Release version 10.4.0 | - | 10.4.0 | 11.0.0 |
| 09-2014 | SA-65 | SP-140559 | 006 | - | Update the link from Solution Set to Information Service due to the end of Release 12 | C | 11.0.0 | 12.0.0 |
| 12-2014 | SA-66 | SP-140801 | 007 | 1 | Alarm quality improvements, solution sets updates for notification support for changed alarm attributes | B | 12.0.0 | 12.1.0 |
| 2016-01 | - | - | - | - | Update to Rel-13 version (MCC) |  | 12.1.0 | **13.0.0** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2016-06 | SA#72 | SP-160407 | 0008 | - | F | Update the link from IRP Solution Set to IRP Information Service | 13.1.0 |
| 2017-03 | SA#75 | - | - | - |  | Promotion to Release 14 without technical change | 14.0.0 |
| 2017-06 | SA#76 | SP-170502 | 0009 | - | F | Update the link from IRP Solution Set to IRP Information Service | 14.1.0 |
| 2018-06 | - | - | - | - | - | Update to Rel-15 version (MCC) | **15.0.0** |
| 2020-07 | - | - | - | - | - | Update to Rel-16 version (MCC) | **16.0.0** |