Technical Specification Group Services and System Aspects
Meeting #16, Marco Island, Florida, 10-13 June 2002

TSGS#16(02)0284

Source: **SA5 (Telecom Management)**

Title: **3 Rel-5 CRs 32.111-4 (Fault Management; Alarm Integration Reference Point; Part 4 CMIP Solution Set)**

Document for: **Approval**

Agenda Item: **7.5.3**

Doc-1st-Level	Spec	CR	R	Phase	Subject	Cat	Ver Cur	Ver New	Doc-2nd-Level	Workitem
SP-020284	32.111-4	008	-	Rel-5	Addition of the parameter alarmListAlignmentRequirement to the notification notifyAlarmListRebuilt in the CMIP SS (32.111-4)	F	5.0.0	5.1.0	S5-026247	OAM-NIM
SP-020284	32.111-4	009	-	Rel-5	Adding the notification notifyPotentialFaultyAlarmList in the CMIP SS (32.111-4)	F	5.0.0	5.1.0	S5-026248	OAM-NIM
SP-020284	32.111-4	010	-	Rel-5	Introduction of SS (32.111-4) to IS (32.111-2) relation and correction of Foreword	F	5.0.0	5.1.0	S5-026249	OAM-NIM

3GPP TSG-SA5 (Telecom Management) **S5-026247**
Meeting #28, Sophia Antipolis, France, 20 - 24 Mai 2002

CR-Form-v5

CHANGE REQUEST

⌘ **32.111-4** **CR** **008** ⌘ **rev** **-** ⌘ Current version: **5.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM ☐ ME/UE ☐ Radio Access Network ☐ Core Network ☐

Title:	⌘	Addition of the parameter *alarmListAlignmentRequirement* to the notification *notifyAlarmListRebuilt* in the CMIP SS (32.111-4)
Source:	⌘	SA5
Work item code:	⌘	OAM-NIM
Date:	⌘	05/04/2002
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
REL-4 (Release 4)
REL-5 (Release 5)

Reason for change:	⌘	Alignment of CMIP SS (32.111-4) with Alarm IRP IS (32.111-2). The parameter *alarmListAlignmentRequirement* has been added to the parameter list of the notification *notifyAlarmListRebuilt* in the Alarm IRP IS (32.111-2). This CR reflects this change in the CMIP SS (32.111-4).
Summary of change:	⌘	The parameter *alarmListAlignmentRequirement* is added to the parameter list of the notification *notifyAlarmListRebuilt*.
Consequences if not approved:	⌘	The CMIP SS (32.111-4) would not be aligned with the IS (32.111-2).

Clauses affected:	⌘	4.7.5, 5.4.1, 6
Other specs affected:	⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ O&M Specifications
Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.
3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

4.7.5 Mapping of Parameters of each notification

The notifications defined in [9] (Alarm IRP: Information Services) have a set of parameters that are common to all the notifications (see [10]: Notification IRP:CMIP Solution Set). Such common set of parameters are:

ManagedObjectClass, ManagedObjectInstance, EventTime, NotificationType, NotificationId.

In the CMIP Solution Set, all the notifications originated within the Agent are reported to the Managers by means of the CMISE "M-EVENT-REPORT" primitive, which is implemented by means of the "m-EventReport OPERATION" (see [2] and[3]). The argument of m-EventReport OPERATION is defined in [3] as follows:

```
EventReportArgument         ::= SEQUENCE {
        managedObjectClass          ObjectClass,
        managedObjectInstance       ObjectInstance,
        eventTime                       [5] IMPLICIT GeneralizedTime OPTIONAL,
        eventType                       EventTypeId,
        eventInfo                               [8] ANY DEFINED BY eventType OPTIONAL
        }
```

where eventinfo is further specified, for each specific notification, by means of specific GDMO/ASN1 definitions.

Therefore the first four parameters of the notification header are mapped to the first four fields of the EventReportArgument. The fifth parameter of the notification header is mapped to the eventinfo field of the EventReportArgument, together with all the other (not common) parameters of the notification.

In the following tables, for the notifications defined in [9], all the parameters (but the common ones) are mapped to their corresponding elements of the CMIP SS notifications.

The parameter SystemDN defined in [9] (Alarm IRP: Information Services) is not used in this CMIP Solution Set.

Table 11: Mapping of Parameters of "notifyNewAlarm"' and "notifyClearedAlarm"'

Notification parameters of Information Services	CMIP Notification equivalences	Qualifier
notificationId	notificationIdentifier (note 1)	M
probableCause	probableCause	M
specificProblems	specificProblems	O
perceivedSeverity	perceivedSeverity	M
backedUpStatus	backedUpStatus	O
backUpObject	backUpObject	O
trendIndication	trendIndication	O
thresholdInfo	thresholdInfo	O
correlatedNotifications	correlatedNotifications	O
stateChangeDefinition	stateChangeDefinition	O
monitoredAttributes	monitoredAttributes	O
proposedRepairActions	proposedRepairActions	O
additionalText	additionalText	O
alarmId	-- (note 2)	
NOTE 1: notificationIdentifier is a parameter of the Notification Header also defined in 3GPP TS 32.302. NOTE 2: In the CMIP Solution Set the alarmId is not used. In the CMIP Solution Set the alarm notifications are univocally identified by means of notificationIdentifier and managedObjectInstance.		

Table 12: Mapping of Parameters of 'notifyAckStateChanged'

Notification parameters of Information Services	CMIP Notification equivalences	Qualifier
notificationId	notificationIdentifier (note 1)	M
probableCause	probableCause	M
specificProblems	specificProblems	O
perceivedSeverity	perceivedSeverity	M
alarmId	-- (note-2)	
ackTime	additionalInformation	M
ackState		M
ackUserId		M
ackSystemId		O
NOTE 1: notificationIdentifier is a parameter of the Notification Header also defined in 3GPP TS 32.302. NOTE 2: In the CMIP Solution Set the alarmId is not used. In the CMIP Solution Set the alarm notifications are univocally identified by means of notificationIdentifier and managedObjectInstance.		

Table 13: Mapping of Parameters of 'notifyAlarmListRebuilt'

Notification parameters of Information Services	CMIP Notification equivalents	Qualifier
notificationId	notificationIdentifier (note)	M
reason	reason	M
objectClass	rebuiltObjectClass	M
objectInstance	rebuiltObjectInstance	M
alarmListAlignmentRequirement	M-EVENT REPORT parameter 'Event information': alarmListAlignmentRequirement (Note 1)	O
NOTE: notificationIdentifier is a parameter of the Notification Header also defined in 3GPP TS 32.302. Note 1: This parameter shall be supported only, if the IRPAgent supports the notification `notifyPotentialFaultyAlarmList`.		

Table 14: Mapping of Parameters of 'notifyComments'

Notification parameters of Information Services	CMIP Notification equivalents	Qualifier
notificationId	notificationIdentifier (note 1)	M
objectClass	alarmedObjectClass	M
objectInstance	alarmedObjectInstance	M
notificationId	notificationIdentifier	M
eventTime	alarmEventTime	M
notificationType	eventType	M
alarmType	alarmType	M
probableCause	alarmProbableCause	M
perceivedSeverity	alarmPerceivedSeverity	M
comments	comments	M
alarmId	-- (note 2)	
NOTE 1: notificationIdentifier is a parameter of the Notification Header also defined in 3GPP TS 32.302. NOTE 2: In the CMIP Solution Set the alarmId is not used. In the CMIP Solution Set the alarm notifications are univocally identified by means of notificationIdentifier and managedObjectInstance.		

5.4.1 alarmListRebuilt (M)

alarmListRebuilt **NOTIFICATION**
BEHAVIOUR
alarmListRebuiltBehaviour;
WITH INFORMATION SYNTAX
TS32-111-4TypeModule .AlarmListRebuiltInfo;
REGISTERED AS { ts32-111AlarmNotification 1};

alarmListRebuiltBehaviour **BEHAVIOUR**
DEFINED AS

"This notification is used by the Agent to inform the NM that the alarm list has been rebuilt.

The 'Event Information' field contains the following data:

- *notificationIdentifier*

  This ITU-T X.721 standardised parameter, together with MOI (Managed Object Instance), unambiguously identifies this notification.

- *rebuiltObjectClass*

  This parameter carries the IRPAgent MOC when the entire AlarmList has been rebuilt. It carries a different MOC when the AlarmList has been partially rebuilt.

- *rebuiltObjectInstance*

  This parameter carries DN of the IRPAgent when the entire AlarmList has been rebuilt. It carries the DN of another MOI when the AlarmList has been partially rebuilt and only the MOIs subordinate of this rebuilt MOI may be affected by this partial rebuilt.

- *reason*

  The parameter indicates the reason for alarm list rebuilding (if applicable).

- *alarmListAlignmentRequirement*

  This parameter indicates, if the IRPManager has to align its alarm list with the IRPAgent. Absence of this parameter means, that an alignment is required.

6 ASN.1 definitions for Alarm IRP

```
TS32-111-4TypeModule {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0) umts-Operation-
Maintenance(3) ts-32-111(111) part4(4) informationModel(0) asn1Module(2) version1(1)}

DEFINITIONS IMPLICIT TAGS ::=
BEGIN
--EXPORTS everything
IMPORTS

NotificationIdentifier, Destination, EventTime, ProbableCause, PerceivedSeverity
FROM Attribute-ASN1Module {joint-iso-ccitt ms(9) smi(3) part2(2) asn1Module(2) 1}

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

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

baseNodeUMTS  OBJECT IDENTIFIER ::= { itu-t (0) identified-organization (4) etsi (0) mobileDomain
(0)
umts-Operation-Maintenance (3) }
ts32-111Prefix          OBJECT IDENTIFIER ::= { baseNodeUMTS ts-32-111(111)}
ts32-111Part4               OBJECT IDENTIFIER ::= { ts32-111Prefix part4(4)}
ts32-111-4InfoModel OBJECT IDENTIFIER ::= { ts32-111Part4 informationModel(0)}

ts32-111AlarmObjectClass        OBJECT IDENTIFIER ::= { ts32-111-4InfoModel managedObjectClass(3)}
ts32-111AlarmPackage            OBJECT IDENTIFIER ::= { ts32-111-4InfoModel package(4)}
ts32-111AlarmParameter      OBJECT IDENTIFIER ::= { ts32-111-4InfoModel parameter(5)}
ts32-111AlarmAttribute          OBJECT IDENTIFIER ::= { ts32-111-4InfoModel attribute(7)}
ts32-111AlarmAction         OBJECT IDENTIFIER ::= { ts32-111-4InfoModel action(9)}
ts32-111AlarmNotification       OBJECT IDENTIFIER ::= { ts32-111-4InfoModel action(10)}



-- Start of 3GPP SA5 own definitions
AckErrorList ::= SET OF ErrorInfo
AlarmReference ::= SEQUENCE
{
  moi ObjectInstance OPTIONAL, -- absent if scope of uniquness of notificationId is across IRPAgent
  notificationIdentifier NotificationIdentifier
}
AckOrUnackAlarms ::= SEQUENCE
    {
    alarmReferenceList  SET OF AlarmReference,  -- ITU-T X.721
    ackUserId               UserId,
    ackSystemId         SystemId OPTIONAL
    }
AckOrUnackAlarmsReply ::= SEQUENCE
    {
    status                  ErrorCauses,
errorAlarmReferenceList     AckErrorList
    }
AckState ::= ENUMERATED
    {
    acknowledged        (0),
    unacknowledged  (1)
}
AckTime ::= GeneralizedTime
AlarmChoice ::= ENUMERATED
    {
    allAlarms                           (0),
    allActiveAlarms                 (1),
    allActiveAndAckAlarms       (2),
    allActiveAndUnackAlarms     (3),
    allClearedAndUnackAlarms    (4)
}
AlarmsCountSummary ::= SEQUENCE
    {
    activeAlarmsCount           INTEGER,    -- this is the sum of criticalCount, majorCount,
minorCount, warningCount
                                                -- and indeterminateCount
    criticalCount                   INTEGER,
    majorCount                  INTEGER,
    minorCount                  INTEGER,
```

```
    warningCount                INTEGER,
    indeterminateCount          INTEGER,
    clearedCount                    INTEGER
    }
AlarmListRebuiltInfo ::= SEQUENCE
    {
    notificationIdentifier      NotificationIdentifier, -- ITU-T X.721
    rebuiltObjectClass          ObjectClass,
    rebuiltObjectInstance       ObjectInstance,
    reason                      ErrorCauses
    alarmListAlignmentRequirement   AlarmListAlignmentRequirement
    }
AlarmListAlignmentRequirement ::= ENUMERATED
    {
    alignmentRequired (0)       An alarm alignment is required.
    alignmentNotRequired (1)    An alarm alignment is not required.
    }
~~Comment ::= GraphicString~~
Comment ::= GraphicString
ErrorCauses ::= ENUMERATED
{
noError (0),    -- operation / notification successfully performed
wrongFilter (1),    -- the value of the filter parameter is not valid
wrongAlarmAckState (2), -- the value of the alarmAckState parameter (e.g. getAlarmCount) is not
valid
ackPartlySuccessful (3),    -- acknowledgment request partly successful
unackPartlySuccessful (4),  -- unacknowledgment request partly successful
wrongAlarmReference (5),    -- alarm identifier used in the alarm reference list not found  (e.g. in
case of acknowledgement request)
wrongAlarmReferenceList (6),    -- the alarm reference list (e.g. in case of acknowledgement
    request) is empty or completely wrong
alarmAlreadyAck (7),    -- alarm to be acknowledged is already in this state
alarmAlreadyUnack (8),  -- alarm to be acknowledged is already in this state
wrongUserId (9),    -- the user identifier in the unacknowledgement operation is    not the same as
in the previous acknowledgementAlarms request
wrongSystemId (10), -- the system identifier in the unacknowledgement operation is  not the same as
in the previous acknowledgementAlarms request
alarmAckNotAllowed (11),    -- current management system not allowed to acknowledge the alarm (e.g.
due to acknowledgement competence rules)
setCommentPartlySuccessful (12),    -- the setComment action partly successful (e.g. some alarmId
are not in the alarmList)
unspecifiedErrorReason (255)    -- operation failed, specific error unknown
}
ErrorInfo ::= SEQUENCE
    {
      moi ObjectInstance OPTIONAL, -- absent if uniqueness of notificationIdentifier is across
IRPAgent
    notificationIdentifier          NotificationIdentifier, -- ITU-T X.721
    reason          ErrorCauses
    }
GeneralObjectId ::= INTEGER
GetAlarmCount ::= SEQUENCE
    {
    alarmAckState       AlarmChoice OPTIONAL,
    filter                  CMISFilter OPTIONAL -- ITU-T X.711
    }
GetAlarmCountReply ::= SEQUENCE
    {
    criticalCount               INTEGER,
    majorCount              INTEGER,
    minorCount              INTEGER,
    warningCount            INTEGER,
    indeterminateCount      INTEGER,
    clearedCount                INTEGER,
    status                      ErrorCauses
    }
GetAlarmIRPVersionReply ::= SEQUENCE
    {
    versionNumberList   SupportedAlarmIRPVersions,
    status                  ErrorCauses
    }
GetAlarmList ::= SEQUENCE
    {
    alarmAckState       AlarmChoice OPTIONAL,
    destination         Destination,                    -- ITU-T X.721
    filter                  CMISFilter OPTIONAL -- ITU-T X.711
    }
GetAlarmListReply ::= SEQUENCE
```

```
    {
    alignmentId     INTEGER,
    status              ErrorCauses
    }
GetNotificationProfileReply ::= SEQUENCE
{
notificationNameProfile         NotificationList,
notificationParameterProfile    ParameterListOfList,
status                              ErrorCauses
}

GetOperationProfileReply ::= SEQUENCE
{
operationNameProfile            OperationList,
operationParameterProfile       ParameterListOfList,
status                              ErrorCauses
}

IRPVersionNumber ::= GraphicString

NotificationList ::= SET OF NotificationName

NotificationName ::= GraphicString

NotifyComments ::= SEQUENCE
{
alarmedObjectClass          ObjectClass,
alarmedObjectInstance       ObjectInstance,
notificationIdentifier      NotificationIdentifier, -- ITU-T X.721
alarmEventTime              EventTime,
alarmType                   EventTypeId,
alarmProbableCause          ProbableCause,
alarmPerceivedSeverity      PerceivedSeverity,
comments                    SET OF Comment
}

OperationList ::= SET OF OperationName

OperationName ::= GraphicString

ParameterList ::= SET OF ParameterName

ParameterListOfList ::= SET OF ParameterList

ParameterName ::= GraphicString

SetComment ::= SEQUENCE
    {
    alarmReferenceList      SET OF AlarmReference,
    commentUserId           UserId,
    commentSystemId     SystemId,
    commentText             Comment
    }
SetCommentReply ::= SEQUENCE
    {
    badAlarmReferenceList       SET OF ErrorInfo,
    status                          ErrorCauses
    }
SystemId ::= GraphicString

SupportedAlarmIRPVersions ::= SET OF IRPVersionNumber
UserId ::= GraphicString

END -- of module TS32-111-4TypeModule
```

3GPP TSG-SA5 (Telecom Management) **S5-026249**
Meeting #27, Cork, IRELAND, 2 - 5 April 2002

CR-Form-v5

CHANGE REQUEST

⌘ **32.111-4** **CR** **010** ⌘ **rev** **-** ⌘ Current version: **5.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM ☐ ME/UE ☐ Radio Access Network **X** Core Network **X**

Title:	⌘	Introduction of SS (32.111-4) to IS (32.111-2) relation and correction of Foreword
Source:	⌘	SA5
Work item code:	⌘	OAM-NIM
Date:	⌘	24/05/2002
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
REL-4 (Release 4)
REL-5 (Release 5)

Reason for change:	⌘	Alignment of CMIP SS (32.111-4) with Alarm IRP IS (32.111-2). The introduction of the relation SS to IS is required. The Foreword has wrong references.
Summary of change:	⌘	The IS to SS relation is added and the wrong references in the Foreword are corrected.
Consequences if not approved:	⌘	It is not possible to identify the IS version on which the subject SS is based. The wrong references in the Foreword might be misleading.

Clauses affected:	⌘	Foreword, 1
Other specs affected:	⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ O&M Specifications
Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Foreword

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

The present document is part 4 of a multi-part TS covering the Telecommunication Management; Fault Management; Part 4: Alarm Integration Reference Point: CMIP solution set, as identifies below:

Part 1: "3G Fault Management Requirements";

Part 2: "Alarm Integration Reference Point: Information Service ~~Version 1~~";

Part 3: "Alarm Integration Reference Point: CORBA Solution Set ~~version 1:1~~";

Part 4: "Alarm Integration Reference Point: CMIP Solution Set".

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.

1 Scope

The present document defines the alarm integration reference point for the CMIP solution set. In detail:

- clause 4 contains an introduction to some basic concepts of the CMIP interfaces;
- clause 5 contains the GDMO definitions for the Alarm Management over the CMIP interfaces;
- clause 6 contains the ASN.1 definitions supporting the GDMO definitions provided in clause 5.

This Solution Set specification is related to 3G TS 32.111-2 V5.0.X.

3GPP TSG-SA5 (Telecom Management) **S5-026248**
Meeting #28, Sophia Antipolis, France, 20 - 24 Mai 2002

CR-Form-v5

CHANGE REQUEST

⌘ **32.111-4** **CR** **009** ⌘ **rev** **-** ⌘ Current version: **5.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM ☐ ME/UE ☐ Radio Access Network ☒ Core Network ☒

Title:	⌘	Adding the notification *notifyPotentialFaultyAlarmList* in the CMIP SS (32.111-4)
Source:	⌘	SA5
Work item code:	⌘	OAM-NIM
Date:	⌘	24/05/2002
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
2 *(GSM Phase 2)*
R96 *(Release 1996)*
R97 *(Release 1997)*
R98 *(Release 1998)*
R99 *(Release 1999)*
REL-4 *(Release 4)*
REL-5 *(Release 5)*

Reason for change:	⌘	Alignment of CMIP SS (32.111-4) with Alarm IRP IS (32.111-2). The notification *notifyPotentialFaultyAlarmList* has been introduced in the Alarm IRP IS (32.111-2). This CR reflects this change in the CMIP SS (32.111-4).
Summary of change:	⌘	The notification *notifyPotentialFaultyAlarmList* is added to the CMIP SS.
Consequences if not approved:	⌘	The CMIP SS would not be aligned with the IS.

Clauses affected:	⌘	4.7.4, 4.7.5, 5, 6
Other specs affected:	⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ O&M Specifications
Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

4.7.4 Mapping of Notifications

Table 10 maps the Notifications defined in the Information Service of the Alarm IRP to the equivalent Notifications of the CMIP solution set for the Alarm IRP. The CMIP Notifications are qualified as Mandatory (M) or Optional (O).

Table 10: Mapping of Notifications

Notifications of Information Services of the Alarm IRP	Equivalent Notifications of the CMIP solution set for the Alarm IRP	Qualifier
notifyNewAlarm	environmentalAlarm ITU-T X.721 [4] equipmentAlarm ITU-T X.721 [4] qualityofServiceAlarm ITU-T X.721 [4] processingErrorAlarm ITU-T X.721 [4] communicationAlarm ITU-T X.721 [4]	M
notifyChangedAlarm	environmentalAlarm ITU-T X.721 [4] equipmentAlarm ITU-T X.721 [4] qualityofServiceAlarm ITU-T X.721 [4] processingErrorAlarm ITU-T X.721 [4] communicationAlarm ITU-T X.721 [4]	O
notifyClearedAlarm	environmentalAlarm ITU-T X.721 [4] equipmentAlarm ITU-T X.721 [4] qualityofServiceAlarm ITU-T X.721 [4] processingErrorAlarm ITU-T X.721 [4] communicationAlarm ITU-T X.721 [4]	M
notifyAckStateChanged	environmentalAlarm ITU-T X.721 [4] equipmentAlarm ITU-T X.721 [4] qualityofServiceAlarm ITU-T X.721 [4] processingErrorAlarm ITU-T X.721 [4] communicationAlarm ITU-T X.721 [4]	M
notifyAlarmListRebuilt	alarmListRebuilt	M
notifyComments	notifyComments	O
notifyPotentialFaultyAlarmList	notifyPotentialFaultyAlarmList	O

4.7.5 Mapping of Parameters of each notification

The notifications defined in [9] (Alarm IRP: Information Services) have a set of parameters that are common to all the notifications (see [10]: Notification IRP:CMIP Solution Set). Such common set of parameters are:

ManagedObjectClass, ManagedObjectInstance, EventTime, NotificationType, NotificationId.

In the CMIP Solution Set, all the notifications originated within the Agent are reported to the Managers by means of the CMISE "M-EVENT-REPORT" primitive, which is implemented by means of the "m-EventReport OPERATION" (see [2] and[3]). The argument of m-EventReport OPERATION is defined in [3] as follows:

```
EventReportArgument         ::= SEQUENCE {
        managedObjectClass          ObjectClass,
        managedObjectInstance       ObjectInstance,
        eventTime                       [5] IMPLICIT GeneralizedTime OPTIONAL,
        eventType                       EventTypeId,
        eventInfo                               [8] ANY DEFINED BY eventType OPTIONAL
        }
```

where eventinfo is further specified, for each specific notification, by means of specific GDMO/ASN1 definitions.

Therefore the first four parameters of the notification header are mapped to the first four fields of the EventReportArgument. The fifth parameter of the notification header is mapped to the eventinfo field of the EventReportArgument, together with all the other (not common) parameters of the notification.

In the following tables, for the notifications defined in [9], all the parameters (but the common ones) are mapped to their corresponding elements of the CMIP SS notifications.

The parameter SystemDN defined in [9] (Alarm IRP: Information Services) is not used in this CMIP Solution Set.

Table 11: Mapping of Parameters of "notifyNewAlarm"' and "notifyClearedAlarm"'

Notification parameters of Information Services	CMIP Notification equivalences	Qualifier
notificationId	notificationIdentifier (note 1)	M
probableCause	probableCause	M
specificProblems	specificProblems	O
perceivedSeverity	perceivedSeverity	M
backedUpStatus	backedUpStatus	O
backUpObject	backUpObject	O
trendIndication	trendIndication	O
thresholdInfo	thresholdInfo	O
correlatedNotifications	correlatedNotifications	O
stateChangeDefinition	stateChangeDefinition	O
monitoredAttributes	monitoredAttributes	O
proposedRepairActions	proposedRepairActions	O
additionalText	additionalText	O
alarmId	-- (note 2)	
NOTE 1: notificationIdentifier is a parameter of the Notification Header also defined in 3GPP TS 32.302. NOTE 2: In the CMIP Solution Set the alarmId is not used. In the CMIP Solution Set the alarm notifications are univocally identified by means of notificationIdentifier and managedObjectInstance.		

Table 12: Mapping of Parameters of 'notifyAckStateChanged'

Notification parameters of Information Services	CMIP Notification equivalences	Qualifier
notificationId	notificationIdentifier (note 1)	M
probableCause	probableCause	M
specificProblems	specificProblems	O
perceivedSeverity	perceivedSeverity	M
alarmId	-- (note-2)	
ackTime	additionalInformation	M
ackState		M
ackUserId		M
ackSystemId		O
NOTE 1: notificationIdentifier is a parameter of the Notification Header also defined in 3GPP TS 32.302. NOTE 2: In the CMIP Solution Set the alarmId is not used. In the CMIP Solution Set the alarm notifications are univocally identified by means of notificationIdentifier and managedObjectInstance.		

Table 13: Mapping of Parameters of 'notifyAlarmListRebuilt'

Notification parameters of Information Services	CMIP Notification equivalents	Qualifier
notificationId	notificationIdentifier (note)	M
reason	reason	M
objectClass	rebuiltObjectClass	M
objectInstance	rebuiltObjectInstance	M
NOTE: notificationIdentifier is a parameter of the Notification Header also defined in 3GPP TS 32.302.		

Table 14: Mapping of Parameters of 'notifyComments'

Notification parameters of Information Services	CMIP Notification equivalents	Qualifier
notificationId	notificationIdentifier (note 1)	M
objectClass	alarmedObjectClass	M
objectInstance	alarmedObjectInstance	M
notificationId	notificationIdentifier	M
eventTime	alarmEventTime	M
notificationType	eventType	M
alarmType	alarmType	M
probableCause	alarmProbableCause	M
perceivedSeverity	alarmPerceivedSeverity	M
comments	comments	M
alarmId	-- (note 2)	
NOTE 1: notificationIdentifier is a parameter of the Notification Header also defined in 3GPP TS 32.302. NOTE 2: In the CMIP Solution Set the alarmId is not used. In the CMIP Solution Set the alarm notifications are univocally identified by means of notificationIdentifier and managedObjectInstance.		

Table 15: Mapping of Parameters of 'notifyPotentialFaultyAlarmList'

IS Parameter Name	CMIP SS Equivalent	Qualifier
objectClass	M-EVENT REPORT parameter 'Event information': potentialFaultyObjectClass	M
objectInstance	M-EVENT REPORT parameter 'Event information': potentialFaultyObjectInstance	M
notificationId	M-EVENT REPORT parameter 'Event information': notificationIdentifier	M
eventTime	M-EVENT REPORT parameter 'Event ´time'	M
systemDN	--	--
notificationType	M-EVENT REPORT parameter: 'Event type'	M
reason	M-EVENT REPORT parameter 'Event information': reason	M

5 GDMO definitions

5.1 Managed Object Classes

5.1.1 alarmControl

This Managed Object Class (MOC) models the alarm information available within the Agent and significant for the NM-EM interface. It deals with both **active** and **cleared but not yet acknowledged** alarms. The NMC may initiate the transfer of current alarms according to the required parameters in the M-ACTION request 'getAlarmList'.

```
alarmControl MANAGED OBJECT CLASS
    DERIVED FROM
        "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        alarmControlBasicPackage,
        alarmAcknowledgementPackage,
        alarmIRPVersionPackage;

    CONDITIONAL PACKAGES
        alarmCountPackage PRESENT IF "an instance supports it",
        alarmCommentPackage PRESENT IF "an instance supports it",
        alarmProfilePackage PRESENT IF "an instance supports it",
        alarmUnacknowledgementPackage PRESENT IF "an instance supports it ",
        alarmPotentialFaultyAlarmListPackage PRESENT IF "an instance supports it ";
REGISTERED AS { ts32-111AlarmObjectClass 1};
```

5.2 Packages

5.2.1 alarmControlBasicPackage

alarmControlBasicPackage **PACKAGE**
BEHAVIOUR
alarmControlBasicPackageBehaviour;
ATTRIBUTES
alarmControlId GET,
alarmsCountSummary GET;
ACTIONS

getAlarmList;
NOTIFICATIONS
alarmListRebuilt;
REGISTERED AS { ts32-111AlarmPackage 1};

alarmControlBasicPackageBehaviour **BEHAVIOUR**
DEFINED AS

"The MOC alarmControl has been defined to provide information to the Manager about the currently alarms controlled by the Agent.

An instance of the 'alarmControl' MOC is identified by the value of the attribute 'alarmControlId'.

The attribute 'alarmsCountSummary' provides a summary of the number of alarms managed in the Agent's alarm list (including the number of cleared but not yet acknowledged alarms).

The action 'getAlarmList' is the means, for the Manager, to trigger an alarm alignment procedure in accordance with the parameter specified in the action request (this may be needed e.g. for first time alignment or after a link interruption between the Agent and the Manager). The alarm list is sent as a sequence of single alarm reports.

The notification 'alarmListRebuilt' is sent by the Agent to the Manager to inform that the alarm list has changed. It is recommended that the Manager subsequently triggers an alarm alignment.";

5.2.2 alarmCountPackage

alarmCountPackage **PACKAGE**
BEHAVIOUR
alarmCountPackageBehaviour;
ACTIONS
getAlarmCount;
REGISTERED AS { ts32-111AlarmPackage 2};

alarmCountPackageBehaviour **BEHAVIOUR**
DEFINED AS

"This package has been defined to allow the Managers to get information from the Agent about the number of alarms currently present in the alarm list.";

5.2.3 alarmAcknowledgementPackage

alarmAcknowledgementPackage **PACKAGE**
BEHAVIOUR
alarmAcknowledgementPackageBehaviour;
ACTIONS
acknowledgeAlarms;
NOTIFICATIONS
"Rec. X.721 | ISO/IEC 10165-2 : 1992":communicationsAlarm,
"Rec. X.721 | ISO/IEC 10165-2 : 1992":environmentalAlarm,
"Rec. X.721 | ISO/IEC 10165-2 : 1992":equipmentAlarm,
"Rec. X.721 | ISO/IEC 10165-2 : 1992":processingErrorAlarm,
"Rec. X.721 | ISO/IEC 10165-2 : 1992":qualityofServiceAlarm;
REGISTERED AS { ts32-111AlarmPackage 3};

alarmAcknowledgementPackageBehaviour **BEHAVIOUR**
DEFINED AS

"This package has been defined to provide information to the Manager about the acknowledgement status of the alarms controlled by the Agent.

The action 'acknowledgeAlarms' allows the NM operator to acknowledge one or several alarms previously sent by the Agent as alarm notifications.

The ITU-T Recommendation X.721 [4] compliant alarm notifications are sent by the Agent to the Manager to inform that one alarm has been acknowledged. The acknowledgement related information is carried in the *additionalInformation* attribute.";

5.2.4 alarmUnacknowledgementPackage

alarmUnacknowledgementPackage **PACKAGE**
BEHAVIOUR
alarmUnacknowledgementPackageBehaviour;
ACTIONS
unacknowledgeAlarms;
REGISTERED AS { ts32-111AlarmPackage 4};

alarmUnacknowledgementPackageBehaviour **BEHAVIOUR**
DEFINED AS

"This package has been defined to provide the Manager with the capability to un-acknowledge alarms.

The action 'unacknowledgeAlarms' allows the NM operator to un-acknowledge one or several alarms previously acknowledged by him.

The ITU-T Recommendation X.721 [4] compliant alarm notifications are sent by the Agent to the Manager to inform that one alarm has been unacknowledged. The acknowledgement related information is carried in the *additionalInformation* attribute.";

5.2.5 alarmCommentPackage

alarmCommentPackage **PACKAGE**
BEHAVIOUR
alarmCommentPackageBehaviour;
ACTIONS
setComment;
NOTIFICATIONS
notifyComments;
REGISTERED AS { ts32-111AlarmPackage 5};

alarmCommentPackageBehaviour **BEHAVIOUR**
DEFINED AS

"This package has been defined to allow the Operators to write comments about alarms that are in the alarm list of the IRP Agent.";

5.2.6 alarmIRPVersionPackage

alarmIRPVersionPackage **PACKAGE**
BEHAVIOUR
alarmIRPVersionPackageBehaviour;
ATTRIBUTES
supportedAlarmIRPVersions GET;
ACTIONS
getAlarmIRPVersion;
REGISTERED AS { ts32-111AlarmPackage 6};

alarmIRPVersionPackageBehaviour **BEHAVIOUR**
DEFINED AS

"This package has been defined to allow the Manager to get information about the Alarm IRP versions supported by the Agent.

The attribute 'supportedAlarmIRPVersions' indicates all versions of the Alarm IRP currently supported by the Agent.

The action 'getAlarmIRPVersion' may be invoked by the Manager to get information about the Alarm IRP versions supported by the Agent. Such Alarm IRP versions must compatible to each other. This means that the Manager may use any one of such Alarm IRP versions";

5.2.7 alarmProfilePackage

alarmProfilePackage **PACKAGE**
BEHAVIOUR
alarmProfilePackageBehaviour;
ACTIONS
getOperationProfile,
getNotificationProfile;

REGISTERED AS { ts32-111AlarmPackage 7};

alarmProfilePackageBehaviour **BEHAVIOUR**

DEFINED AS

"This package has been defined to allow the Manager to get detailed information about the profile of Alarm IRP.

The action 'getOperationProfile' is invoked by the Manager to get detailed information about the operations supported by Alarm IRP.

The action 'getNotificationProfile' is invoked by the Manager to get detailed information about the notifications supported by Alarm IRP.";

5.2.8 alarmPotentialFaultyAlarmListPackage

alarmPotentialFaultyAlarmListPackage **PACKAGE**
BEHAVIOUR
alarmPotentialFaultyAlarmListPackageBehaviour;
NOTIFICATIONS
notifyPotentialFaultyAlarmList;
REGISTERED AS {ts32-111AlarmPackage 8};

alarmPotentialFaultyAlarmListPackageBehaviour **BEHAVIOUR**
DEFINED AS
"This package allows the IRPAgent to inform the IRPManager that the alarm list held by the IRPAgent might be faulty.";

5.3 Actions

5.3.1 acknowledgeAlarms (M)

acknowledgeAlarms **ACTION**
BEHAVIOUR
acknowledgeAlarmsBehaviour;
MODE
CONFIRMED;
WITH INFORMATION SYNTAX
TS32-111-4TypeModule .AckOrUnackAlarms;
WITH REPLY SYNTAX
TS32-111-4TypeModule .AckOrUnackAlarmsReply;
REGISTERED AS { ts32-111AlarmAction 1};

acknowledgeAlarmsBehaviour **BEHAVIOUR**
DEFINED AS

"This action is invoked by the Manager to indicate to the Agent that one or several alarms (previously sent by the Agent as alarm notifications) have to be acknowledged. In the action request the NM supplies the parameter *ackUserId* and *ackSystemId*. The other acknowledgement history parameters, i.e. alarm acknowledgement state (in this case *acknowledged*) and the acknowledgement time are set by the Agent itself.

The 'Action information' field contains the following data:

- *alarmReferenceList*

  This parameter contains a set of MOI (Managed Object Instance) and *notificationIdentifier*. Each pair identifies unambiguously in the scope of the Agent an alarm (previously received by the NM) that have to be now acknowledged. MOI can be absent if scope of uniqueness of notificationIdentifier is across the IRPAgent.

- *ackUserId*

  It contains the name of the operator who acknowledged the alarm or a generic name (dependent on the operational concept). It may have also the value NULL.

- *ackSystemId*

  It indicates the management system where the acknowledgment is triggered. It may have also the value NULL.

  The 'Action response' contains the following data:

- *status*

This parameter contains the results of the NM acknowledgement action. Possible values: noError (0, all alarms found and ack state changed according to the manager request), ackPartlySuccessful (some alarms not found / not changeable, see next parameter), error (value indicates the reason why the complete operation failed).

- *errorAlarmReferenceList*

  This parameter (significant only if *status* = ackPartlySuccessful) contains the list of moi (managed object instance) and notificationIdentifier pairs of the alarms which could not be acknowledged and, for each alarm, also the reason of the error.";

5.3.2 getAlarmCount (O)

getAlarmCount **ACTION**
BEHAVIOUR
getAlarmCountBehaviour;
MODE
CONFIRMED;
WITH INFORMATION SYNTAX
TS32-111-4TypeModule .GetAlarmCount;
WITH REPLY SYNTAX
TS32-111-4TypeModule .GetAlarmCountReply;
REGISTERED AS { ts32-111AlarmAction 2};

getAlarmCountBehaviour **BEHAVIOUR**
DEFINED AS

"The NM invokes this action to receive the number of available alarms in the Agent' alarm list according to the specification in the action request. The Manager may use this action to find out the number of alarms in the alarm list before invoking a synchronisation by means of the *getAlarmList* operation. The request is possible also before the Manager creates an own event forwarding discriminator instance within the Agent.

The 'Action information' field contains the following data:

- *alarmAckState*

  Depending on this optional parameter value, the NM gets the number of alarms of each *perceivedSeverity* value according to the following possible choices:

  - all alarms
  - all active alarms (acknowledged or not yet acknowledged)
  - all active and acknowledged alarms
  - all active and unacknowledged alarms
  - all cleared and unacknowledged alarms.

  If the parameter is absent, all alarms from the Agent's alarm list are taken into consideration.

- *filter*

  The handling of this optional parameter is as follows:

- if present and not NULL, it indicates a filter constraint which shall apply in the calculation of the results
- if its value is NULL, no filter shall be considered and the Agent shall return the number of all alarms according to the value of the parameter *alarmAckState* (see above)
- if absent, the handling depends on the availability of an event forwarding discriminator instance within the Agent. If this instance is valid, the filter construct of the event forwarding discriminator shall apply. If no EFD instance is available, the Agent shall return the number of all alarms according to the value of the above-mentioned parameter *alarmAckState*.

The 'Action response' is composed of:

- The numbers of alarms for each *perceivedSeverity* value (if applicable).
- The parameter *status* containing the results of the NM action. Possible values: noError (0), error (the value indicates the reason of the error).";

5.3.3 getAlarmList (M)

getAlarmList **ACTION**
BEHAVIOUR
getAlarmListBehaviour;
MODE
CONFIRMED;
WITH INFORMATION SYNTAX
TS32-111-4TypeModule .GetAlarmList;
WITH REPLY SYNTAX
TS32-111-4TypeModule .GetAlarmListReply;
REGISTERED AS { ts32-111AlarmAction 3};

getAlarmListBehaviour **BEHAVIOUR**
DEFINED AS

"This action starts an alarm alignment procedure between a NM and Agent, which takes into account the acknowledgment state of the alarms and a dedicated filter (valid only for the current request).

The 'Action information' field contains the following data:

- *alarmAckState*

Depending on this optional parameter value, the NM gets the alarm reports according to the following possible choices:

- all alarms
- all active alarms (acknowledged or not yet acknowledged)
- all active and acknowledged alarms
- all active and unacknowledged alarms
- all cleared and unacknowledged alarms.

If the parameter is absent, all alarms from the Agent's alarm list are taken into consideration.

- *destination*

This parameter identifies the destination to which the alarm reports that have passed the test conditions specified in the parameter 'filter' are sent. According to ITU-T Recommendation X.721 [4], if no destination is specified in the request, then the discriminator is created with the destination defaulted to the AE-Title of the invoker.

- *filter*

The handling of this optional parameter (valid only for the current alignment request) is as follows:

- if present and not NULL, it indicates a filter constraint which shall apply in the forwarding of the alignment-related alarm reports

- if its value is NULL, no real filter shall be considered and the Manager receives the alarms according to the value of the parameter *alarmAckState* (see above).

The 'Action response' contains the following data:

- *alignmentId*

  The parameter is defined by the Agent and identifies unambiguously the current alarm alignment procedure. It allows the Manager to distinguish between alarm reports sent as consequence of several own alignment requests triggered in parallel.

- *status*

  The parameter contains the results of the NM action. Possible values: noError (0), error (the value indicates the reason of the error).

  After the action response is forwarded to the NM, the Agent sends the alarm list as a sequence of single alarm notifications in accordance with the values of the request parameters. Every alarm notification contains all fields of the alarm stored in the alarm list. In particular:

- The field *additionalText* contains at the beginning a string to allow a Manager to recognise that this alarm report is sent due to a previous *getAlarmList* request. The structure of this string is:

- '(ALIGNMENT-alignmentId)' for every alarm report except the last one **or**

- '(ALIGNMENTEND-alignmentId)' for the last alarm report sent by the Agent due to the current *getAlarmList* request.

- If available, the data related to the acknowledgment history (i.e. ackState, ackTime, ackUserId, ackSystemId) are provided in the field *additionalInformation*.

  Further details about the implementation of this operation are provided in the 'Introduction'.";

5.3.4 setComment (O)

setComment **ACTION**
BEHAVIOUR
setCommentBehaviour;
MODE
CONFIRMED;
WITH INFORMATION SYNTAX
TS32-111-4TypeModule .SetComment;
WITH REPLY SYNTAX
TS32-111-4TypeModule .SetCommentReply;
REGISTERED AS { ts32-111AlarmAction 4};

setCommentBehaviour **BEHAVIOUR**
DEFINED AS

"The NM invokes this action to associate a comment to one or more alarms.

The 'Action information' field contains:

- alarmReferenceList
  Contains a list of alarm identifiers to which the comment must be associated.

- commentUserId
  Contains the identity of the User that invokes this operation.

- commentSystemId
  Contains the identity of the NM that invokes this operation.

- commentText
  Contains the text of the comment.

The 'Action response' is composed of the following data:

- errorAlarmReferenceList
  List of pair of `alarmId` and failure reason.

- *status*
  It contains the results of the NM action. Possible values: actionSucceeded (0), actionPartiallyFailed (12) or another value indicating the reason of the error.";

5.3.5 getAlarmIRPVersion (M)

getAlarmIRPVersion **ACTION**
BEHAVIOUR
getAlarmIRPVersionBehaviour;
MODE
CONFIRMED;
WITH REPLY SYNTAX
TS32-111-4TypeModule .GetAlarmIRPVersionReply;
REGISTERED AS { ts32-111AlarmAction 5};

getAlarmIRPVersionBehaviour **BEHAVIOUR**
DEFINED AS

"The NM invokes this action to get information about the Alarm IRP versions supported by the Agent.

The 'Action information' field contains no data.

The 'Action response' is composed of the following data:

- *versionNumbersList*

  It defines a list of Alarm IRP versions supported by the Agent. A list containing no element, i.e. a NULL list means that the concerned Agent doesn't support any version of the Notification IRP.

- *status*

  It contains the results of the NM action. Possible values: noError (0), error (the value indicates the reason of the error).";

5.3.6 getNotificationProfile (O)

getNotificationProfile **ACTION**
BEHAVIOUR
getNotificationProfileBehaviour;
MODE
CONFIRMED;
WITH INFORMATION SYNTAX
TS32-111-4TypeModule.IRPVersionNumber;
WITH REPLY SYNTAX
TS32-111-4TypeModule.GetNotificationProfileReply;
REGISTERED AS { ts32-111AlarmAction 6};

getNotificationProfileBehaviour **BEHAVIOUR**

DEFINED AS

"A Manager invokes this action to enquiry about the notification profile (supported notifications and supported parameters) for this specific Alarm IRP version.

The 'Action information' contains the following data:

- *irpVersionNumber*
  This mandatory parameter identifies the Alarm IRP version.

The 'Action response' is composed of the following data:

- *notificationNameProfile*
  It contains a list of notification names, i.e. a NULL list means that the Alarm IRP doesn't support any notification.

- *notificationParameterProfile*.
  It contains a set of elements, each element corresponds to a notification name and is composed by a set of parameter names.

- *status*
  It contains the results of this action. Possible values: noError (0), error (the value indicates the reason of the error).";

5.3.7 getOperationProfile (O)

getOperationProfile **ACTION**
BEHAVIOUR
getOperationProfileBehaviour;
MODE
CONFIRMED;
WITH INFORMATION SYNTAX
TS32-111-4TypeModule.IRPVersionNumber;
WITH REPLY SYNTAX
TS32-111-4TypeModule.GetOperationProfileReply;
REGISTERED AS { ts32-111AlarmAction 7};

getOperationProfileBehaviour **BEHAVIOUR**

DEFINED AS

"A Manager invokes this action to enquiry about the operation profile (supported operations and supported parameters) for this specific Alarm IRP version.

The 'Action information' contains the following data:

- *irpVersionNumber*
  This mandatory parameter identifies the Alarm IRP version.

The 'Action response' is composed of the following data:

- *operationNameProfile*
  It contains a list of operation names.

- *operationParameterProfile*.
  It contains a set of elements, each element corresponds to an operation name and is composed by a set of parameter names.

  status
  It contains the results of this action. Possible values: noError (0), error (the value indicates the reason of the error).";

5.3.8 unacknowledgeAlarms (O)

unacknowledgeAlarms **ACTION**

BEHAVIOUR
unacknowledgeAlarmsBehaviour;
MODE
CONFIRMED;
WITH INFORMATION SYNTAX
TS32-111-4TypeModule .AckOrUnackAlarms;
WITH REPLY SYNTAX
TS32-111-4TypeModule .AckOrUnackAlarmsReply;
REGISTERED AS { ts32-111AlarmAction 8};
unacknowledgeAlarmsBehaviour **BEHAVIOUR**
DEFINED AS

"This action is used by the Manager to indicate to the Agent that one or several alarms (previously acknowledged) have to be unacknowledged. Subsequently the 'acknowledgement history' information of these alarms in the Agent's alarm list is completely removed (this operation may be used by operators in case of a previous acknowledgement by mistake).

The 'Action information' field contains the following data:

alarmReferenceList
This parameter contains a set of MOI (Managed Object Instance) and *notificationIdentifier pair*. Each of them identifies unambiguously in the scope of the Agent an alarm (previously acknowledged by the NM) that have to be now unacknowledged. MOI can be absent if scope of uniqueness of notificationIdentifier is across the IRPAgent.

- *ackUserId*

  It contains the name of the operator who unacknowledged the alarm or a generic name (dependent on the operational concept). It may have also the value NULL. Note that only the user who previously acknowledged the alarm is allowed to un-acknowledge it later.

- *ackSystemId*
  It indicates the management system where the acknowledgment is triggered. It may have also the value NULL. Note that the un-acknowledgement is allowed only at the management system where previously the acknowledgement took place.

  The 'Action response' contains the following data:

- *status*

  This parameter contains the results of the NM un-acknowledgement action. Possible values: noError (0, all alarms found and ack state changed according to the manager request), unackPartlySuccessful (some alarms not found / not changeable, see next response parameter), error (value indicates the reason why the complete operation failed).

- *errorAlarmReferenceList*

  This parameter (significant only if *status* = unackPartlySuccessful) contains the list of MOI (Managed Object Instance) and notificationIdentifier pairs of the alarms which could not be unacknowledged and, for each alarm, also the reason of the error. MOI can be absent if scope of uniqueness of notificationIdentifier is across the IRPAgent. ";

5.4 Notifications

5.4.1 alarmListRebuilt (M)

alarmListRebuilt **NOTIFICATION**
BEHAVIOUR
alarmListRebuiltBehaviour;
WITH INFORMATION SYNTAX
TS32-111-4TypeModule .AlarmListRebuiltInfo;
REGISTERED AS { ts32-111AlarmNotification 1};

alarmListRebuiltBehaviour **BEHAVIOUR**
DEFINED AS

"This notification is used by the Agent to inform the NM that the alarm list has been rebuilt.

The 'Event Information' field contains the following data:

- *notificationIdentifier*

  This ITU-T X.721 standardised parameter, together with MOI (Managed Object Instance), unambiguously identifies this notification.

- *rebuiltObjectClass*
  This parameter carries the IRPAgent MOC when the entire AlarmList has been rebuilt. It carries a different MOC when the AlarmList has been partially rebuilt.

- *rebuiltObjectInstance*
  This parameter carries DN of the IRPAgent when the entire AlarmList has been rebuilt. It carries the DN of another MOI when the AlarmList has been partially rebuilt and only the MOIs subordinate of this rebuilt MOI may be affected by this partial rebuilt.

- *reason*

  The parameter indicates the reason for alarm list rebuilding (if applicable).";

5.4.2 notifyComments (O)

notifyComments **NOTIFICATION**
BEHAVIOUR
notifyCommentsBehaviour;
WITH INFORMATION SYNTAX
TS32-111-4TypeModule .NotifyComments;
REGISTERED AS { ts32-111AlarmNotification 2};

notifyCommentsBehaviour **BEHAVIOUR**
DEFINED AS

"This notification is used by the Agent to inform the NM that one or more comments have been associated to one alarm.

The 'Event Information' field contains the following data:

- alarmedObjectClass: defined in ITU-T X.710 [2] and X.711 [3]
- alarmedObjectInstance: defined in ITU-T X.710 [2] and X.711 [3]
- notificationIdentifier: This ITU-T X.721 standardised parameter, together with MOI (Managed Object Instance), unambiguously identifies this notification.
- alarmEventTime: defined in ITU-T X.721
- alarmType: the eventType of the alarm to which this comment is associated.
- alarmProbableCause: defined in ITU-T X.721
- alarmPerceivedSeverity: defined in ITU-T X.721
- `comments`: the text of the comment.

";

5.4.3 notifyPotentialFaultyAlarmList (O)

notifyPotentialFaultyAlarmList **NOTIFICATION**
BEHAVIOUR
notifyPotentialFaultyAlarmListBehaviour;
WITH INFORMATION SYNTAX
TS32-111-4TypeModule. NotifyPotentialFaultyAlarmListInfo
REGISTERED AS { ts32-111AlarmNotification 3};

notifyPotentialFaultyAlarmListBehaviour **BEHAVIOUR**
DEFINED AS
"This notification is used by the IRPAgent to inform the IRPManager that the IRPAgent has lost confidence in the integrity of its alarm list.

The 'Event information' field contains the following data:

- *potentialFaultyObjectClass*

This parameter specifies together with the parameter *potentialFaultyObjectInstance* the unreliable alarm information instances in the alarm list.

If this parameter carries the MOC of the IRPAgent, then the entire alarm list is unreliable.

If this parameter carries the MOC of another MO, then only a part of the alarm list is unreliable. The mechanism for identifying the unreliable part is described below.

- *potentialFaultyObjectInstance*

This parameter specifies together with the parameter *potentialFaultyObjectClass* the unreliable alarm information instances in the alarm list.

If *potentialFaultyObjectClass* carries the MOC of the IRPAgent, the this parameter carries the DN of the IRPAgent and the entire alarm list is unreliable.

If *potentialFaultyObjectClass* carries the MOC of another MO, then this parameter carries the DN of an instance of this class. All alarm information instances representing alarms raised by this MOI and its subordinates may be unreliable in this case.

- *notificationIdentifier*

This parameter specifies the notification identifier (ITU-T X.733 [5]), which, together with the instance of the object emitting this notification, unambiguously identifies this notification.

- *reason*

This parameter specifies the reason why the IRPAgent has lost confidence in the integrity of its alarm list and needs to rebuild it.

";

5.5 Attributes

5.5.1 alarmControlId

alarmControlId **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
TS32-111-4TypeModule .GeneralObjectId;
MATCHES FOR
EQUALITY;
BEHAVIOUR
alarmControlIdBehaviour;
REGISTERED AS { ts32-111AlarmAttribute 1};

alarmControlIdBehaviour **BEHAVIOUR**

DEFINED AS
"This attribute names an instance of a 'alarmControl' object class.";

5.5.2 alarmsCountSummary

alarmsCountSummary **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
TS32-111-4TypeModule .AlarmsCountSummary;
MATCHES FOR
EQUALITY;
BEHAVIOUR
alarmsCountSummaryBehaviour;
REGISTERED AS { ts32-111AlarmAttribute 2};

alarmsCountSummaryBehaviour **BEHAVIOUR**
DEFINED AS

"This attribute indicates a summary of number of alarms managed in the Agent's alarm list sorted according to the perceived severity (including the number of cleared but not yet acknowledged alarms). Additionally the number of all currently active alarms is provided.";

5.5.3 supportedAlarmIRPVersions

supportedAlarmIRPVersions **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
TS32-111-4TypeModule . SupportedAlarmIRPVersions;
MATCHES FOR
EQUALITY;
BEHAVIOUR
supportedAlarmIRPVersionsBehaviour;
REGISTERED AS { ts32-111AlarmAttribute 3};

supportedAlarmIRPVersionsBehaviour **BEHAVIOUR**
DEFINED AS
"This attribute provides the information concerning the Alarm IRP versions currently supported by the Agent.";

5.6 Parameters

5.6.1 ackStateParameter

ackStateParameter **PARAMETER**
CONTEXT
TS32-111-4TypeModule .AlarmInfo.additionalInformation;
WITH SYNTAX
TS32-111-4TypeModule .AckState;
BEHAVIOUR
ackStateParameterBehaviour;
REGISTERED AS { ts32-111AlarmParameter 1};

ackStateParameterBehaviour **BEHAVIOUR**
DEFINED AS
"This parameter models the optional *additionalInformation* field of the alarm notification. If present, it informs the NM about the current acknowledgement state of the present alarm.";

5.6.2 ackSystemIdParameter

ackSystemIdParameter **PARAMETER**
CONTEXT
TS32-111-4TypeModule .AlarmInfo.additionalInformation;
WITH SYNTAX
TS32-111-4TypeModule .SystemId;
BEHAVIOUR
ackSystemIdParameterBehaviour;
REGISTERED AS { ts32-111AlarmParameter 2};

ackSystemIdParameterBehaviour **BEHAVIOUR**
DEFINED AS
"This parameter models the optional *additionalInformation* field of the alarm notification. If present, it informs the NM about the identifier of the management system where the present alarm has been acknowledged.";

5.6.3 ackTimeParameter

ackTimeParameter **PARAMETER**
CONTEXT
TS32-111-4TypeModule .AlarmInfo.additionalInformation;
WITH SYNTAX
TS32-111-4TypeModule .AckTime;
BEHAVIOUR
ackTimeParameterBehaviour;
REGISTERED AS { ts32-111AlarmParameter 3};

ackTimeParameterBehaviour **BEHAVIOUR**
DEFINED AS
"This parameter models the optional *additionalInformation* field of the alarm notification. If present, it informs the NM about the time the present alarm has been acknowledged by the Agent.";

5.6.4 ackUserIdParameter

ackUserIdParameter **PARAMETER**
CONTEXT
TS32-111-4TypeModule .AlarmInfo.additionalInformation;
WITH SYNTAX
TS32-111-4TypeModule.UserId;
BEHAVIOUR
ackUserIdParameterBehaviour;
REGISTERED AS { ts32-111AlarmParameter 4};

ackUserIdParameterBehaviour **BEHAVIOUR**
DEFINED AS
"This parameter models the optional *additionalInformation* field of the alarm notification. If present, it informs the NM about the identifier of the user who acknowledged the present alarm.";

6 ASN.1 definitions for Alarm IRP

```
TS32-111-4TypeModule {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0) umts-Operation-
Maintenance(3) ts-32-111(111) part4(4) informationModel(0) asn1Module(2) version1(1)}

DEFINITIONS IMPLICIT TAGS ::=
BEGIN
--EXPORTS everything
IMPORTS

NotificationIdentifier, Destination, EventTime, ProbableCause, PerceivedSeverity
FROM Attribute-ASN1Module {joint-iso-ccitt ms(9) smi(3) part2(2) asn1Module(2) 1}

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

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

baseNodeUMTS  OBJECT IDENTIFIER ::= { itu-t (0) identified-organization (4) etsi (0) mobileDomain
(0)
umts-Operation-Maintenance (3) }
ts32-111Prefix          OBJECT IDENTIFIER ::= { baseNodeUMTS ts-32-111(111)}
ts32-111Part4               OBJECT IDENTIFIER ::= { ts32-111Prefix part4(4)}
ts32-111-4InfoModel OBJECT IDENTIFIER ::= { ts32-111Part4 informationModel(0)}

ts32-111AlarmObjectClass        OBJECT IDENTIFIER ::= { ts32-111-4InfoModel managedObjectClass(3)}
ts32-111AlarmPackage            OBJECT IDENTIFIER ::= { ts32-111-4InfoModel package(4)}
ts32-111AlarmParameter      OBJECT IDENTIFIER ::= { ts32-111-4InfoModel parameter(5)}
ts32-111AlarmAttribute          OBJECT IDENTIFIER ::= { ts32-111-4InfoModel attribute(7)}
ts32-111AlarmAction         OBJECT IDENTIFIER ::= { ts32-111-4InfoModel action(9)}
ts32-111AlarmNotification       OBJECT IDENTIFIER ::= { ts32-111-4InfoModel action(10)}



-- Start of 3GPP SA5 own definitions
AckErrorList ::= SET OF ErrorInfo
AlarmReference ::= SEQUENCE
{
  moi ObjectInstance OPTIONAL, -- absent if scope of uniquness of notificationId is across IRPAgent
  notificationIdentifier NotificationIdentifier
}
AckOrUnackAlarms ::= SEQUENCE
    {
    alarmReferenceList  SET OF AlarmReference,  -- ITU-T X.721
    ackUserId               UserId,
    ackSystemId         SystemId OPTIONAL
    }
AckOrUnackAlarmsReply ::= SEQUENCE
    {
    status                  ErrorCauses,
errorAlarmReferenceList     AckErrorList
    }
AckState ::= ENUMERATED
    {
    acknowledged        (0),
    unacknowledged  (1)
}
AckTime ::= GeneralizedTime
AlarmChoice ::= ENUMERATED
    {
    allAlarms                           (0),
    allActiveAlarms                 (1),
    allActiveAndAckAlarms       (2),
    allActiveAndUnackAlarms     (3),
    allClearedAndUnackAlarms    (4)
}
AlarmsCountSummary ::= SEQUENCE
    {
    activeAlarmsCount           INTEGER,    -- this is the sum of criticalCount, majorCount,
minorCount, warningCount
                                                -- and indeterminateCount
    criticalCount                   INTEGER,
    majorCount                  INTEGER,
    minorCount                  INTEGER,
```

```
    warningCount                INTEGER,
    indeterminateCount          INTEGER,
    clearedCount                    INTEGER
    }
AlarmListRebuiltInfo ::= SEQUENCE
    {
    notificationIdentifier      NotificationIdentifier, -- ITU-T X.721
    rebuiltObjectClass      ObjectClass,
    rebuiltObjectInstance   ObjectInstance,
    reason                  ErrorCauses
    }
Comment ::= GraphicString
ErrorCauses ::= ENUMERATED
{
noError (0),    -- operation / notification successfully performed
wrongFilter (1),    -- the value of the filter parameter is not valid
wrongAlarmAckState (2), -- the value of the alarmAckState parameter (e.g. getAlarmCount) is not
valid
ackPartlySuccessful (3),    -- acknowledgment request partly successful
unackPartlySuccessful (4),  -- unacknowledgment request partly successful
wrongAlarmReference (5),    -- alarm identifier used in the alarm reference list not found  (e.g. in
case of acknowledgement request)
wrongAlarmReferenceList (6),    -- the alarm reference list (e.g. in case of acknowledgement
    request) is empty or completely wrong
alarmAlreadyAck (7),    -- alarm to be acknowledged is already in this state
alarmAlreadyUnack (8),  -- alarm to be acknowledged is already in this state
wrongUserId (9),    -- the user identifier in the unacknowledgement operation is    not the same as
in the previous acknowledgementAlarms request
wrongSystemId (10), -- the system identifier in the unacknowledgement operation is  not the same as
in the previous acknowledgementAlarms request
alarmAckNotAllowed (11),    -- current management system not allowed to acknowledge the alarm (e.g.
due to acknowledgement competence rules)
setCommentPartlySuccessful (12),    -- the setComment action partly successful (e.g. some alarmId
are not in the alarmList)
unspecifiedErrorReason (255)    -- operation failed, specific error unknown
}
ErrorInfo ::= SEQUENCE
    {
      moi ObjectInstance OPTIONAL, -- absent if uniqueness of notificationIdentifier is across
IRPAgent
    notificationIdentifier          NotificationIdentifier, -- ITU-T X.721
    reason          ErrorCauses
    }
GeneralObjectId ::= INTEGER
GetAlarmCount ::= SEQUENCE
    {
    alarmAckState       AlarmChoice OPTIONAL,
    filter                  CMISFilter OPTIONAL -- ITU-T X.711
    }
GetAlarmCountReply ::= SEQUENCE
    {
    criticalCount               INTEGER,
    majorCount              INTEGER,
    minorCount              INTEGER,
    warningCount            INTEGER,
    indeterminateCount      INTEGER,
    clearedCount                INTEGER,
    status                      ErrorCauses
    }
GetAlarmIRPVersionReply ::= SEQUENCE
    {
    versionNumberList   SupportedAlarmIRPVersions,
    status                  ErrorCauses
    }
GetAlarmList ::= SEQUENCE
    {
    alarmAckState       AlarmChoice OPTIONAL,
    destination         Destination,                    -- ITU-T X.721
    filter                  CMISFilter OPTIONAL -- ITU-T X.711
    }
GetAlarmListReply ::= SEQUENCE
    {
    alignmentId     INTEGER,
    status              ErrorCauses
    }
GetNotificationProfileReply ::= SEQUENCE
{
notificationNameProfile         NotificationList,
```

```
notificationParameterProfile    ParameterListOfList,
status                              ErrorCauses
}

GetOperationProfileReply ::= SEQUENCE
{
operationNameProfile            OperationList,
operationParameterProfile       ParameterListOfList,
status                              ErrorCauses
}

IRPVersionNumber ::= GraphicString

NotificationList ::= SET OF NotificationName

NotificationName ::= GraphicString

NotifyComments ::= SEQUENCE
{
alarmedObjectClass          ObjectClass,
alarmedObjectInstance       ObjectInstance,
notificationIdentifier      NotificationIdentifier, -- ITU-T X.721
alarmEventTime              EventTime,
alarmType                   EventTypeId,
alarmProbableCause          ProbableCause,
alarmPerceivedSeverity      PerceivedSeverity,
comments                    SET OF Comment
}

NotifyPotentialFaultyAlarmListInfo ::= SEQUENCE
{
potentialFaultyObjectClass      ObjectClass,            -- ITU-T X.711
potentialFaultyObjectInstance   ObjectInstance,         -- ITU-T X.711
notificationIdentifier          NotificationIdentifier, -- ITU-T X.721
reason                          ReasonPotentialFaultyAlarmList
}

OperationList ::= SET OF OperationName

OperationName ::= GraphicString

ParameterList ::= SET OF ParameterName

ParameterListOfList ::= SET OF ParameterList

ParameterName ::= GraphicString

ReasonPotentialFaultyAlarmList ::= ENUMERATED
{
communicationErrorNEAgent,  (0)     -- A communication error between a NE and the agent has occured.
agentRestart                (1)     -- The agent has restarted and not yet updated its alarm list.
indeterminate               (2)     -- The reasn could not be determined.
}

SetComment ::= SEQUENCE
    {
    alarmReferenceList      SET OF AlarmReference,
    commentUserId           UserId,
    commentSystemId     SystemId,
    commentText             Comment
    }
SetCommentReply ::= SEQUENCE
    {
    badAlarmReferenceList       SET OF ErrorInfo,
    status                          ErrorCauses
    }
SystemId ::= GraphicString

SupportedAlarmIRPVersions ::= SET OF IRPVersionNumber
UserId ::= GraphicString

END -- of module TS32-111-4TypeModule
```