Source: **SA5 (Telecom Management)**

Title: **10 Rel-5/6 CR 32.111-2/3/4 Fault Management; Alarm IRP IS / CORBA / CMIP SSs**

Document for: **Approval**

Agenda Item: **7.5.3**

Doc1stevel	Specific a	CR	R	Phase	Subject	Ca	VersCu	Doc2ndLev	WorkitemsI D
SP-040791	32.111-2	033	--	Rel-5	Correct the time carried by the eventTime parameter of notifyComments	F	5.4.0	S5-046987	OAM-NIM
SP-040791	32.111-2	034	--	Rel-6	Correct the time carried by the eventTime parameter of notifyComments	A	6.2.0	S5-046981	OAM-NIM
SP-040791	32.111-2	035	--	Rel-6	Remove redundant ackTime parameter in notifyAckStateChanged	C	6.2.0	S5-046978	OAM-NIM
SP-040791	32.111-2	036	--	Rel-6	Correct description of thresholdInformation – Align with 32.401	F	6.2.0	S5-047023	OAM-NIM
SP-040791	32.111-2	037	--	Rel-6	Correction of probable cause for alarms.	F	6.2.0	S5-047078	OAM-NIM
SP-040791	32.111-3	034	--	Rel-6	Remove redundant ackTime parameter in notifyAckStateChanged	C	6.0.0	S5-046979	OAM-NIM
SP-040791	32.111-3	035	--	Rel-6	Correction of probable cause definition for AlarmIRP IDL file.	F	6.0.0	S5-047079	OAM-NIM
SP-040791	32.111-3	036	--	Rel-6	Add mandatory exception operationNotSupported for optional operations in AlarmIRP - Align IDL style with IDL Style Guide in 32.150	F	6.0.0	S5-047118	OAM-NIM
SP-040791	32.111-3	037	--	Rel-6	Correction of filterable parameters - Align with the IS in 32.111-2	F	6.0.0	S5-047134	OAM-NIM
SP-040791	32.111-4	029	--	Rel-6	Remove redundant ackTime parameter in notifyAckStateChanged	C	6.2.0	S5-046980	OAM-NIM

CR-Form-v7

CHANGE REQUEST

⌘ 32.111-2 CR 033 ⌘rev - ⌘ Current version: 5.4.0 ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network X Core Network X

Title:	⌘	Correct the time carried by the eventTime parameter of notifyComments
Source:	⌘	**SA5** (olaf.pollakowski@siemens.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	01/10/2004
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)
Rel-6 (Release 6)

Reason for change:	⌘	The eventTime parameter of notifyComments currently carries the alarmChangedTime. However, the eventTime shall carry the time of the event triggering the emission of the related notification which is the Comment.commentTime of the last Comment added in this case.
Summary of change:	⌘	The time stamp carried by the eventTime parameter of notifyComments is changed from alarmChangedTime to Comment.commentTime of the last Comment added.
Consequences if not approved:	⌘	The eventTime carries the wrong time stamp, which may have serious impacts of the fault management procedures at the operation management systems.

Clauses affected:	⌘	6.10.1

Other specs affected: ⌘	Y	N		⌘
		X	Other core specifications	
		X	Test specifications	
	X		O&M Specifications	Rel-6 32.111-2

Other comments:	⌘	Rel-6 Mirror CR in S5-046981.

Change in Clause 6.10.1

6.10.1 notifyComments (O)

6.10.1.1 Definition

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

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

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

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

6.10.1.2 Input Parameters

Parameter Name	Qualifier	Matching Information	Comment
objectClass	M,F	MonitoredEntity.objectClass where the MonitoredEntity is identified by the relation-AlarmedObject-AlarmInformation of the AlarmInformation.	
objectInstance	M,F	MonitoredEntity.objectInstance where the MonitoredEntity is identified by the relation-AlarmedObject-AlarmInformation of the AlarmInformation.	
notificationId	M	This carries the semantics of notification identifier.	
eventTime	M,F	~~AlarmInformation.alarmChangedTime~~Comment.commentTime of the last Comment added	
systemDN	C,F	IRPAgent.systemDN	
notificationType	M,F	"notifyComments"	
alarmType	M,F	AlarmInformation.eventType	
probableCause	M,F	AlarmInformation.probableCause	
perceived Severity	M,F	AlarmInformation.perceivedSeverity	
comments	M	The set of Comment instances involved in a relationship with this AlarmInformation.	
alarmId	M	AlarmInformation.alarmId	

…

End of Change in Clause 6.10.1

CR-Form-v7

CHANGE REQUEST

⌘ 32.111-2 CR 034 ⌘rev - ⌘ Current version: 6.2.0 ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network X Core Network X

Title:	⌘	Correct the time carried by the eventTime parameter of notifyComments
Source:	⌘	**SA5** (olaf.pollakowski@siemens.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	01/10/2004
Category:	⌘	**A**
Release:	⌘	Rel-6

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)
Rel-6 (Release 6)

Reason for change:	⌘	The eventTime parameter of notifyComments currently carries the alarmChangedTime. However, the eventTime shall carry the time of the event triggering the emission of the related notification which is the Comment.commentTime of the last Comment added in this case.
Summary of change:	⌘	The time stamp carried by the eventTime parameter of notifyComments is changed from alarmChangedTime to Comment.commentTime of the last Comment added.
Consequences if not approved:	⌘	The eventTime carries the wrong time stamp.

Clauses affected:	⌘	6.10.1

Other specs affected: ⌘	Y	N	
		X	Other core specifications ⌘
		X	Test specifications
		X	O&M Specifications

Other comments: ⌘ Rel-6 Mirror CR to S5-046987.

Change in Clause 6.10.1

6.10.1 notifyComments (O)

6.10.1.1 Definition

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

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

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

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

6.10.1.2 Input Parameters

Parameter Name	Qualifier	Matching Information	Comment
objectClass	M,F	MonitoredEntity.objectClass where the MonitoredEntity is identified by the relation-AlarmedObject-AlarmInformation of the AlarmInformation.	
objectInstance	M,F	MonitoredEntity.objectInstance where the MonitoredEntity is identified by the relation-AlarmedObject-AlarmInformation of the AlarmInformation.	
notificationId	M	This carries the semantics of notification identifier.	
eventTime	M,F	~~AlarmInformation.alarmChangedTime~~Comment.commentTime of the last Comment added	
systemDN	C,F	IRPAgent.systemDN	
notificationType	M,F	"notifyComments"	
alarmType	M,F	AlarmInformation.eventType	
probableCause	M,F	AlarmInformation.probableCause	
perceived Severity	M,F	AlarmInformation.perceivedSeverity	
comments	M	The set of Comment instances involved in a relationship with this AlarmInformation.	
alarmId	M	AlarmInformation.alarmId	

…

End of Change in Clause 6.10.1

CR-Form-v7

CHANGE REQUEST

⌘ 32.111-2 CR 035 ⌘rev - ⌘ Current version: 6.2.0 ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network X Core Network X

Title:	⌘	Remove redundant ackTime parameter in notifyAckStateChanged
Source:	⌘	**SA5** (olaf.pollakowski@siemens.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	01/10/2004
Category:	⌘	**C**
Release:	⌘	Rel-6

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)
Rel-6 (Release 6)

Reason for change:	⌘	The ackTime is carried twice in notifyAckStateChanged, once in the eventTime and once in a dedicated parameter. This is leading to unnecessary load on the Itf-N.
Summary of change:	⌘	The dedicated ackTime parameter is removed so that the ackTime is carried only once in eventTime.
Consequences if not approved:	⌘	

Clauses affected:	⌘	6.8.2

	Y	N			
Other specs affected:		X	Other core specifications	⌘	
		X	Test specifications		
	X		O&M Specifications		Rel-6 32.111-3, 32.111-4

Other comments:	⌘	Rel-6 Child CR 32.111-3 in S5-046979. Rel-6 Child CR 32.111-4 in S5-046980.

Change in Clause 6.8.2

6.8.2 notifyAckStateChanged (M)

6.8.2.1 Definition

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

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

The IRPManager and the IRPAgent can acknowledge and unacknowledge alarms as defined by 3GPP TS 32.111-1 [9].

6.8.2.2 Input Parameters

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

End of Change in Clause 6.8.2

CR-Form-v7

CHANGE REQUEST

⌘	32.111-2	CR	036	⌘	rev	-	⌘	Current version:	6.2.0	⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network X Core Network X

Title:	⌘	Correct description of thresholdInformation – Align with 32.401
Source:	⌘	**SA5** (edwin.tse@ericsson.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	19/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

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)*
Rel-6	*(Release 6)*

Reason for change:	⌘	The current description of thresholdInformation is wrong.
Summary of change:	⌘	Correct the current description of thresholdInformation and make a reference to that defined in TS 32.401.
Consequences if not approved:	⌘	The description of thresholdInformation is wrong and will lead to interoperability problem between IRPManager and IRPAgent.

Clauses affected:	⌘	2, 5.5.1

Other specs affected: ⌘

Y	N	
	X	Other core specifications
	X	Test specifications
	X	O&M Specifications

⌘

Other comments: ⌘

Change in Clause "2"

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 32.401 "Telecommunication management; Performance Management (PM); Concept and requirements".~~ITU-T Recommendation Q.821: "Stage 2 and Stage 3 description for the Q3 interface - Alarm surveillance". Not used in the body text.~~

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

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

[4] Void.

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

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

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

[8] Void.~~3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects". Not used in the body text.~~

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

[10] 3GPP TS 32.622: "Telecommunication management; Configuration Management (CM); Generic network resources Integration Reference Point (IRP): Network Resource Model (NRM)".

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

[12] Void.~~ITU-T Recommendation X.720: "Information technology - Open Systems Interconnection - Structure of management information: Management information model". Not used in the body text.~~

[13] Void.

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

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

End of change in Clause "2"

Change in Clause “5.5.1”

5.5.1 Definition and legal values

Name	Definition	Legal Values
alarmId	It identifies one AlarmInformation in the AlarmList.	
notificationId	It identifies the notification that carries the AlarmInformation.	
alarmRaisedTime	It indicates the date and time when the alarm is first raised by the alarmed resource.	All values indicating valid time.
alarmChangedTime	It indicates the last date and time when the AlarmInformation is changed by the alarmed resource. Changes to AlarmInformation caused by invocations of the IRPManager would not change this date and time.	All values indicating valid time.
alarmClearedTime	It indicates the date and time when the alarm is Cleared.	All values indicating valid time.
eventType	It indicates the type of event. See Annex A for information on event type.	See Annex A.
probableCause	It qualifies alarm and provides further information than eventType. See Annex B for a complete listing.	See Annex B.
perceivedSeverity	It indicates the relative level of urgency for operator attention.	Critical, Major, Minor, Warning, Indeterminate, Cleared: see ITU-T Recommendation X.733 [2]. This IRP does not recommend the use of indeterminate.
specificProblem	It provides further qualification on the alarm than probableCause. This attribute value shall be single-value and of simple type such as integer or string. See definition in ITU-T Recommendation X.733 [2] clause 8.1.2.2.	Provided by vendor.
backedUpStatus	It indicates if an object (the MonitoredEntity) has a back up. See definition in ITU-T Recommendation X.733 [2] clause 8.1.2.4.	All values that carry the semantics of backedUpStatus defined by ITU-T X.733 [2] clause 8.1.2.4.
trendIndication	It indicates if some observed condition is getting better, worse, or not changing.	"Less severe", "no change", "more severe": see definition in ITU-T Recommendation X.733 [2] clause 8.1.2.6.
thresholdInfo	It indicates the crossed threshold information such as: • The identifier of the monitored attribute whose value has crossed a threshold, • The threshold settings, • The observed value that have crossed a threshold, etc. See definition in ITU-T Recommendation X.733 [2] clause 8.1.2.7. See also for information in TS 32.401 [1] subclause 5.6.~~It indicates the direction of threshold crossing.~~	~~See definitions in ITU-T Recommendation X.733 [2] clause 8.1.2.7.~~
stateChangeDefinition	It indicates MO attribute value changes. See definition in ITU-T Recommendation X.733 [2] clause 8.1.2.10.	
monitoredAttributes	It indicates MO attributes whose value changes are being monitored. See definition in ITU-T Recommendation X.733 [2] clause 8.1.2.11.	
proposedRepairActions	It indicates proposed repair actions. See definition in ITU-T Recommendation X.733 [2] clause 8.1.2.12.	
additionalText	It carries semantics that is outside the scope of this IRP specification. It may provide the identity of the NE (e.g. RNC, Node-B) from which the alarm has been originated. It corresponds to the "user label" attribute of the object class representing the NE in the Generic Network Resource Model [10]. It can contain further information on the alarm.	N/A
additionalInformation	It contains information on the alarm and its semantics is outside the scope of this IRP.	N/A
ackTime	It identifies the time when the alarm has been acknowledged or unacknowledged the last time.	All values that indicate valid time that are later than that carried in alarmRaisedTime.

Name	Definition	Legal Values
ackUserId	It identifies the last user who has changed the Acknowledgement State.	It can be used to identify the human operator such as "John Smith" or it can identify a group, such as "Team Six", or it can contain no information such as "".
ackSystemId	It identifies the system (EM or NM) from which the alarm has been acknowledged or unacknowledged the last time.	It can be used to identify the system, such as "system 6" or it can contain no information such as "".
ackState	It identifies the Acknowledgement State of the alarm.	Acknowledged: the alarm has been acknowledged. Unacknowledged: the alarm has been unacknowledged or the alarm has never been acknowledged.
commentTime	It carries the time when the comment has been added to the alarm.	
commentText	It carries the textual comment.	
commentUserId	It carries the identification of the user who made the comment.	
commentSystemId	It carries the identification of the system (EM or NM) from which the comment is made. That system supports the user that made the comment.	
source	It identifies one MonitoredEntity.	All values that carry the semantics of DN.
notificationIdSet	It carries one or more notification identifiers.	
clearUserId	It carries the identity of the user who invokes the clearAlarms operation.	It can be used to identify the human operator such as "John Smith" or it can identify a group, such as "Team Six", or it can contain no information such as "".
clearSystemId	It carries the identity of the system in which the IRPManager runs. That IRPManager supports the user who invokes the clearAlarms().	It can be used to identify the system, such as "system 6" or it can contain no information such as "".
serviceUser	It identifies the service-user whose request for service provided by the serviceProvider led to the generation of the security alarm.	This attribute may carry no information if the server user is not identifiable.
serviceProvider	It identifies the service-provider whose service is requested by the serviceUser and the service request provokes the generation of the security alarm.	
securityAlarmDetector	It carries the identity of the detector of the security alarm.	This attribute may carry no information if the security alarm detector is not identifiable.

End of change in Clause “5.5.1”

CR-Form-v7

CHANGE REQUEST

⌘ **32.111-2** **CR** **037** ⌘ **rev** **-** ⌘ Current version: **6.2.0** ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network **X** Core Network **X**

Title:	⌘	Correction of probable cause for alarms.
Source:	⌘	**SA5** (llrui@bupt.edu.cn;liyewen@chinamobile.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	19/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

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)*
Rel-6	*(Release 6)*

Reason for change:	⌘	Cross check the probable cause and correct those with spelling error according to the specifications as reference (e.g. ITU-T M.3100, ITU-T X.721).
Summary of change:	⌘	Correction of probable cause for alarms.
Consequences if not approved:	⌘	Incorrect probable cause will lead to bad implementation.

Clauses affected:	⌘	Annex B			
Other specs affected:	⌘	**Y**	**N**		
			X	Other core specifications	⌘
			X	Test specifications	
		X		O&M Specifications	32.111-3, 32.111-4
Other comments:	⌘				

Change in Annex B

Annex B (normative): Probable Causes

This annex lists probable causes and their corresponding event types.

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

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

M.3100 Probable cause	Event type
Indeterminate	Unknown
Alarm Indication Signal (AIS)	Communications
Call Setup Failure	Communications
Degraded Signal	Communications
Far End Receiver Failure (FERF)	Communications
Framing Error	Communications
Loss Of Frame (LOF)	Communications
Loss Of Pointer (LOP)	Communications
Loss Of Signal (LOS)	Communications
Payload Type Mismatch	Communications
Transmission Error	Communications
Remote Alarm Interface	Communications
Excessive Bit Error Rate (EBER)	Communications
Path Trace Mismatch	Communications
Unavailable	Communications
Signal Label Mismatch	Communications
Loss Of Multi Frame	Communications
Communications Receive Failure	Communications
Communications Transmit Failure	Communications
Modulation Failure	Communications
Demodulation Failure	Communications
broadcastChannelFailure	Communications
connectionEstablishmentError	Communications
invalidMessageReceived	Communications
localNodeTransmissionError	Communications
remoteNodeTransmissionError	Communications
routingFailure	Communications
Back Plane Failure	Equipment
Data Set Problem	Equipment
Equipment Identifier Duplication	Equipment
External IF Device Problem	Equipment
Line Card Problem	Equipment
Multiplexer Problem	Equipment
NE Identifier Duplication	Equipment
Power Problem	Equipment
Processor Problem	Equipment
Protection Path Failure	Equipment
Receiver Failure	Equipment
Replaceable Unit Missing	Equipment
Replaceable Unit Type Mismatch	Equipment
Synchronization Source Mismatch	Equipment

Terminal Problem	Equipment
Timing Problem	Equipment
Transmitter Failure	Equipment
Trunk Card Problem	Equipment
Replaceable Unit Problem	Equipment
Real Time Clock Failure	Equipment
antennaFailure	Equipment
batteryChargingFailure	Equipment
diskFailure	Equipment
frequencyHoppingFailure	Equipment
iODeviceError	Equipment
lossOfSynchronisation	Equipment
lossOfRedundancy	Equipment
powerSupplyFailure	Equipment
signalQualityEvaluationFailure	Equipment
tranceiverFailure	Equipment
Protection Mechanism Failure	Equipment
Protection Resource Failure	Equipment
Air Compressor Failure	Environmental
Air Conditioning Failure	Environmental
Air Dryer Failure	Environmental
Battery Discharging	Environmental
Battery Failure	Environmental
Commercial Power Failure	Environmental
Cooling Fan Failure	Environmental
Engine Failure	Environmental
Fire Detector Failure	Environmental
Fuse Failure	Environmental
Generator Failure	Environmental
Low Battery Threshold	Environmental
Pump Failure	Environmental
Rectifier Failure	Environmental
Rectifier High Voltage	Environmental
Rectifier Low F Voltage	Environmental
Ventilation System Failure	Environmental
Enclosure Door Open	Environmental
Explosive Gas	Environmental
Fire	Environmental
Flood	Environmental
High Humidity	Environmental
High Temperature	Environmental
High Wind	Environmental
Ice Build Up	Environmental
Intrusion Detection	Environmental
Low Fuel	Environmental
Low Humidity	Environmental
Low Cable Pressure	Environmental
Low Temperature	Environmental
Low Water	Environmental
Smoke	Environmental
Toxic Gas	Environmental
coolingSystemFailure	Environmental
externalEquipmentFailure	Environmental
ExternalPointFailure	Environmental
Storage Capacity Problem	Processing error
Memory Mismatch	Processing error
Corrupt Data	Processing error
Out Of CPU Cycles	Processing error
Software Environment Problem	Processing error
Software Download Failure	Processing error
Loss Of Real Time	Processing error
Reinitialized	Processing error
applicationSubsystemFailure	Processing error

configurationOrCustomisation Error	Processing error
databaseInconsistency	Processing error
fileError	Processing error
outOfMemory	Processing error
softwareError	Processing error
timeoutExpired	Processing error
underlayingResourceUnavailable	Processing error
versionMismatch	Processing error
bandwidthReduced	Quality of service
congestion	Quality of service
Excessive Error Rate	Quality of service
excessiveResponseTime	Quality of service
excessiveRetransmissionRate	Quality of service
reducedLoggingCapability	Quality of service
systemResourcesOverload	Quality of service

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

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

X.721/X.733/X.736 Probable Cause	Event type
Out of Memory	Processing error
Out of Service	Operational Violation
Output Device Error	Equipment
Performance Degraded	Quality of service
Power Problem	Equipment
Pressure Unacceptable	Environmental
Procedural Error	Operational Violation
Processor Problem	Equipment
Pump Failure	Environmental
Queue Size Exceeded	Quality of service
Receive Failure	Equipment
Receiver Failure	Equipment
Remote Node Transmission Error	Communications
Resource at or Nearing Capacity	Quality of service
Response Time Excessive	Quality of service
Re-transmission Rate Excessive	Quality of service
Software Error	Processing error
Software Program Abnormally Terminated	Processing error
Software Program Error	Processing error
Storage Capacity Problem	Processing error
Temperature Unacceptable	Environmental
Threshold Crossed	Quality of service
Timing Problem	Equipment
Toxic Leak Detected	Environmental
Transmit Failure	Equipment
Transmitter Failure	Equipment
Unauthoriz~~s~~ed Access Attempt	Security Service or Mechanism Violation
Underlying Resource Unavailable	Processing error
Unexpected Information	Integrity Violation
Unspecified Reason	Operational Violation
Unspecified Reason	Physical Violation
Unspecified Reason	Security Service or Mechanism Violation
Version Mismatch	Processing error

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

2G & 3G Wireless Systems	Event Type
A-bis to BTS interface failure	Equipment
A-bis to TRX interface failure	Equipment
Antenna problem	Equipment
Battery breakdown	Equipment
Battery charging fault	Equipment
Clock synchronization problem	Equipment
Combiner problem	Equipment
Disk problem	Equipment
Equipment failure	Equipment
Excessive receiver temperature	Equipment
Excessive transmitter output power	Equipment
Excessive transmitter temperature	Equipment
Frequency hopping degraded	Equipment
Frequency hopping failure	Equipment
Frequency redefinition failed	Equipment
Line interface failure	Equipment
Link failure	Equipment
Loss of synchronization	Equipment
Lost redundancy	Equipment
Mains breakdown with battery back-up	Equipment
Mains breakdown without battery back-up	Equipment
Power supply failure	Equipment
Receiver antenna fault	Equipment
Receiver Failure	Equipment
Receiver multicoupler failure	Equipment
Reduced transmitter output power	Equipment
Signal quality evaluation fault	Equipment
Timeslot hardware failure	Equipment
Transceiver problem	Equipment
Transcoder problem	Equipment
Transcoder or rate adapter problem	Equipment
Transmitter antenna failure	Equipment
Transmitter antenna not adjusted	Equipment
Transmitter failure	Equipment
Transmitter low voltage or current	Equipment
Transmitter off frequency	Equipment
Database inconsistency	Processing error
File system call unsuccessful	Processing error
Input parameter out of range	Processing error
Invalid parameter	Processing error
Invalid pointer	Processing error
Message not expected	Processing error
Message not initialized	Processing error
Message out of sequence	Processing error
System call unsuccessful	Processing error
Timeout expired	Processing error
Variable out of range	Processing error
Watch dog timer expired	Processing error
Cooling system failure	Environmental
External equipment failure	Environmental
External power supply failure	Environmental
External transmission device failure	Environmental
Fan failure	Environmental
High humidity	Environmental
High temperature	Environmental
Intrusion detected	Environmental
Low humidity	Environmental
Low temperature	Environmental
Smoke detected	Environmental
Excessive Error Rate	Quality of service
Reduced alarm reporting	Quality of service
Reduced event reporting	Quality of service

2G & 3G Wireless Systems	Event Type
Reduced logging capability	Quality of service
System resources overload	Quality of service
Broadcast channel failure	Communications
Connection establishment error	Communications
Invalid message received	Communications
Invalid MSU received	Communications
LAPD link protocol failure	Communications
Local alarm indication	Communications
Remote alarm indication	Communications
Routing failure	Communications
SS7 protocol failure	Communications
Transmission error	Communications

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

Table B.4: Duplicated Probable Causes

Duplicated Probable Cause	2G & 3G	X.721 X.733	X.736	M.3100	Event Type
Broadcast Channel Failure	X			X	Communications
Call Establishment Failure (X.721/X.733) Call Setup Failure (M.3100)		X		X	Communications
Connection Establishment Error	X			X	Communications
Degraded Signal		X		X	Communications
Framing Error		X		X	Communications
Invalid Message Received	X			X	Communications
Local Node Transmission Error		X		X	Communications
Loss of Frame		X		X	Communications
Loss of Signal		X		X	Communications
Remote Node Transmission Error		X		X	Communications
Routing Failure	X			X	Communications
Transmission Error	X			X	Communications
Antenna Failure (M.3100) Antenna Problem (2G & 3G)	X			X	Equipment
Battery Charging Failure (M.3100) Battery Charging Fault (2G & 3G)	X			X	Equipment
Disk Failure (M.3100) Disk Problem (2G & 3G)	X			X	Equipment
Equipment Failure (2G & 3G) Equipment Malfunction (X.721/X.733)	X	X			Equipment
Frequency Hopping Failure	X			X	Equipment
IO Device Error (M.3100) Input/Output Device Error (X.721/X.733)		X		X	Equipment
Loss Of Redundancy (M.3100) Lost Redundancy (2G & 3G)	X			X	Equipment
Loss Of Synchronization	X			X	Equipment
Multiplexer Problem		X		X	Equipment
Power Problem		X		X	Equipment
Power Supply Failure	X			X	Equipment
Processor Problem		X		X	Equipment
Receiver Failure	X	X		X	Equipment
Signal Quality Evaluation Failure (M.3100) Signal Quality Evaluation Fault (2G & 3G)	X			X	Equipment
Timing Problem		X		X	Equipment
Transceiver Failure (M.3100) Transceiver Problem (2G & 3G)	X			X	Equipment
Transmitter Failure	X	X		X	Equipment
Cooling System Failure	X			X	Environmental
External Equipment Failure	X			X	Environmental
Enclosure Door Open		X		X	Environmental
Fan Failure (2G & 3G) Cooling Fan Failure (M.3100)	X			X	Environmental
Fire Detected (X.721/X.733) Fire (M.3100)		X		X	Environmental
Flood Detected (X.721/X.733) Flood (M.3100)		X		X	Environmental
High Humidity	X			X	Environmental
High Temperature	X			X	Environmental
Intrusion Detected (2G & 3G) Intrusion Detection (X.736/M.3100)	X		X	X	Environmental (2G & 3G); Physical Violation (X.736/M.3100)
Low Humidity	X			X	Environmental
Low Temperature	X			X	Environmental
Pump Failure		X		X	Environmental
Smoke Detected (2G & 3G) Smoke (M.3100)	X			X	Environmental
Application Subsystem Failure		X		X	Processing Error
Bandwidth Reduced (M.3100) Bandwidth Reduction (X.721/X.733)		X		X	Processing Error

Duplicated Probable Cause	2G & 3G	X.721 X.733	X.736	M.3100	Event Type
Configuration or Customization Error (M.3100) Configuration or Customizing Error (X.721/X.733)		X		X	Processing Error
Database Inconsistency	X			X	Processing Error
File Error		X		X	Processing Error
Storage Capacity Problem		X		X	Processing Error
Excessive Bit Error Rate (M.3100) Excessive Error Rate (2G & 3G)	X			X	Processing Error
Corrupt Data		X		X	Processing Error
Out Of Memory		X		X	Processing Error
Software Error		X		X	Processing Error
Timeout Expired	X			X	Processing Error
Underlaying Resource Unavailable (M.3100) Underlying Resource Unavailable (X.721/X.733)		X		X	Processing Error
Version Mismatch		X		X	Processing Error
Congestion		X		X	Quality of Service
Reduced Logging Capability	X			X	Quality of Service
System Resources Overload	X			X	Quality of Service
Excessive Response Time (M.3100) Response Time Excessive (X.721/X.733)		X		X	Quality of Service
Excessive Retransmission Rate (M.3100) Re-Transmission Rate Excessive (X.721/X,733)		X		X	Quality of Service

End of Change in Annex B
End of Document

CR-Form-v7

CHANGE REQUEST

⌘ 32.111-3 CR 034 ⌘rev - ⌘ Current version: 6.0.0 ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network X Core Network X

Title:	⌘	Remove redundant ackTime parameter in notifyAckStateChanged
Source:	⌘	**SA5** (olaf.pollakowski@siemens.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	01/10/2004
Category:	⌘	**C**
Release:	⌘	Rel-6

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)
Rel-6 (Release 6)

Reason for change:	⌘	The ackTime is carried twice in notifyAckStateChanged, once in the eventTime and once in a dedicated parameter. This is leading to unnecessary load on the Itf-N.
Summary of change:	⌘	The dedicated ackTime parameter is removed so that the ackTime is carried only once in eventTime.
Consequences if not approved:	⌘	

Clauses affected:	⌘	1, 5.3

	Y	N		
Other specs affected: ⌘		X	Other core specifications	⌘
		X	Test specifications	
		X	O&M Specifications	

Other comments:	⌘	Child to S5-046978 CR 32.111-2

Change in Clause 1

1 Scope

The present document specifies the CORBA Solution Set (SS) for the IRP whose semantics is specified in Alarm IRP: Information Service (IS) (TS 32.111-2 [6]).

Clause 1 to 3 provides background information. Clause 4 provides key architectural features supporting the SS. Clause 5 defines the mapping of operations, notification, parameters and attributes defined in IS to their SS equivalents. Clause 6 describes the notification interface containing the push method. Annex A contains the IDL specification.

This Solution Set specification is related to TS 32.111-2 V6.3~~0~~.X.

End of Change in Clause 1

Change in Clause 5.3

5.3 Notification parameter mapping

Reference 3G TS 32.111-2 [6] 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 [1]) equivalents. The composition of OMG Structured Event, as defined in the OMG Notification Service [1], 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 [6] defined notification parameters.

…

Table 13: 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.
notification Id	One NV pair of filterable_ body_fields	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 filterable_ body_fields	M	See that of notifyNewAlarm.
~~ackTime~~	~~One NV pair of filterable_ body_fields~~	~~M~~	~~Name of NV pair is the ACK_TIME of interface AttributeNameValue of module AlarmIRPConstDefs.~~ ~~Value of NV pair is a IRPTime of module ManagedGenericIRPConstDefs.~~
ackUserId	One NV pair of filterable_ body_fields	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 filterable_ body_fields	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 filterable_body_fields	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.
There is no corresponding IS attribute.	remaining_ body		

End of Change in Clause 5.3

CR-Form-v7

CHANGE REQUEST

⌘ 32.111-3 CR 035 ⌘rev - ⌘ Current version: 6.0.0 ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network X Core Network X

Title:	⌘	Correction of probable cause definition for AlarmIRP IDL file.
Source:	⌘	**SA5** (llrui@bupt.edu.cn;liyewen@chinamobile.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	19/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

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)*
Rel-6	*(Release 6)*

Reason for change:	⌘	Cross check the probable cause and correct those with spelling error to keep consistent with AlarmIRP IS.
Summary of change:	⌘	Correction of probable cause definition for AlarmIRP IDL file.
Consequences if not approved:	⌘	Incorrect probable cause will lead to bad implementation.

Clauses affected:	⌘	Annex A.1

	Y	N		
Other specs affected: ⌘		X	Other core specifications	⌘
		X	Test specifications	
		X	O&M Specifications	

Other comments: ⌘

Change in Annex A.1

A.1 IDL specification (file name "AlarmIRPConstDefs.idl")

```
#ifndef AlarmIRPConstDefs_idl
#define AlarmIRPConstDefs_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;

   /* DNTypeOpt is an optional type.
   If the discriminator is true the value is present.
   Otherwise the value is null.
   */
   union DNTypeOpt 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";
};

/*
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_M3100 = 3;
   const short FAR_END_RECEIVER_FAILURE = 4;
   const short FRAMING_ERROR_M3100 = 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;
   const short TRANSMISSION_ERROR = 10;
   // 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_M3100 = 56;
        const short NE_IDENTIFIER_DUPLICATION = 57;
        const short POWER_PROBLEM_M3100 = 58;
        const short PROCESSOR_PROBLEM_M3100 = 59;
        const short PROTECTION_PATH_FAILURE = 60;
        const short RECEIVER_FAILURE_M3100 = 61;
        const short REPLACEABLE_UNIT_MISSING = 62;
        const short REPLACEABLE_UNIT_TYPE_MISMATCH = 63;
        const short SYNCHRONISZATION_SOURCE_MISMATCH = 64;
        const short TERMINAL_PROBLEM = 65;
        const short TIMING_PROBLEM_M3100 = 66;
        const short TRANSMITTER_FAILURE_M3100 = 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 COMMERICALCOMMERCIAL_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_M3100 = 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_M3100 = 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_M3100 = 151;
        const short MEMORY_MISMATCH = 152;
        const short CORRUPT_DATA_M3100 = 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_REDUCEDTION = 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_CUSTOMIZATIONING_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_DETECTEDION = 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_SYNCHRONISZATION_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_SYNCHRONISZATION = 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_INITIALISZED = 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 CALLCONNECTION_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_FAILURE 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
   };

   /*
   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";
};

/*
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 BackedUpStatusType;

/*
It indicates if the threshold crossed was in the up or down direction.
*/
enum ThresholdIndicationType {Up, Down};

/*
It indicates if the AlarmList alignment is required.
*/
enum AlarmListAlignmentRequirementType {Required, NotRequired};

/* FloatTypeOpt is an optional type.
If the discriminator is true the value is present.
Otherwise the value is null.
*/
union FloatTypeOpt switch (boolean)
{
   case TRUE: float value;
};

/* ThresholdLevelIndType 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 ThresholdLevelIndType
{
   ThresholdIndicationType indication;
   FloatTypeOpt low;
   float high;
};

/* ThresholdLevelIndTypeOpt is an optional type.
If the discriminator is true the value is present.
Otherwise, the value is null.
*/
union ThresholdLevelIndTypeOpt switch (boolean)
{
   case TRUE: ThresholdLevelIndType value;
};

/* ThresholdInfoType indicates some gauge or counter
attribute passed a set threshold.
@member attributeID: identifies the attribute that
  crossed the threshold.
@member observedValue: attributes that are of type
  integer will be converted to floats.
@member thresholdlevel: This parameter is for
  multi-level thresholds. Optional.
@member armTime: May contain empty string.
*/
struct ThresholdInfoType
{
   string attributeID;
   float observedValue;
   ThresholdLevelIndTypeOpt thresholdLevel;
   string armTime;
};

/*
It indicates if some observed condition is getting better, worse,
or not changing.
*/
enum TrendIndicationType {LessSevere, NoChange, MoreSevere};

/*
It is used to report a changed attribute value.
*/
struct AttributeValueChangeType
{
   string attribute_name;
   any    old_value;  // type depends on attribute
   any    new_value;  // type depends on attribute
};

typedef sequence <AttributeValueChangeType> AttributeChangeSetType;

/*
It is used to report an attribute and its value.
*/
struct AttributeValueType
{
   string attribute_name;
```

```
   any    value;  // type depends on the attribute
};

typedef sequence <AttributeValueType> AttributeSetType;

typedef sequence <long> NotifIdSetType;

/*
This holds identifiers of notifications that are correlated.
*/
struct CorelatedNotification
{
   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.
   NotifIdSetType notif_id_set;  // Set of related notification ids
};

/*
Correlated Notification sets are sets of Correlated Notification
structures.
*/
typedef sequence <CorelatedNotification> CorrelatedNotificationSetType;

/*
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_information_reference;
   ManagedGenericIRPConstDefs::ShortTypeOpt 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
{
   UnknownAlarmId,
   WrongPerceivedSeverity,
   AcknowledgmentFailed
};

/*
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_information_reference;
      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_information_reference;
      AcknowledgeFailureCategories failure_category;
      string reason;
   };

   typedef sequence <BadAlarmInformationId> BadAlarmInformationIdSeq;
   typedef sequence <BadAcknowledgeAlarmInfo> BadAcknowledgeAlarmInfoSeq;
   typedef sequence <string> AlarmInformationIdSeq;
   typedef CosNotification::EventBatch AlarmInformationSeq;
};
#endif
```

End of Change in Annex A.1
End of Document

3GPP TSG-SA5 (Telecom Management) **S5-047118**
Meeting #40, Sanya, CHINA, 15 - 19 November 2004

CR-Form-v7

CHANGE REQUEST

⌘ **32.111-3** **CR** **036** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network **X** Core Network **X**

Title:	⌘	Add mandatory exception operationNotSupported for optional operations in AlarmIRP - Align IDL style with IDL Style Guide in 32.150
Source:	⌘	**SA5 (XiongKangjian@zte.com.cn, Huangsq@zte.com.cn)**
Work item code:	⌘	OAM-NIM
Date:	⌘	19/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

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)*
Rel-6	*(Release 6)*

Reason for change:	⌘	Alarm IRP can not throw the standard-defined exception when it does not support the optional operations. This non-standard behaviour can confuse IRPManager.
Summary of change:	⌘	Add the mandatory exception operationNotSupported for optional operations in AlarmCMIRP. Align IDL style with IDL Style Guide.
Consequences if not approved:	⌘	The IDL style would not be aligned with the style guide and other CORBA SSs.

Clauses affected:	⌘	Annex A

	Y	N		
Other specs affected: ⌘		X	Other core specifications	⌘
		X	Test specifications	
		X	O&M Specifications	

Other comments: ⌘

Annex A (normative): IDL specifications

A.1 IDL specification (file name "AlarmIRPConstDefs.idl")

```
//File: AlarmIRPConstDefs.idl

#ifndef _ALARMIRPCONSTDEFS_IDL_AlarmIRPConstDefs_idl
#define _ALARMIRPCONSTDEFS_IDL_AlarmIRPConstDefs_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;

   /* DNTypeOpt is an optional type.
   If the discriminator is true the value is present.
   Otherwise the value is null.
   */
   union DNTypeOpt 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";
};

/*
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_M3100 = 3;
   const short FAR_END_RECEIVER_FAILURE = 4;
   const short FRAMING_ERROR_M3100 = 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;
   const short TRANSMISSION_ERROR = 10;
   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_DEVICE_PROBLEM = 54;
const short LINE_CARD_PROBLEM = 55;
const short MULTIPLEXER_PROBLEM_M3100 = 56;
const short NE_IDENTIFIER_DUPLICATION = 57;
const short POWER_PROBLEM_M3100 = 58;
const short PROCESSOR_PROBLEM_M3100 = 59;
const short PROTECTION_PATH_FAILURE = 60;
const short RECEIVER_FAILURE_M3100 = 61;
const short REPLACEABLE_UNIT_MISSING = 62;
const short REPLACEABLE_UNIT_TYPE_MISMATCH = 63;
const short SYNCHRONISATION_SOURCE_MISMATCH = 64;
const short TERMINAL_PROBLEM = 65;
const short TIMING_PROBLEM_M3100 = 66;
const short TRANSMITTER_FAILURE_M3100 = 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 COMMERICAL_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_M3100 = 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_M3100 = 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_M3100 = 151;
const short MEMORY_MISMATCH = 152;
const short CORRUPT_DATA_M3100 = 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_REDUCTION = 303;
// Value 304 corresponds to a duplicated probable cause
const short COMMUNICATION_PROTOCOL_ERROR = 305;
const short COMMUNICATION_SUBSYSTEM_FAILURE = 306;
const short CONFIGURATION_OR_CUSTOMIZING_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_DETECTION = 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_SYNCHRONISATION_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_SYNCHRONISATION = 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_INITIALISED = 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 CALL_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_FAILURE = 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
};

/*
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";
   };

   /*
   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 BackedUpStatusType;

   /*
   It indicates if the threshold crossed was in the up or down direction.
   */
   enum ThresholdIndicationType {UP, DOWNUp, Down};

   /*
   It indicates if the AlarmList alignment is required.
   */
   enum AlarmListAlignmentRequirementType {REQUIRED, NOTREQUIREDRequired,
NotRequired};

   /* FloatTypeOpt is an optional type.
   If the discriminator is true the value is present.
   Otherwise the value is null.
   */
   union FloatTypeOpt switch (boolean)
   {
      case TRUE: float value;
   };
```

```
/* ThresholdLevelIndType 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 ThresholdLevelIndType
{
   ThresholdIndicationType indication;
   FloatTypeOpt low;
   float high;
};

/* ThresholdLevelIndTypeOpt is an optional type.
If the discriminator is true the value is present.
Otherwise, the value is null.
*/
union ThresholdLevelIndTypeOpt switch (boolean)
{
   case TRUE: ThresholdLevelIndType value;
};

/* ThresholdInfoType indicates some gauge or counter
attribute passed a set threshold.
@member attributeID: identifies the attribute that
  crossed the threshold.
@member observedValue: attributes that are of type
  integer will be converted to floats.
@member thresholdlevel: This parameter is for
  multi-level thresholds. Optional.
@member armTime: May contain empty string.
*/
struct ThresholdInfoType
{
   string attributeID;
   float observedValue;
   ThresholdLevelIndTypeOpt thresholdLevel;
   string armTime;
};

/*
It indicates if some observed condition is getting better, worse,
or not changing.
*/
enum TrendIndicationType {LESSSEVERE, NOCHANGE, MORESEVERELessSevere,
NoChange, MoreSevere};

/*
It is used to report a changed attribute value.
*/
struct AttributeValueChangeType
{
   string attribute_name;
   any    old_value;  // type depends on attribute
   any    new_value;  // type depends on attribute
};

typedef sequence <AttributeValueChangeType> AttributeChangeSetType;

/*
```

```
It is used to report an attribute and its value.
*/
struct AttributeValueType
{
   string attribute_name;
   any    value;  // type depends on the attribute
};

typedef sequence <AttributeValueType> AttributeSetType;

typedef sequence <long> NotifIdSetType;

/*
This holds identifiers of notifications that are correlated.
*/
struct CorelatedNotification
{
   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.
   NotifIdSetType notif_id_set;  // Set of related notification ids
};

/*
Correlated Notification sets are sets of Correlated Notification
structures.
*/
typedef sequence <CorelatedNotification> CorrelatedNotificationSetType;

/*
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_information_reference;
   ManagedGenericIRPConstDefs::ShortTypeOpt 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
{
   UNKNOWNALARMID,
   WRONGPERCEIVEDSEVERITY,
   ACKNOWLEDGMENTFAILED
   UnknownAlarmId,
   WrongPerceivedSeverity,
```

```
        AcknowledgmentFailed
    };

    /*
    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_information_reference;
       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_information_reference;
       AcknowledgeFailureCategories failure_category;
       string reason;
    };

    typedef sequence <BadAlarmInformationId> BadAlarmInformationIdSeq;
    typedef sequence <BadAcknowledgeAlarmInfo> BadAcknowledgeAlarmInfoSeq;
    typedef sequence <string> AlarmInformationIdSeq;
    typedef CosNotification::EventBatch AlarmInformationSeq;
};
#endif _ALARMIRPCONSTDEFS_IDL_
```

A.2 IDL specification (file name “AlarmIRPSystem.idl”)

```
//File: AlarmIRPSystem.idl

#ifndef _ALARMIRPSYSTEM_IDL_AlarmIRPSystem_idl
#define _ALARMIRPSYSTEM_IDL_AlarmIRPSystem_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; };

   /*
   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_alarmInformations (
         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::StringTypeOpt 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::StringTypeOpt 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::StringTypeOpt 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::StringTypeOpt clear_system_id,
   out AlarmIRPConstDefs::BadAlarmInformationIdSeq
      bad_alarm_information_id_list
)
raises (ClearAlarms, 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::StringTypeOpt filter,
   in AlarmIRPConstDefs::DNTypeOpt base_object,
   out boolean flag,
   out AlarmInformationIterator iter
)
raises (GetAlarmList, ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
This method returns the count of Alarm Informations.
*/
void get_alarm_count (
   in ManagedGenericIRPConstDefs::StringTypeOpt 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, ManagedGenericIRPSystem::OperationNotSupported,
              ManagedGenericIRPSystem::ParameterNotSupported,
              ManagedGenericIRPSystem::InvalidParameter);
   };
};

#endif _ALARMIRPSYSTEM_IDL_
```

A.3 IDL specification (file name "AlarmIRPNotifications.idl")

```
//File: AlarmIRPNotifications.idl

#ifndef _ALARMIRPNOTIFICATIONS_IDL_
#define _ALARMIRPNOTIFICATIONS_IDL_

#include "AlarmIRPConstDefs.idl"
#include "NotificationIRPConstDefs.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 jobId property,
       * which is transported in the filterable_body fields.
       * 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 jobId property,
       * which is transported in the filterable_body fields.
       * 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 jobId property,
       * which is transported in the filterable_body fields.
       * The data type for the value of this property
       * is string.
       */
      const string SPECIFIC_PROBLEM =
         AlarmIRPConstDefs::AttributeNameValue::SPECIFIC_PROBLEM;
      /**
       * This constant defines the name of the jobId property,
       * which is transported in the filterable_body fields.
       * The data type for the value of this property
       * is AlarmIRPConstDefs::CorrelatedNotificationSetType.
       */
      const string CORRELATED_NOTIFICATIONS =
         AlarmIRPConstDefs::AttributeNameValue::CORRELATED_NOTIFICATIONS;
```

```
        /**
         * This constant defines the name of the jobId property,
         * which is transported in the filterable_body fields.
         * The data type for the value of this property
         * is AlarmIRPConstDefs::BackedUpStatusType.
         */
        const string BACKED_UP_STATUS =
            AlarmIRPConstDefs::AttributeNameValue::BACKED_UP_STATUS;
        /**
         * This constant defines the name of the jobId property,
         * which is transported in the filterable_body fields.
         * 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 jobId property,
         * which is transported in the filterable_body fields.
         * The data type for the value of this property
         * is AlarmIRPConstDefs::TrendIndicationType.
         */
        const string TREND_INDICATION =
            AlarmIRPConstDefs::AttributeNameValue::TREND_INDICATION;
        /**
         * This constant defines the name of the jobId property,
         * which is transported in the filterable_body fields.
         * The data type for the value of this property
         * is AlarmIRPConstDefs::ThresholdInfoType.
         */
        const string THRESHOLD_INFO =
            AlarmIRPConstDefs::AttributeNameValue::THRESHOLD_INFO;
        /**
         * This constant defines the name of the jobId property,
         * which is transported in the filterable_body fields.
         * The data type for the value of this property
         * is AlarmIRPConstDefs::AttributeChangeSetType.
         */
        const string STATE_CHANGE_DEFINITION =
            AlarmIRPConstDefs::AttributeNameValue::STATE_CHANGE_DEFINITION;
        /**
         * This constant defines the name of the jobId property,
         * which is transported in the filterable_body fields.
         * The data type for the value of this property
         * is AlarmIRPConstDefs::AttributeSetType.
         */
        const string MONITORED_ATTRIBUTES =
            AlarmIRPConstDefs::AttributeNameValue::MONITORED_ATTRIBUTES;
        /**
         * This constant defines the name of the jobId property,
         * which is transported in the filterable_body fields.
         * 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 jobId property,
         * which is transported in the filterable_body fields.
         * 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 jobId property,
         * which is transported in the filterable_body fields.
         * 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 jobId property,
         * which is transported in the filterable_body fields.
         * 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 jobId property,
         * which is transported in the filterable_body fields.
         * 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 jobId property,
         * which is transported in the filterable_body fields.
         * 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";

        /**
         * This constant defines the name of the jobId property,
         * which is transported in the filterable_body fields.
         * 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 jobId property,
         * which is transported in the filterable_body fields.
         * 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 jobId property,
         * which is transported in the filterable_body fields.
```

```
        * 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 jobId property,
        * which is transported in the filterable_body fields.
        * The data type for the value of this property
        * is ManagedGenericIRPConstDefs::IRPTime.
        */
       const string ACK_TIME = AlarmIRPConstDefs::AttributeNameValue:: ACK_TIME;
       /**
        * This constant defines the name of the jobId property,
        * which is transported in the filterable_body fields.
        * The data type for the value of this property
        * is string.
        */
       const string ACK_USER_ID =
          AlarmIRPConstDefs::AttributeNameValue::ACK_USER_ID;
       /**
        * This constant defines the name of the jobId property,
        * which is transported in the filterable_body fields.
        * The data type for the value of this property
        * is string.
        */
       const string ACK_SYSTEM_ID =
          AlarmIRPConstDefs::AttributeNameValue::ACK_SYSTEM_ID;
       /**
        * This constant defines the name of the jobId property,
        * which is transported in the filterable_body fields.
        * The data type for the value of this property
        * is AlarmIRPConstDefs::AckState.
        */
       const string ACK_STATE = AlarmIRPConstDefs::AttributeNameValue::ACK_STATE;

   };
   interface NotifyClearedAlarm: NotificationIRPNotifications::Notify
   {
       const string EVENT_TYPE = "notifyClearedAlarm";

       /**
        * This constant defines the name of the jobId property,
        * which is transported in the filterable_body fields.
        * 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 jobId property,
        * which is transported in the filterable_body fields.
        * 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 jobId property,
        * which is transported in the filterable_body fields.
        * 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 jobId property,
        * which is transported in the filterable_body fields.
        * The data type for the value of this property
        * is string.
        */
       const string CLEAR_USER_ID =
          AlarmIRPConstDefs::AttributeNameValue::CLEAR_USER_ID;
       /**
        * This constant defines the name of the jobId property,
        * which is transported in the filterable_body fields.
        * The data type for the value of this property
        * is string.
        */
       const string CLEAR_SYSTEM_ID =
          AlarmIRPConstDefs::AttributeNameValue::CLEAR_SYSTEM_ID;
    };
    interface NotifyAlarmListRebuilt: NotificationIRPNotifications::Notify
    {
       const string EVENT_TYPE = "notifyAlarmListRebuilt";

       /**
        * This constant defines the name of the jobId property,
        * which is transported in the filterable_body fields.
        * The data type for the value of this property
        * is string.
        */
       const string REASON = AlarmIRPConstDefs::AttributeNameValue::REASON;
       /**
        * This constant defines the name of the jobId property,
        * which is transported in the filterable_body fields.
        * The data type for the value of this property
        * is AlarmIRPConstDefs::AlarmListAlignmentRequirementType.
        */
       const string ALARM_LIST_ALIGNMENT_REQUIREMENT =

AlarmIRPConstDefs::AttributeNameValue::ALARM_LIST_ALIGNMENT_REQUIREMENT;
    };
    interface NotifyChangedAlarm: NotificationIRPNotifications::Notify
    {
       const string EVENT_TYPE = "notifyChangedAlarm";

    };
    interface NotifyComments: NotificationIRPNotifications::Notify
    {
       const string EVENT_TYPE = "notifyComments";

       /**
        * This constant defines the name of the jobId property,
        * which is transported in the filterable_body fields.
        * 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 jobId property,
        * which is transported in the filterable_body fields.
        * The data type for the value of this property
        * is string.
        */
       const string REASON = AlarmIRPConstDefs::AttributeNameValue::REASON;
    };


};

#endif _ALARMIRPNOTIFICATIONS_IDL_
```

End of change in Clauses A

3GPP TSG-SA5 (Telecom Management) **S5-047134**
Meeting #40, Sanya, CHINA, 15 - 19 November 2004

CR-Form-v7

CHANGE REQUEST

⌘ **32.111-3** **CR** **037** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network **X** Core Network **X**

Title:	⌘	Correction of filterable parameters - Align with the IS in 32.111-2
Source:	⌘	**SA5** (thomas.tovinger@ericsson.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	19/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

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)
Rel-6 (Release 6)

Reason for change:	⌘	Alignment with IS - Correct the mapping of IS-defined non-filterable parameters to SS-defined non-filterable fields (instead of filterable fields).
Summary of change:	⌘	Place the IS-defined non-filterable parameters into remaining_body of CORBA structured event.
Consequences if not approved:	⌘	This SS would not be aligned with the IS, and the IRPAgent process would waste CPU cycles on non-filterable parameters before emission of notification.

Clauses affected:	⌘	Scope, Subclause 5.3

Other specs affected: ⌘	Y	N	
		X	Other core specifications
		X	Test specifications
		X	O&M Specifications

⌘

Other comments: ⌘

Change in Scope

1 Scope

The present document specifies the CORBA Solution Set (SS) for the IRP whose semantics is specified in Alarm IRP: Information Service (IS) (TS 32.111-2 [6]).

Clause 1 to 3 provides background information. Clause 4 provides key architectural features supporting the SS. Clause 5 defines the mapping of operations, notification, parameters and attributes defined in IS to their SS equivalents. Clause 6 describes the notification interface containing the push method. Annex A contains the IDL specification.

This Solution Set specification is related to TS 32.111-2 V6.3 ~~0~~.X.

End of Change in Scope

Change in subclause 5.3

5.3 Notification parameter mapping

Reference 3G TS 32.111-2 [6] 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 [1]) equivalents. The composition of OMG Structured Event, as defined in the OMG Notification Service [1], 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 [6] defined notification parameters.

Table 11: 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 3.3. 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~~filterable_ body_fields~~	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_	M	Name of NV pair is the PROBABLE_CAUSE of interface

IS Parameters	OMG CORBA Structured Event attribute	Qualifier	Comment
	body_fields		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.
specificProblem	~~One NV pair of~~ One NV pair of remaining_body~~filterable_ body_fields~~	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~~filterable_ body_fields~~	O	Name of NV pair is the CORRELATED_NOTIFICATIONS of interface AttributeNameValue. Value of NV pair is a CorrelatedNotificationSetType of module AlarmIRPConstDefs.
backedUpStatus	One NV pair of remaining_body~~filterable_ body_fields~~	O	Name of NV pair is the BACKED_UP_STATUS of interface AttributeNameValue of module AlarmIRPConstDefs. Value of NV pair is a boolean BackedUpStatusType of module AlarmIRPConstDefs.
backUpObject	One NV pair of remaining_body~~filterable_ body_fields~~	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 [3] for the DN string representation.
trendIndication	One NV pair of remaining_body~~filterable_ body_fields~~	O	Name of NV pair is the TREND_INDICATION of interface AttributeNameValue of module AlarmIRPConstDefs. Value of NV pair is an enum TrendIndicationType of module AlarmIRPConstDefs.
thresholdInfo	One NV pair of remaining_body~~filterable_ body_fields~~	O	Name of NV pair is the THRESHOLD_INFO of interface ParameterNameValue of module AlarmIRPConstDefs. Value of NV pair is a ThresholdInfoType of module AlarmIRPConstDefs.
stateChangeDefinition	One NV pair of remaining_body~~filterable_ body_fields~~	O	Name of NV pair is the STATE_CHANGE_DEFINITION of interface AttributeNameValue of module AlarmIRPConstDefs. Value of NV pair is an AttributeChangeSetType of module AlarmIRPConstDefs.
monitoredAttributes	One NV pair of remaining_body~~filterable_ body_fields~~	O	Name of NV pair is the MONITORED_ATTRIBUTES of interface AttributeNameValue of module AlarmIRPConstDefs. Value of NV pair is an AttributeSetType of module AlarmIRPConstDefs.
proposedRepairActions	~~One NV pair of~~ One NV pair of remaining_body~~filterable_ body_fields~~	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~~One NV pair of filterable_ body_fields~~	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	O	Name and value of all NV pairs are vendor-specific.

IS Parameters	OMG CORBA Structured Event attribute	Qualifier	Comment
	remaining_body~~filterable_body_fields~~		
alarmId	One NV pair of remaining_body~~filterable_body_fields~~	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.
~~There is no corresponding IS attribute.~~	~~remaining_body~~		

Table 12: 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 3.3. 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~~filterable_ body_fields~~	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.
correlatedNotifications	One NV pair of remaining_body~~filterable_ body_fields~~	O	Name of NV pair is the CORRELATED_NOTIFICATIONS of interface AttributeNameValue. Value of NV pair is a CorrelatedNotificationSetType of module AlarmIRPConstDefs.
additionalText	One NV pair of remaining_body~~One NV pair of filterable_ body_fields~~	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~~filterable_~~	O	Name and value of all NV pairs are vendor-specific.

IS Parameters	OMG CORBA Structured Event attribute	Qualifier	Comment
	~~body_fields~~		
alarmId	One NV pair of remaining_body~~filterable_body_fields~~	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~~filterable_body_fields~~	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~~filterable_body_fields~~	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~~filterable_body_fields~~	M	Name of NV pair is the SECURITY_ALARM_DETECTOR of interface AttributeNameValue of module AlarmIRPConstDefs. Value of NV pair is a string.
~~There is no corresponding IS attribute.~~	~~remaining_body~~		

Table 13: 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.
~~notification Id~~notificationId	One NV pair of remaining_body~~filterable_body_fields~~	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~~filterable_body_fields~~	M	See that of notifyNewAlarm.
ackTime	One NV pair of remaining_body~~filterable_body_fields~~	M	Name of NV pair is the ACK_TIME of interface AttributeNameValue of module AlarmIRPConstDefs. Value of NV pair is a IRPTime of module ManagedGenericIRPConstDefs.
ackUserId	One NV pair of remaining_body~~filterable_body_fields~~	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~~filterable_body_fields~~	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~~filterable_body_fields~~	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.
~~There is no corresponding IS attribute.~~	~~remaining_body~~		

Table 14: 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.
~~notification Id~~notificationId	One NV pair of remaining_body~~filterable_body_fields~~	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~~filterable_body_fields~~	M	See that of notifyNewAlarm.
clearUserId	One NV pair of remaining_body~~filterable_body_fields~~	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~~filterable_body_fields~~	O	Name of NV pair is the CLEAR_SYSTEM_ID of interface AttributeNameValue of module AlarmIRPConstDefs. Value of NV pair is a string.
~~There is no corresponding IS attribute.~~	~~remaining_body~~		
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 15: 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.
~~notification Id~~notificationId	One NV pair of remaining_body~~filterable_ body_fields~~	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~~filterable_ body_fields~~	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~~filterable_ body_fields~~	O	Name of NV pair is the ALARM_LIST_ALIGNMENT_REQUIREMENT of interface AttributeNameValue of modue AlarmIRPConstDefs. Value of NV pair is an enum AlarmListAlignmentRequirementType of module AlarmIRPConstDefs.
~~There is no corresponding IS attribute.~~	~~remaining_ body~~		

Table 16: 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.
notification-Id	One NV pair of remaining_body~~filterable_ body_fields~~	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~~filterable_ body_fields~~	M	See that of notifyNewAlarm.
~~There is no corresponding IS attribute.~~	~~remaining_body~~		

Table 17: 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~~filterable_ body_fields~~	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~~filterable_ body_fields~~	M	See that of notifyNewAlarm.
comments	One NV pair of remaining_body~~filterable_ body_fields~~	M	Name of NV pair is the COMMENTS of interface AttributeNameValue of module AlarmIRPConstDefs. Value of NV pair is a CommentSet of module AlarmIRPConstDefs.
~~There is no corresponding IS attribute.~~	~~remaining_body~~		

Table 18: 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 [6] for the description of the ussage of this field to indicate if part or all AlarmList is potentially faulty.
~~notification Id~~notificationId	One NV pair of remaining_body~~filterable_ body_fields~~	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~~filterable_body_fields~~	M	Name of NV pair is the REASON of interface AttributeNameValue of module AlarmIRPConstDefs. Value of NV pair is a string.
~~There is no corresponding IS attribute.~~	~~remaining_body~~		

End of Change in subclause 5.3
End of document.

CR-Form-v7

CHANGE REQUEST

⌘ 32.111-4 CR 029 ⌘rev - ⌘ Current version: 6.2.0 ⌘

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

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network [X] Core Network [X]

Title:	⌘	Remove redundant ackTime parameter in notifyAckStateChanged
Source:	⌘	**SA5** (olaf.pollakowski@siemens.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	01/10/2004
Category:	⌘	**C**
Release:	⌘	Rel-6

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)
Rel-6 (Release 6)

Reason for change:	⌘	The ackTime is carried twice in notifyAckStateChanged, once in the eventTime and once in a dedicated parameter.
Summary of change:	⌘	The dedicated ackTime parameter is removed so that the ackTime is carried only once in eventTime.
Consequences if not approved:	⌘	The ackTime is carried twice in notifyAckStateChanged leading to unnecessary load on the Itf-N.

Clauses affected:	⌘	1, 4.2.5

Other specs affected: ⌘

Y	N	
	X	Other core specifications
	X	Test specifications
	X	O&M Specifications

⌘

Other comments: ⌘ Child to S5-046978 CR 32.111-2

Change in Clause 1

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 3GPP TS 32.111-2 (V6.3~~2~~.X).

End of Change in Clause 1

Change in Clause 4.2.5

4.2.5 Mapping of Notification Parameters

[…]

Table 14: Parameter mapping of the notification *notifyAckStateChanged*

IS Parameter	CMIP SS Equivalent	Qualifier
objectclass	M-EVENT-REPORT parameter 'Managed object class'	M
objectInstance	M-EVENT-REPORT parameter 'Managed object instance'	M
notificationId	M-EVENT-REPORT parameter 'Event information' (AlarmInfo): notificationIdentifier	M
eventTime	M-EVENT-REPORT parameter 'Event time'	M
systemDN	This IS parameter is conditional and not used in the CMIP SS.	--
notificationType	M-EVENT-REPORT parameter 'Event type'	M
probableCause	M-EVENT-REPORT parameter 'Event information' (AlarmInfo): probableCause	M
perceivedSeverity	M-EVENT-REPORT parameter 'Event information' (AlarmInfo): perceivedSeverity	M
alarmType	The semantics of this parameter is conveyed by the notification type.	--
alarmId	M-EVENT-REPORT parameter 'Event information' (AlarmInfo): correlatedNotifications	--
~~ackTime~~	~~M-EVENT-REPORT parameter 'Event information' (AlarmInfo): additionalInformation: ackTimeParameter~~	~~M~~
ackState	M-EVENT-REPORT parameter 'Event information' (AlarmInfo): additionalInformation: ackStateParameter	M
ackUserId	M-EVENT-REPORT parameter 'Event information' (AlarmInfo): additionalInformation: ackUserIdParameter	M
ackSystemId	M-EVENT-REPORT parameter 'Event information' (AlarmInfo): additionalInformation: ackSystemIdParameter	O

[…]

End of Change in Clause 4.2.5