TSGS#28(05)0283

Source: SA5 (Telecom Management)

Title: CR 32111-3-4-5 Fault Management Alarm IRP

Document for: Approval

Agenda Item: 7.5.3

Doc-1st-Level	Spec__#_	CR__#	R	Phase	Subject	Cat	Ver-Cur	Doc-2nd-Level	Workitem
SP-050283	32.111-3	0044	-	Rel-6	Correction of IDL syntax errors in AlarmIRP{ConstDefs	F	6.2.0	S5-056368	OAM-NIM
SP-050283	32.111-4	0032	-	Rel-5	Clarification for Parallel Alarm alignments	F	5.9.0	S5-056369	OAM-NIM
SP-050283	32.111-4	0033	-	Rel-6	Clarification for Parallel Alarm alignments	A	6.4.0	S5-056370	OAM-NIM
SP-050283	32.111-5	0001	-	Rel-6	Align with 32.335	F	6.0.0	S5-056380	OAM-NIM

CR-Form-v7.1

CHANGE REQUEST

⌘ **32.111-3** **CR** **0044** ⌘**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 IDL syntax errors in AlarmIRP{ConstDefs|Notifications}.idl
Source:	⌘	**SA5** (Nortel – Suzèle Lariven – lariven@nortel.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	13/05/2005
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:
Ph2 *(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)*
Rel-7 *(Release 7)*

Reason for change:	⌘	IDL syntax error corrections in AlarmIRP{ConstDefs|Notifications}.idl
Summary of change:	⌘	• Corrections of IDL constant name VENDOR_SPECIFIC_ALARM_TYPE • Alignments with TS 32.150 Style Guide for CORBA SS IDL • Editorial corrections
Consequences if not approved:	⌘	IDLs AlarmIRP{ConstDefs|Notifications}.idl would fail compilation.

Clauses affected:	⌘	A.1, A.2, A.3

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

Other comments: ⌘

Change in Annex Clause A.1

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

```
//File: AlarmIRPConstDefs.idl
#ifndef _AlarmIRPConstDefs_idl_ _ALARM_IRP_CONST_DEFS_IDL_
#define _AlarmIRPConstDefs_idl_ _ALARM_IRP_CONST_DEFS_IDL_

#include "<CosNotification.idl">
#include "<ManagedGenericIRPConstDefs.idl">

// This statement must appear after all include statements
#pragma prefix "3gppsa5.org"

/* ## Module: AlarmIRPConstDefs
This module contains commonly used definitions for Alarm IRP
=================================================================
*/
module AlarmIRPConstDefs
{
   /*
   The format of Distinguished Name (DN) is specified in 3GPP TS 32.300
   "Name Conventions for Managed Objects".
   */
   typedef string DN;

   /* 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 = 3;
   const short FAR_END_RECEIVER_FAILURE = 4;
   const short FRAMING_ERROR = 5;
   const short LOSS_OF_FRAME = 6;
   const short LOSS_OF_POINTER = 7;
   const short LOSS_OF_SIGNAL = 8;
   const short PAYLOAD_TYPE_MISMATCH = 9;

   // Values 10 correspond to a duplicated probable cause
   const short REMOTE_ALARM_INTERFACE = 11;
   const short EXCESSIVE_BIT_ERROR_RATE = 12;
   const short PATH_TRACE_MISMATCH = 13;
   const short UNAVAILABLE = 14;
   const short SIGNAL_LABEL_MISMATCH = 15;
   const short LOSS_OF_MULTI_FRAME = 16;
   const short COMMUNICATIONS_RECEIVE_FAILURE = 17;
   const short COMMUNICATIONS_TRANSMIT_FAILURE = 18;
   const short MODULATION_FAILURE = 19;
   const short DEMODULATION_FAILURE = 20;
   // Values 21-26 correspond to duplicated probable causes
   // Values 27-50 are reserved for M.3100 potential future extensions
   const short BACK_PLANE_FAILURE = 51;
   const short DATA_SET_PROBLEM = 52;
   const short EQUIPMENT_IDENTIFIER_DUPLICATION = 53;
   const short EXTERNAL_IF_DEVICE_PROBLEM = 54;
   const short LINE_CARD_PROBLEM = 55;
   const short MULTIPLEXER_PROBLEM = 56;
   const short NE_IDENTIFIER_DUPLICATION = 57;
   const short POWER_PROBLEM = 58;
   const short PROCESSOR_PROBLEM = 59;
   const short PROTECTION_PATH_FAILURE = 60;
   const short RECEIVER_FAILURE = 61;
   const short REPLACEABLE_UNIT_MISSING = 62;
   const short REPLACEABLE_UNIT_TYPE_MISMATCH = 63;
   const short SYNCHRONIZATION_SOURCE_MISMATCH = 64;
   const short TERMINAL_PROBLEM = 65;
   const short TIMING_PROBLEM = 66;
   const short TRANSMITTER_FAILURE = 67;
   const short TRUNK_CARD_PROBLEM = 68;
   const short REPLACEABLE_UNIT_PROBLEM = 69;
   const short REAL_TIME_CLOCK_FAILURE = 70;
   // Values 71-80 correspond to duplicated probable causes
   const short PROTECTION_MECHANISM_FAILURE = 81;
   const short PROTECTING_RESOURCE_FAILURE = 82;
   // Values 83-100 are reserved for M.3100 potential future extensions
   const short AIR_COMPRESSOR_FAILURE = 101;
   const short AIR_CONDITIONING_FAILURE = 102;
   const short AIR_DRYER_FAILURE = 103;
   const short BATTERY_DISCHARGING = 104;
   const short BATTERY_FAILURE = 105;
   const short COMMERCIAL_POWER_FAILURE = 106;
   const short COOLING_FAN_FAILURE = 107;
   const short ENGINE_FAILURE = 108;
   const short FIRE_DETECTOR_FAILURE = 109;
   const short FUSE_FAILURE = 110;
```

```
const short GENERATOR_FAILURE = 111;
const short LOW_BATTERY_THRESHOLD = 112;
const short PUMP_FAILURE = 113;
const short RECTIFIER_FAILURE = 114;
const short RECTIFIER_HIGH_VOLTAGE = 115;
const short RECTIFIER_LOW_F_VOLTAGE = 116;
const short VENTILATION_SYSTEM_FAILURE = 117;
const short ENCLOSURE_DOOR_OPEN = 118;
const short EXPLOSIVE_GAS = 119;
const short FIRE = 120;
const short FLOOD = 121;
const short HIGH_HUMIDITY = 122;
const short HIGH_TEMPERATURE = 123;
const short HIGH_WIND = 124;
const short ICE_BUILD_UP = 125;
const short INTRUSION_DETECTION = 126;
const short LOW_FUEL = 127;
const short LOW_HUMIDITY = 128;
const short LOW_CABLE_PRESSURE = 129;
const short LOW_TEMPERATURE = 130;
const short LOW_WATER = 131;
const short SMOKE = 132;
const short TOXIC_GAS = 133;
// Values 134-135 correspond to duplicated probable causes
const short EXTERNAL_POINT_FAILURE = 136;
// Values 137-150 are reserved for potential M.3100 future extensions
const short STORAGE_CAPACITY_PROBLEM = 151;
const short MEMORY_MISMATCH = 152;
const short CORRUPT_DATA = 153;
const short OUT_OF_CPU_CYCLES = 154;
const short SOFTWARE_ENVIRONMENT_PROBLEM = 155;
const short SOFTWARE_DOWNLOAD_FAILURE = 156;
const short LOSS_OF_REAL_TIME = 157;
const short REINITIALIZED = 158;
// Values 159-167 correspond to duplicated probable causes
// Values 168-200 are reserved for potential M.3100 future extensions
// Values 201-202 correspond to duplicated probable causes
const short EXCESSIVE_ERROR_RATE = 203;
// Values 204-207 correspond to duplicated probable causes
// Values 208-300 are reserved for potential M.3100 future extensions
/*
Probable causes originating from X.721.
Values below correspond to X.721 values with an offset of 300.
*/
const short ADAPTER_ERROR = 301;
const short APPLICATION_SUBSYSTEM_FAILURE = 302;
const short BANDWIDTH_REDUCED = 303;
// Value 304 corresponds to a duplicated probable cause
const short COMMUNICATIONS_PROTOCOL_ERROR = 305;
const short COMMUNICATIONS_SUBSYSTEM_FAILURE = 306;
const short CONFIGURATION_OR_CUSTOMIZATION_ERROR = 307;
const short CONGESTION = 308;
// Value 309 corresponds to a duplicated probable cause
const short CPU_CYCLES_LIMIT_EXCEEDED = 310;
const short DATA_SET_OR_MODEM_ERROR = 311;
// Value 312 corresponds to a duplicated probable cause
const short DTE_DCE_INTERFACE_ERROR = 313;
// Value 314 corresponds to a duplicated probable cause
const short EQUIPMENT_MALFUNCTION = 315;
const short EXCESSIVE_VIBRATION = 316;
const short FILE_ERROR = 317;
// Values 318-320 correspond to duplicated probable causes
const short HEATING_OR_VENTILATION_OR_COOLING_SYSTEM_PROBLEM = 321;
const short HUMIDITY_UNACCEPTABLE = 322;
const short INPUT_OUTPUT_DEVICE_ERROR = 323;
const short INPUT_DEVICE_ERROR = 324;
const short LAN_ERROR = 325;
const short LEAK_DETECTED = 326;
const short LOCAL_NODE_TRANSMISSION_ERROR = 327;
// Values 328-329 correspond to duplicated probable causes
const short MATERIAL_SUPPLY_EXHAUSTED = 330;
// Value 331 corresponds to a duplicated probable cause
const short OUT_OF_MEMORY = 332;
const short OUTPUT_DEVICE_ERROR = 333;
const short PERFORMANCE_DEGRADED = 334;
// Value 335 corresponds to a duplicated probable cause
const short PRESSURE_UNACCEPTABLE = 336;
// Values 337-338 correspond to duplicated probable causes
```

```
const short QUEUE_SIZE_EXCEEDED = 339;
const short RECEIVE_FAILURE = 340;
// Value 341 corresponds to a duplicated probable cause
const short REMOTE_NODE_TRANSMISSION_ERROR = 342;
const short RESOURCE_AT_OR_NEARING_CAPACITY = 343;
const short RESPONSE_TIME_EXCESSIVE = 344;
const short RETRANSMISSION_RATE_EXCESSIVE = 345;
const short SOFTWARE_ERROR = 346;
const short SOFTWARE_PROGRAM_ABNORMALLY_TERMINATED = 347;
const short SOFTWARE_PROGRAM_ERROR = 348;
// Value 349 corresponds to a duplicated probable cause
const short TEMPERATURE_UNACCEPTABLE = 350;
const short THRESHOLD_CROSSED = 351;
// Value 352 corresponds to a duplicated probable cause
const short TOXIC_LEAK_DETECTED = 353;
const short TRANSMIT_FAILURE = 354;
// Value 355 corresponds to a duplicated probable cause
const short UNDERLYING_RESOURCE_UNAVAILABLE = 356;
const short VERSION_MISMATCH = 357;
// Values 358-500 are reserved for potential X.721 future extensions
/*
Probable causes for 2G & 3G wireless systems.
*/
const short A_BIS_TO_BTS_INTERFACE_FAILURE = 501;
const short A_BIS_TO_TRX_INTERFACE_FAILURE = 502;
const short ANTENNA_PROBLEM = 503;
const short BATTERY_BREAKDOWN = 504;
const short BATTERY_CHARGING_FAULT = 505;
const short CLOCK_SYNCHRONIZATION_PROBLEM = 506;
const short COMBINER_PROBLEM = 507;
const short DISK_PROBLEM = 508;
// Value 509 corresponds to a duplicated probable cause
const short EXCESSIVE_RECEIVER_TEMPERATURE = 510;
const short EXCESSIVE_TRANSMITTER_OUTPUT_POWER = 511;
const short EXCESSIVE_TRANSMITTER_TEMPERATURE = 512;
const short FREQUENCY_HOPPING_DEGRADED = 513;
const short FREQUENCY_HOPPING_FAILURE = 514;
const short FREQUENCY_REDEFINITION_FAILED = 515;
const short LINE_INTERFACE_FAILURE = 516;
const short LINK_FAILURE = 517;
const short LOSS_OF_SYNCHRONIZATION = 518;
const short LOST_REDUNDANCY = 519;
const short MAINS_BREAKDOWN_WITH_BATTERY_BACKUP = 520;
const short MAINS_BREAKDOWN_WITHOUT_BATTERY_BACKUP = 521;
const short POWER_SUPPLY_FAILURE = 522;
const short RECEIVER_ANTENNA_FAULT = 523;
// Value 524 corresponds to a duplicated probable cause
const short RECEIVER_MULTICOUPLER_FAILURE = 525;
const short REDUCED_TRANSMITTER_OUTPUT_POWER = 526;
const short SIGNAL_QUALITY_EVALUATION_FAULT = 527;
const short TIMESLOT_HARDWARE_FAILURE = 528;
const short TRANSCEIVER_PROBLEM = 529;
const short TRANSCODER_PROBLEM = 530;
const short TRANSCODER_OR_RATE_ADAPTER_PROBLEM = 531;
const short TRANSMITTER_ANTENNA_FAILURE = 532;
const short TRANSMITTER_ANTENNA_NOT_ADJUSTED = 533;
// Value 534 corresponds to a duplicated probable cause
const short TRANSMITTER_LOW_VOLTAGE_OR_CURRENT = 535;
const short TRANSMITTER_OFF_FREQUENCY = 536;
const short DATABASE_INCONSISTENCY = 537;
const short FILE_SYSTEM_CALL_UNSUCCESSFUL = 538;
const short INPUT_PARAMETER_OUT_OF_RANGE = 539;
const short INVALID_PARAMETER = 540;
const short INVALID_POINTER = 541;
const short MESSAGE_NOT_EXPECTED = 542;
const short MESSAGE_NOT_INITIALIZED = 543;
const short MESSAGE_OUT_OF_SEQUENCE = 544;
const short SYSTEM_CALL_UNSUCCESSFUL = 545;
const short TIMEOUT_EXPIRED = 546;
const short VARIABLE_OUT_OF_RANGE = 547;
const short WATCH_DOG_TIMER_EXPIRED = 548;
const short COOLING_SYSTEM_FAILURE = 549;
const short EXTERNAL_EQUIPMENT_FAILURE = 550;
const short EXTERNAL_POWER_SUPPLY_FAILURE = 551;
const short EXTERNAL_TRANSMISSION_DEVICE_FAILURE = 552;
// Values 553-560 correspond to duplicated probable causes
const short REDUCED_ALARM_REPORTING = 561;
const short REDUCED_EVENT_REPORTING = 562;
```

```
   const short RECUCED_LOGGING_CAPABILITY = 563;
   const short SYSTEM_RESOURCES_OVERLOAD = 564;
   const short BROADCAST_CHANNEL_FAILURE = 565;
   const short CONNECTION_ESTABLISHMENT_ERROR = 566;
   const short INVALID_MESSAGE_RECEIVED = 567;
   const short INVALID_MSU_RECEIVED = 568;
   const short LAPD_LINK_PROTOCOL_FAILURE = 569;
   const short LOCAL_ALARM_INDICATION = 570;
   const short REMOTE_ALARM_INDICATION = 571;
   const short ROUTING_FAILURE = 572;
   const short SS7_PROTOCOL_FAILURE = 573;
   const short TRANSMISSION_ERROR = 574;
   // Value 575 corresponds to a duplicated probable cause
   // Values 576-700 are reserved for potential future extensions
   // for 2G & 3G wireless systems
   /*
   Probable causes originating from M.3100 security alarm causes.
   Values below correspond to M.3100 values with an offset of 700.
   */
   const short Authentication_Failure = 701;
   const short Breach_of_Confidentiality = 702;
   const short Cable_Tamper = 703;
   const short Delayed_Information = 704;
   const short Denial_of_Service = 705;
   const short Duplicate_Information = 706;
   const short Information_Missing = 707;
   const short Information_Modification_detected = 708;
   const short Information_out_of_Sequence = 709;
   // Value 710 corresponds to a duplicated probable cause
   const short Key_Expired = 711;
   const short Non_Repudiation_Failure = 712;
   const short Out_of_Hours_Activity = 713;
   const short Out_of_Service = 714;
   const short Procedural_Error = 715;
   const short Unauthorised_Access_Attempt = 716;
   const short Unexpected_Information = 717;
   const short Unspecified_Reason = 718;
   // Values 719-800 are reserved for potential M.3100 future extensions
};

/*
This block identifies the acknowledgement state of a reported alarm.
*/
interface AckState
{
   const short ACKNOWLEDGED = 1;
   const short UNACKNOWLEDGED = 2;
};

/*
This block identifies attributes which are included as part of the Alarm IRP
These attribute values should not clash with those defined for the attributes
of notification header (see IDL of Notification IRP).
*/
interface AttributeNameValue
{
   const string ALARM_ID = "f";
   const string PROBABLE_CAUSE = "g";
   const string PERCEIVED_SEVERITY = "h";
   const string SPECIFIC_PROBLEM = "i";
   const string ADDITIONAL_TEXT = "j";
   const string ACK_TIME = "k";
   const string ACK_USER_ID = "l";
   const string ACK_SYSTEM_ID = "m";
   const string ACK_STATE = "n";
   const string COMMENTS = "o";
   const string BACKED_UP_STATUS = "p";
   const string BACK_UP_OBJECT = "q";
   const string THRESHOLD_INFO = "r";
   const string TREND_INDICATION = "s";
   const string STATE_CHANGE_DEFINITION = "t";
   const string MONITORED_ATTRIBUTES = "u";
   const string PROPOSED_REPAIR_ACTIONS = "v";
   const string CORRELATED_NOTIFICATIONS = "w";
   const string REASON = "x";
   const string CLEAR_USER_ID = "y";
   const string CLEAR_SYSTEM_ID = "z";
   const string ALARM_LIST_ALIGNMENT_REQUIREMENT = "ff";
```

```
   const string SERVICE_USER = "gg";
   const string SERVICE_PROVIDER = "hh";
   const string SECURITY_ALARM_DETECTOR = "ii";
   const string VENDOR_SPECIFIC_ALARM_TYPE = "JJjj";
};

/*
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;

   /*
   Define the cause values of notifyPotentialFaultyAlarmList and
   notifyAlarmListRebuilt, which have been defined in 32111-2.
   */
   const string AGENT_NE_COMMUNICATION_ERROR = "Agent-NE communication error";
   const string AGENT_RESTARTS = "Agent restarts";
   const string INDETERMINATE = "Indeterminate";

};

#endif // _AlarmIRPConstDefs_idl_ _ALARM_IRP_CONST_DEFS_IDL_
```

End of Change in Annex Clause A.1

Change in Annex Clause A.2

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

```
//File: AlarmIRPSystem.idl

#ifndef _ALARM_IRP_SYSTEM_IDL_
#define _ALARM_IRP_SYSTEM_IDL_

#include "<AlarmIRPConstDefs.idl">
#include "<ManagedGenericIRPSystem.idl">

// This statement must appear after all include statements
#pragma prefix "3gppsa5.org"
```

```
/* ## Module: AlarmIRPSystem
This module contains the specification of all operations of Alarm IRP Agent.
================================================================
*/
module AlarmIRPSystem
{
   /*
   System fails to complete the operation. —System can provide reason
   to qualify the exception. —The semantics carried in reason
   is outside the scope of this IRP.
   */
   exception GetAlarmIRPVersions { string reason; };
   exception GetAlarmIRPOperationsProfile { string reason; };
   exception GetAlarmIRPNotificationProfile { string reason; };
   exception AcknowledgeAlarms { string reason; };
   exception UnacknowledgeAlarms { string reason; };
   exception CommentAlarms { string reason; };
   exception ClearAlarms { string reason; };
   exception GetAlarmList { string reason; };
   exception GetAlarmCount { string reason; };
   exception NextAlarmInformations { string reason; };

   /*
   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::OperationNotSupported,
        ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
This method returns Alarm Informations.
If flag is TRUE, all returned Alarm Informations shall be
in AlarmInformationSeq that contains 0 or more Alarm Informations.
Output parameter iter shall be useless.
If flag is FALSE, no Alarm Informations shall be in AlarmInformationSeq.
IRPAgent needs to use iter to retrieve them.
```

```
      */
      AlarmIRPConstDefs::AlarmInformationSeq get_alarm_list (
         in ManagedGenericIRPConstDefs::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_ _ALARM_IRP_SYSTEM_IDL_
```

End of Change in Annex Clause A.2

Change in Annex Clause A.3

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

```
//File: AlarmIRPNotifications.idl

#ifndef _ALARM_IRP_NOTIFICATIONS_IDL_
#define _ALARM_IRP_NOTIFICATIONS_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 probableCause property.
       * The data type for the value of this property
       * is short.
       */
      const string PROBABLE_CAUSE =
         AlarmIRPConstDefs::AttributeNameValue::PROBABLE_CAUSE;

      /**
```

```
         * This constant defines the name of the
         * perceivedSeverity property.
         * The data type for the value of this property
         * is short.
         */
        const string PERCEIVED_SEVERITY =
           AlarmIRPConstDefs::AttributeNameValue::PERCEIVED_SEVERITY;

        /**
         * This constant defines the name of the specificProblem
         * property.  The data type for the value of this property
         * is string.
         */

        /**
         * This constant identifies the NE alarm type or NE related alarm type.
         * The data type for the value of this property is string.
         */
        const string VENDOR_SPECIFIC_ALARM_TYPE =
           AlarmIRPConstDefs::AttributeNameValue::VENDOR_SPECIFIC_ALARM_TYPE;

        /**
This constant identifies the NE alarm type or NE related alarm type.
The data type for the value of this property is string.
         */

        /**
         * This constant defines the name of the specificProblem
         * property. The data type for the value of this property
         * is string.
         */
        const string SPECIFIC_PROBLEM =
           AlarmIRPConstDefs::AttributeNameValue::SPECIFIC_PROBLEM;

        /**
         * This constant defines the name of the
         * correlatedNotifications property.
         * 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
         * backedUpStatus property.
         * 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 backUpObject property.
         * The data type for the value of this property
         * is a string carrying of DN of the back-up object.
         */
        const string BACK_UP_OBJECT =
           AlarmIRPConstDefs::AttributeNameValue::BACK_UP_OBJECT;

        /**
         * This constant defines the name of the
         * trendIndication property.
         * The data type for the value of this property
         * is AlarmIRPConstDefs::TrendIndicationType.
         */
        const string TREND_INDICATION =
           AlarmIRPConstDefs::AttributeNameValue::TREND_INDICATION;

        /**
         * This constant defines the name of the thresholdInfo property.
         * The data type for the value of this property
         * is AlarmIRPConstDefs::ThresholdInfoType.
         */
        const string THRESHOLD_INFO =
           AlarmIRPConstDefs::AttributeNameValue::THRESHOLD_INFO;
```

```
        /**
         * This constant defines the name of the
         * stateChangeDefinition property.
         * The data type for the value of this property
         * is AlarmIRPConstDefs::AttributeChangeSetType.
         */
        const string STATE_CHANGE_DEFINITION =
          AlarmIRPConstDefs::AttributeNameValue::STATE_CHANGE_DEFINITION;

        /**
         * This constant defines the name of the
         * monitoredAttributes property.
         * The data type for the value of this property
         * is AlarmIRPConstDefs::AttributeSetType.
         */
        const string MONITORED_ATTRIBUTES =
           AlarmIRPConstDefs::AttributeNameValue::MONITORED_ATTRIBUTES;

        /**
         * This constant defines the name of the
         * proposedRepairActions property.
         * The data type for the value of this property
         * is string.
         */
        const string PROPOSED_REPAIR_ACTIONS =
          AlarmIRPConstDefs::AttributeNameValue::PROPOSED_REPAIR_ACTIONS;

        /**
         * This constant defines the name of the additiionalText
         * property.
         * The data type for the value of this property
         * is string.
         */
        const string ADDITIONAL_TEXT =
           AlarmIRPConstDefs::AttributeNameValue::ADDITIONAL_TEXT;

        /**
         * This constant defines the name of the alarmId property.
         * The data type for the value of this property
         * is string. –If the string is a zero-length string or if
         * this NV pair is absent, the default semantics is that
         * alarmId is a concatenation of
         * managedObjectInstance, eventType, probableCause and
         * specificProblem, if present, of this Structured Event.
         * Since probableCause is encoded as a short, it shall be
         * converted into string before concatenation.
         * The resultant string shall not contain spaces.
         */
        const string ALARM_ID =
          AlarmIRPConstDefs::AttributeNameValue::ALARM_ID;

        /**
         * This constant defines the name of the serviceUser property.
         * The data type for the value of this property
         * is string.
         */
        const string SERVICE_USER =
           AlarmIRPConstDefs::AttributeNameValue::SERVICE_USER;

        /**
         * This constant defines the name of the serviceProvider
         * property.
         * The data type for the value of this property
         * is string.
         */
        const string SERVICE_PROVIDER =
           AlarmIRPConstDefs::AttributeNameValue::SERVICE_PROVIDER;

        /**
         * This constant defines the name of the
         * securityAlarmDetector property.
         * The data type for the value of this property
         * is string.
         */
        const string SECURITY_ALARM_DETECTOR =
          AlarmIRPConstDefs::AttributeNameValue::SECURITY_ALARM_DETECTOR;
    };
```

```
interface NotifyAckStateChanged:
    NotificationIRPNotifications::Notify
{
   const string EVENT_TYPE = "notifyAckStateChanged";

   const string PROBABLE_CAUSE =
      AlarmIRPConstDefs::AttributeNameValue::PROBABLE_CAUSE;

   const string PERCEIVED_SEVERITY =
      AlarmIRPConstDefs::AttributeNameValue::PERCEIVED_SEVERITY;

   const string ALARM_ID =
      AlarmIRPConstDefs::AttributeNameValue::ALARM_ID;

   /**
    * This constant defines the name of the ackTime property.
    * The data type for the value of this property
    * is ManagedGenericIRPConstDefs::IRPTime.
    */
   const string ACK_TIME =
     AlarmIRPConstDefs::AttributeNameValue:: ACK_TIME;

   const string ACK_USER_ID =
      AlarmIRPConstDefs::AttributeNameValue::ACK_USER_ID;

   const string ACK_SYSTEM_ID =
      AlarmIRPConstDefs::AttributeNameValue::ACK_SYSTEM_ID;

   const string ACK_STATE =
       AlarmIRPConstDefs::AttributeNameValue::ACK_STATE;

};


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

   const string PROBABLE_CAUSE =
      AlarmIRPConstDefs::AttributeNameValue::PROBABLE_CAUSE;

   const string PERCEIVED_SEVERITY =
      AlarmIRPConstDefs::AttributeNameValue::PERCEIVED_SEVERITY;

   const string ALARM_ID =
     AlarmIRPConstDefs::AttributeNameValue::ALARM_ID;

   const string CLEAR_USER_ID =
      AlarmIRPConstDefs::AttributeNameValue::CLEAR_USER_ID;

   const string CLEAR_SYSTEM_ID =
      AlarmIRPConstDefs::AttributeNameValue::CLEAR_SYSTEM_ID;
};


interface NotifyAlarmListRebuilt:
    NotificationIRPNotifications::Notify
{
   const string EVENT_TYPE = "notifyAlarmListRebuilt";

   const string REASON =
     AlarmIRPConstDefs::AttributeNameValue::REASON;

   const string ALARM_LIST_ALIGNMENT_REQUIREMENT =
     AlarmIRPConstDefs::AttributeNameValue::
       ALARM_LIST_ALIGNMENT_REQUIREMENT;
};


interface NotifyChangedAlarm: NotificationIRPNotifications::Notify
{
   const string EVENT_TYPE = "notifyChangedAlarm";

   const string PROBABLE_CAUSE =
      AlarmIRPConstDefs::AttributeNameValue::PROBABLE_CAUSE;

   const string PERCEIVED_SEVERITY =
```

```
        AlarmIRPConstDefs::AttributeNameValue::PERCEIVED_SEVERITY;

      const string ALARM_ID =
        AlarmIRPConstDefs::AttributeNameValue::ALARM_ID;


   };


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

      const string PROBABLE_CAUSE =
         AlarmIRPConstDefs::AttributeNameValue::PROBABLE_CAUSE;

      const string PERCEIVED_SEVERITY =
         AlarmIRPConstDefs::AttributeNameValue::PERCEIVED_SEVERITY;

      const string ALARM_ID =
        AlarmIRPConstDefs::AttributeNameValue::ALARM_ID;

      /**
       * This constant defines the name of the comments property.
       * The data type for the value of this property
       * is AlarmIRPConstDefs::CommentSet.
       */
      const string COMMENTS =
        AlarmIRPConstDefs::AttributeNameValue::COMMENTS;
   };


   interface NotifyPotentialFaultyAlarmList:
       NotificationIRPNotifications::Notify
   {
      const string EVENT_TYPE = "notifyPotentialFaultyAlarmList";

      /**
       * This constant defines the name of the reason property.
       * The data type for the value of this property
       * is string.
       */
      const string REASON =
        AlarmIRPConstDefs::AttributeNameValue::REASON;
   };

};

#endif // _ALARM_IRP_NOTIFICATIONS_IDL_
```

End of Change in Annex Clause A.3
End of Document

Annex B (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Mar 2005	SA_27	SP-050021	041	--	Add definitions in IDLs - Align with the IS (TS 32.111-2)	6.1.0	6.2.0
Mar 2005	SA_27	SP-050021	043	--	Update the IS-SS relatioship in the Alarm IRP CORBA SS	6.1.0	6.2.0

CR-Form-v7.1

CHANGE REQUEST

⌘ **32.111-4** **CR** **0032** ⌘ **rev** **-** ⌘ Current version: **5.9.0** ⌘

*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:	⌘	Clarification for Parallel Alarm alignments
Source:	⌘	SA5 (clemens.suerbaum@siemens.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	13/05/2005
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:
Ph2 *(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)*
Rel-7 *(Release 7)*

Reason for change:	⌘	The description of alarm alignment is not complete and precise enough for simultaneous alignments initiated by the same or different IRP managers.
Summary of change:	⌘	Add descriptions to make text clearer and unambigous
Consequences if not approved:	⌘	Incompatible implementations. Duplicate sending or unwanted reception or wrong interpretation of notifyAlarmAlignmentEnd notification possible.

Clauses affected:	⌘	4.1.6

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

Other comments: ⌘

Change in Clause 4.1.6

4.1.6 Alignment of alarm conditions over the Itf-N

The IRP Manager is able to trigger the alarm conditions alignment using the Action *getAlarmList*

The following specifies the logical steps of the alignment procedure, by describing a possible implementation. Any other implementation showing the same behaviour on the Itf-N interface is compliant with the present document.

- The Manager sends to the Agent a *getAlarmList* request containing the following information:
  - *alarmAckState*, used to select the alarms from the Agent's alarm list for the current alignment (e.g. all active alarms).
  - *destination*, identifying the destination to which event reports that have passed the filter conditions are sent.
  - *filter*, this optional parameter defines the conditions an alarm notification shall fulfil in order to be forwarded to the Manager. It applies only for the current alignment request.
- After evaluation of the request, the Agent first generates an *alignmentId* value, which unambiguously identifies this alignment process. This value is used by the Manager to correlate alarm reports to the corresponding alignment requests, in case this Manager issues several alarm alignments in parallel.
- The Agent creates a temporary Event Forwarding Discriminator (EFD) instance for the purpose of this alarm alignment, using the parameters *destination* and *filter* received in the request. If the *filter* parameter is absent in the alarm synchronisation request, all alarm notifications are forwarded to the Manager through this EFD, taking into account both the *filter* constraint currently active for the event reporting to the manager having invoked the synchronisation request and the value of the parameter *alarmAckState*.
  The filter is set by the Agent automatically in order to forward only those alarm notifications containing, at the beginning of the field *additionalText*, the string "(ALIGNMENT-<alignmentId>)". The filter must also forward the notification *notifyAlarmAlignmentEnd* indicating the end of the alarm alignment process. The alarm alignment end notifications of other alignment processes shall be filtered out using the *alignmentId* carried by the event information parameter of *notifyAlarmAlignmentEnd*.
- The Agent sends back a *getAlarmList* response, which contains the *alignmentId* described above and the *status* information, indicating the result of the request. (see the message flow in Figure 1).
- The Agent scans now its alarm list. For every alarm, which matches the criteria defined by the *alarmAckState* parameter and the *filter* parameter, the Agent inserts, at the beginning of the field *additionalText*, the string "(ALIGNMENT-<alignmentId>)".
- Depending on the event being reported, the *additionalInformation* field of every alarm notification shall carry the parameters *ackTimeParameter*, *ackStateParameter*, *ackUserIdParameter*, *ackSystemIdParameter*, *clearUserIdParameter*, *clearSystemIdParameter*, *commentsParameter*, *alarmRaisedTimeParameter* or *alarmClearedTimeParameter*.
- According to ITU-T Recommendation X.734 [6], the Agent forwards these alarm notifications towards all EFDs.

  NOTE: These alarm notifications can reach the current Manager only via the temporary EFD created for the current alignment. They are filtered out:

  a) By all the EFD instances used for "real-time" alarm reporting, due to the presence of the sub-string "ALIGNMENT" in the field *additionalText* (see 3GPP TS 32.304 [10]).

  b) By all temporary EFD instances possibly created for parallel alignments, due to the presence of the unambiguous sub-string "<alignmentId>" in the *additionalText* field.
- At the end of the alarm alignment process the Agent shall send the dedicated notification *notifyAlarmAlignmentEnd* in order to indicate the end of the current alignment process (unambiguously identified by the *alignmentId*). In case the alarm list is empty or no alarm matches the criteria defined by the *alarmAckState* parameter and the *filter* paramter the notification *notifyAlarmAlignmentEnd* shall be emitted directly after the the agent has send the *getAlarmList* response.

- The temporary EFD of the current alarm alignment process shall forward only alarm alignment end notifications carrying in the event information field the *alignmnentId* of this alignment process. All other alarm alignment end notifications shall be filtered out.

- Each NMC has to set the filter of its permanent EFD instance in order to block the notifyAlarmAlignmentEnd notification (otherwise the NMC would receive this notification twice: Once by the termporary EFD, once by the permanent)

- In case of several alignments running in parallel, each NMC has to evaluate the alignmentId value of every received notifyAlarmAlignmentEnd notification (passed via all "temporary" EFD instances) and to ignore those notifications containing alignmentId values that do not correspond to one of its own alignments.

- After sending the notification *notifyAlarmAlignmentEnd* the Agent automatically deletes the temporary EFD instance (see figure 1).

At the end of the alarm conditions alignment the acknowledgement state and the comments assigned to each alarm are implicitly synchronised between the IRPAgent and the IRPManager that has requested the alignment.

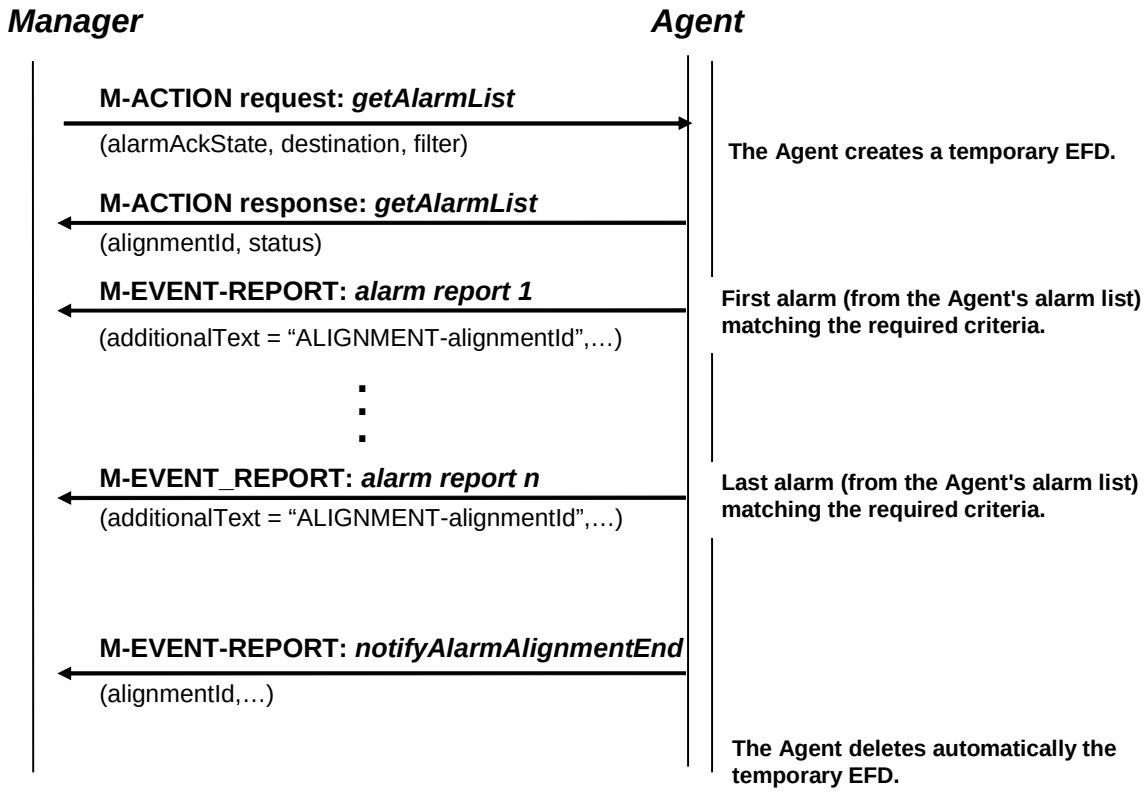


Figure 1: Alignment arrow diagram

Figure 2 shows the handling of a "real-time" alarm notification (occurred during the execution of the *getAlarmList* operation), which is forwarded by the Agent (according to ITU-T Recommendation X.734 [6]) to all currently available EFD instances. Dependent on the *discriminatorConstruct* setting of every EFD, such an alarm may or may not reach the related Manager. In any case, this alarm is filtered out by the temporary EFD assigned to the Manager, which triggered the *getAlarmList* request.

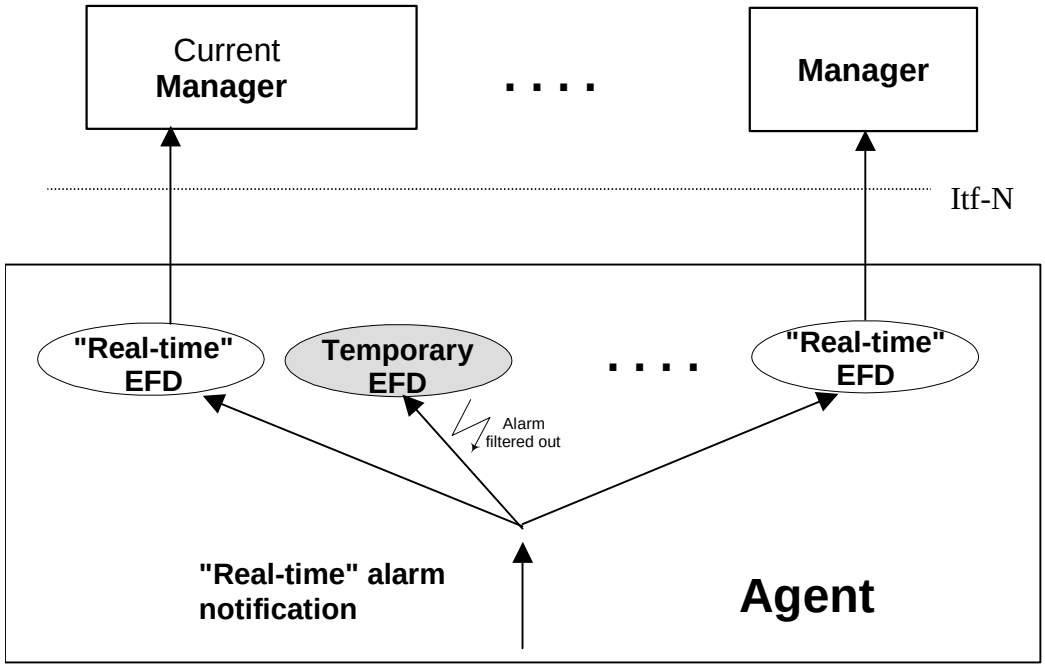


Figure 2: Treatment of "real time" alarms

Figure 3 shows the handling of an alarm notification from the alarm list, matching the criteria defined in the parameters *alarmAckState* of the *getAlarmList* request and forwarded by the Agent to all EFD instances as well. This alarm is filtered out by all EFD instances in charge of discrimination of "real-time" alarms and can reach only the Manager, which triggered the *getAlarmList* request, because it passes the temporary EFD instance assigned to this Manager.

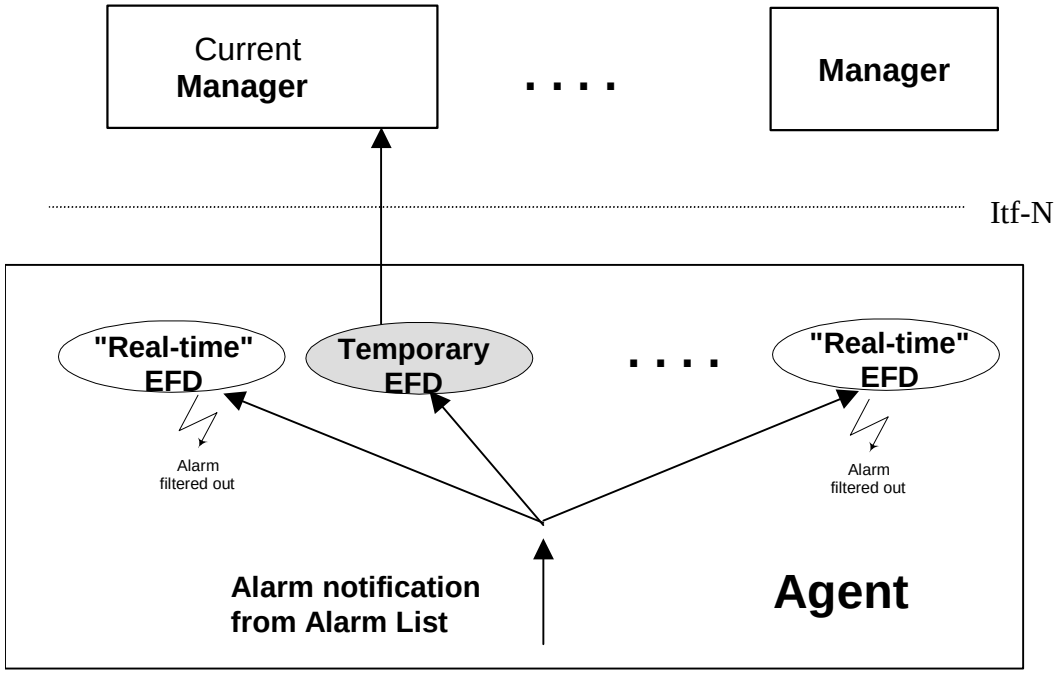


Figure 3: Treatment of "alignment" alarms

End of Change in Clause 4.1.6

End of Document

Annex B (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Mar 2000	SA_07	SP-000012	--	--	Approved at TSG SA #7 and placed under Change Control	2.0.0	3.0.0
Mar 2000	--	--	--	--	cosmetic	3.0.0	3.0.1
Jun 2000	SA_08	SP-000254	005	--	Split of TS - Part 4: Alarm Integration Reference Point (IRP): CMIP Solution Set (SS)	3.0.1	3.1.0
Sep 2000	--	--	--	--	cosmetic	3.1.0	3.1.1
Jun 2001	SA_12	SP-010282	001	--	Alarm IRP: CMIP SS Rel4 - Addition of feature. As SA5 had not reviewed this part, it is submitted to SA#12 for Information only.	3.1.1	--
Sep 2001	SA_13	SP-010470	001	1	Addition of features	3.1.1	4.0.0
Dec 2001	SA_14	SP-010640	003	--	Change of qualifier for setComment and notifyComment	4.0.0	4.1.0
Dec 2001	SA_14	SP-010640	004	--	Addition of missing parameter in notifyComments	4.0.0	4.1.0
Mar 2002	SA_15	SP-020028	005	--	Addition of "perceivedSeverity" as parameter to "acknowledgeAlarms" operation (CMIP SS)	4.1.0	4.2.0
Mar 2002	SA_15	--	--	--	Automatic upgrade to Rel-5 (no Rel-5 CR)	4.2.0	5.0.0
Jun 2002	SA_16	SP-020283	007	--	Correction of errors and ambiguities in the Parameter Mapping Tables and ASN.1 Definitions	5.0.0	5.1.0
Jun 2002	SA_16	SP-020284	008	--	Addition of the parameter alarmListAlignmentRequirement to the notification notifyAlarmListRebuilt in the CMIP SS (32.111-4)	5.0.0	5.1.0
Jun 2002	SA_16	SP-020284	009	--	Adding the notification notifyPotentialFaultyAlarmList in the CMIP SS (32.111-4)	5.0.0	5.1.0
Jun 2002	SA_16	SP-020284	010	--	Introduction of SS (32.111-4) to IS (32.111-2) relation and correction of Foreword	5.0.0	5.1.0
Sep 2002	SA_17	SP-020480	011	--	Alignment with 32.111-2 on Alarm Clearance Functionality	5.1.0	5.2.0
Dec 2002	SA_18	SP-020751	013	--	Add the additionalInformation parameter in notifyNewAlarms to the Alarm IRP CMIP SS (Alignment with Information Service in Rel-5 32111-2)	5.2.0	5.3.0
Dec 2002	SA_18	SP-020753	014	--	Addition of Security Alarm Support to the Alarm IRP CMIP SS (Alignment with Information Service in Rel-5 32111-2)	5.2.0	5.3.0
Mar 2003	SA_19	SP-030063	016	--	Correction to Alarm Comments- alignment with 32.111-1	5.3.0	5.4.0
Mar 2003	SA_19	SP-030138	017	--	Add missing x721AlarmNotificationsPackage	5.3.0	5.4.0
Mar 2003	SA_19	SP-030138	018	--	Corrections to GDMO and ASN.1 definitions in the Alarm IRP CMIP SS	5.3.0	5.4.0
Jun 2003	SA_20	SP-030277	019	--	Correction of Compilation Errors	5.4.0	5.5.0
Jun 2003	SA_20	SP-030277	020	--	Addition of missing reasons for the emission of notifyAlarmListRebuilt	5.4.0	5.5.0
Sep 2003	SA_21	SP-030416	022	--	Correction of syntax error in type SetCommentInfo	5.5.0	5.6.0
Dec 2003	SA_22	SP-030627	023	--	Add missing parts for the support of security alarms	5.6.0	5.7.0
Dec 2003	SA_22	SP-030627	024	--	Mapping completion of getAlarmList	5.6.0	5.7.0
Jan 2004	--	--	--	--	Editorial (Tables & CMIP code cosmetics)	5.7.0	5.7.1
Sep 2004	SA_25	SP-040561	027	--	Align with the IS 32.111-2 the possibility to apply filters to notification parameters	5.7.1	5.8.0
Mar 2005	SA_27	SP-050021	030	--	Add missing definition of getAlarmList return value - Align with the IS (TS 32.111-2)	5.8.0	5.9.0

Tdoc ⌘S5-056370

CR-Form-v7.1

CHANGE REQUEST

⌘ **32.111-4** **CR** **0033** ⌘**rev** **-** ⌘ Current version: **6.4.0** ⌘

*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:	⌘	Clarification for Parallel Alarm alignments
Source:	⌘	SA5 (clemens.suerbaum@siemens.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	13/05/2005
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:
Ph2 *(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)*
Rel-7 *(Release 7)*

Reason for change:	⌘	The description of alarm alignment is not complete and precise enough for simultaneous alignments initiated by the same or different IRP managers.
Summary of change:	⌘	Add descriptions to make text clearer and unambigous
Consequences if not approved:	⌘	Incompatible implementations. Duplicate sending or unwanted reception or wrong interpretation of notifyAlarmAlignmentEnd notification possible.

Clauses affected:	⌘	4.1.6

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

Other comments: ⌘

Change in Clause 4.1.6

4.1.6 Alignment of alarm conditions over the Itf-N

The IRP Manager is able to trigger the alarm conditions alignment using the Action *getAlarmList*

The following specifies the logical steps of the alignment procedure, by describing a possible implementation. Any other implementation showing the same behaviour on the Itf-N interface is compliant with the present document.

- The Manager sends to the Agent a *getAlarmList* request containing the following information:
  - *alarmAckState,* used to select the alarms from the Agent's alarm list for the current alignment (e.g. all active alarms).
  - *baseObjectClass, baseObjectInstance,* identifies the part of the alarm list to be uploaded.
  - *destination,* identifying the destination to which event reports that have passed the filter conditions are sent.
  - *filter,* this optional parameter defines the conditions an alarm notification shall fulfil in order to be forwarded to the Manager. It applies only for the current alignment request.
- After evaluation of the request, the Agent first generates an *alignmentId* value, which unambiguously identifies this alignment process. This value is used by the Manager to correlate alarm reports to the corresponding alignment requests, in case this Manager issues several alarm alignments in parallel.
- The Agent creates a temporary Event Forwarding Discriminator (EFD) instance for the purpose of this alarm alignment, using the parameters *destination* and *filter* received in the request. If the *filter* parameter is absent in the alarm synchronisation request, all alarm notifications are forwarded to the Manager through this EFD, taking into account both the *filter* constraint currently active for the event reporting to the manager having invoked the synchronisation request and the value of the parameter *alarmAckState*.
  The filter is set by the Agent automatically in order to forward only those alarm notifications containing, at the beginning of the field *additionalText*, the string "(ALIGNMENT-<alignmentId>)". The filter must also forward the notification *notifyAlarmAlignmentEnd* indicating the end of the alarm alignment process. The alarm alignment end notifications of other alignment processes shall be filtered out using the *alignmentId* carried by the event information parameter of *notifyAlarmAlignmentEnd*.
- The Agent sends back a *getAlarmList* response, which contains the *alignmentId* described above and the *status* information, indicating the result of the request. (see the message flow in Figure 1).
- The Agent scans now its alarm list. For every alarm, which matches the criteria defined by the *alarmAckState* parameter and the *filter* parameter, the Agent inserts, at the beginning of the field *additionalText*, the string "(ALIGNMENT-<alignmentId>)".
- Depending on the event being reported, the *additionalInformation* field of every alarm notification shall carry the parameters *ackTimeParameter*, *ackStateParameter*, *ackUserIdParameter*, *ackSystemIdParameter*, *clearUserIdParameter*, *clearSystemIdParameter*, *commentsParameter*, *alarmRaisedTimeParameter* or *alarmClearedTimeParameter*.
- According to ITU-T Recommendation X.734 [6], the Agent forwards these alarm notifications towards all EFDs.

  NOTE: These alarm notifications can reach the current Manager only via the temporary EFD created for the current alignment. They are filtered out:

  a) By all the EFD instances used for "real-time" alarm reporting, due to the presence of the sub-string "ALIGNMENT" in the field *additionalText* (see 3GPP TS 32.304 [10]).

  b) By all temporary EFD instances possibly created for parallel alignments, due to the presence of the unambiguous sub-string "<alignmentId>" in the additionalText field.
- At the end of the alarm alignment process the Agent shall send the dedicated notification *notifyAlarmAlignmentEnd* in order to indicate the end of the current alignment process (unambiguously identified by the *alignmentId*). In case the alarm list is empty or no alarm matches the criteria defined by the *alarmAckState* parameter and the *filter* parameter the notification *notifyAlarmAlignmentEnd* shall be emitted directly after the agent has send the *getAlarmList* response.

- The temporary EFD of the current alarm alignment process shall forward only alarm alignment end notifications carrying in the event information field the *alignmnentId* of this alignment process. All other alarm alignment end notifications shall be filtered out.

- Each NMC has to set the filter of its permanent EFD instance in order to block the notifyAlarmAlignmentEnd notification (otherwise the NMC would receive this notification twice: Once by the termporary EFD, once by the permanent)

- In case of several alignments running in parallel, each NMC has to evaluate the alignmentId value of every received notifyAlarmAlignmentEnd notification (passed via all "temporary" EFD instances) and to ignore those notifications containing alignmentId values that do not correspond to one of its own alignments.

- After sending the notification *notifyAlarmAlignmentEnd* the Agent automatically deletes the temporary EFD instance (see figure 1).

At the end of the alarm conditions alignment the acknowledgement state and the comments assigned to each alarm are implicitly synchronised between the IRPAgent and the IRPManager that has requested the alignment.

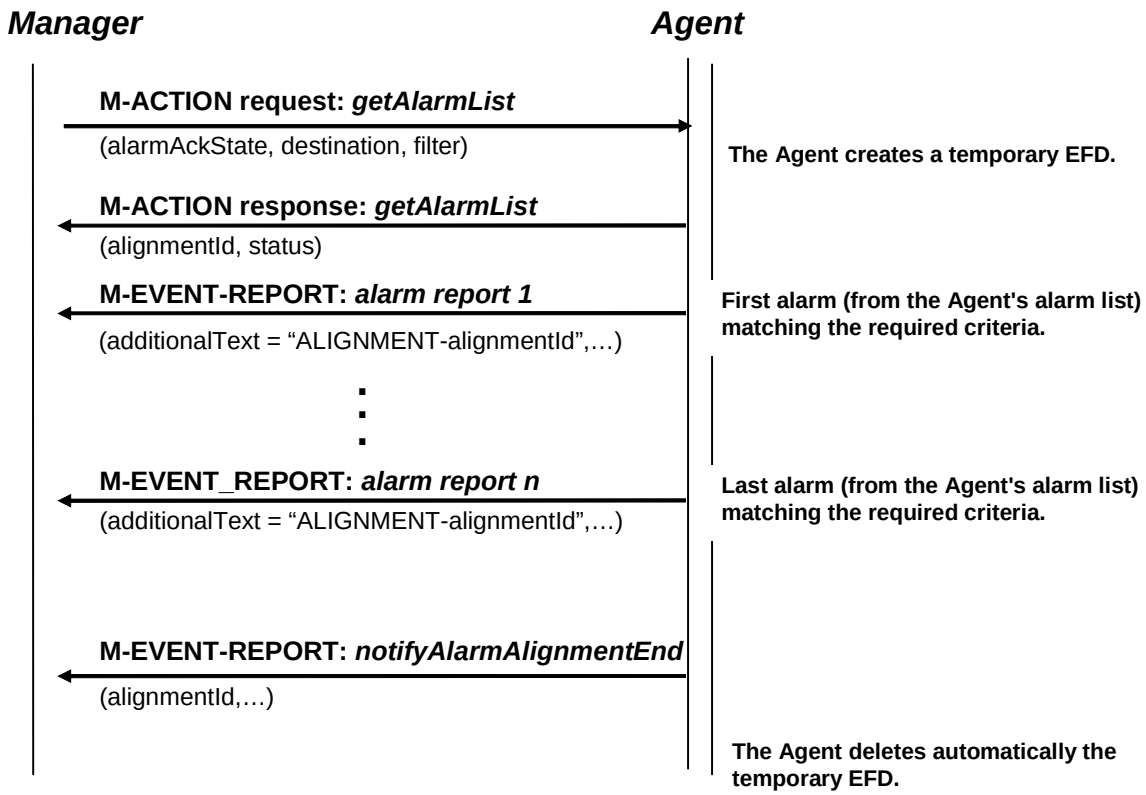


Figure 1: Alignment arrow diagram

Figure 2 shows the handling of a "real-time" alarm notification (occurred during the execution of the *getAlarmList* operation), which is forwarded by the Agent (according to ITU-T Recommendation X.734 [6]) to all currently available EFD instances. Dependent on the *discriminatorConstruct* setting of every EFD, such an alarm may or may not reach the related Manager. In any case, this alarm is filtered out by the temporary EFD assigned to the Manager, which triggered the *getAlarmList* request.

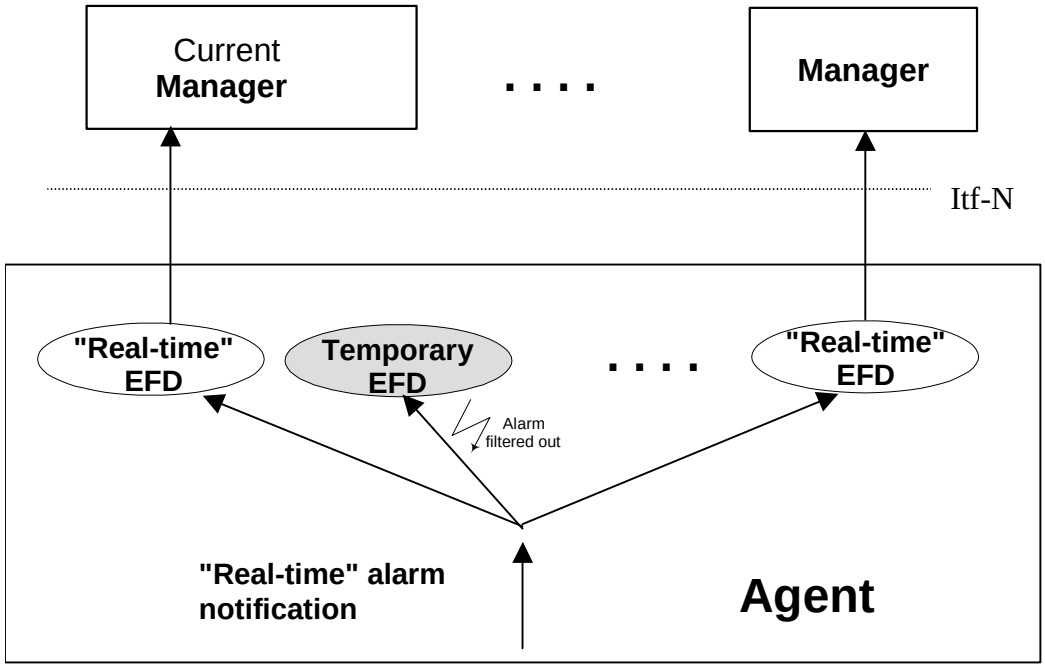


Figure 2: Treatment of "real time" alarms

Figure 3 shows the handling of an alarm notification from the alarm list, matching the criteria defined in the parameters *alarmAckState* of the *getAlarmList* request and forwarded by the Agent to all EFD instances as well. This alarm is filtered out by all EFD instances in charge of discrimination of "real-time" alarms and can reach only the Manager, which triggered the *getAlarmList* request, because it passes the temporary EFD instance assigned to this Manager.

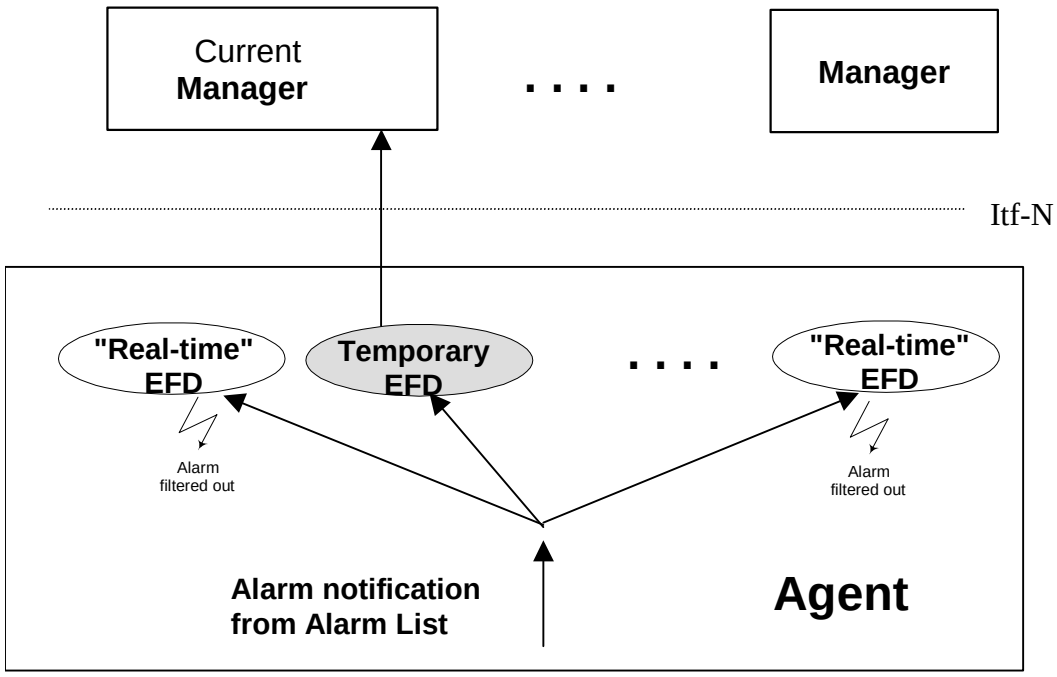


Figure 3: Treatment of "alignment" alarms

It is possible to abort an ongoing alarm alignment process by invoking the action *abortGetAlarmList*. Also in this case the notification *notifyAlarmAlignmentEnd* is emitted.

End of Change in Clause 4.1.6
End of Document

Annex B (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Mar 2000	SA_07	SP-000012	--	--	Approved at TSG SA #7 and placed under Change Control	2.0.0	3.0.0
Mar 2000	--	--	--	--	cosmetic	3.0.0	3.0.1
Jun 2000	SA_08	SP-000254	005	--	Split of TS - Part 4: Alarm Integration Reference Point (IRP): CMIP Solution Set (SS)	3.0.1	3.1.0
Sep 2000	--	--	--	--	cosmetic	3.1.0	3.1.1
Jun 2001	SA_12	SP-010282	001	--	Alarm IRP: CMIP SS Rel4 - Addition of feature. As SA5 had not reviewed this part, it is submitted to SA#12 for Information only.	3.1.1	--
Sep 2001	SA_13	SP-010470	001	1	Addition of features	3.1.1	4.0.0
Dec 2001	SA_14	SP-010640	003	--	Change of qualifier for setComment and notifyComment	4.0.0	4.1.0
Dec 2001	SA_14	SP-010640	004	--	Addition of missing parameter in notifyComments	4.0.0	4.1.0
Mar 2002	SA_15	SP-020028	005	--	Addition of "perceivedSeverity" as parameter to "acknowledgeAlarms" operation (CMIP SS)	4.1.0	4.2.0
Mar 2002	SA_15	--	--	--	Automatic upgrade to Rel-5 (no Rel-5 CR)	4.2.0	5.0.0
Jun 2002	SA_16	SP-020283	007	--	Correction of errors and ambiguities in the Parameter Mapping Tables and ASN.1 Definitions	5.0.0	5.1.0
Jun 2002	SA_16	SP-020284	008	--	Addition of the parameter alarmListAlignmentRequirement to the notification notifyAlarmListRebuilt in the CMIP SS (32.111-4)	5.0.0	5.1.0
Jun 2002	SA_16	SP-020284	009	--	Adding the notification notifyPotentialFaultyAlarmList in the CMIP SS (32.111-4)	5.0.0	5.1.0
Jun 2002	SA_16	SP-020284	010	--	Introduction of SS (32.111-4) to IS (32.111-2) relation and correction of Foreword	5.0.0	5.1.0
Sep 2002	SA_17	SP-020480	011	--	Alignment with 32.111-2 on Alarm Clearance Functionality	5.1.0	5.2.0
Dec 2002	SA_18	SP-020751	013	--	Add the additionalInformation parameter in notifyNewAlarms to the Alarm IRP CMIP SS (Alignment with Information Service in Rel-5 32111-2)	5.2.0	5.3.0
Dec 2002	SA_18	SP-020753	014	--	Addition of Security Alarm Support to the Alarm IRP CMIP SS (Alignment with Information Service in Rel-5 32111-2)	5.2.0	5.3.0
Mar 2003	SA_19	SP-030063	016	--	Correction to Alarm Comments- alignment with 32.111-1	5.3.0	5.4.0
Mar 2003	SA_19	SP-030138	017	--	Add missing x721AlarmNotificationsPackage	5.3.0	5.4.0
Mar 2003	SA_19	SP-030138	018	--	Corrections to GDMO and ASN.1 definitions in the Alarm IRP CMIP SS	5.3.0	5.4.0
Jun 2003	SA_20	SP-030277	019	--	Correction of Compilation Errors	5.4.0	5.5.0
Jun 2003	SA_20	SP-030277	020	--	Addition of missing reasons for the emission of notifyAlarmListRebuilt	5.4.0	5.5.0
Sep 2003	SA_21	SP-030416	022	--	Correction of syntax error in type SetCommentInfo	5.5.0	5.6.0
Dec 2003	SA_22	SP-030627	023	--	Add missing parts for the support of security alarms	5.6.0	5.7.0
Dec 2003	SA_22	SP-030627	024	--	Mapping completion of getAlarmList	5.6.0	5.7.0
Dec 2003	SA_22	SP-030629	025	--	Align operation getAlarmList with the notification notifyAlarmListRebuilt	5.7.0	6.0.0
Jan 2004	--	--	--	--	Editorial (Tables & CMIP code cosmetics)	6.0.0	6.0.1
Mar 2004	SA_23	SP-040120	026	--	Addition of a method to abort an ongoing alarm alignment process in the asynchronous mode of the operation getAlarmList	6.0.1	6.1.0
Sep 2004	SA_25	SP-040561	028	--	Align with the IS 32.111-2 the possibility to apply filters to notification parameters	6.1.0	6.2.0
Dec 2004	SA_26	SP-040791	029	--	Remove redundant ackTime parameter in notifyAckStateChanged	6.2.0	6.3.0
Mar 2005	SA_27	SP-050021	031	--	Add missing definition of getAlarmList return value - Align with the IS (TS 32.111-2)	6.3.0	6.4.0

CR-Form-v7.1

CHANGE REQUEST

⌘ **32.111-5** **CR** **0001** ⌘**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:	⌘	Align with 32.335
Source:	⌘	SA5
Work item code:	⌘	OAM-NIM
Date:	⌘	13/05/2005
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:

Ph2	*(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)*
Rel-7	*(Release 7)*

Reason for change:	⌘	To align 32.111-5 with 32.335 and correct some editorial errors
Summary of change:	⌘	1. "ObjectClass" in figures of section 4.1 Alarm IRP XML Definitions are removed. 2. "SystemDN" in figures of section 4.1 Alarm IRP XML Definitions are changed to reflect "minOccors=0" attribute. 3. Type of element "alarmId" should be "string" instead of "long" 4. Type of element "perceivedSeverity" should be "PerceivedSeverity" instead of "ProbableCause". 5. Type of element "notification-id refers to "xe:NotificationID" defined in 32.305 to avoid duplicated definition. 6. Element "body" of complexType "NotifyClearedAlarm" should refers to element definition of "probableCause", "perceivedSeverity", "alarmType", "correlatedNotifications", and "alarmId" to avoid duplicated definitions. 7. Attribute substitutionGroup of element "NotifyAckStateChanged", "NotifyAlarmListRebuilt", "NotifyChangedAlarm", "NotifyClearedAlarm", "NotifyComments", "NotifyNewAlarm", and "NotifyPotentialFaultAlarmList" removed. 8. change namespace of "http://www.w3.org/2001/XMLSchema" to be the default namespace of this XML schema. 9. change namespace of "http://www.3gpp.org/ftp/specs/latest/rel-6/32_series/32111-5#alarmIRPNotif" to be xmlns:xa namespace of this XML schema.
Consequences if not approved:	⌘	32.111-5 won't align with 32.335.

Clauses affected:	⌘	4.1, 4.2

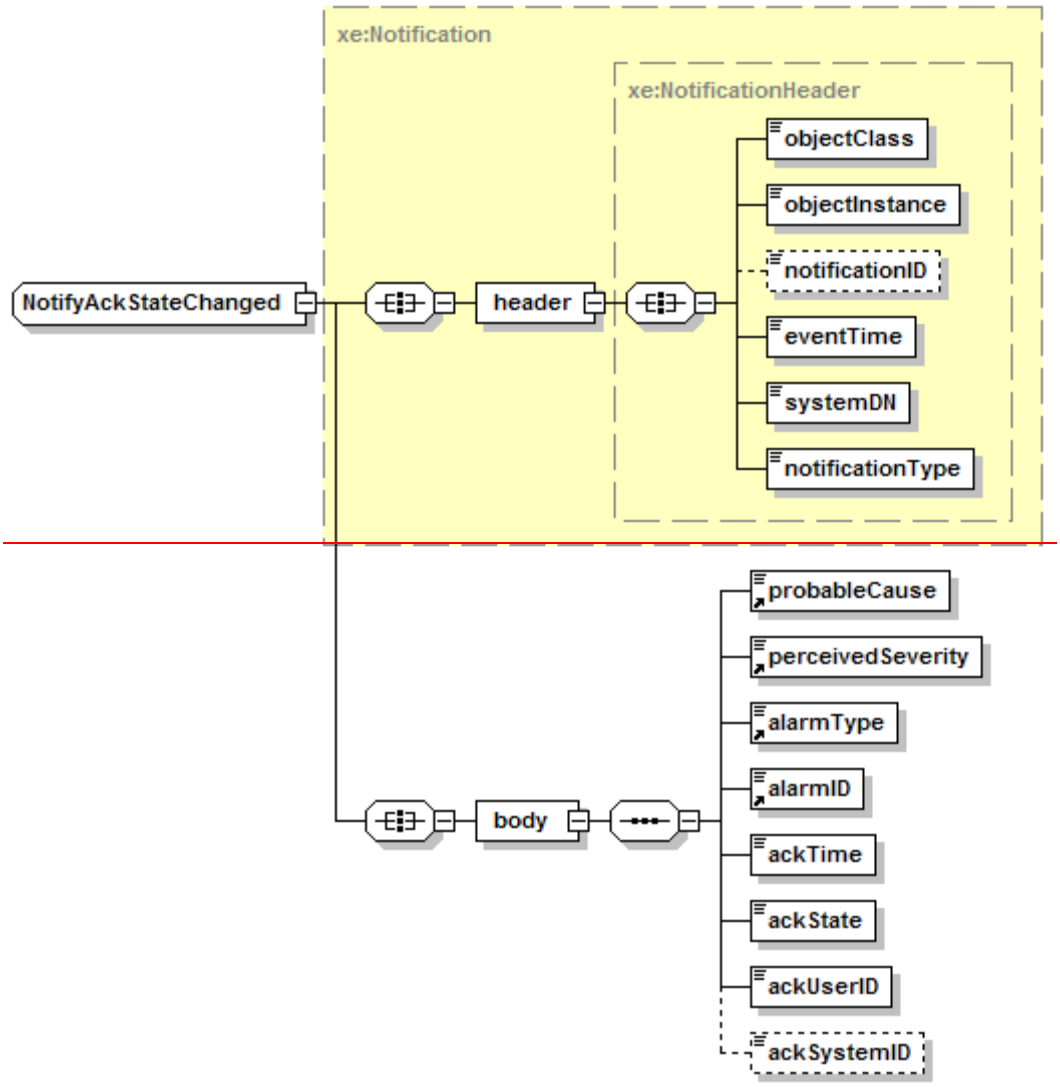
Other specs affected:	⌘	Y	N		⌘
				Other core specifications	
				Test specifications	
				O&M Specifications	

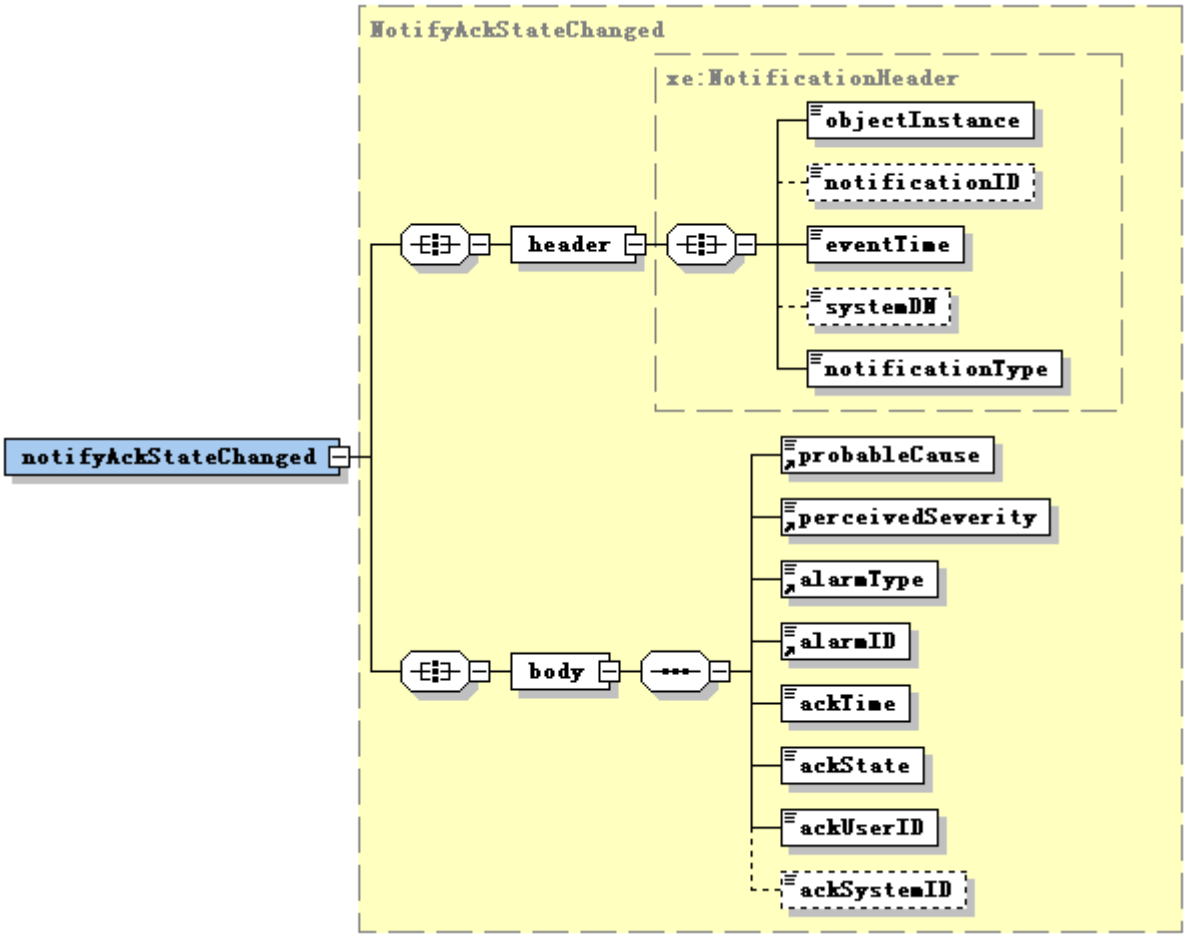
Other comments:	⌘	

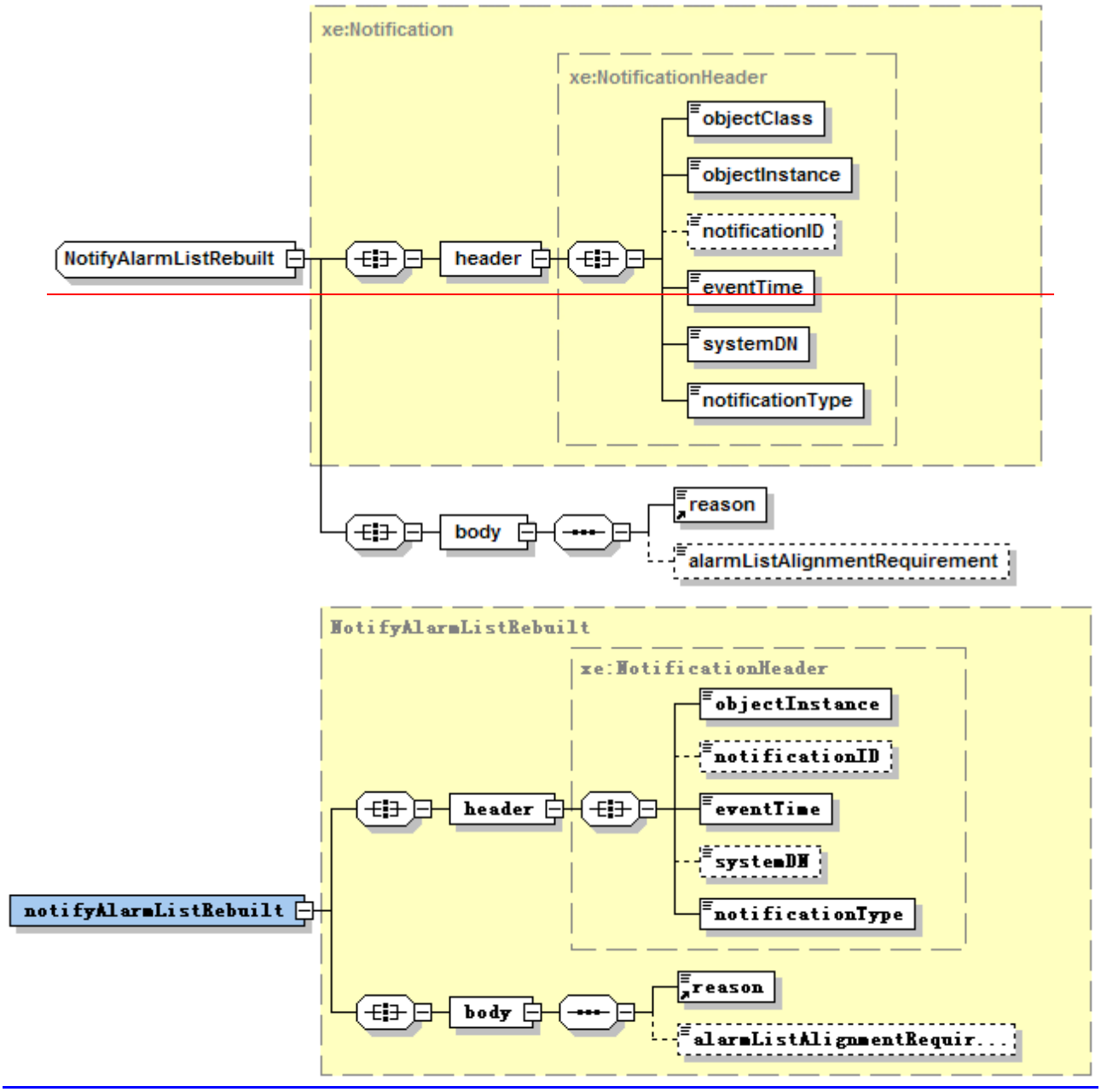
Change in Clause 4.1

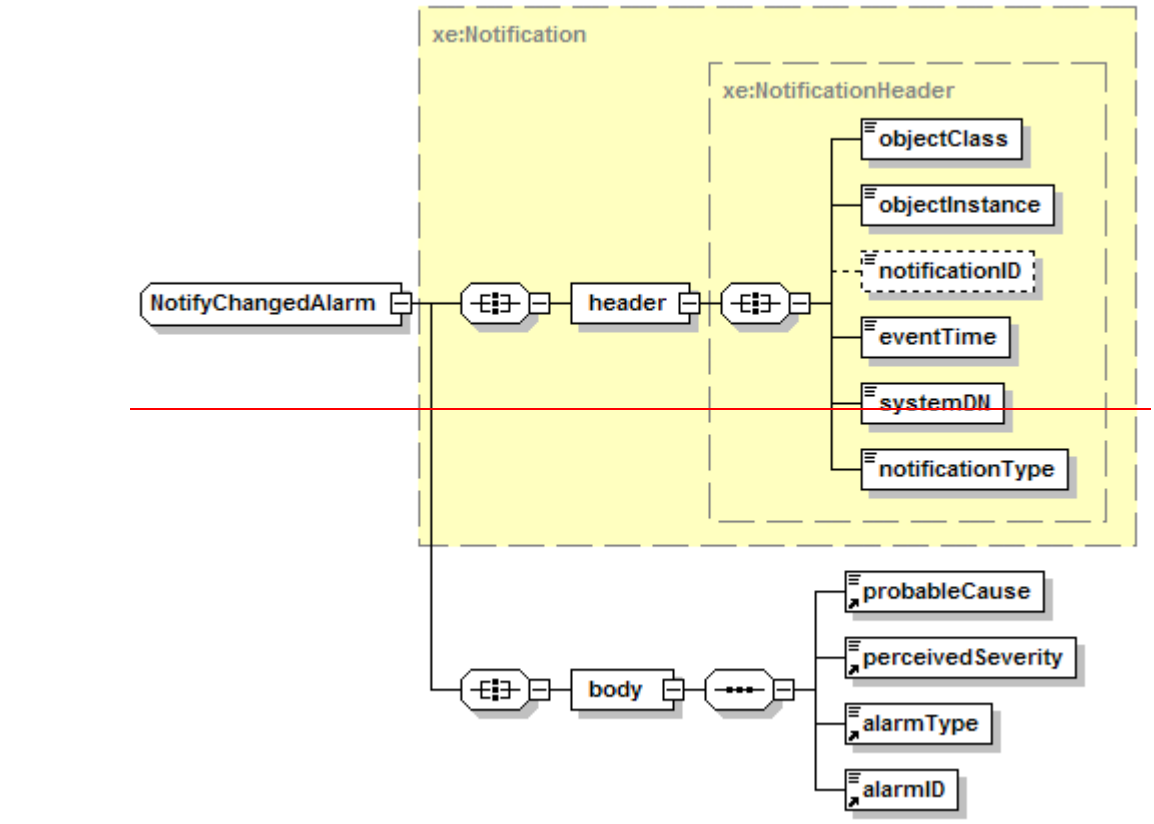
4.1 Alarm IRP XML Definitions

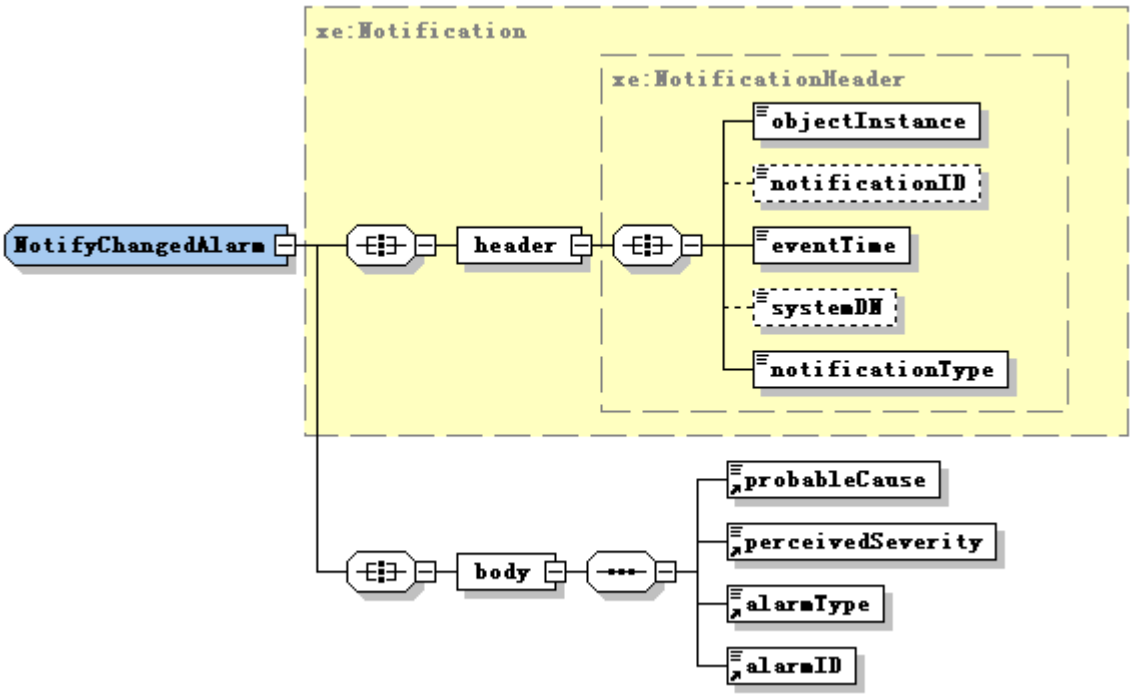
This section provides XML definitions of Alarm IRP notifications as defined in [3]. These definitions are to be used in conjunction with Notification Log IRP XML Definitions for Notification Log IRP XML Data File and the NL IRP XML Notification Format [4], as well as considerations for NL IRP XML File Name Conventions defined therein.

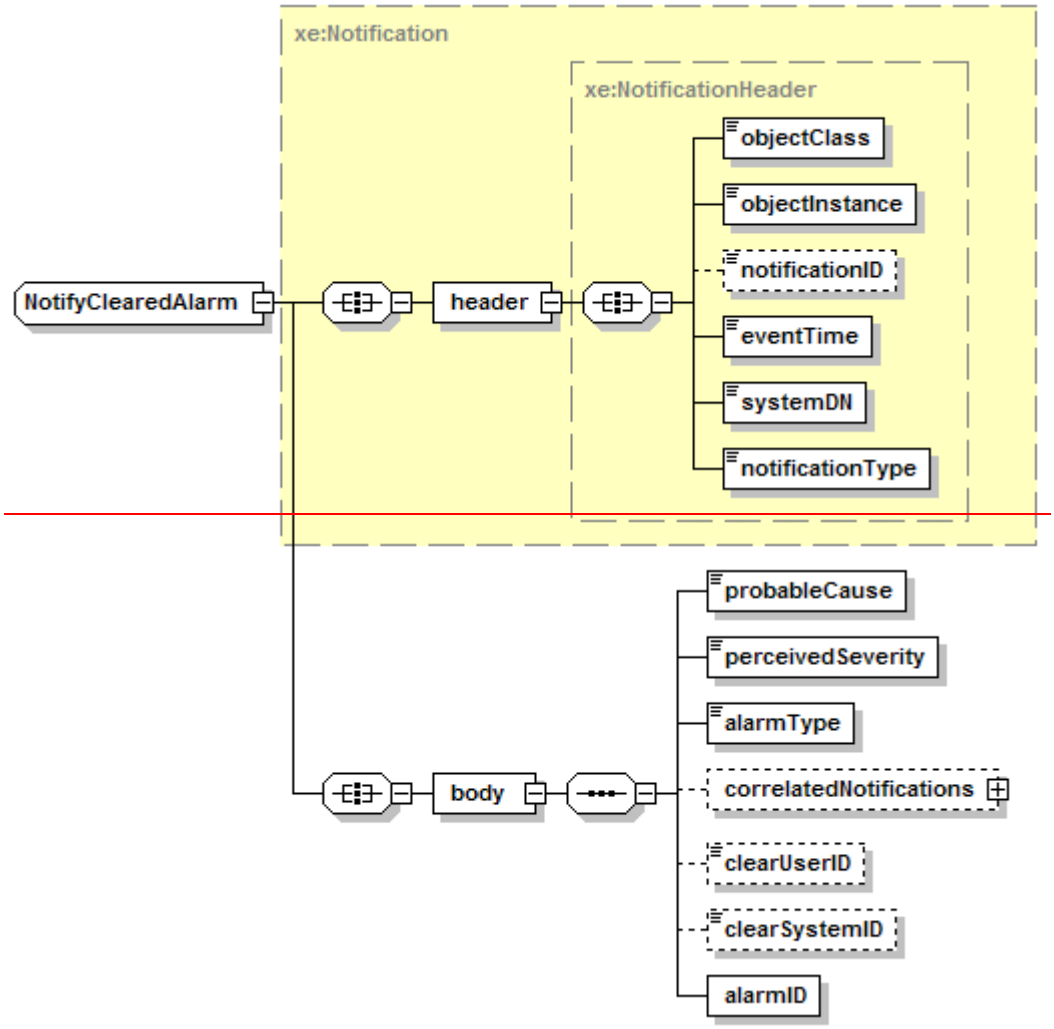

xe:Notification
xe:NotificationHeader
objectClass
objectInstance
notificationID
eventTime
systemDN
notificationType
NotifyAckStateChanged
header
probableCause
perceivedSeverity
alarmType
alarmID
body
ackTime
ackState
ackUserID
ackSystemID

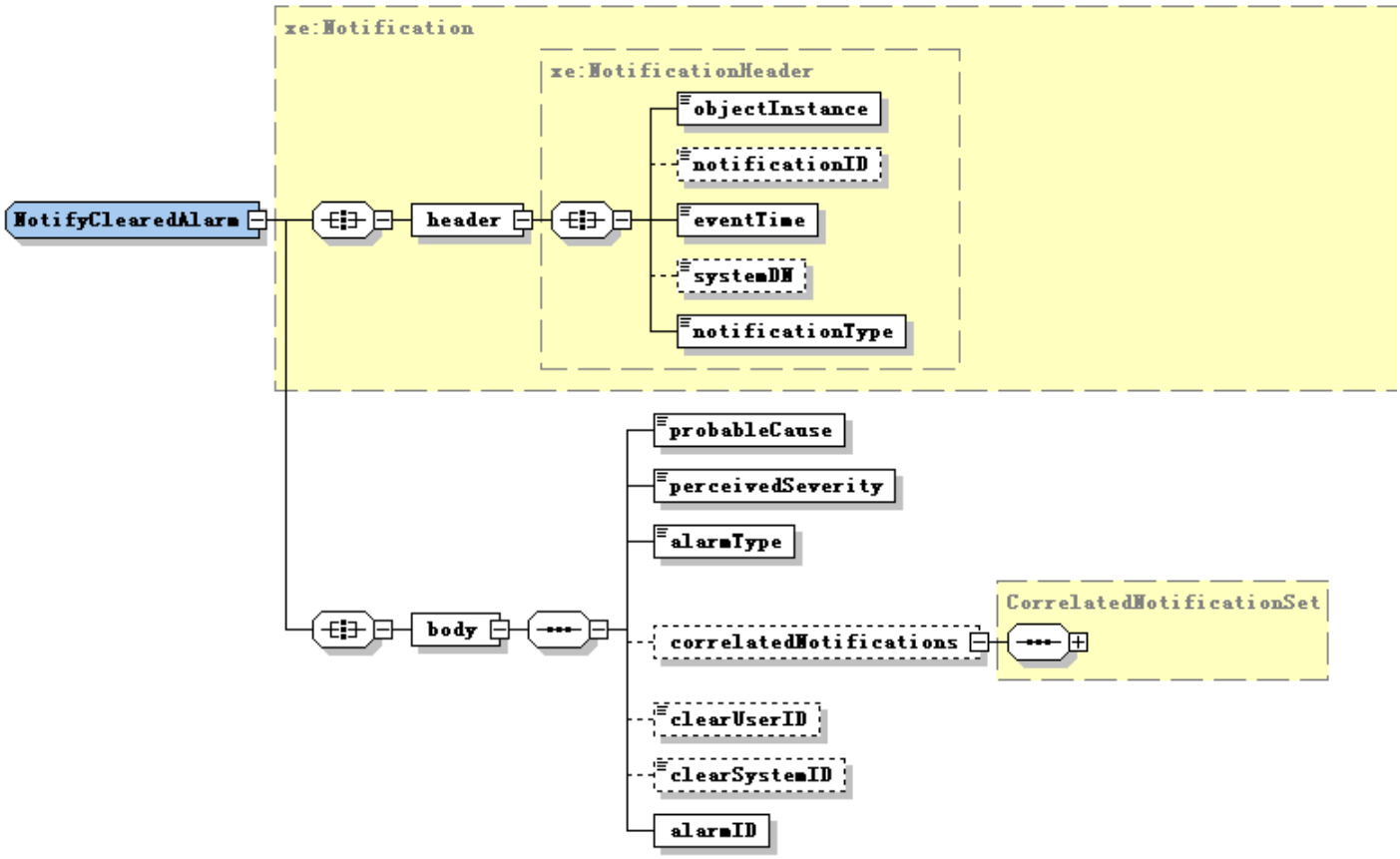

NotifyAckStateChanged
xe:NotificationHeader
objectInstance
notificationID
eventTime
systemDN
notificationType
header
notifyAckStateChanged
probableCause
perceivedSeverity
alarmType
alarmID
ackTime
ackState
ackUserID
ackSystemID
body

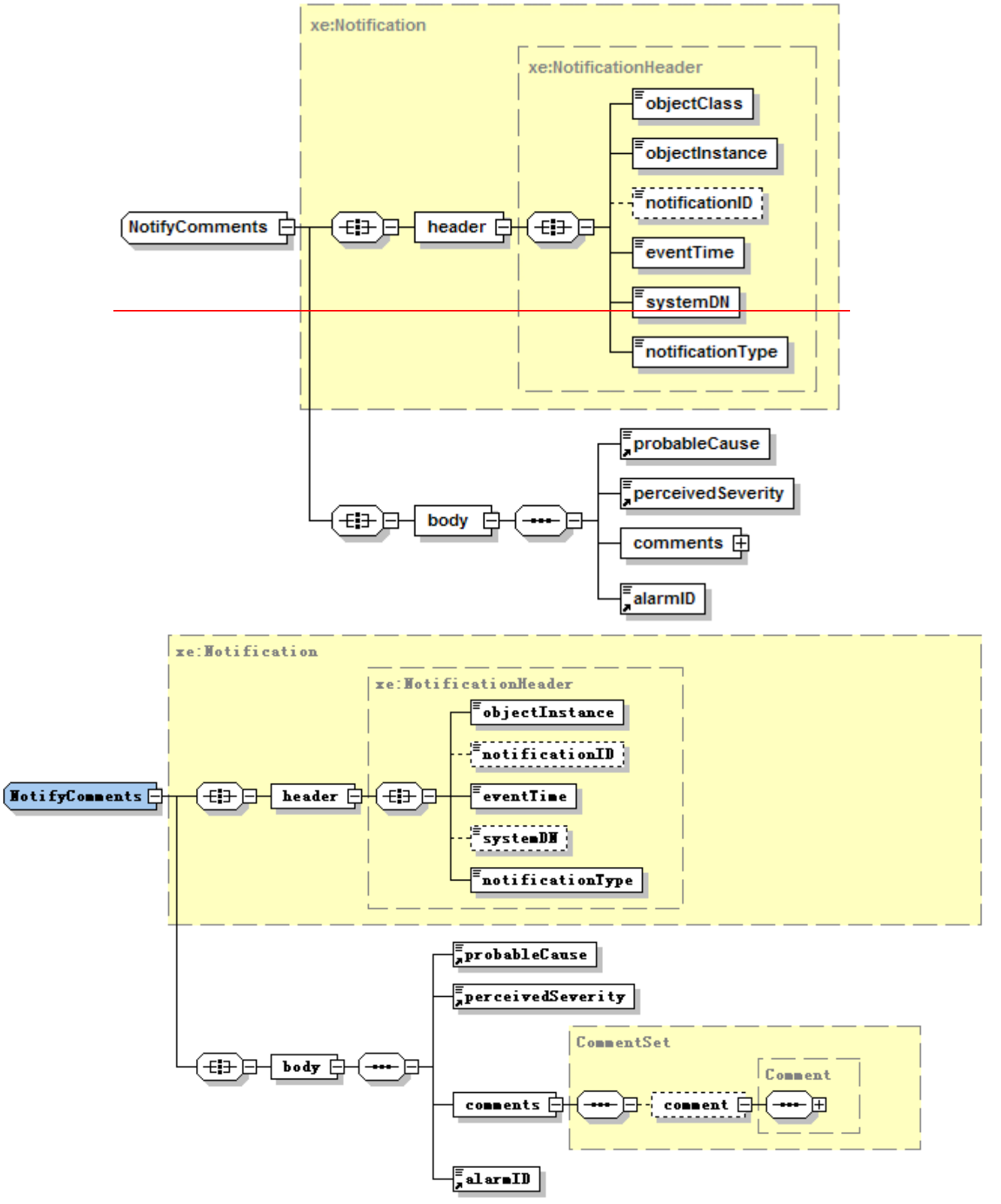

xe:Notification
xe:NotificationHeader
objectClass
objectInstance
notificationID
eventTime
systemDN
notificationType
NotifyAlarmListRebuilt
header
body
reason
alarmListAlignmentRequirement
NotifyAlarmListRebuilt
xe:NotificationHeader
objectInstance
notificationID
eventTime
systemDN
notificationType
notifyAlarmListRebuilt
header
body
reason
alarmListAlignmentRequir...

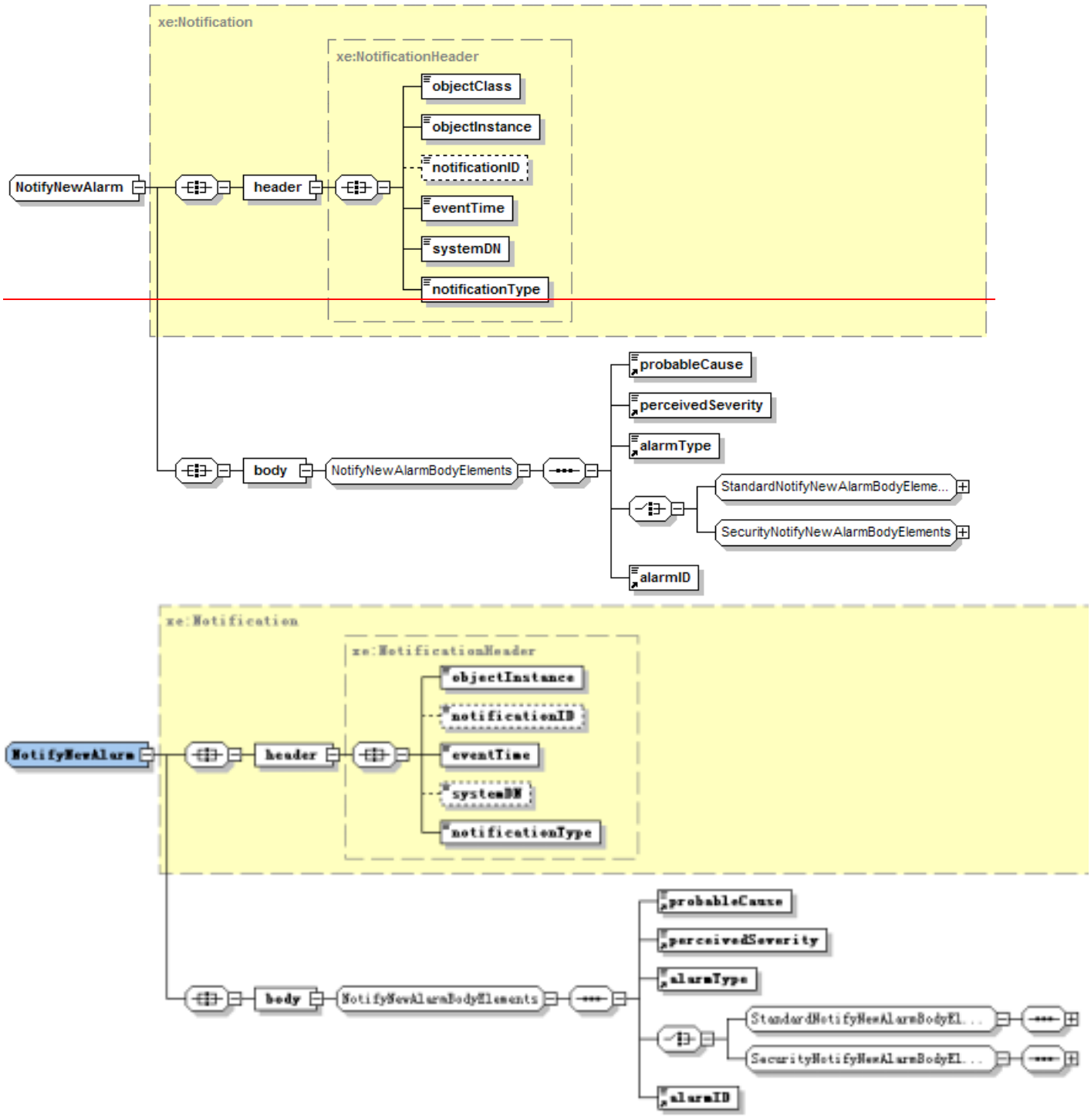

xe:Notification
xe:NotificationHeader
objectClass
objectInstance
notificationID
eventTime
systemDN
notificationType
NotifyChangedAlarm
header
body
probableCause
perceivedSeverity
alarmType
alarmID


xe:Notification
xe:NotificationHeader
objectInstance
notificationID
eventTime
systemDN
notificationType
NotifyChangedAlarm
header
body
probableCause
perceivedSeverity
alarmType
alarmID

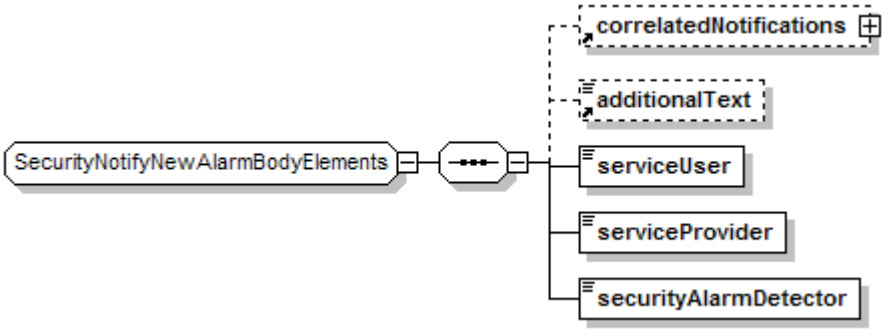

xe:Notification
xe:NotificationHeader
objectClass
objectInstance
notificationID
eventTime
systemDN
notificationType
NotifyClearedAlarm
header
probableCause
perceivedSeverity
alarmType
body
correlatedNotifications
clearUserID
clearSystemID
alarmID

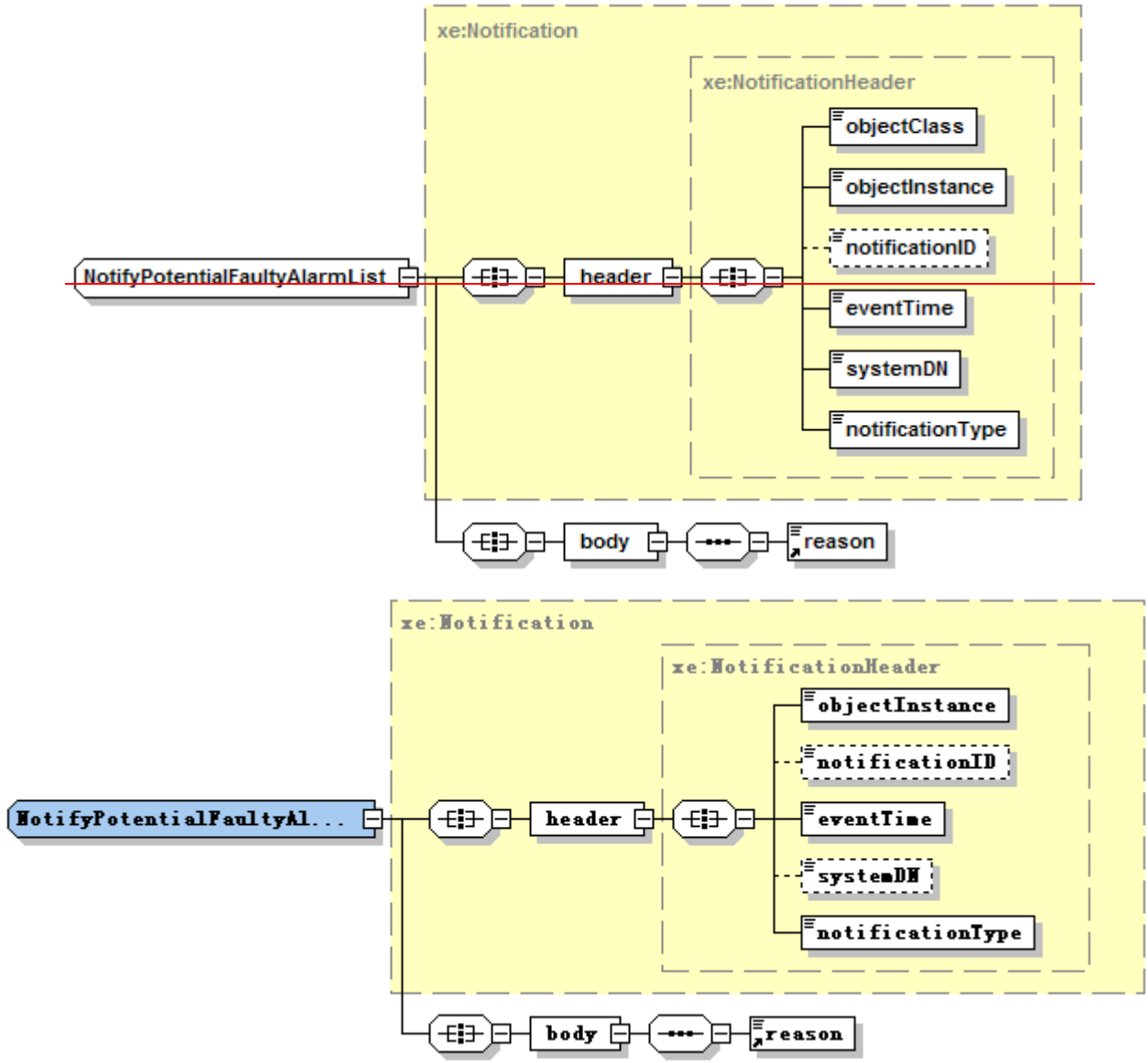

xe:Notification
xe:NotificationHeader
objectInstance
notificationID
eventTime
systemDN
notificationType
NotifyClearedAlarm
header
probableCause
perceivedSeverity
alarmType
CorrelatedNotificationSet
body
correlatedNotifications
clearUserID
clearSystemID
alarmID


xe:Notification
xe:NotificationHeader
objectClass
objectInstance
notificationID
eventTime
systemDN
notificationType
NotifyComments
header
body
probableCause
perceivedSeverity
comments
alarmID
xe:Notification
xe:NotificationHeader
objectInstance
notificationID
eventTime
systemDN
notificationType
NotifyComments
header
body
probableCause
perceivedSeverity
CommentSet
Comment
comments
comment
alarmID


xe:Notification
xe:NotificationHeader
objectClass
objectInstance
notificationID
eventTime
systemDN
notificationType
NotifyNewAlarm
header
body
NotifyNewAlarmBodyElements
probableCause
perceivedSeverity
alarmType
StandardNotifyNewAlarmBodyEleme...
SecurityNotifyNewAlarmBodyElements
alarmID
xe:Notification
xe:NotificationHeader
objectInstance
notificationID
eventTime
systemDN
notificationType
NotifyNewAlarm
header
body
NotifyNewAlarmBodyElements
probableCause
perceivedSeverity
alarmType
StandardNotifyNewAlarmBodyEl...
SecurityNotifyNewAlarmBodyEl...
alarmID


StandardNotifyNewAlarmBodyEleme...
specificProblem
correlatedNotifications
backedUpStatus
backupObject
trendIndication
thresholdInfo
stateChangeDefinition
monitoredAttributes
proposedRepairActions
additionalText
additionalInformation


SecurityNotifyNewAlarmBodyElements
correlatedNotifications
additionalText
serviceUser
serviceProvider
securityAlarmDetector



End of Change in Clause 4.1

Change in Clause 4.2

4.2 Alarm IRP XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
 targetNamespace="http://www.3gpp.org/ftp/specs/latest/rel-6/32_series/32111-5-600.zip#alarmIRPNotif"
 elementFormDefault="qualified" attributeFormDefault="unqualified"
 xmlns:xa="http://www.3gpp.org/ftp/specs/latest/rel-6/32_series/32111-5-600.zip#alarmIRPNotif"
 xmlns:xe="http://www.3gpp.org/ftp/specs/latest/rel-6/32_series/323305-600.zip#notification"
 xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:import
   namespace="http://www.3gpp.org/ftp/specs/latest/rel-6/32_series/323305-600.zip#notification"/>

  <xs:simpleType name="AckState">
    <xs:restriction base="xs:string">
      <xs:enumeration value="ACKNOWLEDGED"/>
      <xs:enumeration value="UNACKNOWLEDGED"/>
```

```
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="AlarmListAlignmentRequirement">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Alignment required"/>
      <xs:enumeration value="Alignment not required"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="AlarmListRebuiltReason">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Agent-NE communications error"/>
      <xs:enumeration value="Agent restarts"/>
      <xs:enumeration value="Indeterminate"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="EventType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Communications Alarm"/>
      <xs:enumeration value="Processing Error Alarm"/>
      <xs:enumeration value="Environmental Alarm"/>
      <xs:enumeration value="Quality of Service Alarm"/>
      <xs:enumeration value="Equipment Alarm"/>
      <xs:enumeration value="Integrity Violation"/>
      <xs:enumeration value="Operational Violation"/>
      <xs:enumeration value="Physical Violation"/>
      <xs:enumeration value="Security Service or Mechanism Violation"/>
      <xs:enumeration value="Time Domain Violation"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="PerceivedSeverity">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Critical"/>
      <xs:enumeration value="Major"/>
      <xs:enumeration value="Minor"/>
      <xs:enumeration value="Warning"/>
      <xs:enumeration value="Indeterminate"/>
      <xs:enumeration value="Cleared"/>
      <xs:enumeration value=""/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ProbableCause">
    <xs:restriction base="xs:string">
      <xs:enumeration value="INDETERMINATE"/>
      <xs:enumeration value="ALARM INDICATION SIGNAL"/>
      <xs:enumeration value="CALL SETUP FAILURE"/>
      <xs:enumeration value="DEGRADED SIGNAL M3100"/>
      <xs:enumeration value="FAR END RECEIVER FAILURE"/>
      <xs:enumeration value="FRAMING ERROR M3100"/>
      <xs:enumeration value="LOSS OF FRAME"/>
      <xs:enumeration value="LOSS OF POINTER"/>
      <xs:enumeration value="LOSS OF SIGNAL"/>
      <xs:enumeration value="PAYLOAD TYPE MISMATCH"/>
      <xs:enumeration value="TRANSMISSION ERROR"/>
      <xs:enumeration value="REMOTE ALARM INTERFACE"/>
      <xs:enumeration value="EXCESSIVE BIT ERROR RATE"/>
      <xs:enumeration value="PATH TRACE MISMATCH"/>
      <xs:enumeration value="UNAVAILABLE"/>
      <xs:enumeration value="SIGNAL LABEL MISMATCH"/>
      <xs:enumeration value="LOSS OF MULTI FRAME"/>
      <xs:enumeration value="COMMUNICATIONS RECEIVE FAILURE"/>
      <xs:enumeration value="COMMUNICATIONS TRANSMIT FAILURE"/>
      <xs:enumeration value="MODULATION FAILURE"/>
      <xs:enumeration value="DEMODULATION FAILURE"/>
      <xs:enumeration value="BACK PLANE FAILURE"/>
      <xs:enumeration value="DATA SET PROBLEM"/>
      <xs:enumeration value="EQUIPMENT IDENTIFIER DUPLICATION"/>
      <xs:enumeration value="EXTERNAL DEVICE PROBLEM"/>
      <xs:enumeration value="LINE CARD PROBLEM"/>
      <xs:enumeration value="MULTIPLEXER PROBLEM M3100"/>
      <xs:enumeration value="NE IDENTIFIER DUPLICATION"/>
      <xs:enumeration value="POWER PROBLEM M3100"/>
      <xs:enumeration value="PROCESSOR PROBLEM M3100"/>
      <xs:enumeration value="PROTECTION PATH FAILURE"/>
      <xs:enumeration value="RECEIVER FAILURE M3100"/>
      <xs:enumeration value="REPLACEABLE UNIT MISSING"/>
      <xs:enumeration value="REPLACEABLE UNIT TYPE MISMATCH"/>
      <xs:enumeration value="SYNCHRONISATION SOURCE MISMATCH"/>
      <xs:enumeration value="TERMINAL PROBLEM"/>
```

```
<xs:enumeration value="TIMING PROBLEM M3100"/>
<xs:enumeration value="TRANSMITTER FAILURE M3100"/>
<xs:enumeration value="TRUNK CARD PROBLEM"/>
<xs:enumeration value="REPLACEABLE UNIT PROBLEM"/>
<xs:enumeration value="REAL TIME CLOCK FAILURE"/>
<xs:enumeration value="PROTECTION MECHANISM FAILURE"/>
<xs:enumeration value="PROTECTING RESOURCE FAILURE"/>
<xs:enumeration value="AIR COMPRESSOR FAILURE"/>
<xs:enumeration value="AIR CONDITIONING FAILURE"/>
<xs:enumeration value="AIR DRYER FAILURE"/>
<xs:enumeration value="BATTERY DISCHARGING"/>
<xs:enumeration value="BATTERY FAILURE"/>
<xs:enumeration value="COMMERICAL POWER FAILURE"/>
<xs:enumeration value="COOLING FAN FAILURE"/>
<xs:enumeration value="ENGINE FAILURE"/>
<xs:enumeration value="FIRE DETECTOR FAILURE"/>
<xs:enumeration value="FUSE FAILURE"/>
<xs:enumeration value="GENERATOR FAILURE"/>
<xs:enumeration value="LOW BATTERY THRESHOLD"/>
<xs:enumeration value="PUMP FAILURE M3100"/>
<xs:enumeration value="RECTIFIER FAILURE"/>
<xs:enumeration value="RECTIFIER HIGH VOLTAGE"/>
<xs:enumeration value="RECTIFIER LOW F VOLTAGE"/>
<xs:enumeration value="VENTILATION SYSTEM FAILURE"/>
<xs:enumeration value="ENCLOSURE DOOR OPEN M3100"/>
<xs:enumeration value="EXPLOSIVE GAS"/>
<xs:enumeration value="FIRE"/>
<xs:enumeration value="FLOOD"/>
<xs:enumeration value="HIGH HUMIDITY"/>
<xs:enumeration value="HIGH TEMPERATURE"/>
<xs:enumeration value="HIGH WIND"/>
<xs:enumeration value="ICE BUILD UP"/>
<xs:enumeration value="INTRUSION DETECTION"/>
<xs:enumeration value="LOW FUEL"/>
<xs:enumeration value="LOW HUMIDITY"/>
<xs:enumeration value="LOW CABLE PRESSURE"/>
<xs:enumeration value="LOW TEMPERATURE"/>
<xs:enumeration value="LOW WATER"/>
<xs:enumeration value="SMOKE"/>
<xs:enumeration value="TOXIC GAS"/>
<xs:enumeration value="EXTERNAL POINT FAILURE"/>
<xs:enumeration value="STORAGE CAPACITY PROBLEM M3100"/>
<xs:enumeration value="MEMORY MISMATCH"/>
<xs:enumeration value="CORRUPT DATA M3100"/>
<xs:enumeration value="OUT OF CPU CYCLES"/>
<xs:enumeration value="SOFTWARE ENVIRONMENT PROBLEM"/>
<xs:enumeration value="SOFTWARE DOWNLOAD FAILURE"/>
<xs:enumeration value="LOSS OF REAL TIME"/>
<xs:enumeration value="REINITIALIZED"/>
<xs:enumeration value="EXCESSIVE ERROR RATE"/>
<xs:enumeration value="ADAPTER ERROR"/>
<xs:enumeration value="APPLICATION SUBSYSTEM FAILURE"/>
<xs:enumeration value="BANDWIDTH REDUCTION"/>
<xs:enumeration value="COMMUNICATION PROTOCOL ERROR"/>
<xs:enumeration value="COMMUNICATION SUBSYSTEM FAILURE"/>
<xs:enumeration value="CONFIGURATION OR CUSTOMIZING ERROR"/>
<xs:enumeration value="CONGESTION"/>
<xs:enumeration value="CPU CYCLES LIMIT EXCEEDED"/>
<xs:enumeration value="DATA SET OR MODEM ERROR"/>
<xs:enumeration value="DTE DCE INTERFACE ERROR"/>
<xs:enumeration value="EQUIPMENT MALFUNCTION"/>
<xs:enumeration value="EXCESSIVE VIBRATION"/>
<xs:enumeration value="FILE ERROR"/>
<xs:enumeration value="HEATING OR VENTILATION OR COOLING SYSTEM PROBLEM"/>
<xs:enumeration value="HUMIDITY UNACCEPTABLE"/>
<xs:enumeration value="INPUT OUTPUT DEVICE ERROR"/>
<xs:enumeration value="INPUT DEVICE ERROR"/>
<xs:enumeration value="LAN ERROR"/>
<xs:enumeration value="LEAK DETECTION"/>
<xs:enumeration value="LOCAL NODE TRANSMISSION ERROR"/>
<xs:enumeration value="MATERIAL SUPPLY EXHAUSTED"/>
<xs:enumeration value="OUT OF MEMORY"/>
<xs:enumeration value="OUTPUT DEVICE ERROR"/>
<xs:enumeration value="PERFORMANCE DEGRADED"/>
<xs:enumeration value="PRESSURE UNACCEPTABLE"/>
<xs:enumeration value="QUEUE SIZE EXCEEDED"/>
<xs:enumeration value="RECEIVE FAILURE"/>
<xs:enumeration value="REMOTE NODE TRANSMISSION ERROR"/>
```

```
<xs:enumeration value="RESOURCE AT OR NEARING CAPACITY"/>
<xs:enumeration value="RESPONSE TIME EXCESSIVE"/>
<xs:enumeration value="RETRANSMISSION RATE EXCESSIVE"/>
<xs:enumeration value="SOFTWARE ERROR"/>
<xs:enumeration value="SOFTWARE PROGRAM ABNORMALLY TERMINATED"/>
<xs:enumeration value="SOFTWARE PROGRAM ERROR"/>
<xs:enumeration value="TEMPERATURE UNACCEPTABLE"/>
<xs:enumeration value="THRESHOLD CROSSED"/>
<xs:enumeration value="TOXIC LEAK DETECTED"/>
<xs:enumeration value="TRANSMIT FAILURE"/>
<xs:enumeration value="UNDERLYING RESOURCE UNAVAILABLE"/>
<xs:enumeration value="VERSION MISMATCH"/>
<xs:enumeration value="A BIS TO BTS INTERFACE FAILURE"/>
<xs:enumeration value="A BIS TO TRX INTERFACE FAILURE"/>
<xs:enumeration value="ANTENNA PROBLEM"/>
<xs:enumeration value="BATTERY BREAKDOWN"/>
<xs:enumeration value="BATTERY CHARGING FAULT"/>
<xs:enumeration value="CLOCK SYNCHRONISATION PROBLEM"/>
<xs:enumeration value="COMBINER PROBLEM"/>
<xs:enumeration value="DISK PROBLEM"/>
<xs:enumeration value="EXCESSIVE RECEIVER TEMPERATURE"/>
<xs:enumeration value="EXCESSIVE TRANSMITTER OUTPUT POWER"/>
<xs:enumeration value="EXCESSIVE TRANSMITTER TEMPERATURE"/>
<xs:enumeration value="FREQUENCY HOPPING DEGRADED"/>
<xs:enumeration value="FREQUENCY HOPPING FAILURE"/>
<xs:enumeration value="FREQUENCY REDEFINITION FAILED"/>
<xs:enumeration value="LINE INTERFACE FAILURE"/>
<xs:enumeration value="LINK FAILURE"/>
<xs:enumeration value="LOSS OF SYNCHRONISATION"/>
<xs:enumeration value="LOST REDUNDANCY"/>
<xs:enumeration value="MAINS BREAKDOWN WITH BATTERY BACKUP"/>
<xs:enumeration value="MAINS BREAKDOWN WITHOUT BATTERY BACKUP"/>
<xs:enumeration value="POWER SUPPLY FAILURE"/>
<xs:enumeration value="RECEIVER ANTENNA FAULT"/>
<xs:enumeration value="RECEIVER MULTICOUPLER FAILURE"/>
<xs:enumeration value="REDUCED TRANSMITTER OUTPUT POWER"/>
<xs:enumeration value="SIGNAL QUALITY EVALUATION FAULT"/>
<xs:enumeration value="TIMESLOT HARDWARE FAILURE"/>
<xs:enumeration value="TRANSCEIVER PROBLEM"/>
<xs:enumeration value="TRANSCODER PROBLEM"/>
<xs:enumeration value="TRANSCODER OR RATE ADAPTER PROBLEM"/>
<xs:enumeration value="TRANSMITTER ANTENNA FAILURE"/>
<xs:enumeration value="TRANSMITTER ANTENNA NOT ADJUSTED"/>
<xs:enumeration value="TRANSMITTER LOW VOLTAGE OR CURRENT"/>
<xs:enumeration value="TRANSMITTER OFF FREQUENCY"/>
<xs:enumeration value="DATABASE INCONSISTENCY"/>
<xs:enumeration value="FILE SYSTEM CALL UNSUCCESSFUL"/>
<xs:enumeration value="INPUT PARAMETER OUT OF RANGE"/>
<xs:enumeration value="INVALID PARAMETER"/>
<xs:enumeration value="INVALID POINTER"/>
<xs:enumeration value="MESSAGE NOT EXPECTED"/>
<xs:enumeration value="MESSAGE NOT INITIALISED"/>
<xs:enumeration value="MESSAGE OUT OF SEQUENCE"/>
<xs:enumeration value="SYSTEM CALL UNSUCCESSFUL"/>
<xs:enumeration value="TIMEOUT EXPIRED"/>
<xs:enumeration value="VARIABLE OUT OF RANGE"/>
<xs:enumeration value="WATCH DOG TIMER EXPIRED"/>
<xs:enumeration value="COOLING SYSTEM FAILURE"/>
<xs:enumeration value="EXTERNAL EQUIPMENT FAILURE"/>
<xs:enumeration value="EXTERNAL POWER SUPPLY FAILURE"/>
<xs:enumeration value="EXTERNAL TRANSMISSION DEVICE FAILURE"/>
<xs:enumeration value="REDUCED ALARM REPORTING"/>
<xs:enumeration value="REDUCED EVENT REPORTING"/>
<xs:enumeration value="RECUCED LOGGING CAPABILITY"/>
<xs:enumeration value="SYSTEM RESOURCES OVERLOAD"/>
<xs:enumeration value="BROADCAST CHANNEL FAILURE"/>
<xs:enumeration value="CALL ESTABLISHMENT ERROR"/>
<xs:enumeration value="INVALID MESSAGE RECEIVED"/>
<xs:enumeration value="INVALID MSU RECEIVED"/>
<xs:enumeration value="LAPD LINK PROTOCOL FAILURE"/>
<xs:enumeration value="LOCAL ALARM INDICATION"/>
<xs:enumeration value="REMOTE ALARM INDICATION"/>
<xs:enumeration value="ROUTING FAILURE"/>
<xs:enumeration value="SS7 PROTOCOL FAILURE"/>
<xs:enumeration value="TRANSMISSION FAILURE"/>
<xs:enumeration value="AUTHENTICATION FAILURE"/>
<xs:enumeration value="BREACH OF CONFIDENTIALITY"/>
<xs:enumeration value="CABLE TAMPER"/>
```

```
      <xs:enumeration value="DELAYED INFORMATION"/>
      <xs:enumeration value="DENIAL OF SERVICE"/>
      <xs:enumeration value="DUPLICATE INFORMATION"/>
      <xs:enumeration value="INFORMATION MISSING"/>
      <xs:enumeration value="INFORMATION MODIFICATION DETECTED"/>
      <xs:enumeration value="INFORMATION OUT OF SEQUENCE"/>
      <xs:enumeration value="KEY EXPIRED"/>
      <xs:enumeration value="NON REPUDIATION FAILURE"/>
      <xs:enumeration value="OUT OF HOURS ACTIVITY"/>
      <xs:enumeration value="OUT OF SERVICE"/>
      <xs:enumeration value="PROCEDURAL ERROR"/>
      <xs:enumeration value="UNAUTHORISED ACCESS ATTEMPT"/>
      <xs:enumeration value="UNEXPECTED INFORMATION"/>
      <xs:enumeration value="UNSPECIFIED REASON"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ThresholdIndicator">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UP"/>
      <xs:enumeration value="DOWN"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="TrendIndicator">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Less Severe"/>
      <xs:enumeration value="No Change"/>
      <xs:enumeration value="More Severe"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="additionalInformation" type="xs:string"/>
  <xs:element name="additionalText" type="xs:string"/>
  <xs:element name="alarmID" type="xs:longstring"/>
  <xs:element name="alarmType" type="xa:EventType"/>
  <xs:element name="correlatedNotifications" type="xa:CorrelatedNotificationSet"/>
  <xs:element name="perceivedSeverity" type="xa:PerceivedSeverityProbableCause"/>
  <xs:element name="probableCause" type="xa:ProbableCause"/>
  <xs:element name="reason" type="xa:AlarmListRebuiltReason"/>
  <xs:complexType name="AttributeValue">
    <xs:all>
      <xs:element name="attribute-name" type="xs:string"/>
      <xs:element name="attribute-value" type="xs:anySimpleType"/>
    </xs:all>
  </xs:complexType>
  <xs:complexType name="AttributeValueSet">
    <xs:sequence>
      <xs:element name="attribueValue" type="xa:AttributeValue" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="AttributeValueChange">
    <xs:all>
      <xs:element name="attribute-name" type="xs:string"/>
      <xs:element name="old-value" type="xs:anySimpleType"/>
      <xs:element name="new-value" type="xs:anySimpleType"/>
    </xs:all>
  </xs:complexType>
  <xs:complexType name="AttributeValueChangeSet">
    <xs:sequence>
      <xs:element name="attributeValueChange" type="xa:AttributeValueChange" minOccurs="0"
                  maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Comment">
    <xs:sequence>
      <xs:element name="comment-date" type="xs:dateTime"/>
      <xs:element name="comment-text" type="xs:string"/>
      <xs:element name="user-id" type="xs:string"/>
      <xs:element name="system-id" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="CommentSet">
    <xs:sequence>
      <xs:element name="comment" type="xa:Comment" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="CorrelatedNotification">
    <xs:sequence>
      <xs:element name="source" type="xs:string"/>
```

```
      <xs:element name="notification-id-set">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="notification-id" type="xe:NotificationIDxs:long" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="CorrelatedNotificationSet">
    <xs:sequence>
      <xs:element name="correlatedNotification" type="xa:CorrelatedNotification" minOccurs="0"
                  maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ThresholdInfo">
    <xs:all>
      <xs:element name="attribute-id" type="xs:string"/>
      <xs:element name="observed-value" type="xs:float"/>
      <xs:element name="threshold-level" type="xa:ThresholdLevel" minOccurs="0"/>
      <xs:element name="arm-time" type="xs:dateTime"/>
    </xs:all>
  </xs:complexType>
  <xs:complexType name="ThresholdLevel">
    <xs:all>
      <xs:element name="indication" type="xa:ThresholdIndicator"/>
      <xs:element name="low" type="xs:float" minOccurs="0"/>
      <xs:element name="high" type="xs:float"/>
    </xs:all>
  </xs:complexType>
  <xs:group name="NotifyNewAlarmBodyElements">
    <xs:sequence>
      <xs:element ref="xa:probableCause"/>
      <xs:element ref="xa:perceivedSeverity"/>
      <xs:element ref="xa:alarmType"/>
      <xs:choice>
        <xs:group ref="xa:StandardNotifyNewAlarmBodyElements"/>
        <xs:group ref="xa:SecurityNotifyNewAlarmBodyElements"/>
      </xs:choice>
      <xs:element ref="xa:alarmID"/>
    </xs:sequence>
  </xs:group>
  <xs:group name="StandardNotifyNewAlarmBodyElements">
    <xs:sequence>
      <xs:element name="specificProblem" type="xs:string" minOccurs="0"/>
      <xs:element ref="xa:correlatedNotifications" minOccurs="0"/>
      <xs:element name="backedUpStatus" type="xs:boolean" minOccurs="0"/>
      <xs:element name="backupObject" type="xe:DN" minOccurs="0"/>
      <xs:element name="trendIndication" type="xa:TrendIndicator" minOccurs="0"/>
      <xs:element name="thresholdInfo" type="xa:ThresholdInfo" minOccurs="0"/>
      <xs:element name="stateChangeDefinition" type="xa:AttributeValueChangeSet" minOccurs="0"/>
      <xs:element name="monitoredAttributes" type="xa:AttributeValueSet" minOccurs="0"/>
      <xs:element name="proposedRepairActions" type="xs:string" minOccurs="0"/>
      <xs:element ref="xa:additionalText" minOccurs="0"/>
      <xs:element ref="xa:additionalInformation" minOccurs="0"/>
    </xs:sequence>
  </xs:group>
  <xs:group name="SecurityNotifyNewAlarmBodyElements">
    <xs:sequence>
      <xs:element ref="xa:correlatedNotifications" minOccurs="0"/>
      <xs:element ref="xa:additionalText" minOccurs="0"/>
      <xs:element name="serviceUser" type="xs:string"/>
      <xs:element name="serviceProvider" type="xs:string"/>
      <xs:element name="securityAlarmDetector" type="xs:string"/>
    </xs:sequence>
  </xs:group>
  <xs:complexType name="NotifyAckStateChanged">
    <xs:complexContent>
      <xs:extension base="xe:Notification">
        <xs:all>
          <xs:element name="body">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="xa:probableCause"/>
                <xs:element ref="xa:perceivedSeverity"/>
                <xs:element ref="xa:alarmType"/>
                <xs:element ref="xa:alarmID"/>
```

```
              <xs:element name="ackTime" type="xs:dateTime"/>
              <xs:element name="ackState" type="xa:AckState"/>
              <xs:element name="ackUserID" type="xs:string"/>
              <xs:element name="ackSystemID" type="xs:string" minOccurs="0"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:all>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="NotifyAlarmListRebuilt">
  <xs:complexContent>
    <xs:extension base="xe:Notification">
      <xs:all>
        <xs:element name="body">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="xa:reason"/>
              <xs:element name="alarmListAlignmentRequirement"
                          type="xa:AlarmListAlignmentRequirement" minOccurs="0"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:all>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="NotifyChangedAlarm">
  <xs:complexContent>
    <xs:extension base="xe:Notification">
      <xs:all>
        <xs:element name="body">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="xa:probableCause"/>
              <xs:element ref="xa:perceivedSeverity"/>
              <xs:element ref="xa:alarmType"/>
              <xs:element ref="xa:alarmID"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:all>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="NotifyClearedAlarm">
  <xs:complexContent>
    <xs:extension base="xe:Notification">
      <xs:all>
        <xs:element name="body">
          <xs:complexType>
            <xs:sequence>
              <xs:element refname="xa:probableCause" type="ProbableCause"/>
              <xs:element refname="xa:perceivedSeverity" type="PerceivedSeverity"/>
              <xs:element refname="xa:alarmType" type="EventType"/>
              <xs:element refname="xa:correlatedNotifications" type="CorrelatedNotificationSet"
minOccurs="0"/>
              <xs:element name="clearUserID" type="xs:string" minOccurs="0"/>
              <xs:element name="clearSystemID" type="xs:string" minOccurs="0"/>
              <xs:element nameref="xa:alarmID"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:all>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="NotifyComments">
  <xs:complexContent>
    <xs:extension base="xe:Notification">
      <xs:all>
        <xs:element name="body">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="xa:probableCause"/>
              <xs:element ref="xa:perceivedSeverity"/>
              <xs:element name="comments" type="xa:CommentSet"/>
```

```
                <xs:element ref="xa:alarmID"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:all>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="NotifyNewAlarm">
    <xs:complexContent>
      <xs:extension base="xe:Notification">
        <xs:all>
          <xs:element name="body">
            <xs:complexType>
              <xs:group ref="xa:NotifyNewAlarmBodyElements"/>
            </xs:complexType>
          </xs:element>
        </xs:all>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="NotifyPotentialFaultyAlarmList">
    <xs:complexContent>
      <xs:extension base="xe:Notification">
        <xs:all>
          <xs:element name="body">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="xa:reason"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:all>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:element name="NotifyAckStateChanged" type="xa:NotifyAckStateChanged"
substitutionGroup="xe:Notification"/>
  <xs:element name="NotifyAlarmListRebuilt" type="xa:NotifyAlarmListRebuilt"
substitutionGroup="xe:Notification"/>
  <xs:element name="NotifyChangedAlarm" type="xa:NotifyChangedAlarm"
substitutionGroup="xe:Notification"/>
  <xs:element name="NotifyClearedAlarm" type="xa:NotifyClearedAlarm"
substitutionGroup="xe:Notification"/>
  <xs:element name="NotifyComments" type="xa:NotifyComments" substitutionGroup="xe:Notification"/>
  <xs:element name="NotifyNewAlarm" type="xa:NotifyNewAlarm" substitutionGroup="xe:Notification"/>
  <xs:element name="NotifyPotentialFaultAlarmList" type="xa:NotifyPotentialFaultyAlarmList"
substitutionGroup="xe:Notification"/>
</xs:schema>
```

End of Change in Clause 4.2
End of Document

Annex B (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Dec 2004	S_26	SP-040792	--	--	Submitted to TSG SA#26 for Information	1.0.0	
Mar 2005	S_27	SP-050022	--	--	Submitted to TSG SA#27 for Approval	2.0.0	6.0.0