**Source:** **SA5 (Telecom Management)**

**Title:** **Rel-6 CR 32.302-510 Update Notification IRP IS for new methodology**

**Document for:** **Decision**

**Agenda Item:** **7.5.3**

| Doc-1st-Level | Spec | CR | Rev | Phase | Subject | Cat | Version-Current | Doc-2nd-Level | Workitem |
|---|---|---|---|---|---|---|---|---|---|
| SP-040118 | 32.302 | 005 | - | Rel-6 | Update Ntf IRP IS using new Template and UML Repertoire | F | 5.1.0 | S5-046142 | OAM-NIM |

**3GPP TSG-SA5 (Telecom Management)** **S5-046142**
**Meeting #37, Malaga, SPAIN, February 23-27, 2004**

*CR-Form-v7*

# CHANGE REQUEST

⌘ **32.302** **CR** **005** ⌘ **rev** **-** ⌘ Current version: **5.1.0** ⌘

*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 ☒ Core Network ☒

| | | |
|---|---|---|
| ***Title:*** | ⌘ | Update Ntf IRP IS using new Template and UML Repertoire |
| ***Source:*** | ⌘ | **SA5** Ericsson (edwin.tse@ericsson.com) |
| ***Work item code:*** | ⌘ | OAM-NIM |
| ***Date:*** | ⌘ | 27/02/2004 |
| ***Category:*** | ⌘ | **F** |
| ***Release:*** | ⌘ | Rel-6 |

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

*Use one of the following releases:*

| | |
|---|---|
| *2* | *(GSM Phase 2)* |
| *R96* | *(Release 1996)* |
| *R97* | *(Release 1997)* |
| *R98* | *(Release 1998)* |
| *R99* | *(Release 1999)* |
| *Rel-4* | *(Release 4)* |
| *Rel-5* | *(Release 5)* |
| *Rel-6* | *(Release 6)* |

| | | |
|---|---|---|
| ***Reason for change:*** | ⌘ | Update IS specification such that it aligns with specification template and UML Repertoire guidance. |
| ***Summary of change:*** | ⌘ | Modify the UML class diagram according to UML Repertoire guideline. Modify class tables according to new specification template. |
| ***Consequences if not approved:*** | ⌘ | The IS specification is not documented using the agreed specification template and UML Repertoire guidelines. |

| | | |
|---|---|---|
| ***Clauses affected:*** | ⌘ | 5.2, 6 |

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

⌘

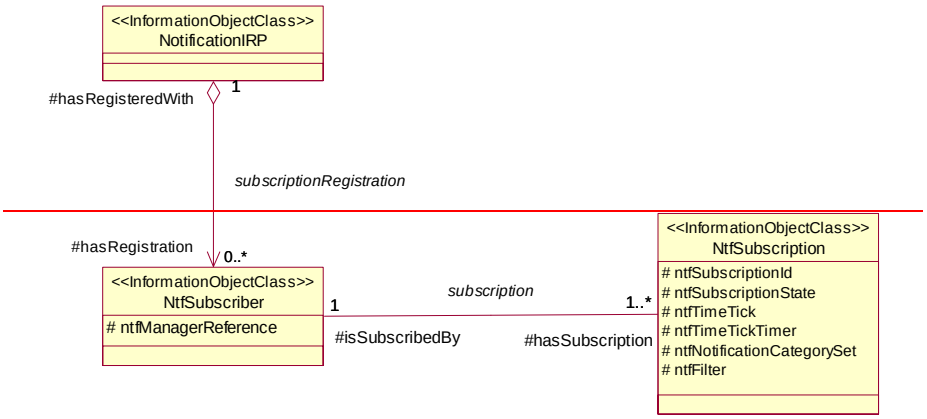
***Other comments:*** ⌘

**How to create CRs using this form:**

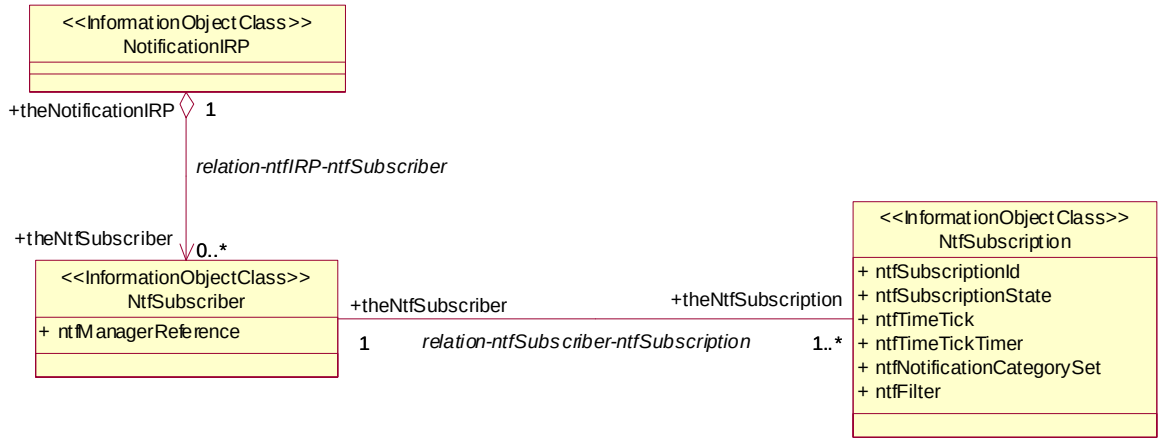
## 5.2 Class Diagram

### 5.2.1 Attributes and relationships

This subclause depicts the set of IOCs that encapsulate information within the notification IRP. The intent is to identify the information required for the notification IRP implementation of its operations and notification emission. This subclause provides the overview of all information object classes in UML. Subsequent subclauses provides more detailed specification of various aspects of these information object classes.

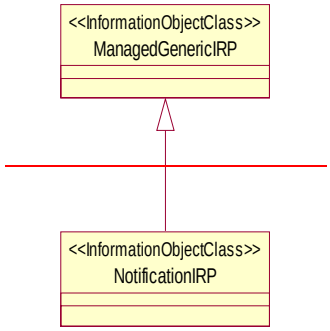



**~~Figure :~~**

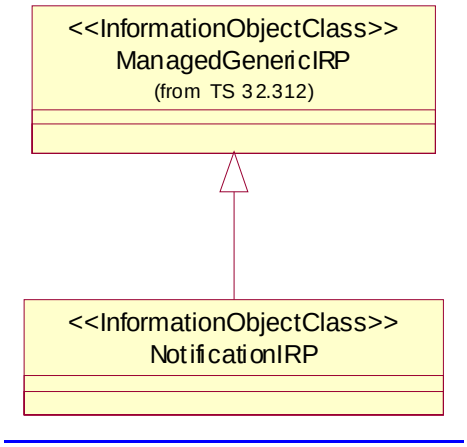


### 5.2.2 Inheritance

This subclause depicts the inheritance relationships that exists between information object classes.




**~~Figure :~~**




## 5.3 Information object classes definition

### 5.3.1 NtfSubscriber

#### 5.3.1.1 Definition

This information object represents a Subscriber from a notification IRP perspective : a subscriber is fully identified by a manager reference. An IRPManager using multiple managerReference attributes to subscribe will result in multiple NtfSubscriber instances. It inherits from IOC Top.

#### 5.3.1.2 Attributes

| Attribute name | Visibility | Support Qualifier | Read Qualifier | Write Qualifier |
|---|---|---|---|---|
| ntfManagerReference | % | M | - | - |

### 5.3.2 NtfSubscription

#### 5.3.2.1 Definition

This information object represents a subscription that have been requested by an IRPManager and created. It inherits from IOC Top.

### 5.3.2.2 Attributes

| Attribute name | Visibility | Support Qualifier | Read Qualifier | Write Qualifier |
|---|---|---|---|---|
| ntfSubscriptionId | + | M | M | - |
| subscriptionState | + | M | M | - |
| ntfTimeTick | + | M | M | - |
| ntfTimeTickTimer | % | M | - | - |
| ntfNotificationCategorySet | + | M | M | - |
| ntfFilter | + | M | M | - |

### 5.3.2.3 State diagram

The diagram below depicts states that can be supported by a NtfSubscription.

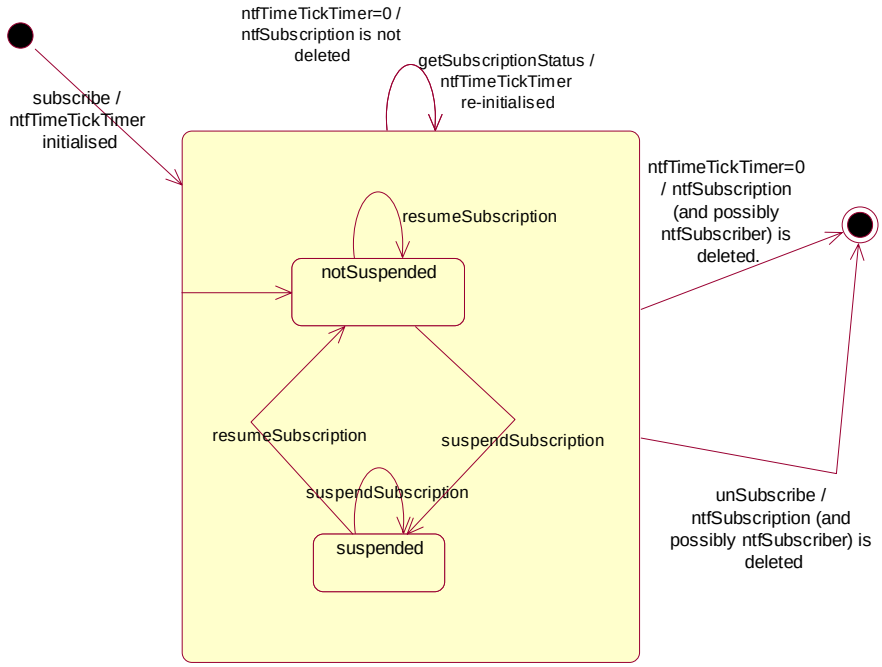



**Figure :**

NotificationIRP can lose the list of `managerReference` that identifies current IRPManagers under subscription. Under this condition, IRPAgent is incapable of sending events to the affected subscriber(s).

This Notification IRP recommends that IRPManager should invoke the `getSubscriptionStatus` operation periodically to confirm that IRPAgent still has the IRPManager's reference in its list. In case getSubscriptionStatus returns the exception operation_failed, IRPManager should assume that IRPAgent has lost the IRPManager's reference.

This IRP does not recommend the frequency IRPManager should use to invoke `getSubscriptionStatus` operation.

## 5.3.3 NotificationIRP

### 5.3.3.1 Definition

This information object represents a notification IRP. It inherits from IOC managedGenericIRP.

## 5.4 Information relationships definition

### 5.4.1 relation-ntfSubscriber-ntfSsubscription (M)

#### 5.4.1.1 Definition

This relationship defines the relationship between a NtfSubscriber and its current subscriptions.

#### 5.4.1.2 Roles

| Name | Definition |
|---|---|
| theNtfSubscriberIsSubscribedBy | This role represents the one who has subscribed. It can be played by instances of IOC NtfSubscriber |
| theNtfSubscriptionhasSubscription | This role represents the subscriptions which were made and not unsubscribed. It can be played by instances of IOC NtfSubscription |

#### 5.4.1.3 Constraints

| Name | Definition |
|---|---|
| inv_notificationCategoriesAllDistinct | T"the notification categories contained in the ntfNotificationCategorySet attribute of NtfSubscription playing the role theNtfSubscriptionhasSubscription are all distinct from each other." |

### 5.4.2 relation-ntfIRP-ntfSubscribersubscriptionRegistration (M)

#### 5.4.2.1 Definition

This relationship defines the relationship between the NotificationIRP and the current subscribers of notifications.

#### 5.4.2.2 Roles

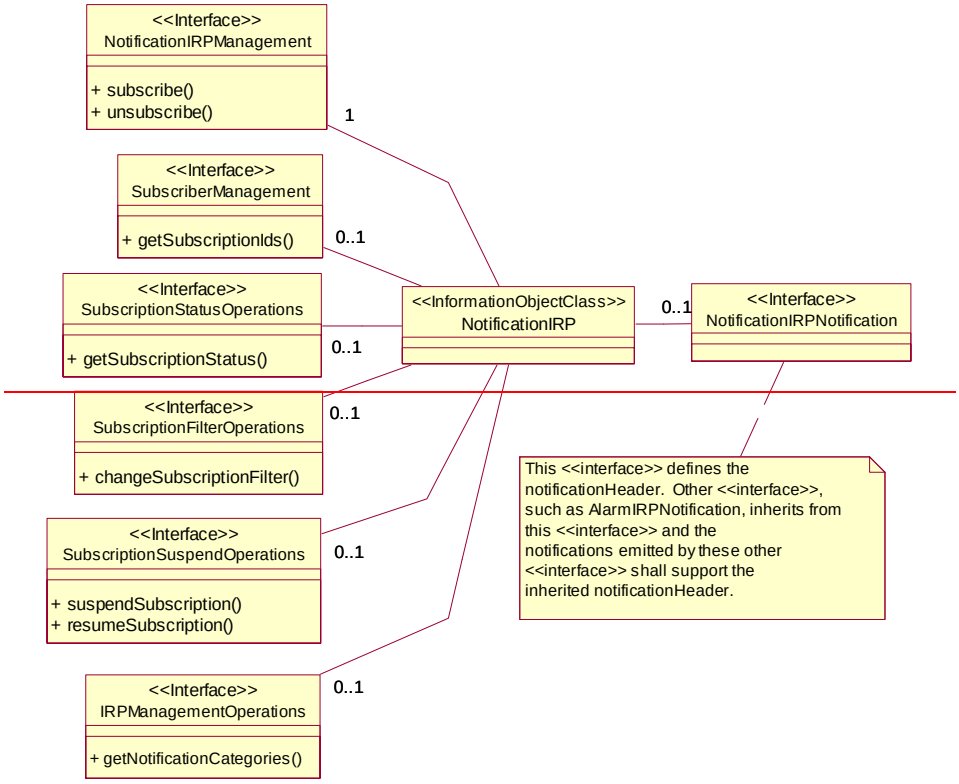
| Name | Definition |
|---|---|
| theNtfSubscriberhasRegistration | This role represents the entities to which IRPAgent will notify events. It is played by instances of IOC NtfSubscriber |
| theNotificationIRPHasRegisteredWith | This role represents the NotificationIRP to which an IRPManager has subscribed. It is played by instances of IOC NotificationIRP |

#### 5.4.2.3 Constraints

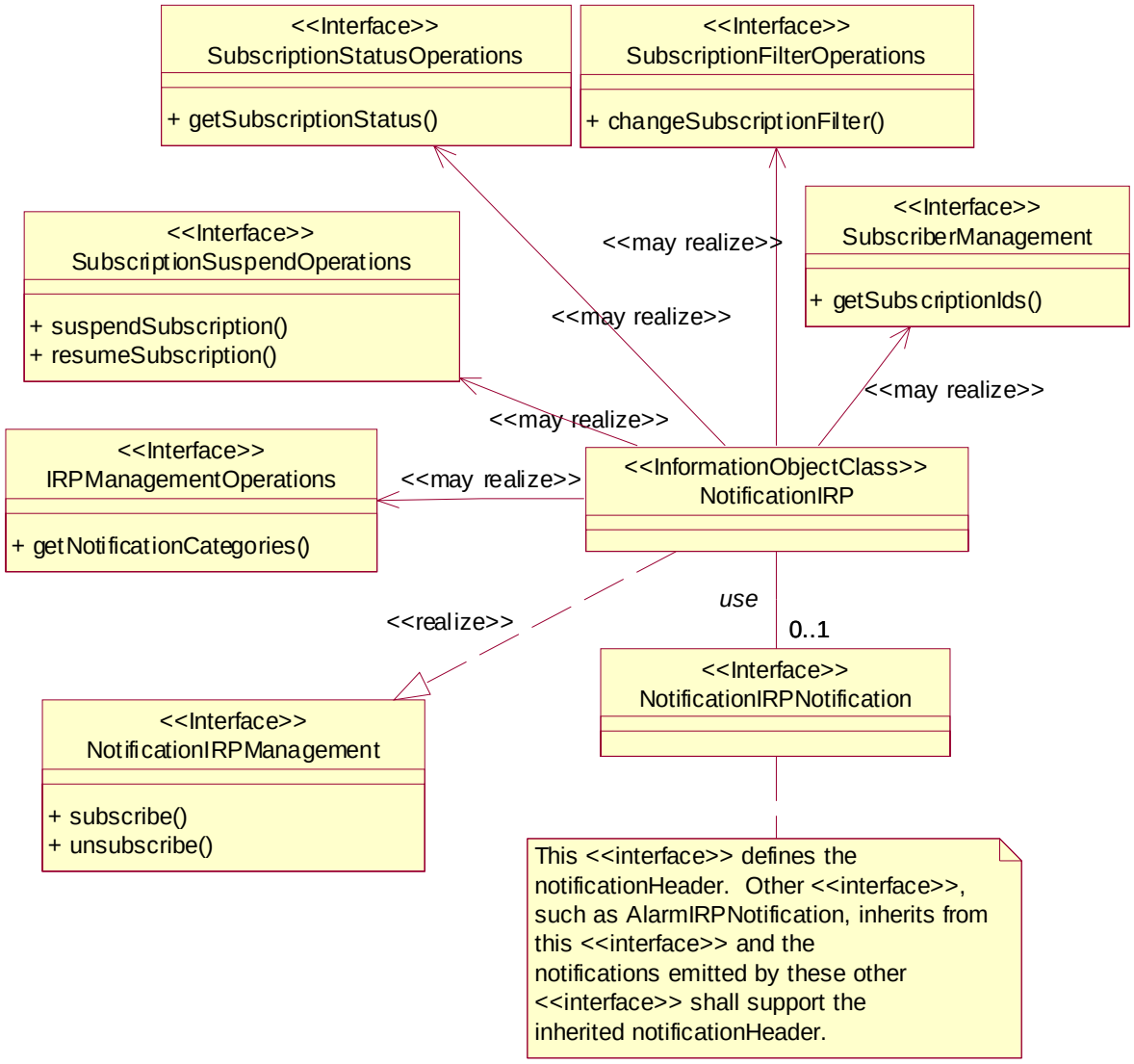
| Name | Definition |
|---|---|
| inv_uniqueManagerReference | A"all NtfSubscriber involved in the subscriptionRegistration relationship with NotificationIRP are distinguished from each other by their ntfManagerReference Attribute." |

# 6 Interface Definition

## 6.1 Class diagram representing interfaces




**~~Figure :~~**




## 6.2 Generic rules

**Rule 1:** Each operation with at least one input parameter supports a pre-condition valid_input_parameter which indicates that all input parameters shall be valid with regards to their information type. Additionally, each such operation supports an exception operation_failed_invalid_input_parameter which is raised when pre-condition valid_input_parameter is false. The exception has the same entry and exit state.

**Rule 2:** Each operation with at least one optional input parameter supports a set of pre-conditions supported_optional_input_parameter_xxx where "xxx" is the name of the optional input parameter and the pre-condition indicates that the operation supports the named optional input parameter. Additionally, each such operation supports an exception operation_failed_unsupported_optional_input_parameter_xxx which is raised when (a) the pre-condition supported_optional_input_parameter_xxx is false and (b) the named optional input parameter is carrying information. The exception has the same entry and exit state.

**Rule 3:** Each operation shall support a generic exception operation_failed_internal_problem which is raised when an internal problem occurs and that the operation cannot be completed. The exception has the same entry and exit state.

# 6.3 notificationIRPManagement Interface (M)

## 6.3.1 Operation `subscribe` (M)

### 6.3.1.1 Definition

IRPManager invokes this operation to establish subscription to receive network events via notifications, under the filter constraint specified in this operation.

### 6.3.1.2 Input parameters

| Parameter Name | Qualifier | Information Type | Comment |
|---|---|---|---|
| managerReference | M | NtfSubscriber.ntfManagerReference | It specifies the reference of IRPManager to which notifications shall be sent. |
| timeTick | O | NtfSubscription.ntfTimeTick | It specifies the value of a timer hold by NotificationIRP for the subject IRPManager.<br>The value is in unit of whole minute.<br>A special infinite value is assumed when parameter is absent or present but equal to zero. |
| notification Categories | O | SET OF (name of IRP, version of IRP) | It identifies one or more Notification Category (see also Definition in subclause 3.1) |
| filter | O | NtfSubscription.ntfFilter<br><br>Filter constraint grammar is SS dependent | It specifies a filter constraint that IRPAgent shall use to filter notification of the category specified in `notificationCategories` parameter.<br>If this parameter is absent, then no filter constraint shall be applied. |

### 6.3.1.3 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
|---|---|---|---|
| subscriptionId | M | NtfSubscription.ntfSubscriptionId | It holds an unambiguous identity of this subscription. |
| status | M | ENUM (OperationSucceeded, OperationFailedExistingSubscription, OperationFailed) | If subscriptionCreated is true, status = OperationSuceeded.<br>If operation_failed_existing_subscription is true, status = OperationFailedExistingSubscription<br>If operation_failed is true, status = OperationFailed. |

### 6.3.1.4 Pre-condition

notificationCategoriesNotAllSubscribed OR notificationCategoriesParameterAbsentAndNotAllSubscribed.

| Assertion Name | Definition |
|---|---|
| notificationCategoriesNotAllSubscribed | "Aat least one notificationCategory identified in the notificationCategories input parameter is supported by IRPAgent and is not a member of the ntfNotificationCategorySet attribute of an NtfSubscription which is involved in a subscription relationship with the NtfSubscriber identified by the managerReference input parameter". |
| notificationCategoriesParameterAbsentAndNotAllSubscribed | The" notificationCategories input parameter is absent and at least one notificationCategory supported by IRPAgent is not a member of the ntfNotificationCategorySet attribute of an ntfSsubscription which is involved in a subscription relationship with the NtfSubscriber identified by the managerReference input parameter." |

#### 6.3.1.5 Post-condition

subscriberPossiblyCreated AND subscriptionCreated.

| Assertion Name | Definition |
|---|---|
| subscriberPossiblyCreated | A "an NtfSubscriber with a ntfManagerReference attribute equal to the value of the managerReference input parameter is involved in a subscriptionRegistration relationship with NotificationIRP." |
| subscriptionCreated | A "an NtfSubscription has been created according to the following rules:<br>• subscriptionState attribute value has been set to "notSuspended";<br>• ntfTimeTick attribute value has been set to the value of the timeTick input parameter if this value was higher or equal to 15, or set to 15 if this parameter value was between 1 and 15, or set to a special infinite value if the parameter value was lower or equal to 0 or if parameter was absent;<br>• ntfTimeTickTimer has been reset with the value of timeTick attribute;<br>• ntfFilter attribute value has been set to the value of the filter input parameter if present;<br>• NtfSubscription is involved in a subscription relationship with the NtfSubscriber identified by the managerReference input parameter;<br>• attribute ntfNotificationCategorySet of NtfSubscription contains EITHER the notification categories identified by the notificationCategories input parameter that were not already contained in the ntfNotificationCategorySet attribute of other NtfSubscription of the same NtfSubscriber identified by the managerReference input parameter OR if notificationCategories input parameter is absent, all notification categories supported by IRPAgent that were not already contained in the ntfNotificationCategorySet attribute of other subscriptions of the same NtfSubscriber identified by the managerReference input parameter". |

#### 6.3.1.6 Exceptions

| Name | Definition |
|---|---|
| operation_failed_existing_subscription | **Condition:** (notificationCategoriesNotAllSubscribed OR notificationCategoriesParameterAbsentAndNotAllSubscribed) not true<br>**Returned Information:** The output parameter status<br>**Exit state:** Entry State |
| Operation_failed | **Condition:** Post-condition is false<br>**Returned Information:** The output parameter status<br>**Exit state:** Entry State |

### 6.3.2 Operation `unsubscribe` (M)

#### 6.3.2.1 Definition

The IRPManager invokes this operation to cancel subscriptions. The IRPManager can cancel one subscription made with a managerReference by providing the corresponding subscriptionId or all subscriptions made with the same managerReference by leaving the subscriptionId parameter absent.

#### 6.3.2.2 Input parameters

| Parameter Name | Qualifier | Information Type | Comment |
|---|---|---|---|
| managerReference | M | NtfSubscriber.ntfManagerReference | It specifies the reference of an IRPManager. |
| subscriptionId | O | NtfSubscription.ntfSubscriptionId | It holds a subscriptionId carried as the output parameter in the subscribe operation. |

#### 6.3.2.3 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
|---|---|---|---|
| status | M | ENUM (OperationSucceeded, OperationFailed) | If (subscriptionDeleted OR allSubscriptionDeleted) is true, status = OperationSucceeded.<br>If operation_failed is true, status = OperationFailed. |

#### 6.3.2.4 Pre-condition

validSubscriptionId&ManagerReference OR SubscriptionIdAbsent&ValidManagerReference

| Assertion Name | Definition |
|---|---|
| validSubscriptionId&ManagerReference | T"the NtfSubscription identified by subscriptionId input parameter is involved in a subscription relationship with the NtfSubscriber identified by the managerReference input parameter." |
| SubscriptionIdAbsent&ValidManagerReference | The "subscriptionId input parameter is absent and the NtfSubscriber identified by the managerReference input parameter exists." |

#### 6.3.2.5 Post-condition

subscriptionDeleted OR allSubscriptionDeleted.

| Assertion Name | Definition |
|---|---|
| subscriptionDeleted | T"the NtfSubscription identified by subscriptionId input parameter is no more involved in a subscription relationship with the NtfSubscriber identified by the managerReference input parameter and has been deleted. If this NtfSubscriber has no more NtfSubscription, it is deleted as well." |
| allSubscriptionDeleted | ""in the case subscriptionId input parameter was absent, the NtfSubscriber identified by the managerReference input parameter is no more involved in any subscription relationship and is deleted, the corresponding NtfSubscription have been deleted as well." |

#### 6.3.2.6 Exceptions

| Name | Definition |
|---|---|
| Operation_failed | **Condition:** Pre-condition is false or post-condition is false<br>**Returned Information:** The output parameter status<br>**Exit state:** Entry State |

## 6.4 subscriberManagement Interface (O)

### 6.4.1 Operation `getSubscriptionIds` (MO)

#### 6.4.1.1 Definition

IRPManager invokes this operation to get the values of all still valid (not unsubscribed or removed by IRPAgent) `subscriptionIds` assigned by NotificationIRP as result of previously `subscribe` operations performed by this IRPManager.

#### 6.4.1.2 Input parameters

| Parameter Name | Qualifier | Information Type | Comment |
|---|---|---|---|
| managerReference | M | NtfSubscriber.ntfManagerReference | It specifies the reference of IRPManager that requests the set of identifiers of active subscriptions related to this IRPManager. |

#### 6.4.1.3 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
|---|---|---|---|
| subscriptionIdSet | M | SET OF NtfSubscription.ntfSubscriptionId where NtfSubscription is involved in a subscription relationship with the NtfSubscriber identified by the managerReference input parameter | It holds a set of the subscriptionId, each assigned as output parameter in previous subscribe operations invoked by the current IRPManager. This value should contain no information if the IRPManager did not yet subscribed to that System or System lost all subscription related information. |
| status | M | ENUM (Operation succeeded, Operation failed) | If validSubscriptionIdSet is true, status = OperationSuceeded.<br>If operation_failed is true, status = OperationFailed. |

#### 6.4.1.4 Pre-condition

validManagerReference.

| Assertion Name | Definition |
|---|---|
| validManagerReference | T"The NtfSubscriber identified by the managerReference input parameter exists." |

#### 6.4.1.5 Post-condition

None specific

#### 6.4.1.6 Exceptions

| Name | Definition |
|---|---|
| Operation_failed | **Condition:** Pre-condition is false<br>**Returned Information:** The output parameter status<br>**Exit state:** Entry State |

## 6.5 subscriptionStatusOperations Interface (O)

### 6.5.1 Operation getSubscriptionStatus (MO)

#### 6.5.1.1 Definition

IRPManager invokes this operation to query the subscription status of a particular subscription. IRPManager can use getSubscriptionStatus operation to know about the filter constraint in effect, the state of subscription (i.e. if subscription is suspended/inactive or resumed/active), the timeTick value that may be set at subscribe invocation time and the notificationCategory currently in used in the subscription.

#### 6.5.1.2 Input parameters

| Parameter Name | Qualifier | Information Type | Comment |
|---|---|---|---|
| subscriptionId | M | NtfSubscription.ntfSubscriptionId | It holds the subscriptionId carried as the output parameter in the subscribe operation |

#### 6.5.1.3 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
|---|---|---|---|
| notification CategorySet | C | NtfSubscription.ntfNotificationCategorySet | It identifies the notification Category(ies) supported in this subscription. |
| filterInEffect | O | NtfSubscription.ntfFilter | It contains the filter constraint currently set. |
| SubscriptionState | O | NtfSubscription.ntfSubscriptionState | |
| timeTick | O | NtfSubscription.ntfTimeTick | It carries the same value as the one in subscribe operation |
| status | M | ENUM (Operation succeeded, Operation failed) | If (timeTickReset) is true, status = OperationSucceeded.<br>If operation_failed is true, status = OperationFailed. |

#### 6.5.1.4 Pre-condition

validSubscriptionId.

| Assertion Name | Definition |
|---|---|
| validSubscriptionId | "the NtfSubscription identified by subscriptionId input parameter is involved in a subscription relationship" |

#### 6.5.1.5 Post-condition

timeTickReset

| Assertion Name | Definition |
|---|---|
| timeTickReset | T"the ntfTimeTickTimer attribute of NtfSubscription identified as input parameter has been reset with the value of ntfTimeTick attribute of the same NtfSubscription." |

#### 6.5.1.6 Exceptions

| Name | Definition |
|---|---|
| Operation_failed | **Condition:** Pre-condition is false or post-condition is false<br>**Returned Information:** The output parameter status<br>**Exit state:** Entry State |

## 6.6 subscriptionFilterOperations Interface (O)

### 6.6.1 Operation changeSubscriptionFilter (MO)

#### 6.6.1.1 Definition

IRPManager invokes this operation to replace the present filter constraint with a new one.

#### 6.6.1.2 Input parameters

| Parameter Name | Qualifier | Information Type | Comment |
|---|---|---|---|
| subscriptionId | M | NtfSubscription.ntfSubscriptionId | It carries the subscriptionId carried as the output parameter in the subscribe operation. |
| filter | M | NtfSubscription.ntfFilter | It specifies a filter constraint |

#### 6.6.1.3 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
|---|---|---|---|
| status | M | ENUM (Operation succeeded, Operation failed) | If filterUpdated is true, status = OperationSuceeded.<br>If operation_failed is true, status = OperationFailed. |

#### 6.6.1.4 Pre-condition

validNtfSubscriptionId.

| Assertion Name | Definition |
|---|---|
| validNtfSubscriptionId | T"the NtfSubscription identified by subscriptionId input parameter is involved in a subscription relationship." |

#### 6.6.1.5 Post-condition

filterUpdated.

| Assertion Name | Definition |
|---|---|
| filterUpdated | The "ntfFilter attribute value of the NtfSubscription identified by subscriptionId input parameter has been set to the value of the filter input parameter." |

#### 6.6.1.6 Exceptions

| Name | Definition |
|---|---|
| Operation_failed | **Condition:** Pre-condition is false or post-condition is false<br>**Returned Information:** The output parameter status<br>**Exit state:** Entry State |

## 6.7 subscriptionSuspendOperations Interface (O)

### 6.7.1 Operation `suspendSubscription` (MO)

#### 6.7.1.1 Definition

IRPManager invokes this operation to request IRPAgent to stop emission of notifications. IRPAgent may lose notification(s) if subscription is suspended.

#### 6.7.1.2 Input parameters

| Parameter Name | Qualifier | Information Type | Comment |
|---|---|---|---|
| subscriptionId | M | NtfSubscription.ntfSubscriptionId | It carries the subscriptionId carried as the output parameter in the subscribe operation. |

### 6.7.1.3 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
|---|---|---|---|
| status | M | ENUM (Operation succeeded, Operation failed) | If subscriptionStateSuspended is true, status = OperationSuceeded.<br>If operation_failed is true, status = OperationFailed. |

### 6.7.1.4 Pre-condition

validSubscriptionId.

| Assertion Name | Definition |
|---|---|
| validSubscriptionId | T"tThe NtfSubscription identified by subscriptionId input parameter is involved in a subscription relationship." |

### 6.7.1.5 Post-condition

subscriptionStateSuspended.

| Assertion Name | Definition |
|---|---|
| subscriptionStateSuspended | Tne "ntfSubscriptionState attribute value of the NtfSubscription identified by subscriptionId input parameter has been set to or kept as "suspended"." |

### 6.7.1.6 Exceptions

| Name | Definition |
|---|---|
| Operation_failed | **Condition:** Pre-condition is false or post-condition is false<br>**Returned Information:** The output parameter status<br>**Exit state:** Entry State |

## 6.7.2 Operation `resumeSubscription` (MO)

### 6.7.2.1 Definition

IRPManager invokes this operation to request IRPAgent to resume emission of notifications.

### 6.7.2.2 Input parameters

| Parameter Name | Qualifier | Information Type | Comment |
|---|---|---|---|
| subscriptionId | M | NtfSubscription.ntfSubscriptionId | It carries the subscriptionId carried as the output parameter in the subscribe operation. |

### 6.7.2.3 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
|---|---|---|---|
| status | M | ENUM (Operation succeeded, Operation failed) | If subscriptionStateNotSuspended is true, status = OperationSuceeded.<br>If operation_failed is true, status = OperationFailed. |

### 6.7.2.4 Pre-condition

validSubscriptionId.

| Assertion Name | Definition |
|---|---|
| validSubscriptionId | T"The NtfSubscription identified by subscriptionId input parameter is involved in a subscription relationship." |

#### 6.7.2.5 Post-condition

subscriptionStateNotSuspended.

| Assertion Name | Definition |
|---|---|
| subscriptionStateNotSuspended | The "ntfSubscriptionState attribute value of the NtfSubscription identified by subscriptionId input parameter has been set to or kept as "notSuspended"." |

#### 6.7.2.6 Exceptions

| Name | Definition |
|---|---|
| Operation_failed | **Condition:** Pre-condition is false or post-condition is false<br>**Returned Information:** The output parameter status<br>**Exit state:** Entry State |

## 6.8 IRPManagementOperations Interface (O)

### 6.8.1 Operation getNotificationCategories (MO)

#### 6.8.1.1 Definition

IRPManager invokes this operation to query the categories of notification supported by IRPAgent. IRPAgent returns the list of categories of notification supported. Each category of notification defines the name and the version of the IRP specification. The list of category of notification returned shall only contain the name and version of the IRP specifications that actually have notifications defined.

IRPManager does not need to be in subscription to invoke this operation.

#### 6.8.1.2 Input parameters

None.

#### 6.8.1.3 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
|---|---|---|---|
| NotificationCategoryList | M | SET OF (name and version of IRP specification) where each IRP is contained by IRPAgent and the attribute notificationNameProfile of the ManagedGenericIRP is not empty. | |
| status | M | ENUM (Operation succeeded, Operation failed) | OperationFailed only if operation_failed_internal_problem |

#### 6.8.1.4 Pre-condition

None specific.

#### 6.8.1.5 Post-condition

None specific.

### 6.8.1.6 Exceptions

None specific.

## 6.9 NotificationIRPNotification Interface

IRPAgent notifies the subscribed IRPManager that an event has occurred and that the event has satisfied the filter constraints used for this subscription. One event example is the notification defined in Alarm IRP: IS (3GPP TS 32.111-2 [1]).

It should be possible to pack multiple notifications together for sending to NM. This provides more efficient use of data communication resources. In order to pack multiple notifications, an EM/NE configurable parameter defines the maximum number of notifications to be packed together. Additionally an EM/NE configurable parameter defines the maximum time delay before the notifications have to be sent.

Under normal operations, an IRPAgent shall send, to each IRPManager, notifications in the same order they were generated, i.e. in the First-In, First-Out order. There shall not be any priority given to types of notifications.

This interface doesn't define any specific notification but instead defines information that is commonly found in notifications defined by other IRPs. This information is called notificationHeader. Notification interfaces defined in other IRPs, such as Alarm IRP: IS (3GPP TS 32.111-2 [1]), shall inherit from this interface and define their notifications by:

- Identifying and qualifying the Notification Header attributes for their use;
- Specify additional attributes specific to their use.

Despite the fact that the semantic of notifications is defined by other IRP ISs, it is notification IRP and not those IRP that is responsible for the emission of those notifications.

The Notification Header is defined here below.

| Attribute Name | Qualifier | Comment |
|---|---|---|
| objectClass | M, Y~~F~~ | It specifies the class name of the IOC. A network event has occurred in an instance of this class. |
| objectInstance | M, Y~~F~~ | It specifies the instance of the above IOC in which the network event occurred by carrying the Distinguished Name (DN) of this object instance. This object may or may not be identical to the object instance actually emitting the notification. |
| notificationId | O,N | This is an identifier for the notification, which may be used to correlate notifications. The identifier of the notification shall be chosen to be unique across all notifications of a particular managed object throughout the time that correlation is significant, it uniquely identifies the notification from other notifications generated by the subject Information Object.<br>If IRPManager receives notifications from one IRPAgent, IRPManager shall use the identifier of the notification and the `objectInstance` to uniquely identify all received notifications.<br>If IRPManager receives notifications from multiple IRPAgents and notifications of each Information Object are reported at most through one IRPAgent, IRPManager shall use the identifier of the notification and `objectInstance` to uniquely identify all received notifications.<br>If IRPManager receives notifications from multiple IRPAgents and notifications of one or more Information Objects are reported through two or more IRPAgents, IRPManager shall use the identifier of the notification together with `objectInstance` and the identity of IRPAgent (systemDN), to uniquely identify all received notifications. If the information `systemDN` is absent, IRPManager needs other means, which are outside the scope of this IRP, to determine the identity of IRPAgent.<br>How identifiers of notifications are re-used to correlate notifications is outside of the scope of this recommendation. |
| eventTime | M, ~~F~~Y | It indicates the event occurrence time. The semantics of Generalised Time specified by ITU-T shall be used here. |
| systemDN | C, Y~~F~~ | It carries the Distinguished Name (DN) of IRPAgent that detects the network event and generates the notification. See "Name Convention for Managed Objects" [3] for name convention regarding DN. |
| notificationType | M, Y~~F~~ | The type of notification which is reported by the notification |