International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

M.3702

(01/2010)

SERIES M: TELECOMMUNICATION MANAGEMENT, INCLUDING TMN AND NETWORK MAINTENANCE

Integrated services digital networks

Common management services – Notification management – Protocol neutral requirement and analysis

Recommendation ITU-T M.3702

ITU-T M-SERIES RECOMMENDATIONS

TELECOMMUNICATION MANAGEMENT, INCLUDING TMN AND NETWORK MAINTENANCE

Introduction and general principles of maintenance and maintenance organization	M.10–M.299
International transmission systems	M.300–M.559
International telephone circuits	M.560–M.759
Common channel signalling systems	M.760–M.799
International telegraph systems and phototelegraph transmission	M.800–M.899
International leased group and supergroup links	M.900–M.999
International leased circuits	M.1000–M.1099
Mobile telecommunication systems and services	M.1100–M.1199
International public telephone network	M.1200–M.1299
International data transmission systems	M.1300–M.1399
Designations and information exchange	M.1400–M.1999
International transport network	M.2000–M.2999
Telecommunications management network	M.3000–M.3599
Integrated services digital networks	**M.3600–M.3999**
Common channel signalling systems	M.4000–M.4999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T M.3702

Common management services – Notification management – Protocol neutral requirement and analysis

Summary

Recommendation ITU-T M.3702 provides the requirements and analysis for one of the common management services – notification management. The functional requirements for notification management interface include the management functions for notification subscription, notification type, notification storage and recovery. In the analysis part, the detail information model supporting the above function across the management interface is provided.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T M.3702	2010-01-13	2

Keywords

Common management service, notification management, notification IRP.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

CONTENTS

Recommendation ITU-T M.3702

Common management services – Notification management – Protocol neutral requirement and analysis

1 Scope

The purpose of this Recommendation is to define an interface through which a manager can subscribe to an agent for receiving notifications. This Recommendation provides the requirements and analysis phase of one of the common management services: Notification management in a protocol neutral way. It defines, for the purpose of subscribing to an agent for receiving notifications, the basic requirements that shall be fulfilled by a notification interface.

Agents generate notifications of events about occurrences within the network. Different kinds of events carry different kinds of information. An agent emits notifications. A manager receives notifications. This Recommendation defines an interface through which a manager can subscribe to an agent for receiving notifications and the information model used to define specific notification types in other Recommendations.

Notifications are sent to managers without the need for managers to periodically check for new notifications.

This Recommendation provides the protocol neutral model definition for notification. It defines, for the purpose of subscribing to an agent for receiving notifications, the information observable and controlled by the management system's client, and it also specifies the semantics of the interactions used to carry this information. It also defines the information common to all notifications.

An agent may emit one or multiple types of notifications, defined as extensions of the notification information object class (IOC) defined herein. This Recommendation defines a mechanism that a manager can use to determine the types of notifications supported by an agent. It also defines a mechanism (subscribe and unsubscribe operations) that a manager can use to specify the type of notifications an agent should emit to a manager during subscription. It also defines a mechanism (getSubscriptionIds operation) that a manager can use to check which types of notifications it has subscribed to. A manager can set and change filtering criteria applicable during the life-cycle of a subscription and subscribe to filters already configured by another manager. A manager can also exercise flow control on an agent's emission of notifications (suspendSubscription, resumeSubscription operations).

A manager can create several subscriptions. This will result in multiple subscriptions. As far as the agent is concerned, notifications are sent to multiple destinations with different filtering criteria.

This Recommendation specifies the information that is carried in all notifications. Further information is defined by specializations of the Notification IOC.

How a manager discovers the agent's address or reference (so that the manager can invoke an operation) is outside the scope of this Recommendation.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T M.3020] Recommendation ITU-T M.3020 (2009), *Management interface specification methodology.*

[ITU-T M.3160] Recommendation ITU-T M.3160 (2008), *Generic, protocol-neutral management information model*.

[ITU-T X.680] Recommendation ITU-T X.680 (2008) | ISO/IEC 8824-1:2008, *Information technology – Abstract syntax Notation One (ASN.1): Specification of basic notation.*

[ITU-T X.701] Recommendation ITU-T X.701 (1997) | ISO/IEC 10040:1998, *Information technology – Open Systems Interconnection – Systems management overview.*

[ITU-T X.734] Recommendation ITU-T X.734 (1992) | ISO/IEC 10164-5:1993, *Information technology – Open Systems Interconnection – Systems Management: Event report management function.*

[ITU-T X.790] Recommendation ITU-T X.790 (1995), *Trouble management function for ITU-T applications.*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 agent [ITU-T M.3020]: Encapsulates a well-defined subset of management functionality. It interacts with managers using a management interface. From the manager's perspective, the agent behaviour is only visible via the management interface.

3.1.2 event [ITU-T X.790]: An instantaneous occurrence that changes the global status of an object. This status change may be persistent or temporary, thus allowing for surveillance, monitoring, and performance measurement functionality, etc. Events may or may not generate reports; they may be spontaneous or planned; they may trigger other events or may be triggered by one or more other events.

3.1.3 information object class [ITU-T M.3020]: Describes the information that can be passed/used in management interfaces and is modelled using the stereotype "Class" in the UML meta-model. For a formal definition of information object class and its structure of specification, see Annex B of [ITU-T M.3020].

3.1.4 manager [ITU-T M.3020]: Models a user of agent(s) and it interacts directly with the agent(s) using management interfaces.

Since the manager represents an agent user, it gives a clear picture of what the agent is supposed to do. From the agent perspective, the manager behaviour is only visible via the management interface.

NOTE – Considered equivalent to IRPManager 3GPP TS 32.150.

3.1.5 notification [ITU-T X.701]: Information emitted by a managed object relating to an event that has occurred within the managed object.

3.2 Terms defined in this Recommendation

This Recommendation has no new definitions.

4 Abbreviations

This Recommendation uses the following abbreviations:

DN	Distinguished Name
EM	Element Manager
IOC	Information Object Class
IRP	Integration Reference Point
NE	Network Element
NM	Network Manager
UML	Unified Modelling Language

5 Conventions

This Recommendation follows the conventions defined in [ITU-T M.3020].

In this Recommendation, the term "subscription" indicates a subscription of a notification service, and the term "subscriber" is the representation in an agent of a receiver of notifications, Notification IRP is an interface in an agent providing notification management functions.

6 Requirements

6.1 Concepts and background

Information of an event is carried in notification. An agent (typically an EM or a NE) emits notifications. A manager (typically a network management system) receives notifications. The purpose of Notification IRP is to define an interface through which a manager can subscribe to an agent for receiving notifications.

This notification IRP bases its design on work captured in [ITU-T X.734].

6.2 Business-level requirements

6.2.1 Requirements

An event may occur in the network at any time. The agent should send notifications to the manager indicating these events. The notification management requirements can be grouped into one of the following categories:

– General, i.e., the notification management requirements with general purpose, e.g., notification report, notification mechanism, notification header, etc.

– Subscription management, e.g., subscribe notification, unsubscribe notifications, suspend subscription, resume subscription, modify subscription, query subscription.

– Notification type, e.g., query notification type.

– Notification storage and recovery, e.g., temporary notification storage and recover.

6.2.1.1 General

REQ-NO-FUN-01	The agent shall provide a mechanism to send notifications to managers.
REQ-NO-FUN-02	The notification mechanism will allow the definition of notification types in other Recommendations.

REQ-NO-FUN-03 It is required that all notifications emitted by the notification interface support the same header that contains enough information to identify the type of notification, the source of the notification and the time at which the notification originated.

6.2.1.2 Subscription management

The following requirements apply to the notification management interface.

REQ-NO-FUN-O4 The agent shall provide the managers with the capability to subscribe and unsubscribe to the notification mechanism.

REQ-NO-FUN-05 A manager shall be able to specify filtering criteria (including the types of notifications) of a subscription that shall be applied by the notification mechanism.

REQ-NO-FUN-06 A manager shall be able to specify the destination of a subscription to which notifications should be forwarded by the notification mechanism in the agent.

REQ-NO-FUN-07 A manager shall be able to suspend and resume the notifications pertaining to a subscription. Events may be lost when notifications are suspended.

REQ-NO-FUN-08 A manager shall be able to suspend and resume notifications from all subscriptions for a given agent. Events may be lost when notifications are suspended.

REQ-NO-FUN-09 Any manager shall be able to set and change filtering criteria applicable during the life-cycle of one of its subscriptions in order to change the pre-defined criteria which determine what notifications are sent.

REQ-NO-FUN-10 A manager shall be able to query the information of a specified subscription in an agent.

REQ-NO-FUN-11 A manager shall be able to query the information of all the subscriptions of the manager for the given agent.

6.2.1.3 Notification type

REQ-NO-FUN-12 The agent shall provide managers with the capability to discover the notification types supported by the agent.

The above requirements are documented in the subsequent use cases.

6.2.1.4 Notification storage and recovery

REQ-NO-FUN-13 If forwarding is not possible at the time a notification is requested, e.g., due to communication breakdown, then the notifications shall be sent as soon as the communication capability has been restored. The storage space is limited. The storage capacity is operator and implementation dependent. If the number of delayed notifications exceeds the storage space, then notifications may be lost and a synchronization procedure may be required. Such a synchronization procedure is outside the scope of this Recommendation.

6.2.2 Actor roles

The capabilities described in this Recommendation are available and relevant to all agents and managers.

6.2.3 Telecommunication resources

The notification management functionality is applicable to all types of telecommunication resources.

6.2.4 High-level use case diagrams

The first overview use case diagram in Figure 1 shows the overall interaction of the notification management interface.

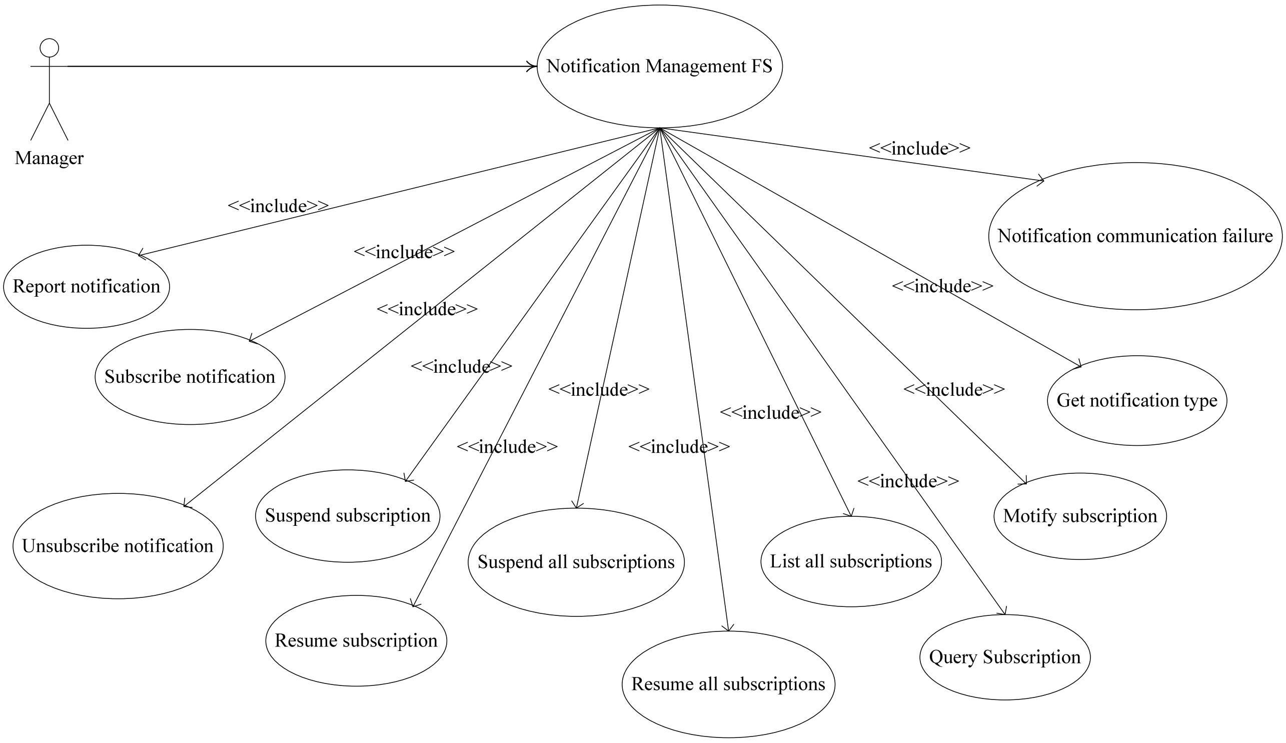

Figure 1 – High-level use case diagram of the notification management interface

6.3 Specification-level requirements

6.3.1 Requirements

There are no specification-level requirements.

6.3.2 Actor roles

See clause 6.2.2.

6.3.3 Telecommunications resources

See clause 6.2.3.

6.3.4 Use cases

The general exceptions (e.g., communication error, processing error) related to all uses cases will not be described in the following uses cases, and they will only be handled in design phases.

6.3.4.1 Report notification

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	Agents can report notifications on any reportable event to managers, through the management interface.	
Actor and roles	All agents and managers.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the agent and the manager is available.	
Pre-conditions	The manager has subscribed to this type of notifications and the subscription is not suspended.	Subscribe notification
Begins when	A reportable event occurs in the agent.	
Step 1	When any reportable event occurs in an agent, a corresponding notification is generated.	
Step 2	The notification is then processed in the notification report mechanism of the agent. For each manager that has subscribed to this type of notification, the filtering criteria will be checked.	
Step 3	If the notification meets the filtering criteria of the managers' subscription, it will be delivered to those managers. Otherwise, the notification is discarded.	
Ends when	A notification of the event is delivered to all subscribed managers, for subscriptions whose filters match the notification. Otherwise, the notification is discarded, if no filters match this notification.	
Exceptions	Communication error.	
Post-conditions	The manager receives the notification if it matches the filtering criteria of the managers' subscription. Otherwise, the notification is discarded.	
Traceability	REQ-NO-FUN-01	

6.3.4.2 Subscribe notification

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	The manager can send a request to an agent to subscribe to notifications from the agent, through the management interface.	
Actor and roles	All agents and managers	
Telecom resources	All types of telecommunication resources.	
Assumptions	Communication between the manager and the agent is available.	
Pre-conditions	The notification forwarding mechanism in the agent is created and ready to work.	

Use case stage	Evolution/Specification	<<Uses>> Related use
Begins when	The manager sends a request to an agent in order to subscribe to notifications from the agent.	

Step 1	The manager sends a request to the agent to subscribe to notifications; the request should include the following information:
	– manager identifier;
	– notification destination;
	– notification types;
	– filtering criteria.	

Step 2.1	If the subscription is created successfully, the agent will return the identifier of the subscription. The subsequent notifications will be forwarded according to the destination and filtering criteria of the subscription.	
Step 2.2	If the subscription creation fails, the agent will return error information. The possible errors are listed under "Exceptions".	
Ends when	The agent gives the manager a response indicating that the subscription creation is completed, or the subscription is failed because of errors.	
Exceptions	Already Subscribed; Invalid parameter.	
Post-conditions	The subscription is created by the agent at the request of the manager. Through this new subscription, notifications will be sent to the manager accordingly.	
Traceability	REQ-NO-FUN-04, REQ-NO-FUN-05, REQ-NO-FUN-06	

6.3.4.3 Unsubscribe notification

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	A manager can send a request to an agent to unsubscribe an existing notification subscription from the agent, through the management interface.	
Actor and roles	All agents and managers.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available.	
Pre-conditions	The manager has already subscribed to the agent.	
Begins when	A manager sends a request to an agent in order to unsubscribe notifications from the agent.	
Step 1	The manager sends a request to the agent to unsubscribe an existing notification subscription; the request should include the following information: – subscription identifier.	

Use case stage	Evolution/Specification	<<Uses>> Related use
Step 2.1	If the operation succeeds, the subscription will be removed. The agent will return operation success information, and subsequent notifications will no longer be forwarded to the manager through this subscription.	
Step 2.2	If the subscription removal fails, the agent will return error information. The possible errors are listed under "Exceptions".	
Ends when	The agent gives the manager a response indicating that the specified notification subscription is removed, or the unsubscribe operation fails because of errors.	
Exceptions	Unknown subscription.	
Post-conditions	The specified notification subscription is removed by the agent at the request of the manager. No more notifications will be forwarded to the manager, because of this deleted notification subscription.	
Traceability	REQ-NO-FUN-04	

6.3.4.4 Suspend subscription

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	A manager can send a request to an agent to suspend a notification subscription in the agent, through the management interface. Thereafter, notifications will not be forwarded to the manager through this subscription.	
Actor and roles	All agents and managers.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available. The agent supports the function of suspending a notification subscription. The manager has already subscribed to the agent.	
Pre-conditions	The specified notification subscription exists in the agent and it is not suspended.	
Begins when	A manager sends a request to an agent to suspend a notification subscription.	
Step 1	The manager sends a request to the agent to suspend an existing notification subscription. The request should include the following information: – subscription identifier.	
Step 2.1	If the operation succeeds, the subscription will temporarily be suspended, and no notifications will be forwarded to the manager through this subscription until it is resumed.	

Use case stage	Evolution/Specification	<<Uses>> Related use
Step 2.2	If the operation fails, the agent will return error information. The possible errors are listed under "Exceptions".	
Ends when	The agent gives the manager a response indicating that the specified notification subscription is suspended, or the suspension fails because of errors.	
Exceptions	Unknown subscription; subscription already suspended.	
Post-conditions	The specified notification subscription is suspended by the agent at the request of the manager. No more notifications will be forwarded to the manager through this notification subscription, until it is resumed.	
Traceability	REC-NO-FUN-07	

6.3.4.5 Resume subscription

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	A manager can send a request to an agent to resume a suspended notification subscription in the agent through the management interface. Thereafter, notifications will be forwarded to the manager through this subscription again.	
Actor and roles	All agents and managers.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available. The agent supports the function of resuming a notification subscription. The manager has already subscribed to the agent.	
Pre-conditions	The specified notification subscription exists in the agent and it is suspended.	
Begins when	A manager sends a request to an agent to resume a suspended notification subscription.	
Step 1	The manager sends a request to the agent to resume a suspended notification subscription. The request should include the following information: – subscription identifier.	
Step 2.1	If the operation succeeds, the subscription will be resumed, and notifications will be forwarded to the manager through this subscription again.	
Step 2.2	If the operation fails, the agent will return error information. The possible errors are listed under "Exceptions".	

Use case stage	Evolution/Specification	<<Uses>> Related use
Ends when	The agent gives the manager a response indicating that the specified notification subscription is resumed, or the resumption fails because of some errors.	
Exceptions	Unknown subscription; subscription not suspended.	
Post-conditions	The specified notification subscription is resumed by the agent on the request of the manager. Thereafter, notifications will be forwarded to the manager, through this notification subscription, again.	
Traceability	REC-NO-FUN-07	

6.3.4.6 Suspend all subscriptions

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	A manager can send a request to an agent to suspend all the notification subscriptions in the agent, corresponding to the manager, through the management interface. Thereafter, notifications will not be forwarded to the manager despite all its subscriptions. NOTE – The notification subscriptions for other managers are not affected.	
Actor and roles	All agents and managers.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available. The agent supports the function of suspending notification subscriptions. The manager has already subscribed to the agent.	
Pre-conditions	The specified manager identifier exists in the agent and not all the subscriptions are suspended.	
Begins when	A manager sends a request to an agent to suspend all its notification subscriptions.	
Step 1	The manager sends a request to the agent to suspend all the exiting notification subscriptions of the manager. The request should include the following information: – manager identifier.	
Step 2.1	If the operation succeeds, all the subscriptions of the manager will temporarily be suspended, and no notifications will be forwarded to the manager through these subscriptions until they are resumed.	
Step 2.2	If the operation fails, the agent will return error information. The possible errors are listed under "Exceptions".	

Use case stage	Evolution/Specification	<<Uses>> Related use
Ends when	The agent gives the manager a response indicating that all its notification subscriptions are suspended, or the suspension fails because of errors.	
Exceptions	Unknown manager; all subscriptions already suspended.	
Post-conditions	All the notification subscriptions corresponding to the manager are suspended by the agent at the request of the manager. No more notifications will be forwarded to the manager through these notification subscriptions, until they are resumed.	
Traceability	REQ-NO-FUN-08	

6.3.4.7 Resume all subscriptions

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	A manager can send a request to an agent to resume all the suspended notification subscriptions in the agent, corresponding to the manager, through the management interface. Thereafter, all subscribed notifications will be forwarded again to the manager. NOTE – The notification subscriptions for other managers are not affected.	
Actor and roles	All agents and managers.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available. The agent supports the function of suspending notification subscriptions. The manager has already subscribed to the agent.	
Pre-conditions	The specified manager identifier exists in the agent and not all the subscriptions are resumed.	
Begins when	A manager sends a request to an agent to resume all suspended notification subscriptions.	
Step 1	A manager sends a request to an agent to resume all suspended notification subscriptions of the manager. The request should include the following information: – manager identifier.	
Step 2.1	If the operation succeeds, all the subscriptions of the manager will be resumed, and notifications will be forwarded to the manager through these subscriptions again.	
Step 2.2	If the operation fails, the agent will return error information. The possible errors are listed under "Exceptions".	

Use case stage	Evolution/Specification	<<Uses>> Related use
Ends when	The agent gives the manager a response indicating that all the notification subscriptions are resumed, or the resumption fails because of some errors.	
Exceptions	Unknown manager; all subscriptions already resumed.	
Post-conditions	All the notification subscriptions corresponding to the manager are resumed by the agent on the request of the manager. Thereafter, all subscribed notifications will be again forwarded to the manager.	
Traceability	REC-NO-FUN-08	

6.3.4.8 Modify subscription

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	A manager can send a request to an agent to modify the attribute values of a notification subscription in the agent, through the management interface.	
Actor and roles	All agents and managers.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available. The agent supports the function of modifying a notification subscription. The manager has already subscribed to the agent.	
Pre-conditions	The specified notification subscription exists in the agent and it is suspended.	
Begins when	A manager sends a request to an agent to modify the attribute values of a suspended notification subscription.	
Step 1	The manager sends a request to the agent to modify the attribute values of a notification subscription. The filtering criteria and the destination can be modified. The request should include the following information: – subscription identifier; – the list of attribute names to be modified (include "destination" , "notification type" and "filtering criteria"); – the list of attribute values corresponding to the specified attribute names to be modified.	
Step 2.1	If the operation succeeds, the specified attribute values of the subscription will be modified, and notifications will be forwarded to the new destination with the new filtering criteria.	
Step 2.2	If the operation fails, the agent will return error information. The possible errors are listed under "Exceptions".	

Use case stage	Evolution/Specification	<<Uses>> Related use
Ends when	The agent gives the manager a response indicating that the specified attribute values of the notification subscription are modified successfully, or the modification fails because of errors.	
Exceptions	Unknown subscription; invalid parameter; subscription not suspended.	
Post-conditions	The attribute values of the specified notification subscription are modified by the agent on the request of the manager. The agent may send the corresponding attribute value change notification to the manager. The notification dispatcher in the agent then forwards notifications with the new filtering criteria to the new destination of the newly modified notification subscription.	
Traceability	REQ-NO-FUN-09	

6.3.4.9 Query subscription

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	A manager can send a request to an agent to query the attribute values of a notification subscription in the agent.	
Actor and roles	All agents and managers.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available. The agent supports the function of querying a notification subscription. The manager has already subscribed to the agent.	
Pre-conditions	The specified notification subscription exists in the agent.	
Begins when	A manager sends a request to an agent to query the attribute values of a notification subscription.	
Step 1	The manager sends a request to the agent to query the attribute values of a notification subscription, such as subscription state, filtering criteria, destination, subscribed notification types and so on. The request should include the following information: – subscription identifier.	
Step 2.1	If the operation succeeds, the all attribute values of the subscription will be returned to the manager.	
Step 2.2	If the operation fails, the agent will return error information. The possible errors are listed under "Exceptions".	
Ends when	The agent gives the manager a response with the specified attribute values, or an indication that the query fails because of errors.	

Use case stage	Evolution/Specification	<<Uses>> Related use
Exceptions	Unknown subscription.	
Post-conditions	The specified attribute values of the specified notification subscription are returned by the agent at the request of the manager.	
Traceability	REQ-NO-FUN-10	

6.3.4.10 List all subscription Ids

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	A manager can send a request to an agent to query the list of all the notification subscriptions of the manager in the agent.	
Actor and roles	All agents and managers.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The communication between the manager and the agent is available. The agent supports the function of querying a notification subscription. The manager has already subscribed to the agent.	
Pre-conditions	The specified manager identifier exists in the agent.	
Begins when	A manager sends a request to an agent to query the list of all the notification subscriptions of the manager in the agent.	
Step 1	The manager sends a request to the agent to query the list of all the notification subscriptions. The request should include the following information: – manager identifier.	
Step 2.1	If the operation succeeds, the list of the notification subscription identifiers of the manager will be returned to the manager.	
Step 2.2	If the operation fails, the agent will return error information. The possible errors are listed under "Exceptions".	
Ends when	The agent gives the manager a response with the list of subscription identifiers, or an indication that the query fails because of errors.	
Exceptions	Unknown manager.	
Post-conditions	The list of identifiers of the notification subscriptions are returned by the agent at the request of the manager.	
Traceability	REQ-NO-FUN-11	

6.3.4.11 Get Notification type

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	A manager can send a request to an agent to query the notification types currently supported by a Notification IRP in the agent.	
Actor and roles	All agents and managers.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The agent supports the function of getting notification types.	
Pre-conditions	None.	
Begins when	A manager sends a request to an agent to query the notification types supported by a Notification IRP in the agent.	
Step 1	The manager sends a request to the agent to query the notification types supported by a Notification IRP in the agent. The request should include the following information: – notification IRP identifier.	Subscribe notification
Step 2.1	If the operation succeeds, the list of supported notification types of the specified Notification IRP will be returned to the manager.	
Step 2.2	If the operation fails, the agent will return error information. The possible errors are listed under "Exceptions".	
Ends when	The agent gives the manager a response with the specified list of supported notification types, or an indication that the query fails because of some errors.	
Exceptions	Unknown notification IRP.	
Post-conditions	The list of the supported notification types of the specified Notification IRP is returned by the agent at the request of the manager.	
Traceability	REQ-NO-FUN-12	

6.3.4.12 Notification communication failure

Use case stage	Evolution/Specification	<<Uses>> Related use
Goal	Agents may provide a mechanism to ensure no notifications are lost due to communications failure.	
Actor and roles	All agents and managers.	
Telecom resources	All types of telecommunication resources.	
Assumptions	The agent supports the function of temporary notification storage and recovery.	
Pre-conditions	Communications from the agent to the manager have failed.	

Use case stage	Evolution/Specification	<<Uses>> Related use
Begins when	An agent detects a communications failure to the manager.	
Step 1	The agent detects a communications failure to the manager, and stores the notifications which are not forwarded using some temporary storage mechanism.	
Step 2	When the communications between the agent and the manager are available again, the agent will then recover those notifications from temporary storage, and forward them to the manager.	
Ends when	The notifications missed during communications failure are forwarded to the manager.	
Exceptions	None.	
Post-conditions	The notifications missed during communications failure are successively received by the manager.	
Traceability	REQ-NO-FUN-13	

7 Analysis

7.1 Concepts and background

The system contexts for the notification management service are shown in Figures 2 and 3.

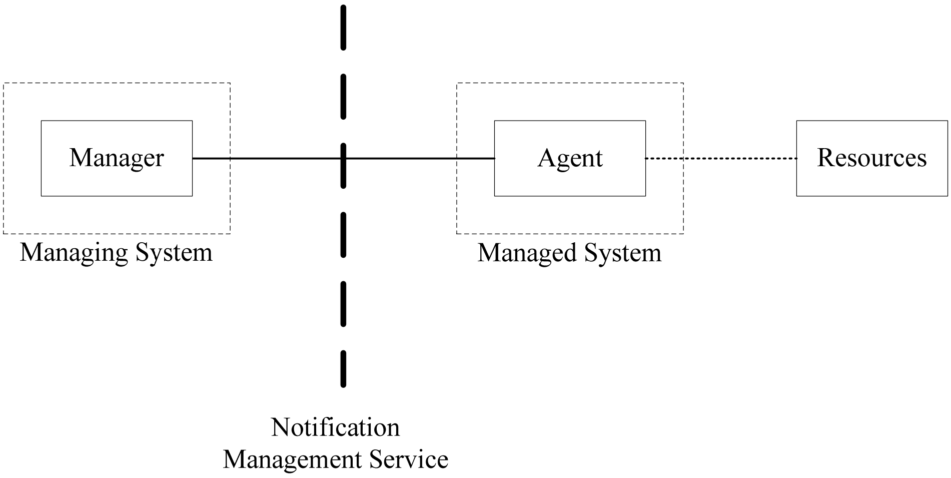

Figure 2 – System context A for notification management service

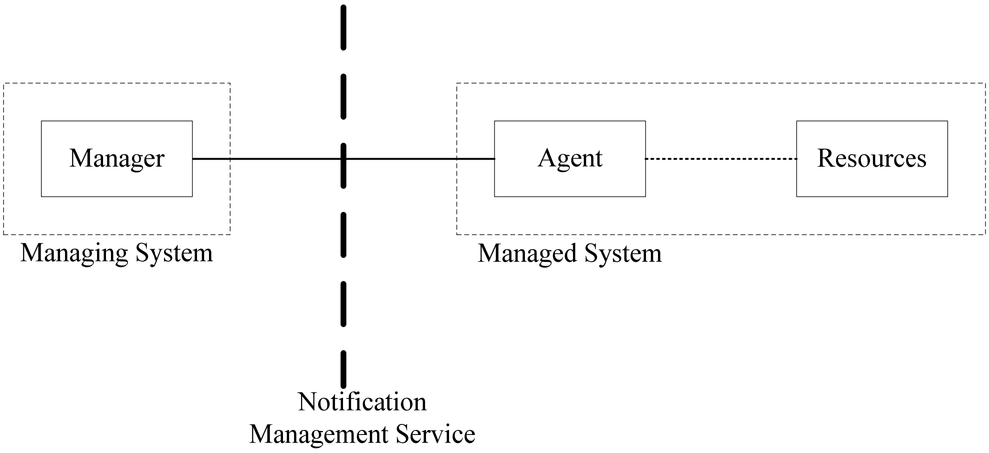

Figure 3 – System context B for notification management service

7.2 Information object classes

7.2.1 Information entities imported and local label

Label reference	Local label
[ITU-T M.3160], information object class, Top	Top
[ITU-T M.3160], information object class, Network	Network

7.2.2 Class diagram

7.2.2.1 Attributes and relationships

Figure 4 shows the class containment relationships of the notification management related information object classes.

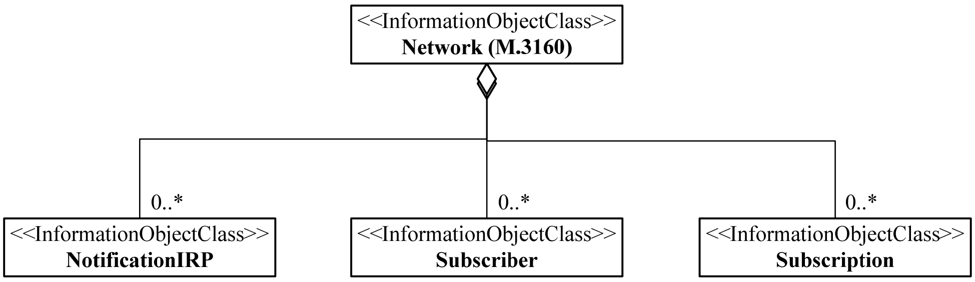
<<InformationObjectClass>>
Network (M.3160)
0..*
0..*
0..*
<<InformationObjectClass>>
NotificationIRP
<<InformationObjectClass>>
Subscriber
<<InformationObjectClass>>
Subscription

Figure 4 – Containment diagram of notification management IOCs

Figure 5 shows the detailed UML class diagram of the information object classes `NotificationIRP`, `Subscriber`, `Subscription`, and the association relationships between them.

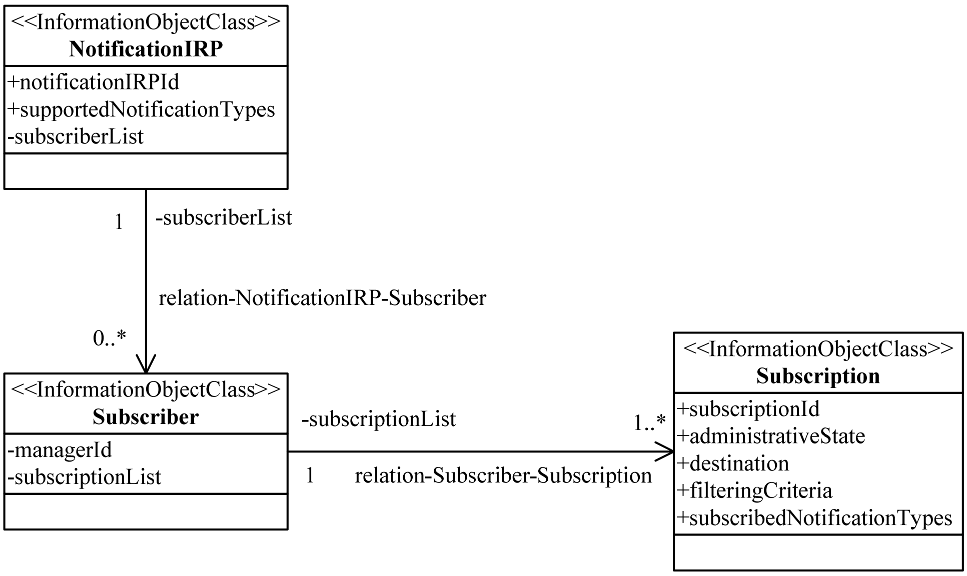

Figure 5 – Class diagram of notification management IOCs

7.2.2.2 Inheritance

Figure 6 is the inheritance diagram of notification management IOCs.

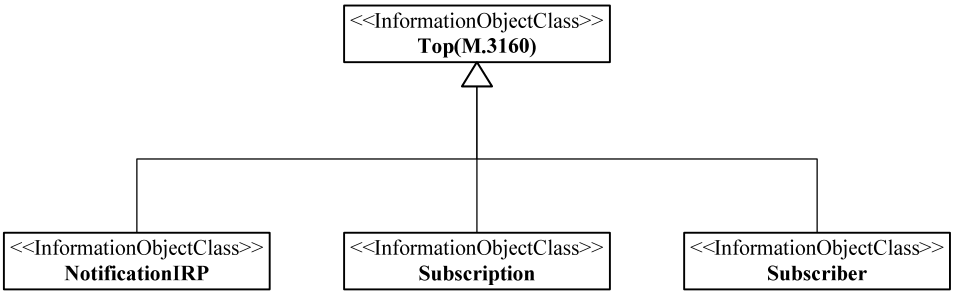
<<InformationObjectClass>>
Top(M.3160)
<<InformationObjectClass>>
NotificationIRP
<<InformationObjectClass>>
Subscription
<<InformationObjectClass>>
Subscriber

Figure 6 – Inheritance diagram for IOCs in notification management

7.2.3 Information object class definitions

Class name	Qualifier	Requirement IDs
`Subscription`	M	REQ-NO-FUN-04, REQ-NO-FUN-05, REQ-NO-FUN-06
`Subscriber`	M	REQ-NO-FUN-04, REQ-NO-FUN-05
`NotificationIRP`	M	REQ-NO-FUN-01

7.2.3.1 Subscriber

7.2.3.1.1 Definition

This IOC represents a subscriber from a notificationIRP perspective: a subscriber is fully identified by a manager identifier. A manager using multiple managerId attributes to subscribe will result in multiple subscriber instances. It inherits from IOC Top.

7.2.3.1.2 Attributes

Attribute name	Support qualifier	Read qualifier	Write qualifier	Requirement IDs
managerId	M	–	–	REQ-NO-FUN-04
subscriptionList	M	–	–	REQ-NO-FUN-04, REQ-NO-FUN-08, REQ-NO-FUN-11

7.2.3.2 Subscription

7.2.3.2.1 Definition

This IOC represents a subscription that has been requested by a manager and created. It inherits from IOC Top.

7.2.3.2.2 Attributes

Attribute name	Support qualifier	Read qualifier	Write qualifier	Requirement IDs
subscriptionId	M	M	–	REQ-NO-FUN-04,
administrativeState	M	M	M	REQ-NO-FUN-07, REQ-NO-FUN-08
filteringCriteria	M	M	M	REQ-NO-FUN-05
destination	M	M	M	REQ-NO-FUN-06
subscribedNotificationTypes	M	M	M	REQ-NO-FUN-05

7.2.3.2.3 State diagram

Figure 7 depicts the states that can be supported by a subscription.

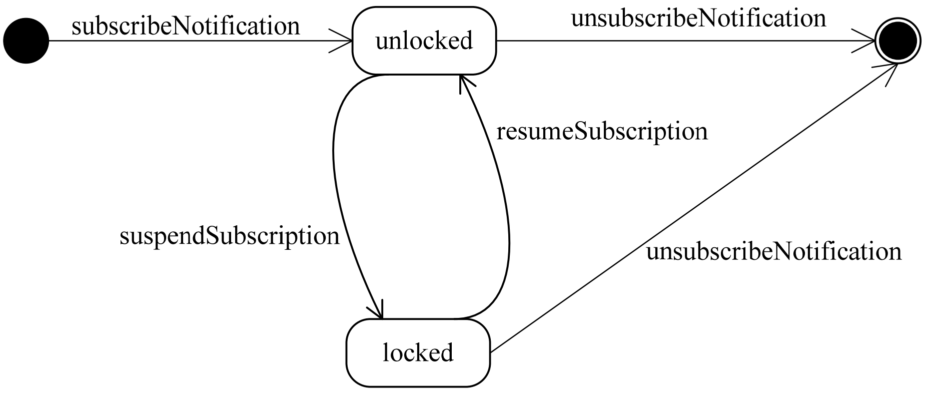

Figure 7 – State diagram of notification subscription

7.2.3.3 NotificationIRP

7.2.3.3.1 Definition

This IOC represents a notification IRP. It inherits from IOC Top.

7.2.3.3.2 Attributes

Attribute name	Support qualifier	Read qualifier	Write qualifier	Requirement IDs
notificationIRPId	M	M	–	REQ-NO-FUN-01
supportedNotificationTypes	M	M	–	REQ-NO-FUN-12
subscriberList	O	–	–	REQ-NO-FUN-08, REQ-NO-FUN-11

7.2.4 Information relationship definitions

Relationship	Support qualifier	Requirement IDs
Relation-NotificationIRP-Subscriber	O	REQ-NO-FUN-08, REQ-NO-FUN-11
Relation-Subscriber-Subscription	O	REQ-NO-FUN-08, REQ-NO-FUN-11

7.2.4.1 Relation NotificationIRP-Subscriber (O)

7.2.4.1.1 Definition

This represents a unidirectional relation from the IOC `NotificationIRP` to the IOC `Subscriber`.

The role of the relation shall be mapped to a reference attribute of the IOC. The name of the reference attribute shall be the role name.

7.2.4.1.2 Roles

Name	Definition
subscriberList	This role (when present) represents the `NotificationIRP` capability to identify the list of Subscribers which subscribed to the notification through this NotificationIRP. When this role is present, the `NotificationIRP.subscriberList` shall carry the list of DNs of the IOC `Subscriber`.

7.2.4.1.3 Constraint

Name	Definition
unique_ManagerId	All Subscribers involved in this relationship with NotificationIRP are distinguished from each other by their managerId attribute.

7.2.4.2 Relation Subscriber-Subscription (O)

7.2.4.2.1 Definition

This represents a unidirectional relation from the IOC `Subscriber` to the IOC `Subscription`.

The role of the relation shall be mapped to a reference attribute of the IOC. The name of the reference attribute shall be the role name.

7.2.4.2.2 Roles

Name	Definition
subscriptionList	This role represents the `Subscriber` capability to identify the list of Subscriptions belonging to this `Subscriber`. When this role is present, the `Subscriber.subscriptionList` shall carry the list of DNs of the IOC `Subscriber`.

7.2.4.2.3 Constraint

Name	Definition
–	–

7.2.5 Information attribute definitions

7.2.5.1 Definition and legal values

Attribute name	Definition	Information type/Legal values
notificationIRPId	The identifier of a NotificationIRP	`String`
subscriberList	The list of the subscriber instances which subscribed to a NotificationIRP.	`SET OF Name`

supportedNotificationTypes	The list of supported notification types by a NotificationIRP.	`SET OF NotificationType`
		`NotificationType ::= String`
		The string for NotificationType can be one of the following list, and can also be expended:
		{*attributeValueChange, communicationAlarm, environmentalAlarm, equipmentAlarm, integrityViolation, objectCreation, objectDeletion, operationalViolation, physicalViolation, qualityOfServiceAlarm, stateChange, processingErrorAlarm, relationshipChange, securityViolation, timeDomainViolation*
		}
		An empty set indicates that all the notification types are supported by this NotificationIRP.

Attribute name	Definition	Information type/Legal values
managerId	This attribute indicates the identifier of a manager. It should be unique among different managers.	`String`
subscriptionList	The list of subscriptions which belong to the same subscriber.	`SET OF Name`
subscriptionId	The identifier of a subscription	`String`
administrativeState	It indicates the state of a notification subscription.	`ENUMERATED /` "*locked*": the subscription is locked. "*unlocked*": the subscription is active.
subscribedNotificationTypes	The list of notification types that are subscribed in a subscription.	`SET OF NotificationType` An empty set indicates that all the supported notification types are subscribed.
filteringCriteria	This attribute provides the filtering constraints for notifications to be forwarded.	`SET OF String` (NOTE – The information type of this attribute can be protocol-specific.)
destination	This attribute indicates the destination in a Subscription, where appropriate notifications are sent to.	`String` (NOTE – The information type of this attribute can be protocol-specific.)

7.2.5.2 Constraints

None.

7.3 Interface definition

7.3.1 Class diagram representing interfaces

The class diagram for the notification management interface can be found in Figure 8.

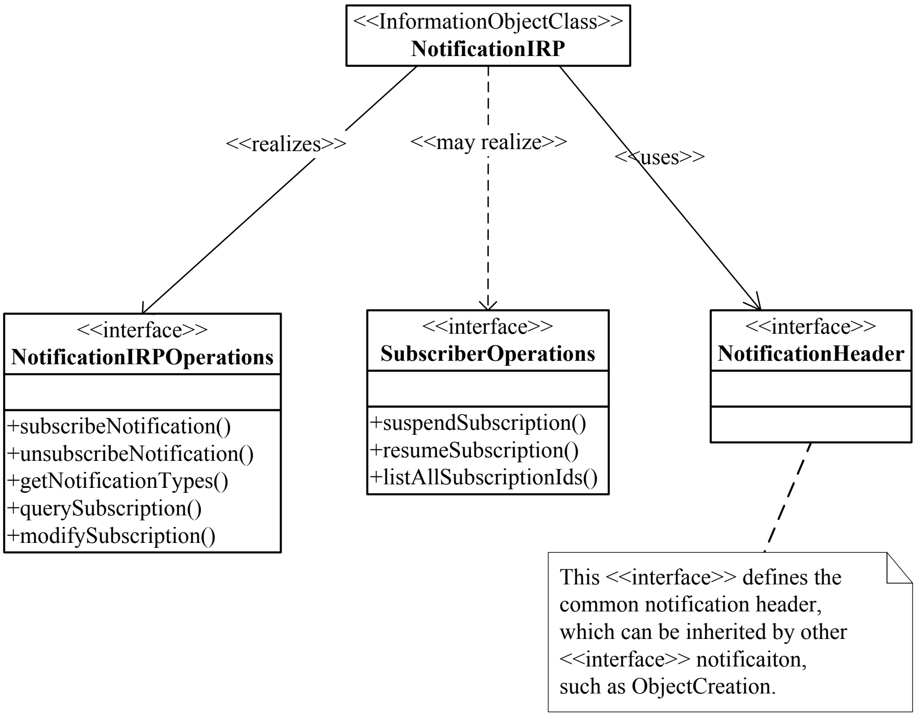

Figure 8 – Class diagram for the notification interface

7.3.2 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 regard 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 the operation cannot be completed. The exception has the same entry and exit state.

7.3.3 Interface NotificationIRPOperations (M)

The NotificationIRPOperations interface provides operations for the management of subscriptions to receive notifications. The operations in this interface can be found in the following table:

Operation name	Qualifier	Requirement IDs
subscribeNotification	M	REQ-NO-FUN-04, REQ-NO-FUN-05, REQ-NO-FUN-06
unsubscribeNotification	M	REQ-NO-FUN-04
getNotificationTypes	M	REQ-NO-FUN-12
querySubscription	M	REQ-NO-FUN-10
modifySubscription	M	REQ-NO-FUN-09

7.3.3.1 Operation subscribeNotification (M)

7.3.3.1.1 Definition

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

7.3.3.1.2 Input parameters

Parameter name	Support qualifier	Information type/ Legal values	Comment
managerId	M	`String`	It specifies the identifier of the manager who sends this request.

notificationTypes	O	`SET OF NotificationType`
		`NotificationType ::= String`
		An empty set indicates all the supported notification types are subscribed.
			It specifies the notification types to be subscribed by the manager.
			If this parameter is absent, all supported notification types are subscribed.

filteringCriteria	O	`String` (The type of filteringCriteria is protocol specific in the design phase)	It specifies the filter constraint that the agent shall use to filter notifications. If this parameter is absent, then no filter constraint shall be applied.
destination	M	`Destination ::= String` (The type of "Destination" is protocol specific in the design phase).	It specifies the destination of a manager to which notifications shall be sent to.

7.3.3.1.3 Output parameters

Parameter name	Support qualifier	Matching information/ Information type/ Legal values	Comment
subscriptionId	M	Subscription.SubscriptionId	It holds a unique identity of this subscription within the scope of a manager identifier.
status	M	`ReturnStatusType ::= ENUMERATED {`*`operationSucceeded, operationFailed`*` } /` "*operationSucceeded*": subscriptionCreated is true. "*operationFailed*": operation_failed is true.	

7.3.3.1.4 Pre-condition

notificationForwardingReadyInAgent AND notificationTypesNotAllSubscribed.

Assertion name	Definition
notificationForwardingReadyInAgent	The notification forwarding mechanism in the agent is created and ready to work.
notificationTypesNotAllSubscribed	At least one notificationType identified in the notificationTypes input parameter is supported by the agent and it is not a member of the subscribedNotificationTypes attribute of a subscription which is involved in a subscription relationship with the subscriber identified by the managerId input parameter.

7.3.3.1.5 Post-condition

subscriberPossiblyCreated AND subscriptionCreated.

Assertion name	Definition
subscriberPossiblyCreated	A subscriber with a managerId attribute equal to the value of the managerId input parameter is involved in a subscriptionRegistration relationship with NotificationIRP.
subscriptionCreated	A subscription has been created according to the following rules: • administrativeState attribute value has been set to "unlocked"; • filteringCriteria attribute value has been set to the value of the filteringCriteria input parameter, if present; • subscription is involved in a subscription relationship with the subscriber identified by the managerId input parameter; • attribute NotificationTypes of subscription contains the notification types identified by the notificationTypes input parameter; • destination attribute value has been set to the value of the destination input parameter.

7.3.3.1.6 Exceptions

Name	Definition
already_subscribed	Condition: The value of "destination" input parameter in this request has already been subscribed in another subscription. Returned Information: "Already subscribed". Exit state: Entry state.
invalid_parameter	Condition: There are some invalid values of input parameters in this request. Returned Information: "Invalid parameter", with an indicator for the specific parameters, which can be either "notificationTypes", "destination", or "filteringCriteria". Exit state: Entry state.

7.3.3.2 Operation unsubscribeNotification (M)

7.3.3.2.1 Definition

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

7.3.3.2.2 Input parameters

Parameter name	Support qualifier	Information type/Legal values	Comment
managerId	M	`String`	It specifies the identifier of a manager.
subscriptionId	O	`Name`	It holds a subscriptionId carried as the output parameter in the subscribe operation.

7.3.3.2.3 Output parameters

Parameter name	Support qualifier	Matching information/ Information type/Legal values	Comment
status	M	`ENUMERATED /` "*operationSucceeded*": subscriptionDeleted or allSubscriptionDeleted is true. "*operationFailed*": operation_failed is true.	

7.3.3.2.4 Pre-condition

validSubscriptionId&ManagerId OR SubscriptionIdAbsent&ValidManagerId

Assertion name	Definition
managerAlreadySubscribed	The manager has already subscribed to notifications from the agent.
validSubscriptionId&ManagerId	The subscription identified by the subscriptionId input parameter is involved in a subscription relationship with the subscriber identified by the managerId input parameter.
SubscriptionIdAbsent&ValidManagerId	The subscriptionId input parameter is absent and the subscriber identified by the managerId input parameter exists.

7.3.3.2.5 Post-condition

subscriptionDeleted OR allSubscriptionDeleted.

Assertion name	Definition
subscriptionDeleted	The subscription identified by the subscriptionId input parameter is no longer involved in a subscription relationship with the subscriber identified by the managerId input parameter and has been deleted. If this subscriber has no more subscriptions, it is deleted as well.
allSubscriptionDeleted	In the case subscriptionId input parameter was absent, the subscriber identified by the managerId input parameter is no longer involved in any subscription relationship and is deleted. The corresponding subscription has been deleted as well.

7.3.3.2.6 Exceptions

Name	Definition
unknown_subscription	Condition: The value of "subscriptionId" input parameter in this request is not known to the agent. Returned information: "Unknown_subscription". Exit state: Entry state.

7.3.3.3 Operation getNotificationTypes (M)

7.3.3.3.1 Definition

A manager invokes this operation to query the notification types supported by an agent. The agent returns the list of notification types supported. The list of notification types returned shall only contain the notifications that may be emitted from this agent.

The manager does not need to be subscribed in order to invoke this operation.

7.3.3.3.2 Input parameters

Parameter name	Support qualifier	Information type/ Legal values	Comment
notificationIRPId	M	`Name`	The identified of the NotificationIRP.

7.3.3.3.3 Output parameters

Parameter name	Support qualifier	Matching information/ Information type/ Legal values	Comment
notificationTypeList	M	SET OF NotificationType NotificationType ::= String	The returned list of supported notifications by the specified NotificationIRP.
status	M	ENUMERATED / "*operationSucceeded*": validNotificationIRPId is true. "*operationFailed*": operation_failed is true.	OperationFailed only if operation_failed_internal_problem

7.3.3.3.4 Pre-condition

validNotificationIRPId

Assertion name	Definition
validNotificationIRPId	The NotificationIRP identified by the notificationIRPId input parameter is valid.

7.3.3.3.5 Post-condition

None.

7.3.3.3.6 Exceptions

Name	Definition
unknown_notificationIRP	Condition: The value of "notificationIRPId" input parameter in this request is not known to the agent. Returned information: "Unknown_notificationIRP". Exit state: Entry state.

7.3.3.4 Operation querySubscription (M)

7.3.3.4.1 Definition

The manager invokes this operation to query all the attribute values of a `Subscription`.

7.3.3.4.2 Input parameters

Parameter name	Support qualifier	Information type/Legal values	Comment
subscriptionId	M	Name	It specifies the identifier of the subscription whose attribute values are to be queried.

7.3.3.4.3 Output parameters

Parameter name	Support qualifier	Matching information/ Information type/ Legal values	Comment
subscribedNotification Types	M	Subscription.subscribed NotificationTypes	It specifies the notification types that are subscribed in this subscription.
subscriptionStatus	M	Subscription.subscriptionStatus	It indicates the status of the subscriptions, which can be locked, or unlocked.
destination	M	Subscription.destination	It specifies the destination where the notification will be sent to.
filteringCriteria	M	Subscription.filteringCriteria	It specifies the filtering criteria to filter notifications.
status	M	`ReturnStatusType /` "*operationSucceeded*": validSubscriptionId is true. "*operationFailed*": operation_failed is true.	

7.3.3.4.4 Pre-condition

validSubscriptionId.

Assertion name	Definition
validSubscriptionId	The subscription identifier specified by the subscriptionId input parameter exists in the agent.

7.3.3.4.5 Post-condition

None.

7.3.3.4.6 Exceptions

Name	Definition
unknown_subscription	Condition: The identifier of the subscription specified in the "subscriptionId" input parameter is unknown to the agent. Returned information: "Unknown Subscription". Exit state: Entry state.

7.3.3.5 Operation modifySubscription (M)

7.3.3.5.1 Definition

The manager invokes this operation to replace the present attribute values of a subscription with a new one(s). The attributes that can be modified include "destination", "filtering criteria", and "notification type".

7.3.3.5.2 Input parameters

Parameter name	Support qualifier	Information type/Legal Values	Comment
subscriptionId	M	`Name`	It carries the subscriptionId carried as the output parameter in the subscribe operation.
filteringCriteria	O	`SET OF String` (NOTE – The information type of the attribute can be protocol-specific.)	If this parameter exists, it specifies the new filtering criteria to replace the old one.
destination	O	`String` (NOTE – The information type of this attribute can be protocol-specific.)	If this parameter exists, it specifies the new destination to which the notification will be sent.
notificationTypes	O	`SET OF NotificationType` An empty set indicates all the supported notification types are subscribed.	If this parameter exists, it specifies the new subscription notification types for this Subscription.

7.3.3.5.3 Output parameters

Parameter name	Support qualifier	Matching information/ Information type/ Legal values	Comment
status	M	`ReturnStatusType /` "*operationSucceeded*": subscriptionUpdated is true. "*operationFailed*": operation_failed is true.	

7.3.3.5.4 Pre-condition

validSubscriptionId.

Assertion name	Definition
validSubscriptionId	The subscription identified by the subscriptionId input parameter is involved in a subscription relationship.
subscriptionLocked	The subscription identified by the subscriptionId input parameter is locked.

7.3.3.5.5 Post-condition

subscriptionUpdated

Assertion name	Definition
subscriptionUpdated	Either the destination, notificationTypes, or filteringCriteria attribute value(s) of the subscription identified by the subscriptionId input parameter have been set to the value of the corresponding input parameter.

7.3.3.5.6 Exceptions

Name	Definition
unknown_subscription	Condition: The identifier of the subscription specified in the "subscriptionId" input parameter is unknown to the agent. Returned information: "Unknown subscription". Exit state: Entry state.
subscription_not_locked	Condition: The specified subscription is not locked. Returned information: "Subscription not locked". Exit state: Entry state.
invalid_parameter	Condition: There are some invalid values of input parameters in this request. Returned information: "Invalid parameter", with an indicator for the specific parameters, which can be either "notificationTypes", "destination", or "filteringCriteria". Exit state: Entry state.

7.3.4 subscriberOperations interface (O)

The SubscriberOperations interface provides operations for subscribers. The operations in this interface can be found in the following table:

Operation name	Qualifier	Requirement IDs
listAllSubscriptionIds	M	REQ-NO-FUN-11
suspendSubscription	M	REC-NO-FUN-07, REQ-NO-FUN-08
resumeSubscription	M	REC-NO-FUN-07, REQ-NO-FUN-08

7.3.4.1 Operation listAllSubscriptionIds (M)

7.3.4.1.1 Definition

The manager invokes this operation to get the values of all still valid (not unsubscribed or removed by the agent) `subscriptionIds` assigned by NotificationIRP as a result of previous `subscribe` operations performed by this manager.

7.3.4.1.2 Input parameters

Parameter name	Support qualifier	Information type/Legal values	Comment
managerId	M	`String`	It specifies the identifier of the manager that requests the set of identifiers of active subscriptions related to this manager.

7.3.4.1.3 Output parameters

Parameter Name	Support qualifier	Matching information/ Information type/Legal values	Comment
subscriptionIdSet	M	`SET OF Name` where subscription is involved in a subscription relationship with the subscriber identified by the managerId input parameter.	It holds a set of subscriptionId, each assigned as output parameter in previous subscribe operations invoked by the current manager. This value should contain no information if the manager has not subscribed to that system or if the system has lost all subscription-related information.
status	M	`ReturnStatusType /` "*operationSucceeded*": validManagerId is true. "*operationFailed*": operation_failed is true.	

7.3.4.1.4 Pre-condition

validManagerId.

Assertion name	Definition
validManagerId	The Subscriber identified by the managerId input parameter exists.

7.3.4.1.5 Post-condition

None.

7.3.4.1.6 Exceptions

Name	Definition
unknown_manager	Condition: The identifier of the manager specified in the "managerId" input parameter is unknown to the agent. Returned information: "Unknown manager". Exit state: Entry state.

7.3.4.2 Operation suspendSubscription (M)

7.3.4.2.1 Definition

A manager invokes this operation to request an agent to stop the emission of notifications from one specified subscription by providing the corresponding subscriptionId, or all subscriptions made with the same managerId by leaving the subscriptionId parameter absent. The agent may lose notification(s) if subscriptions are locked.

7.3.4.2.2 Input parameters

Parameter name	Support qualifier	Information type/Legal values	Comment
managerId	M	`String`	It carries the managerId which can be used to identify a subscriber corresponding to this manager.
subscriptionId	O	`Name`	It carries the subscriptionId carried as the output parameter in the subscribe operation.

7.3.4.2.3 Output parameters

Parameter name	Support qualifier	Matching information/ Information type/ Legal values	Comment
status	M	`ReturnStatusType /` "*operationSucceeded*": administrativeStateLocked is true. "*operationFailed*": operation_failed is true.	

7.3.4.2.4 Pre-condition

validManagerId and validSubscriptionId.

Assertion name	Definition
validManagerId	The manager identified by the managerId input parameter is valid and a corresponding subscriber was created for this manager before.
validSubscriptionId	When the subscriptionId input parameter exists, the subscription identified by the subscriptionId input parameter is valid.

7.3.4.2.5 Post-condition

administrativeStateLocked.

Assertion name	Definition
administrativeStateLocked	The administrativeState attribute value of the specified subscription identified by the subscriptionId input parameter, or all the subscriptions with this subscriber identifier by the "managerId" input parameter have been set to or kept as "locked".

7.3.4.2.6 Exceptions

Name	Definition
unknown_manager	Condition: The identifier of the manager specified in the "managerId" input parameter is unknown to the agent. Returned information: "Unknown Manager". Exit state: Entry state.
unknown_subscription	Condition: The identifier of the subscription specified in the "subscriptionId" input parameter is unknown to the agent. Returned information: "Unknown Subscription". Exit state: Entry state.
subscription_already_suspended	Condition: The specified subscription identified by the subscriptionId input parameter is already suspended. Returned information: "Subscription Already Suspended". Exit state: Entry state.

7.3.4.3 Operation resumeSubscription (M)

7.3.4.3.1 Definition

A manager invokes this operation to request an agent to resume the emission of notifications from one specified subscription by providing the corresponding subscriptionId, or all subscriptions that belong to this subscriber identified by the "managerId" input parameter.

7.3.4.3.2 Input parameters

Parameter name	Support qualifier	Information type/Legal values	Comment
managerId	M	`String`	It carries the managerId which can be used to identify a subscriber corresponding to this manager.
subscriptionId	O	`Name`	It carries the subscriptionId carried as the output parameter in the subscribe operation.

7.3.4.3.3 Output parameters

Parameter name	Support qualifier	Matching information/ Information type/Legal values	Comment
status	M	`ReturnStatusType /` "*operationSucceeded*": administrativeStateUnlocked is true. "*operationFailed*": operation_failed is true.	

7.3.4.3.4 Pre-condition

validManagerId and validSubscriptionId.

Assertion name	Definition
validManagerId	The manager identified by the managerId input parameter is valid and a corresponding subscriber was created for this manager before.
validSubscriptionId	When the subscriptionId input parameter exists, the subscription identified by subscriptionId input parameter is valid.

7.3.4.3.5 Post-condition

administrativeStateUnlocked.

Assertion name	Definition
administrativeStateUnlocked	The administrativeState attribute value of the specified subscription identified by the subscriptionId input parameter, or all the subscriptions with this subscriber identified by the "managerId" input parameter have been set to or kept as "unlocked".

7.3.4.3.6 Exceptions

Name	Definition
unknown_subscription	Condition: The identifier of the subscription specified in the "subscriptionId" input parameter is unknown to the agent. Returned information: "Unknown Subscription". Exit state: Entry state.
unknown_subscription	Condition: The identifier of the subscription specified in the "subscriptionId" input parameter is unknown to the agent. Returned information: "Unknown Subscription". Exit state: Entry state.
subscription_not_suspended	Condition: The specified subscription identified by the subscriptionId input parameter is not suspended. Returned information: "Subscription Not Suspended". Exit state: Entry state.

7.3.5 NotificationHeader interface

An agent notifies the subscribed managers that an event has occurred and that the event has satisfied the filtering criteria (filter constraints) used for this subscription.

It should be possible to pack multiple notifications together to send to the manager. 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 agent shall send, to each manager, 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 does not define any specific notifications but instead defines information that is commonly found in notifications defined in other Recommendations. This information is called notificationHeader. Notification interfaces defined in other Recommendations shall inherit from this interface and define their notifications by:

- identifying and qualifying the notification header attributes for their use;
- specifying additional attributes specific to their use.

Despite the fact that the semantics of notifications are defined by other Recommendations, it is the notificationIRP and not other IRPs that are responsible for the emission of those notifications.

The common notification header is defined below.

Parameter name	Qualifiers	Matching information/ Information type/ Legal values	Comment
objectClass	M	`String`	It specifies the class name of the IOC. A network event has occurred in an instance of this class.
objectInstance	M	`Name`	It specifies the instance of the above IOC in which the network event occurred by carrying the name of this object instance. This object may or may not be identical to the object instance actually emitting the notification.
notificationId	M	`String`	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.

Parameter name	Qualifiers	Matching information/ Information type/ Legal values	Comment
			If a manager receives notifications from one agent, or a manager receives notifications from multiple agents, and notifications of each information object are reported at most through one agent, the manager shall use the identifier of the notification and the `objectInstance` to uniquely identify all received notifications. If a manager receives notifications from multiple agents and notifications of one or more information objects are reported through two or more agents, the manager shall use the identifier of the notification together with `objectInstance` and the identity of the agent (systemDN), to uniquely identify all received notifications. How identifiers of notifications are re-used to correlate notifications is outside of the scope of this Recommendation.
eventTime	M	`GeneralizedTime`	It indicates the event occurrence time. The semantics of generalized time specified in [ITU-T X.680] shall be used here.
systemDN	M	`Name`	It carries the name of the agent that detects the network event and generates the notification.
notificationType	M	`String`	The type of notification which is reported by the notification.

Bibliography

[b-ITU-T X.754]	Recommendation ITU-T X.754 (2000), *Enhanced Event Control Function*.
[b-3GPP TS 32.150]	3GPP TS 32.150 V8.1.0 (2008), *Telecommunication management; Integration Reference Point (IRP) Concept and definitions*.
[b-3GPP 32.301]	3GPP TS 32.301 V8.0.0 (2008), *Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Requirements*.
[b-3GPP 32.302]	3GPP TS 32.302 V8.0.0 (2008), *Configuration Management (CM); Notification Integration Reference Point (IRP): Information Service (IS).*
[b-3GPP 32.602]	3GPP TS 32.602 V8.0.0 (2008), *Telecommunication management; Configuration Management (CM); Basic CM Integration Reference Point (IRP): Information Service (IS).*

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	**Telecommunication management, including TMN and network maintenance**
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2010