ETSI TS 132 150 V8.2.0 (2009-01)

Technical Specification

Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Telecommunication management; Integration Reference Point (IRP) Concept and definitions (3GPP TS 32.150 version 8.2.0 Release 8)

Reference
RTS/TSGS-0532150v820

Keywords
GSM, LTE, UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

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

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Contents

Foreword

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

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

Introduction

The present document is part of a TS-family covering the 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication management; as identified below:

TS 32.150:	**Integration Reference Point (IRP) Concept and definitions**
TS 32.151:	Integration Reference Point (IRP) Information Service (IS) template
TS 32.152:	Integration Reference Point (IRP) Information Service (IS) Unified Modelling Language (UML) repertoire
TS 32.153	Integration Reference Point (IRP) technology specific templates
TS 32.154	Backward and Forward Compatibility (BFC); Concept and definitions
TS 32.155	Telecommunication management; Requirements template

1 Scope

The present document provides the overall concept for all Integration Reference Point (IRP) specifications. Relevant IRP overview and high-level definitions are already provided in 3GPP TS 32.101 [1] and TS 32.102 [2].

IRP specifications are intended to be applicable to any management interface (see definition of Integration Reference Point in subclause 3.1).

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TS 32.101: "Telecommunication management; Principles and high level requirements".

[2] 3GPP TS 32.102: "Telecommunication management; Architecture".

[3] 3GPP TS 32.151: "Telecommunication management; Integration Reference Point (IRP) Information Service (IS) template".

[4] 3GPP TS 32.152: "Telecommunication management; Integration Reference Point (IRP) Information Service (IS) Unified Modelling Language (UML) repertoire".

[5] ITU-T Recommendation M.3020: "TMN Interface Specification Methodology".

[6] OMG IDL Style Guide, ab/98-06-03, June 17, 1998.

[7] 3GPP TS 32.111-2: "Telecommunication management; Fault Management; Part 2: Alarm Integration Reference Point: Information Service (IS)".

[8] 3GPP TS 32.601: "Telecommunication management; Configuration Management (CM); Basic CM Integration Reference Point (IRP); Requirements".

[9] 3GPP TS 32.602: "Telecommunication management; Configuration Management (CM); Basic CM Integration Reference Point (IRP): Information Service (IS)".

[10] 3GPP TS 32.603: "Telecommunication management; Configuration Management (CM); Basic CM Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)".

[11] 3GPP TS 32.621: "Telecommunication management; Configuration Management (CM); Generic network resources Integration Reference Point (IRP); Requirements".

[12] 3GPP TS 32.622: "Telecommunication management; Configuration Management (CM); Generic network resources Integration Reference Point (IRP): Network Resource Model (NRM)".

[13] 3GPP TS 32.623: "Telecommunication management; Configuration Management (CM); Generic network resources Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)".

[14] 3GPP TS 32.671: "Telecommunication management; Configuration Management (CM); State Management Integration Reference Point (IRP): Requirements".

[15] 3GPP TS 32.672: "Telecommunication management; Configuration Management (CM); State Management Integration Reference Point (IRP): Information Service (IS)".

[16] 3GPP TS 32.673: "Telecommunication management; Configuration Management (CM); State Management Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 32.101 [1], 3GPP TS 32.102 [2], 3GPP TS 32.151 [3] and the following apply:

Data Definition IRP: 3GPP publishes IRP specifications relating to commonly used data definitions that can be imported for use by Interface IRP and/or NRM IRP. This term represents all such specifications. An example of a Data Definition IRP is the State Management IRP (32.671 [14], 32.672 [15], 32.673 [16], etc).

Information Object Class (IOC): 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 3GPP TS 32.151[3].

Integration Reference Point (IRP): An architectural concept that is described by a set of specifications for definition of a certain aspect of a management interface, comprising a **Requirements** specification, an **Information Service** specification, and one or more **Solution Set** specifications.

Interface IRP: 3GPP publishes a number of IRP specifications each of which is related to a set of operations and notifications for a specific telecom management domain such as alarm management, configuration management, etc. Interface IRPs also contain definitions of Support IOCs. This term represents all such specifications. An example of an Interface IRP is the Basic CM IRP (the set of TSs 32.601 [8], 32.602 [9], 32.603 [10], etc.).

IRPAgent: Encapsulates a well-defined subset of network (element) functions. It interacts with IRPManagers using one or more IRPs. From the IRPManager's perspective, the IRPAgent behaviour is only visible via the IRP(s).

Information Service (IS): an IRP Information Service describes the information related to the entities (either network resources or support objects) to be managed and the way that the information may be managed for a certain functional area (e.g. the Alarm IRP Information Service in the fault management area). Information Services are defined for all IRPs.

IRPManager: Models a user of IRPAgent(s) and it interacts directly with the IRPAgent(s) using IRP(s).
Since the IRPManager represents an IRPAgent user, it gives a clear picture of what the IRPAgent is supposed to do. From the IRPAgent perspective, the IRPManager behaviour is only visible via the IRP.

Solution Set (SS): contains a mapping of the IRP Information Service (IS) to one of several technologies. An IS can be mapped to several different Solution Sets. Different technology selections may be made for different IRP Information Services. The functionality and information specified in a Solution Set is constrained by the functionality and information specified in the associated Information Service.

Managed Object: Entity used to represent information in a Solution Set. The Managed Objects (MO) are obtained as the result of a mapping exercise of Information Objects defined in IS, taking into account some engineering choices and technology specificity.

Network Resource Model (NRM): An Information Service describing Information Object Classes representing the manageable aspects of network resources, e.g. an RNC or NodeB.

NRM IRP: 3GPP publishes a number of IRP specifications each of which is related to a particular NRM (Network Resource Model) as defined in 3GPP TS 32.101 [1]. NRM IRPs do not define any operations or notifications. This term represents all such specifications. Note: In some NRM IRP titles, for historic reasons, they are named "…network resources IRP"). An example of an NRM IRP is the Generic NRM IRP (32.621 [11], 32.622 [12], 32.623 [13], etc.).

Support IOC: IOC that represents a particular capability, introduced to model a management service. As an example of Support IOC, in the Alarm IRP Information Service [59] there are the AlarmInformation and AlarmList IOCs.

YyyIRP: 3GPP defines a number of support-IOCs (defined in Interface IRPs) such as AlarmIRP, BasicCMIRP and EPIRP. This term represents all such support-IOCs.

Yyy IRP: For a specific Interface IRP such as the Basic CM IRP, when the letters Yyy are replaced by the specific key words naming that IRP (in the given example the Yyy is replaced by 'Basic CM'), this term represents all specifications that are part of that Interface IRP.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TS 32.101 [1], 3GPP TS 32.102 [2], 3GPP TS 32.151 [3] and the following apply:

CORBA	Common Object Request Broker Architecture
EM	Element Manager
GDMO	Guidelines for the Definition of Managed Objects
GUI	Graphical User Interface
IDL	Interface Definition Language
IOC	Information Object Class
IRP	Integration Reference Point
IS	Information Service
NE	Network Element
NM	Network Manager
NRM	Network Resource Model
OMG	Object Management Group
ORB	Object Request Broker
PSA	Product Specific Application
SMP	System Management Processes
SNM	Sub-Network Manager
SS	Solution Set
TMF	TeleManagement Forum
TOM	Telecom Operations Map
UML	Unified Modelling Language

4 Integration Reference Points (IRPs)

4.1 Introduction

For the purpose of management interface development 3GPP has developed an interface concept known as Integration Reference Point (IRP) to promote the wider adoption of standardized management interfaces in telecommunication networks. The IRP concept and associated methodology employs protocol and technology neutral modelling methods as well as protocol specific solution sets to achieve its goals.

4.1.1 General

The three cornerstones of the IRP concept are:

- **Top-down, process-driven modelling approach:** The purpose of each IRP is automation of one specific task, related to TMF TOM. This allows taking a "one step at a time" approach with a focus on the most important tasks.
- **Technology-independent modelling:** To create from the requirements an interface technology independent model. This is specified in the IRP Information Service.
- **Standards-based technology-dependent modelling:** To create one or more interface technology dependent models from the technology independent model. This is specified in the IRP Solution Set(s).

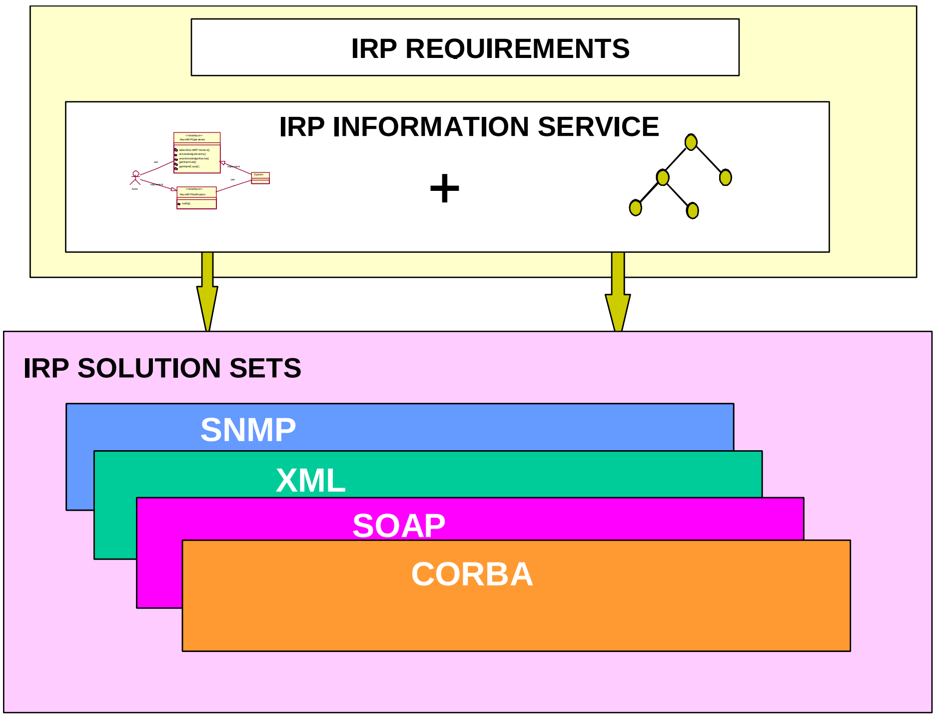

Figure 4.1: IRP components (with example Solution Sets; for definition of valid 3GPP Solution Sets, see Annex C in TS 32.101 [1])

4.1.2 IRP Specifications Approach

As highlighted in the previous subclause, IRP specifications are specified using a 3-level approach: Requirements, IS-level and SS-level.

Furthermore, there are three categories of IRP specifications (see formal and more detailed definitions in subclause 3.1):

- Interface IRPs
- NRM IRPs
- Data Definition IRPs.

Each category is partitioned into Requirements, IS-level and SS-level specifications.

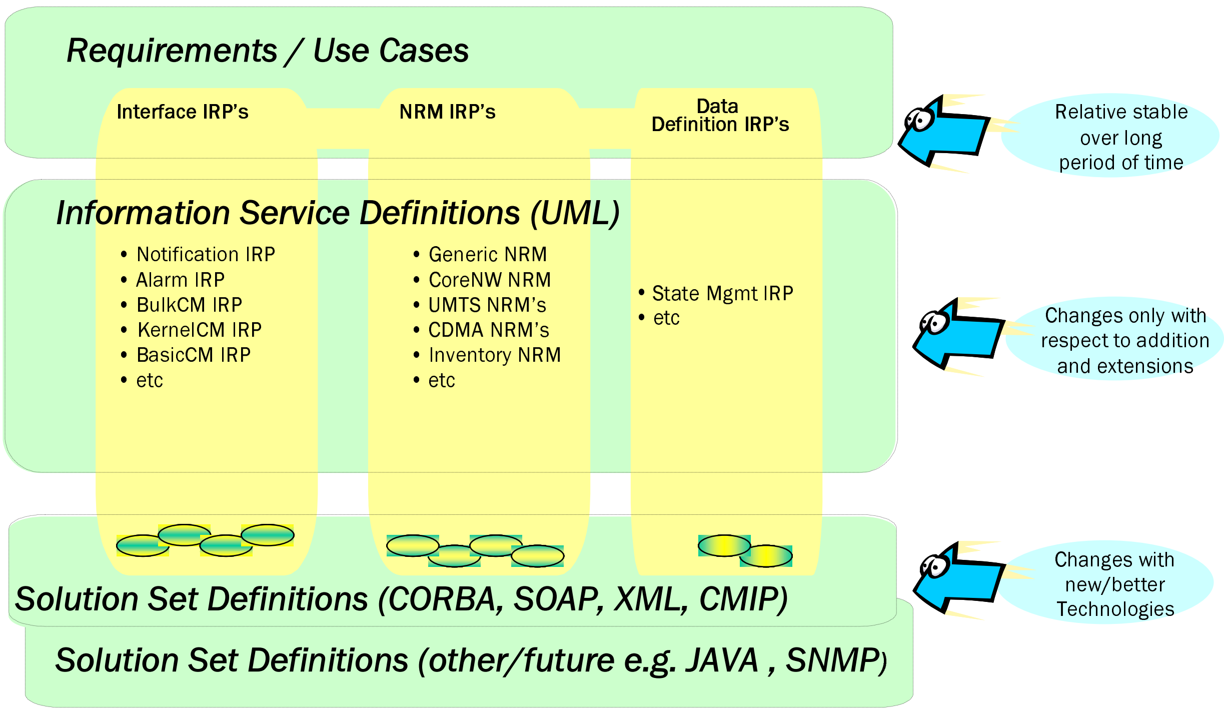

Figure 4.2: The IRP 3-Level Specifications Approach combined with the three IRP categories.

Level 1:

The "Requirements-level" intends to provide conceptual and use cases definitions for a specific management interface aspect as well as defining subsequent requirements for this IRP.

Level 2:

- The "IS-level" provides the technology independent specification of an IRP.

Level 3:

The "SS-level" finally provides the mapping of IS definitions into one or more technology-specific Solution Sets. This concept provides support for multiple interface technologies as applicable on a vendor and/or network type basis and also enables accommodation of future interface technologies - without the need to redefine requirements and IS-level definitions.

4.2 Integration levels

Virtually all types of telecom/datacom networks comprise many different technologies purchased from several different vendors. This implies that the corresponding management solution need to be built by integrating product-specific applications from different vendors with a number of generic applications that each provide some aspect of multi-vendor and/or multi-technology support. A complete management solution is thus composed of several independent applications.

The following levels of integration are defined:

- **Screen Integration:** Each application provides its own specific Graphical User Interface (GUI) that need to be accessible from a single, unified screen (a common desktop). A seamless integration between the various GUIs is then required. Screen Integration is not specified in the present document.

- **Application Integration:** Applications need to interwork, on a machine-machine basis, in order to automate various end-to-end processes of a communication provider.

4.2.1 Application integration

Interfaces related to application integration can be divided in the following three categories:

1) **High-level generic interfaces:** between generic applications on the network and service management layers. The same approach and concepts apply for these as the next category.

2) **High-level (technology-independent to the extent possible) interfaces:** between product-specific and generic applications are needed in order to automate and streamline frequently occurring tasks applicable to several types of network elements. A top-down approach shall be taken when defining these interfaces, where the main input is:

    a) business processes of a communication provider; and

    b) the types of generic applications that are used to implement the process support.

3) **Detailed (product-specific) interfaces:** between product-specific applications and the corresponding network elements are of course also needed. These interfaces are defined using the traditional bottom-up approach, where the actual network infrastructure is modelled. This is the traditional TMN approach to element management. The management information in these interfaces is not further discussed in the present document, as it is internal to a specific development organization and does not need to be open. In fact, by publishing the management information in these interfaces, too much of the internal design may be revealed and it may become impossible to later enhance the systems that are using the interfaces. The management services (operations and notifications) and protocol shall however be open and standardized as long as they are independent of the NRM describing the managed NEs/NRs.

4.3 Application of IRPs

When providing integrated management solutions for multi-vendor networks, there is a strong requirement that the NEs and the management solutions that go together with them are systems integratable.

It should be noted that these IRPs could be provided by an IRPAgent on any management interface.

These IRPs are introduced to ensure interoperability, for example between Product-Specific Applications (PSA) and the Network and System Management Processes (SMP) of the Network Manager (NM) - see figure 4.3 from TS 32.101 [1]. These IRPs are considered to cover the most basic needs of task automation.

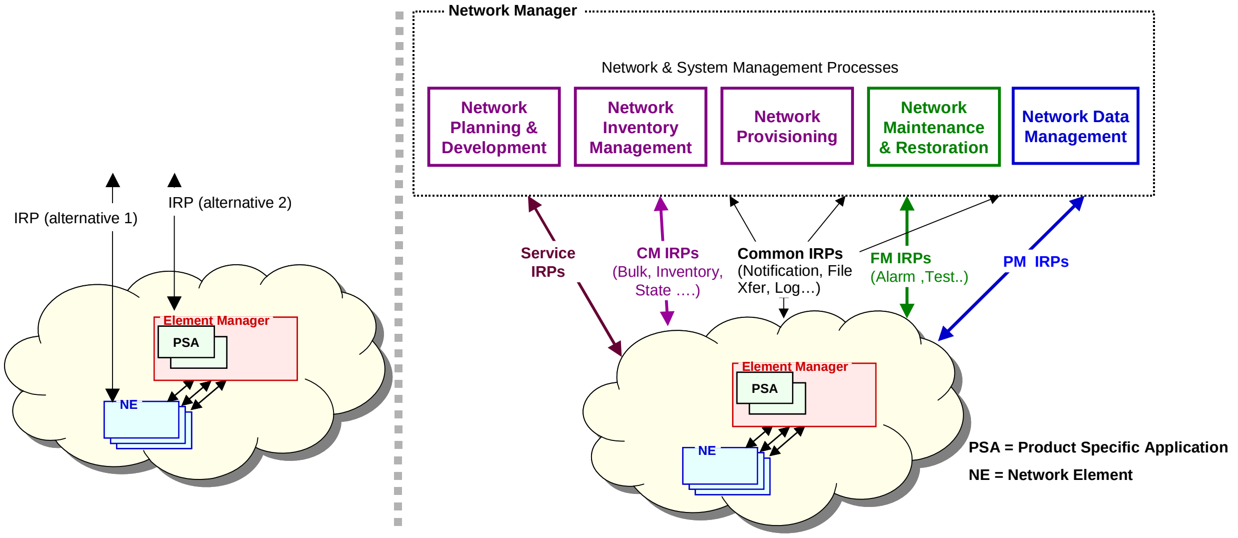

Figure 4.3: Examples of IRPs for application integration

Taking one of the above mentioned IRPs as an example, the Network and System Management Processes have similar need to receive notifications from various PSAs. The corresponding service is formalized as a *Notification IRP*. It specifies: firstly, an interface through which subscriptions to different types of notifications can be set-up (or cancelled), and secondly, common attributes for all notifications.

Further, applying a common *Name Convention for Managed Objects* is useful for co-operating applications that require identical interpretation of names assigned to network resources under management.

4.4 Defining the IRPs

It is important to accommodate more than one specific technology, as the technologies will change over time. Applications need to be future-proof. One fundamental principle for achieving this is to clearly separate the semantics of information definition from the protocols definitions (accessing the information) for the external interfaces.

The framework being used to define IRPs allows the implementation of user requirements for each management capability (e.g. configuration management), by modelling the information related to the resources to be managed and the way that the information may be accessed and manipulated. Such modelling is done in a way that is independent of the technology and distribution used in the implementation of a management system.

The IRP methodology uses the following steps:

a) Capture the management requirements.

b) Specify the semantics of the information to describe the system. Trace back to item (a).

c) Specify the semantics of the interactions between the management system and its clients. Trace back to item (a).

d) Specify the syntaxes of the information and interactions identified in (b) and (c). The specification is technology dependent. Trace back to items (b) and (c).

Figure 4.4 shows an example of how an IRP can be structured (the Alarm IRP).

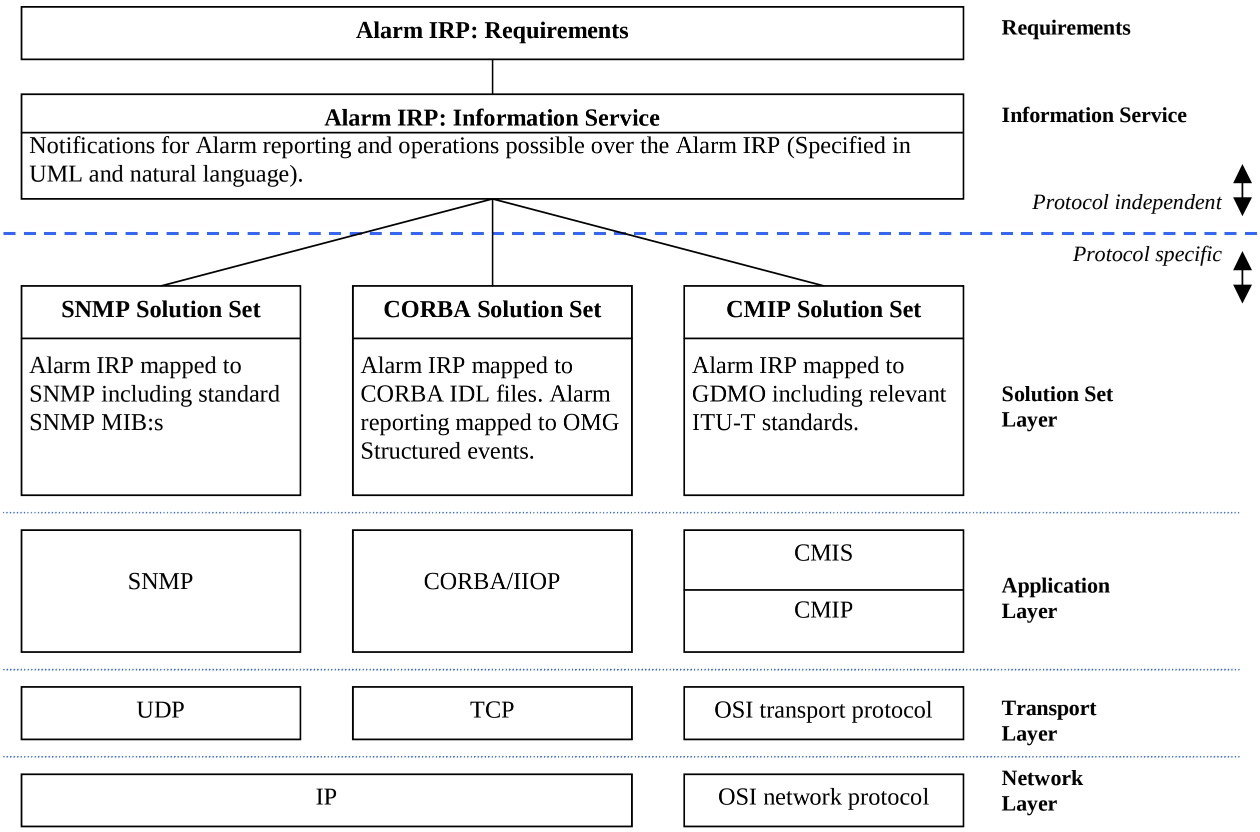

Figure 4.4: Example of an IRP (Alarm IRP)

4.5 Void

4.6 Mandatory, Optional and Conditional qualifiers

This subclause defines a number of terms used to qualify the relationship between the Information Service, the Solution Sets and their impact on the IRP implementations. The qualifiers defined in this clause are used to qualify IRPAgent behaviour only. This is considered sufficient for the specification of the IRPs.

IS specifications define IOC attributes, interfaces, operations, notifications, operation parameters and notification parameters. They can have the following support/read/write qualifiers: M, O, CM, CO, C.

Definition of qualifier M (Mandatory):

- Used for items that shall be supported.

Definition of qualifier O (Optional):

- Used for items which may or may not be supported.

Definition of qualifier CM (Conditional-Mandatory):

- Used for items that are mandatory under certain conditions, specifically:
  - All items having the support qualifier CM shall have a corresponding constraint defined in the IS specification. If the specified constraint is met then the items shall be supported.

Definition of qualifier CO (Conditional-Optional):

- Used for items that are optional under certain conditions, specifically:
  - All items having the support qualifier CO shall have a corresponding constraint defined in the IS specification. If the specified constraint is met then the items may be supported.

Definition of qualifier C (SS-Conditional):

- Used for items that are only applicable for certain but not all Solutions Sets (SSs).

SS specifications define the SS-equivalents of the IS-defined IOC attributes, operations, notifications, operation parameters and notification parameters. These SS-equivalents can have the following support/read/write qualifiers: M, O, CM and CO.

The mapping of the qualifiers of IS-defined constructs to the qualifiers of the corresponding SS-constructs is defined as follows:

- For qualifier M, O, CM and CO, each IS-defined item (operation and notification, input and output parameter of operations, input parameter of notifications, information relationship and information attribute) shall be mapped to its equivalent(s) in all SSs. Mapped equivalent(s) shall have the same qualifier as the IS-defined qualifier.

- For qualifier C, each IS-defined item shall be mapped to its equivalent(s) in at least one SS. Mapped equivalent(s) can have support qualifier M or O.

Table 4.6 defines the semantics of qualifiers of the Interface IRP SS equivalents, in terms of support from the IRPAgent perspective.

Table 4.6: Semantics for Mandatory, Optional and Conditional qualifiers used in Solution Sets

Mapped SS Equivalent	Mandatory	Optional	Conditional-Mandatory (CM)	Conditional-Optional (CO)
Mapped notification equivalent	The IRPAgent shall generate the notification.	The IRPAgent may or may not generate it.	The IRPAgent shall generate this notification if the constraint described in the IS for this item is satisfied.	The IRPAgent may choose whether or not to generate it. If the IRPAgent chooses to generate it, the constraint described in the IS for this notification must be satisfied.
Mapped operation equivalent	The IRPAgent shall support it.	The IRPAgent may or may not support this operation. If the IRPAgent does not support this operation, the IRPAgent shall reject the operation invocation with a reason indicating that the IRPAgent does not support this operation. The rejection, together with a reason, shall be returned to the IRPManager.	The IRPAgent shall support this operation if the constraint described in the IS for this item is satisfied.	The IRPAgent may support this operation if the constraint described in the IS for this item is satisfied.
Input parameter of the mapped operation equivalent	The IRPAgent shall accept and behave according to its value.	The IRPAgent may or may not support this input parameter. If the IRPAgent does not support this input parameter and if it carries meaning (i.e. it does not carry no-information semantics), the IRPAgent shall reject the invocation with a reason (that it does not support the parameter). The rejection, together with the reason, shall be returned to the IRPManager.	The IRPAgent shall accept and behave according to its value if the constraint described in the IS for this item is satisfied.	The IRPAgent may accept and behave according to its value if the constraint described in the IS for this item is satisfied.
Input parameter of mapped notification equivalent AND output parameter of mapped operation equivalent	The IRPAgent shall supply this parameter.	The IRPAgent may supply this parameter.	The IRPAgent shall supply this parameter if the constraint described in the IS for this item is satisfied.	The IRPAgent may supply this parameter if the constraint described in the IS for this item is satisfied.
Mapped IOC attribute equivalent	The IRPAgent shall support it.	The IRPAgent may support it.	The IRPAgent shall support this attribute if the constraint described in the IS for this item is satisfied.	The IRPAgent may support this attribute if the constraint described in the IS for this item is satisfied.

4.7 System context for Interface IRPs

Every Interface IRP on a management interface (e.g. Alarm IRP, Notification IRP, Basic CM IRP, Bulk CM IRP) is subject to a System Context as described in this subclause (also consistent with 3GPP TS 32.102 [2] clause 8).

Figure 4.7.1 and 4.7.2 identify system contexts of the Interface IRP in terms of its implementation, called IRPAgent, and the user of the IRPAgent, called IRPManager.

Each IRPAgent implements and supports one or more IRPs. The set of IRPs that is related to each Interface IRP is defined by the System Context subclause in each individual Interface IRP IS specification, e.g. subclause 4.2 in the Alarm IRP IS [7].

An NE can be managed via System Context A or B. The criterion for choosing System Context A or B to manage a particular NE is implementation dependent. An IRPAgent shall support one of the two System Contexts. By observing the interaction across the management interface, an IRPManager cannot deduce if the EM and NE are integrated in a single system or if they run in separate systems.

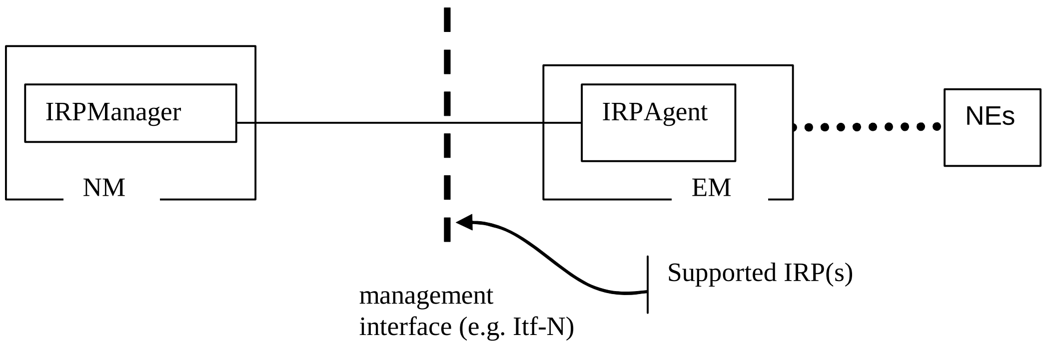

Figure 4.7.1: System Context A

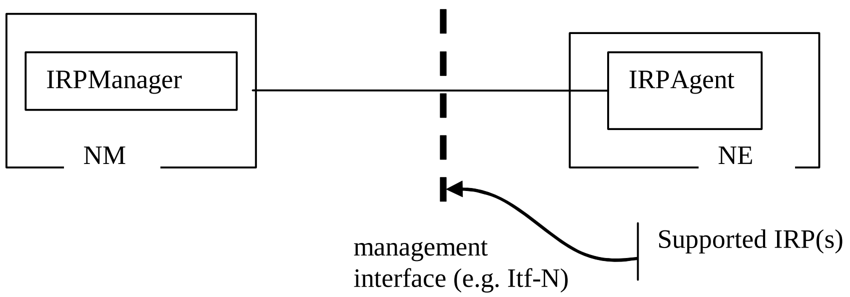

Figure 4.7.2: System Context B

Annex A (informative): General rules for Solution Sets

A.1 Introduction

The intent of this annex is twofold. The first intent is for 3GPP-internal use to document how a 3GPP Solution Set is produced and what it shall contain. The second intent with the annex is to give the reader of an Information Service (IS) or a Solution Set (SS) a better understanding on how to interpret the IS or SS specifications.

A.2 Solution Set (SS) versioning

For further study.

A.3 Referenced Information Service (IS) specification

A sentence shall be included in the clause "Scope" of all Solution Set specifications. The sentence shall read as follows:

"This Solution Set specification is related to Z".

where Z is the 3GPP Information Service (IS) specification number including the version, such as "TS 32.111-2 V4.1.X" for the case of Alarm Integration Reference Point (IRP): Information Service.

NOTE: that "X", rather than the actual digit, is actually used in the sentence. This is because the value of X is not relevant for the reference purpose since different values of X identify different 3GPP published specifications that reflect only minor editorial changes.

Annex B (normative): Rules for CORBA Solution Sets

B.1 Introduction

The intent of this annex is threefold.

1) The first intent is for 3GPP internal use to document how a 3GPP CORBA SS is produced and how it is structured.
2) The second intent with the annex is to give the reader or implementer of a CORBA SS a better understanding on how to interpret the CORBA SS specification.
3) The third and maybe most important intent is to put requirement on an implementer of a CORBA SS.

It is expected that this annex is to be extended in later versions of the present document.

B.2 Rules for specification of CORBA Solution Sets

B.2.1 Introduction

This subclause identifies rules for specification of CORBA SSs. This subclause is mainly for 3GPP-internal use. It is only for information for the implementer of a CORBA SS.

B.2.2 Pragma prefix

All IDL-code shall define the pragma prefix using the following statement:

#pragma prefix "3gppsa5.org"

See clause D.1.4.3 for information of this `#pragma` statement in relation to other IDL statements.

B.3 Implementation aspects of CORBA Solution Sets

B.3.1 Introduction

This subclause identifies rules for the implementation of CORBA SSs. This subclause is normative for the implementer of a CORBA SS.

B.3.2 IRPAgent behaviour on incoming optional method

The IRPAgent, claiming compliance to a particular SS version of a particular IRP such as the Alarm IRP, shall implement all Mandatory and all Optional methods. Each method implementation shall have a signature specifying all Mandatory and all Optional parameters.

- If the IRPAgent does not support a particular optional method, it shall throw the `OperationNotSupported` exception when the IRPManager invokes that method.
- If the IRPAgent have not implemented a particular method (because it is compiled with an IDL version that does not define the method), the CORBA ORB of the IRPAgent shall throw a system exception if the IRPManager invokes that method.

In all the above cases when an exception is thrown, the IRPAgent shall restore its state before the method invocation.

B.3.3 IRPAgent Behaviour on incoming optional parameter of operation

An IRPAgent must implement all optional parameters, as well as mandatory parameters, in all methods.

If the IRPAgent supports the implemented method but does not support its (one or more) optional input parameters, upon method invocation, the IRPAgent shall check if those parameters carry "no information" or absence semantics (defined later in subclause B.3.5). If the check is negative, the IRPAgent shall throw the `ParameterNotSupported` exception with a string carrying the name of the unsupported optional parameter.

B.3.4 IRPAgent Behaviour on outgoing attributes of Notification

CORBA SS uses OMG defined structured event to carry notification. The structured event is partitioned into header and body.

The absence semantics of attribute in the header is realized by a string of zero length.

The body consists of one or more name-value pair attributes. The absence semantics of these attributes is realized by their absence.

For optional sub-attributes of an attribute carried by the name-value pair, their absence semantics is realized by the encoding rule of "absence semantics". See subclause B.3.5.

B.3.5 Encoding rule of absence semantics

The operation parameters are mapped to method parameters of CORBA SS. The absence semantics for an operation (input and output) parameter is method parameter type dependent.

- For a string type, if the parameter is specified as a string type, the absence semantics is a string of zero length. If the parameter is specified as a union structure (preferred), the absence semantics is conveyed via a FALSE Boolean value switch.
- For an integer type, if the parameter is specified as a signed, unsigned, long, etc type, the absence semantics is the highest possible positive number. If the parameter is specified as a union structure (preferred), the absence semantics is conveyed via a FALSE Boolean value switch.
- For a boxed valueType (supported by CORBA 2.3), it is the null value.

The notification parameters are mapped to attributes of the CORBA Structured Events. The absence semantics for a notification parameter is attribute position (within the Structured Event) dependent.

- For the fixed header of the Structured Event header, the absence semantics is realized by a string of zero length.
- For the filterable body fields of the Structured Event body, the absence semantics is realized by the absence of the corresponding attribute.

B.4 Void

Annex C (informative): Void

Annex D (informative): Style Guide for CORBA SS IDL

This annex is the style guide for writing IDL statements for Interface IRP and NRM IRP. The guidelines are largely based on the OMG IDL Style Guide (OMG document: ab/98-06-03) [6] with extensions for IRP use.

The guide sets out consistent naming, structural conventions and usage of SS interface for the IDL in IRP CORBA SS specifications.

D.1 Modules and File

D.1.1 Use of Modules

All declarations of IDL shall be contained in modules. No declarations of interfaces and definitions shall appear in the global scope.

Nesting modules is a useful technique when dealing with large namespaces to avoid name clashes and clarify relationships. A module nested within another module shall not have the same name as a top-level module in any other IRP CORBA SS specification.

D.1.2 File Names

CORBA SS specifications contain IDL statements.

The rule defined below specifies:

a) How to partition/extract these IDL statements to be placed in a file; and

b) How to name the file.

Note that IDL uses `"#include "X""` statement where `X` is a name of a file containing IDL statements.

Rule:

In the annex where IDL statements are defined, use a special marker to indicate that a set of IDL statements shall be contained in one file. The name of the file shall be the name of the first IDL module, concatenated with four characters '`.idl`'. Within a CORBA SS, multiple markers (implying multiple files), can be used.

It is not allowed to have an IDL module split into multiple files.

D.1.3 Include Conventions

All included IDL files shall be specified using the <…> form of `#include`. For example:

```
#include <ManagedGenericIRPConstDefs.idl>
```

D.1.4 File Structure

D.1.4.1 File Internal Identification

The first line of the IDL file shall contain '`//File:`' followed by a single space followed by the name of the file. For example,

```
//File: ExampleIRPConstDefs.idl
```

D.1.4.2 File Guard

An IDL file shall use a *guard* (consisting of three pre-processor lines) to avoid multiple definition errors. An example of a guard for the file called `TestManagementIRPConstDefs.idl` is:

```
#ifndef _TestManagementIRPConstDefs_idl_
#define _TestManagementIRPConstDefs_idl_

...remainder of the IDL

#endif // _TestManagementIRPConstDefs_idl_
```

D.1.4.3 Required Contents

If any other files are to be included, the `#include` statements come after the guard.

After `#include` lines, if any, and immediately before the `module` statement, the following line shall appear:

```
#pragma prefix "3gppsa5.org"
```

D.1.4.4 Example illustrating a File Structure

```
//File: ExampleIRPConstDefs.idl
#ifndef _EXAMPLE_IRP_CONST_DEFS_IDL_
#define _EXAMPLE_IRP_CONST_DEFS_IDL_

// This module describes/is part of…
#include "ExampleIncludeOne.idl"
#include "ExampleIncludeTwo.idl"

#pragma prefix "3gppsa5.org"
module ExampleIRPConstDefs {

// IDL Definitions here

};
#endif // _EXAMPLE_IRP_CONST_DEFS_IDL_
```

D.2 Identifiers

D.2.1 Mixed Case, Beginning Upper, No Underscores

The following categories of identifiers follow the *Mixed Case, Beginning Upper, No Underscores* rules:

- `module`
- `interface`
- `typedef`
- Constructed types (`struct, union, enum`)
- `exception`

The 'No underscores' rule is also applicable to all words that begin with an upper case letter with the remaining letters being lower case.

As a further note on naming, it is not necessary to append the value '*Type*' to an identifier. The fact that it is a type is obvious from the consistent application of this naming convention.

Examples:

```
module PMIRPConstDefs(…);
interface AttributeNameValue(…);
```

D.2.2 Lower Case with Underscores

The following categories of identifiers follow the *Lower Case with Underscores* rules. All letters are lower case and words (if more than one) are separated with underscores.

- Operation name and notification name
- Attribute name
- Parameter name
- Structure member name

Examples:

```
get_notification_categories(…);
string comment_text;
void get_alarm_count (…, out unsigned long critical_count,..);
  struct Comment {…; string user_id; string system_id;..};
```

D.2.3 Upper Case with Underscores

The following categories of identifiers follow *Upper Case with Underscores* rules. All letters are in upper case and words have an underscore separating them.

- Enum value
- Constant

Examples:

```
enum SubscriptionState {ACTIVE, SUSPENDED, INVALID};
const string JOB_ID = "JOB_ID";
```

D.2.4 Naming IDL Sequence Types

Typically a new type declared as an IDL sequence of another type will have the text '`List`' appended to the name of the base type. Another convention is to declare such types as unordered sequences or ordered sets for consistency with ASN.1 notation. In this case they should have the '`Seq`' or '`Set`' (instead of '`List`') appended respectively.

Example of an 'ordered set':

```
typedef sequence <SubscriptionId> SubscriptionIdSet;
```

D.3 Interface IRP

Every Interface IRP should have 3 IDL modules (each specified in a separate IDL file):

```
module YyyIRPConstDefs {…};  // no change from Rel-5 practice.

module YyyIRPSystem {…}; // no change from Rel-5 practice.

module YyyIRPNotifications {…}; // new compared to Rel-5 practice
```

The first module defines all necessary IDL constructs, such as constant strings and type definitions, for the methods and notifications. The second module defines the methods. The third module defines the notifications.

D.3.1. Constant String and Type Definitions

This first module defines all necessary IDL constructs used by the methods (defined in the second module) and notifications (defined in the third module). The name of this module is `YyyIRPConstDefs` where `Xxx` is the name of the subject Interface IRP. An example is '`PMIRPConstDefs`'.

Within this module, define data types used in the methods.

Also, define the data types of the attribute values used in the notifications.

CORBA SS authors should always check the generic types defined in '`ManagedGenericIRPConstDefs`' before creating a new type.

For the attribute names of the structured notifications, define an interface `AttributeNameValue` that captures the string definitions. Make sure these definitions do not clash with those defined for the notification header, i.e. notification id, event time, system DN, managed object class and managed object instance (see `NotificationIRPNotification::Notify`).

An example from `PMIRPConstDefs`:

```
/**
 * This block identifies attributes which are included as part of the
 * PMIRP. These attribute values should not
 * clash with those defined for the attributes of notification
 * header (see IDL of Notification IRP).
 */

interface AttributeNameValue
{
    const string JOB_ID = "JOB_ID";
    const string JOB_STATUS = "JOB_STATUS";
    const string REASON = "REASON";
    const string MONITOR_ID = "MONITOR_ID";
    const string MONITOR_STATUS = "MONITOR_STATUS";
};
```

D.3.2 Operations

The second module defines the methods. The name of the module is `YyyIRPSystem` where `Yyy` is the name of the subject Interface IRP. An example is `AlarmIRPSystem`.

At the beginning of this module, define all required exceptions. Naming conventions for `exception` are covered in D.2.1 above. CORBA SS authors should always check if the generic exceptions defined in the `ManagedGenericIRPSystem` can be reused before declaring new `exception` types.

Then define one interface called `YyyIRP` encapsulating all methods of the subject `Yyy` Interface IRP. If the subject Interface IRP IS specifies that its `YyyIRP` inherits from `XxxIRP`, then reflect the inheritance relation in the interface definition. The following is an example of `AlarmIRP` that inherits from `ManagedGenericIRP`.

```
module AlarmIRPSystem
{
…
…
interface AlarmIRP : ManagedGenericIRPSystem:: ManagedGenericIRP {…};
…
};
```

Naming conventions for operations are covered in D.2.2 above.

D.3.3 Notifications

Use a separate module to define the notifications. The name the module is `YyyIRPNotifications` where `Yyy` is the name of the subject Interface IRP. Examples are `KernelCMIRPNotifications` and `PMIRPNotifications`.

For `NotificationIRPNotifications`, do:

- Define one IDL interface `Notify`. Capture the four constant strings that are the names of the four NV (name value) pairs of `filterable_body_field` of the CORBA structured event. These four CORBA NV pairs are mapped from the five notification header attributes (defined by the Notification IRP IS), i.e. the `objectClass, objectInstance, notificationId`, `eventTime` and `systemDN`.

For `YyyIRPNotifications` where `Yyy` is not `Notification`, do:

- At the beginning of this module, define the const strings for the notification types that correspond to the set of notifications specified by (and not inherited by and not imported by) the subject Interface IRP.
- Then define a number of IDL interfaces corresponding to notifications specified in the subject Interface IRP. These interfaces should inherit from `NotificationIRPNotifications::Notify`. Within each interface, the first IDL statement defines the notification type (that is used as the second field of the fixed header of the structured notification). The second and subsequent IDL statements define the attribute names of this notification type, excepting those already defined by `NotificationIRPNotifications::Notify`. The data type of the attribute value, which is defined in `YyyIRPConstDefs`, should be mentioned in the comment block of this IDL statement.
- Then define a number of IDL interfaces corresponding to notifications imported, if any. These interfaces should inherit from the imported interface. An example is `interface NotifyObjectCreation : KernelCMIRPNotifications:: NotifyObjectCreation`. Within this interface, define all necessary IDL constructs, if any, which are not defined in the imported interface. This interface may contain no IDL statement if the IDL constructs defined in the imported interface are sufficient. For each interface imported, insert a comment 'The first field of this notification carries the IRPVersion of this CORBA SS.'

- There is no need to re-define interfaces for notifications that are already specified in other Interface IRP, and from which the subject IRP inherits.

The following is an extract from `PMIRPNotifications`.

```
module PMIRPNotifications
{

   const string ET_MEASUREMENT_JOB_STATUS_CHANGED = "notifyMeasurementJobStatusChanged";
   const string ET_THRESHOLD_MONITOR_STATUS_CHANGED = "notifyThresholdMonitorStatusChanged";

   interface NotifyMeasurementJobStatusChanged: NotificationIRPNotifications::Notify
   {
     const string EVENT_TYPE = ET_MEASUREMENT_JOB_STATUS_CHANGED;

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

     …
     …
   };


   interface NotifyXXX : NotificationIRPNotifications::Notify
   {
       …
   };

   …
};
```

D.4 NRM IRP

Use one module to define the IDL constructs for the managed object classes. The name of this module is `XxxNRIRPConstDefs` where `Xxx` is the name of the subject NRM IRP.

An example is `UtranNRIRPConstDefs`.

Within the module, define a set of IDL interfaces each of which corresponds to a managed object class specified. The interface definition respects the inheritance relation specified. An example of managed object class `RncFunction`, which inherits from `GenericNRIRPConstDefs::ManagedFunction`, is shown below.

```
module UtranNRIRPConstDefs
{
…

/**
       *  Definitions for MO class RncFunction
       */
      interface RncFunction : GenericNRIRPConstDefs::ManagedFunction
      {
         const string CLASS = "RncFunction";

         // Attribute Names
         //
         const string rncFunctionId = "rncFunctionId";

         const string mcc= "mcc";
         const string mnc= "mnc";
         const string rncId= "rncId";
      };
…
};
```

Annex E (normative): XSD Use Cases

E.1 Background

3GPP defines a number of IOCs, say in Release-N 3GPP also defined an XSD schema that can capture these IOCs, in Release-N

Release-N+1 authors can/may also extend these Release-N IOCs to capture the newly agreed capabilities and publish them as Release-N+1 IOCs. Authors also defined a Release-N+1 XSD schema.

The Use Cases here illustrate the capabilities of the Release-N+1 XSD schema.

The '?' in the Use Case tables are for further study.

E.2 Use Cases Set 1

Suppose the old-XSD has SubNetwork (SN) containing ManagedElement (ME).

Suppose the new-XSD has enhanced-SN containing enhanced-ME.

Suppose we want the XML-instance-documents (doc-1, doc-2, etc) to contain instances as identified below:

- Doc-1 has SN instance containing ME instances
- Doc-2 has SN instance containing enhanced-ME instances
- Doc-3 has SN instance containing ME instances and enhanced-ME instances
- Doc-4 has enhanced-SN containing enhanced-ME instances
- Doc-5 has enhanced-SN containing ME instances
- Doc-6 has enhanced-SN containing enhanced-ME instances and ME instances

Can a XML-instance-doc-creator (IRPAgent or IRPManager) produce the doc (column) using the XSD (row) identified?

	Doc-1	Doc-2	Doc-3	Doc-4	Doc-5	Doc-6
Old-XSD	?	?	?	?	?	?
New-XSD	?	?	?	?	?	?

Can a XML-instance-doc-reader (IRPAgent or IRPManager) validate the doc (i.e. confirm that the document is well-formed) (column) using the XSD (row) identified?

	Doc-1	Doc-2	Doc-3	Doc-4	Doc-5	Doc-6
Old-XSD	?	?	?	?	?	?
New-XSD	?	?	?	?	?	?

E.3 Use Cases Set 2

Suppose one old-XSD has SubNetwork (SN) containing ManagedElement (ME) that in turn, contained an RNCFunction (RNC) defined by the other Old-XSD.

Suppose the new-XSD has enhanced-SN containing enhanced-ME. Suppose also another new-XSD has enhanced-RNC.

- Doc-7 has ME instance containing RNC instances
- Doc-8 has ME instances containing enhanced-RNC instances
- Doc-9 has ME instances containing RNC instances and enhanced-RNC instances
- Doc-10 has enhanced-ME instances containing enhanced-RNC instances
- Doc-11 has enhanced-ME instances containing RNC instances
- Doc-12 has enhanced-ME instances containing RNC instances and enhanced-RNC instances

Can a XML-instance-doc creator (IRPAgent or IRPManager) produce the doc (column) using the XSD (row) identified?

	Doc-7	**Doc-8**	**Doc-9**	**Doc-10**	**Doc-11**	**Doc-12**
Old-XSD	?	?	?	?	?	?
New-XSD	?	?	?	?	?	?

Can a reader (IRPAgent or IRPManager) validate the doc (i.e. confirm that the document is well-formed) (column) using the XSD (row) identified?

	Doc-1	**Doc-2**	**Doc-3**	**Doc-4**	**Doc-5**	**Doc-6**
Old-XSD	?	?	?	?	?	?
New-XSD	?	?	?	?	?	?

Annex F (informative): Style Guide for XSD

This annex is the style guide for writing XSD statements for Interface IRP and NRM IRP.

Editor's Note: This annex is For Further Study.

Annex G (normative): IOC Properties, Inheritance and Import

G.1 Property

The properties of an IOC (excluding Support IOC) are specified in terms of the following:

a) An IOC attribute(s) including its semantics and syntax, its legal value ranges and support qualifications. The IOC attributes are not restricted to Configuration Management but also include those related to, for example, 1) Performance Management (i.e., measurement types), 2) Trace Management and 3) Accounting Management.

b) The non-attribute-specific behaviour associated with an IOC (see Note 1).

NOTE 1: As an example, the Link between MscServerFunction and CsMgwFunction is optional. It is mandatory if the MscServerFunction instance belongs to one ManagedElement instance while the CsMgwFunction instance belongs to another ManagedElement instance. This Link behaviour is a non-attribute-specific behaviour. It is expected that this behaviour, like others, will be inherited.

c) An IOC relationship(s) with another IOC(s).

d) An IOC notification type(s) and their qualifications.

e) An IOC"s relation with its parents (see Note 2). There are three mutually exclusive cases:

1) The IOC is abstract and no parents have yet been designated.

2) The IOC is abstract and all of the possible parent(s) have been designated and whether subclass IOCs can be designated as a root IOC.

3) The IOC is not abstract and all of the possible parent(s) have been designated and whether the IOC can be designated as a root IOC.

An IOC instance is either a root IOC or it has one and only one parent. Only 3GPP SA5 may designate an IOC class as a potential root IOC. Currently, only SubNetwork, ManagedElement or MeContext IOCs are potential root IOCs.

NOTE 2: The parent and child relation in this clause is the parent name-containing the child relation.

f) An IOC"s relation with its children. There are three mutually exclusive cases:

1) An IOC shall not have any children (name-containment relation) IOCs.

2) An IOC can have children IOC(s). The maximum number of instances per children IOC can be specified. An IOC may designate that vendor specific objects are not allowed as children IOCs.

3) An IOC can only have the specific children IOC(s) (or their subclasses). The maximum number of instances per children IOC can be specified. An IOC may designate that vendor specific objects are not allowed as children IOCs.

g) Whether An IOC can be instantiated or not (i.e., whether An IOC is an abstract IOC).

h) An attribute for naming purpose.

i) An optional attribute for holding IOC name. An IOC name is unique among all 3GPP-defined IOCs and vendor-defined IOCs. An IOC name must be unambiguous.

Editor's Note: No agreement on above point i) achieved yet. This is FFS.]

G.2 Inheritance

An IOC (the subclass) inherits from another IOC (the superclass) in that the subclass shall have all the properties of the superclass.

The subclass can change the inherited support-qualification(s) from optional to mandatory but not vice versa. The subclass can change the inherited support-qualification from conditional-optional to conditional-mandatory but not vice versa.

An IOC can be a superclass of many IOC(s). A subclass cannot have more than one superclass.

The subclass can:

a) Add (compared to those of its superclass) unique attributes including their behaviour, legal value ranges and support-qualifications. Each additional attribute shall have its own unique attribute name (among all added and inherited attributes).

b) Add non-attribute behaviour on an IOC basis. This behaviour may not contradict inherited superclass behaviour.

c) Add relationship(s) with IOC(s). Each additional relationship shall have its own unique name (among all added and inherited relations).

d) Add additional notification types and their qualifications.

e) Designate all of the possible parent(s) (and their subclasses) if the superclass has Property-e-1 such that an IOC will have Property-e-2 or Property-e-3. Restrict possible parent(s) (and their subclasses) and/or remove the capability of the subclass from being a root IOC, if the superclass has Property-e-2 or Property-e-3.

f) Add children IOC(s) if the superclass has Property-f-2 such that an IOC will have Property-f-3. Restrict the allowed children IOC(s) (or their subclasses) if the superclass has Property-f-3.

g) Specify whether an IOC can be instantiated or not (i.e. the IOC is an abstract IOC).

h) Restrict the legal value range of a superclass attribute that has a legal value range.

G.3 Import

NRM IRP specifications define IOCs. To facilitate re-use of NRM IOC definitions among IRP specifications, an import mechanism is used by one NRM IRP (called the subject NRM IRP) specification to reuse IOC definition defined in another NRM IRP specification. When the subject NRM IRP specification imports an IOC, it cannot change the imported IOC property. If it requires changes to the imported IOC, it must use inheritance to define its own new class.

Annex H (informative): Change history

Change history								
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Cat**	**Old**	**New**
Mar 2006	SP-31	SP-060099	0006	--	Extension/Generalization of the IRP definition and concept (OAM7-NIM-NGN)	C	6.5.0	7.0.0
Dec 2006	SP-34	SP-060723	0007	--	Correct existing definitions of the IRPs and gather all IRP-related definitions	F	7.0.0	7.1.0
Jan 2007	--	--	--	--	Editorial: added two returns in 3.1 Definitions	--	7.1.0	7.1.1
Mar 2007	SP-35	SP-070045	0008	--	Delete the incorrect reference	F	7.1.1	7.2.0
Jun 2007	SP-36	SP-070309	0009	--	Identify the use case of XSD schema	B	7.2.0	8.0.0
Jun 2007	SP-36	SP-070309	0010	--	Add IOC Property Inheritance and Import definitions	B	7.2.0	8.0.0
Mar 2008	SP-39	SP-080058	0012	--	Generalization of the IRP definition for NGN management - Align with 32.101 and TISPAN	A	8.0.0	8.1.0
Jun 2008	SP-40	SP-080329	0013	--	Remove CMIP and add SOAP reference as supported technology for IRP Solution Sets - Align with 32.101	F	8.1.0	8.2.0

History

Document history		
V8.2.0	January 2009	Publication