ETSI TS 132 152 V12.0.0 (2014-10)

Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS);
LTE;
Telecommunication management;
Integration Reference Point (IRP) Information Service (IS)
Unified Modelling Language (UML) repertoire
(3GPP TS 32.152 version 12.0.0 Release 12)

Reference
RTS/TSGS-0532152vc00

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

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://ipr.etsi.org).

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.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**may not**", "**need**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

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	Requirements template

1 Scope

3GPP SA5 has chosen UML to capture systems behaviour in the IRP IS context.

UML provides a rich set of concepts, notations and model elements to model distributive systems. Usage of all UML notations and model elements is not necessary for the purpose of IRP IS specifications. This TS documents the necessary and sufficient set of UML notations and model elements, including the ones built by the UML extension mechanism <<stereotype>>, for use by 3GPP IRP IS authors. Collectively, this set of notations and model elements is called the 3GPP IRP IS modelling repertoire.

The selection of the UML notations and model elements in this repertoire is based on the needs of the existing 3GPP IRP IS specifications. Future IRP IS releases may require the use of additional UML notations or model elements.

IRP IS specifications shall employ the UML notation and model elements of this repertoire and may also employ other UML notation and model elements considered necessary. However, before any other UML notation and model elements may be employed in an approved 3GPP IRP specification, the other notation and model elements should be agreed for inclusion first in this repertoire.

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 32.150: "Telecommunication management; Integration Reference Point (IRP) Concept and definitions".

[4] 3GPP TS 32.151: "Telecommunication management; Integration Reference Point (IRP) Information Service (IS) template".

[5] OMG: "Unified Modelling Language Specification, Version 1.5". http://www.omg.org/spec/UML/1.5/

[6] 3GPP TS 32.602 "Telecommunication management; Configuration Management (CM); Basic CM Integration Reference Point (IRP): Information Service (IS)".

[7] 3GPP TS 32.612: "Telecommunication management; Configuration Management (CM); Bulk CM Integration Reference Point (IRP): Information Service (IS)".

[8] 3GPP TS 32.302: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Information Service (IS)".

[9] 3GPP TS 32.404: "Telecommunication management; Performance Management (PM); Performance measurements - Definitions and template".

[10] 3GPP TS 32.300: “Name convention for Managed Objects”.

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 32.150 [3] and the following apply:

Distinguished Name: See 3GPP TS 32.300 [10].

IRPAgent: See 3GPP TS 32.150 [3].

IRPManager: See 3GPP TS 32.150 [3].

Naming attribute: See 3GPP TS 32.300 [10].

Notification: See 3GPP TS 32.302 [8].

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 32.150 [3] and the following apply:

CM	Configuration Management
DN	Distinguished Name
GERAN	GSM/EDGE Radio Access Network
IRP	Integration Reference Point
IS	Information Service
NRM	Network Resource Model
OMG	Object Management Group
UML	Unified Modelling Language (OMG)

4 Requirements

IRPAgent can be characterized by several different but related models. The models can be exterior or interior to the IRPAgent. Exterior models are use case models and interior models are object models.

Current version of this TS focuses on the interior model aspects of IRPAgents.

The notation elements captured in this repertoire shall be used to model all aspects of NRM IRP IS (such as GERAN NRM IRP: IS) and (interface) IRP (such as Alarm IRP: IS).

All quotes are from [5].

Capitalized words are defined by various 3GPP IRP IS specifications or the reference [5].

5 Model Elements and Notations

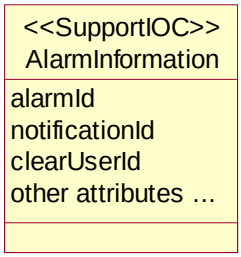
5.1 Basic model elements

UML defined a number of basic model elements. This subclause lists the selected subset for use in the repertoire. The semantics of the selected ones are defined in [5].

5.1.1 attribute

See subclause 3.25 of [5].

This sample shows someattributes, listed as strings in the attribute compartment of the class `AlarmInformation`.



5.1.2 aggregation

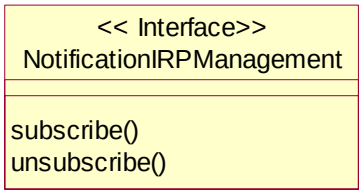
See subclause 3.43.2.5 of [5].

This sample shows a hollow diamond attached to the end of a path to indicate aggregation. The diamond is attached to the class that is the aggregate.

5.1.3 operation

See subclause 3.26 of [5].

This sample shows two operations, shown as strings in the operation compartment of class NotificationIRPManagement, that the instance of NotificationIRPManagement may be requested to perform. The operation has a name, e.g. subscribe and a list of arguments (not shown).



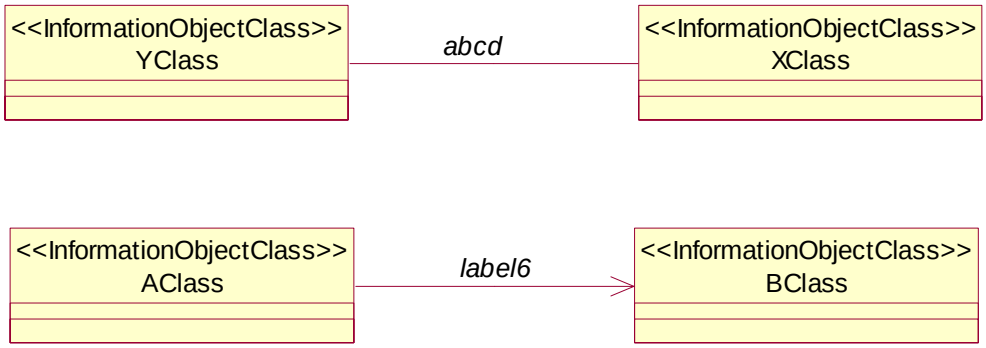
5.1.4 association and association name

See subclause 3.41 of [5].

These two samples show a binary association between exactly two model elements. The association can include the possibility of relating a model element to itself. The first sample shows a bi-directional association in that each model

element is aware of the other. The second sample shows a unidirectional association (shown with an open arrow at the target model element end) in that only the source model element is aware of the target model element and not vice-versa.

Association can be named, such as *abcd* and *label6* in the following samples.



5.1.5 realization relationship

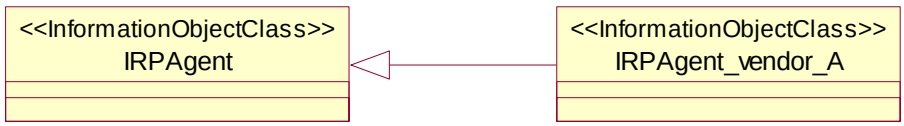
See subclause 2.5.2.1 of [5].

This sample shows the realization relationship between a model element `AlarmIRPOperations_1` and another model element, `AlarmIRP.` The latter (the target model element) implements the former. The target model element must be a <<Interface>>.

5.1.6 generalization relationship

See subclause 3.50 of [5].

This sample shows a generalization relationship between a more general element (the `IRPAgent`) and a more specific element (the `IRPAgent_vendor_A`) that is fully consistent with the first element and that adds additional information.



5.1.7 dependency relationship

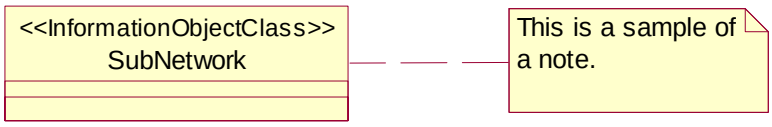
See subclause 3.51 of [5].

This sample shows that `BClass` instances have a semantic relationship with `AClass` instances. It indicates a situation in which a change to the target element will require a change to the source element in the dependency.

5.1.8 note

See subclause 3.11 of [5].

This sample shows a note, as a rectangle with a "bent corner" in the upper right corner. The note contains arbitrary text. It appears on a particular diagram and may be attached to zero or more modelling elements by dashed lines.

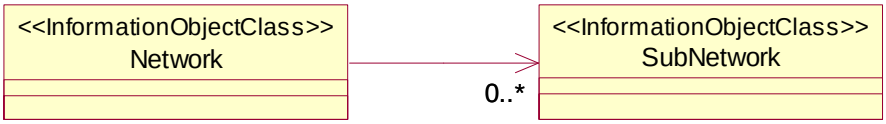


5.1.9 multiplicity, a.k.a. cardinality

See subclause 3.44 of [5].

This sample shows a multiplicity attached to the end of an association path. The meaning of this multiplicity is that one to many. `Network` instance(s) is associated with zero, one or more `SubNetwork` instances.

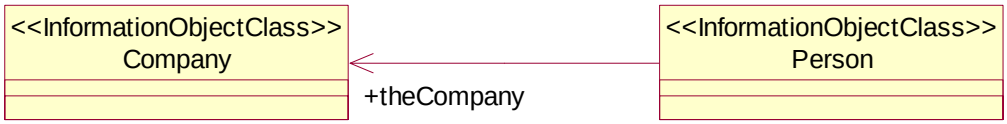
In Release 7 and earlier versions, the cardinality zero can indicate that the IOC has the so-called “transient state” characteristic. For example, it indicates that the instance is not yet created but it is in the process of being created. In Release 8 and onwards, the cardinality zero will not be used to indicate this characteristic since such characteristic is considered inherent in all IOCs. In Release 8 and onwards, all IOCs defined are considered to have such inherent “transient state” characteristics.



5.1.10 rolename

See subclause 3.43.2.6 of [5].

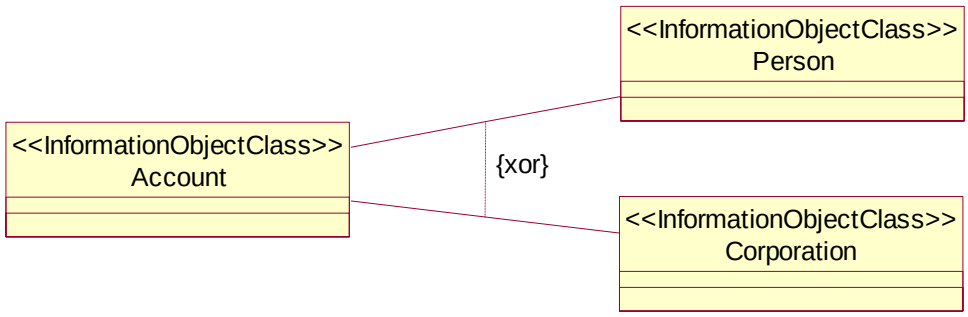
This sample shows a `Person` (say instance John) is associated with a `Company` (say whose DN is “Company=XYZ”). We navigate the association by using the opposite association-end such that John’s Person.theCompany would hold the DN, i.e."Company=XYZ". Use noun for the rolename.



5.1.11 Xor constraint

See subclause 2.5.2.3 and 3.42.5.1 of [5].

This sample shows an Account (e.g. account 0960) that is associated with a Person (e.g. John Smith) or a Corporation (e.g. ABC Inc).



5.2 Stereotype

This subclause lists all allowable stereotypes to be used in IRP IS specifications. One stereotype <<Interface>> is defined in [5]. This document lists it out for ease of reference and completeness. Other stereotypes are defined in this document.

Table: Stereotypes

Stereotype	Base Class	Affected Metamodel Elements
Interface	Class	
ProxyClass	Class	
Notification	Class	
Archetype	Classifier (subclause 2.5.2.10 of [5])	
InformationObjectClass	Classifier	
SupportIOC	Classifier	
agent-internal-usage	Association	
use	Association	
may use	Association	
may realize	Association	
names	Composition	--

5.2.1 <<Interface>>

Subclause 2.5.2.25 of [5]:

"An interface is a named set of operations that characterize the behaviour of an element. In the metamodel, an Interface contains a set of Operations that together define a service offered by a Classifier realizing the Interface. A Classifier may offer several services, which means that it may realize several Interfaces, and several Classifiers may realize the same Interface.

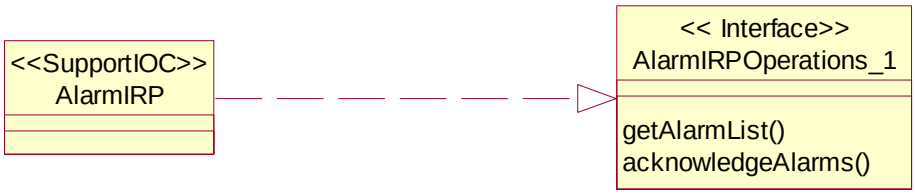
Interfaces may not have Attributes, Associations, or Methods. An Interface may participate in an Association provided the Interface cannot see the Association; that is, a Classifier (other than an Interface) may have an Association to an Interface that is navigable from the Classifier but not from the Interface."

Subclause 2.5.4.6 of [5]: "The purpose of an interface is to collect a set of operations that constitute a coherent service offered by classifiers. Interfaces provided a way to partition and characterize groups of operations. An interface is only a collection of operations with a name. It cannot be directly instantiated. Instantiable classifiers, such as class or use case, may use interfaces for specifying different services offered by their instances. Several classifiers may realize the same interface. All of them must contain at least the operations matching those contained in the interface. The specification of an operation contains the signature of the operation (i.e. its name, the types of the parameters and the return type). An interface does not imply any internal structure of the realizing classifier. For example, it does not include which algorithm to use for realizing an operation."

An operation may, however, include a specification of the effects [e.g. with pre and post-conditions] of its invocation.

5.2.1.1 Sample

This sample shows an `AlarmIRPOperations_1` <<Interface>> that has two operations. The input and output parameters of the operations are hidden (i.e. not shown). The `AlarmIRP` has a unidirectional mandatory realization relationship with the <<Interface>>.



<<Interface>> Notation

5.2.2 Void

5.2.3 <<ProxyClass>>

It is a form or template representing a number of <<InformationObjectClass>>. It encapsulates attributes, links, methods (or operations), and interactions that are present in the represented <<InformationObjectClass>>.

The semantics of a <<ProxyClass>> is that all behaviour of the <<ProxyClass>> is present in the represented <<InformationObjectClass>>. Since this class is simply a representation of other classes, this class cannot define its own behaviour other than those already defined by the represented <<InformationObjectClass>>.

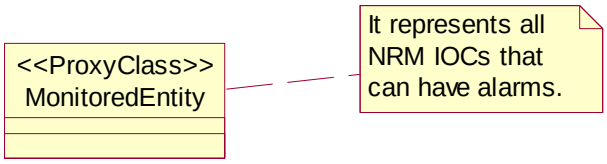
A particular <<InformationObjectClass>> can be represented by zero, one or more <<ProxyClass>> or <<Archetype>>. For example, the ManagedElement <<InformationObjectClass>> can have MonitoredEntity <<ProxyClass>> and ManagedEntity <<ProxyClass>>.

The attributes of the <<ProxyClass>> are accessible by the source entity that has an association with the <<ProxyClass>>.

5.2.3.1 Sample

This shows a <<ProxyClass>> named `MonitoredEntity`. It represents all NRM <<InformationObjectClass>> (e.g. `GgsnFunction` <<InformationObjectClass>>) whose instances are being monitored for alarm conditions.

Note that <<MonitoredEntity>> does not define any attribute. The attributes are already defined by all NRM <<InformationObjectClass>>.



<<ProxyClass>> Notation

See Annex A for more samples that use <<ProxyClass>>.

5.2.4 <<Archetype>>

It is a form or template representing common class properties (e.g. attributes, links, operations and interactions) of a number of represented <<InformationObjectClass>>.

The semantics of an <<Archetype>> is that all attributes, links operations and interactions encapsulated by the <<Archetype>> may or may not be present in the represented <<InformationObjectClass>>. The <<Archetype>>

represents a placeholder class that is most useful in technology neutral analysis models that will require further specification and/or mapping within a more complete construction model.

5.2.4.1 Sample

This shows an <<Archetype>> `StateManagement`. It also shows an <<InformationObjectClass>> `IRPAgent` that depends on this `StateManagement`. Note that the `StateManagement` has defined a number of attributes (not shown in the UML diagram).

The classes that depend on this `StateManagement` may or may not use all of the `StateManagement` attributes. In other words, at least one of the attributes of `StateManagement` is present in the `IRPAgent`. The precise set of `StateManagement` attributes used by the `IRPAgent` is specified in specification that defines the `IRPAgent`.

<<Archetype>> Notation

5.2.5 <<InformationObjectClass>>

It is the descriptor for a set of network resources and network management capabilities. Each <<InformationObjectClass>> represents a set of instances with similar structure, behaviour and relationships.

This <<InformationObjectClass>> and other information classes such as <<Interface>> are mapped into technology specific model elements. The mapping of IS modelling constructs to technology specific modelling constructs are captured in the corresponding IRP Solution Set specifications.

The name of an <<InformationObjectClass>> has scope within the 3GPP IRP IS document in which it is specified and the name must be unique among all <<InformationObjectClass>> names within that 3GPP IRP IS document. The IRP IS document name is considered in the similar way as the UML Package-name.

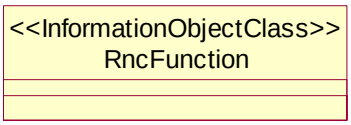
The <<InformationObjectClass>> is identical to UML *class* except that it does not include/define methods or operations.

Subclause 3.22.1 of [5]: "A *class* represents a concept within the system being modelled. Classes have data structure and behaviour and relationships to other elements."

See Annex B for information on various application of this Stereotype when compared to that of <<SupportIOC>>.

5.2.5.1 Sample

This sample shows an `RncFunction` <<InformationObjectClass>>.



<<InformationObjectClass>> Notation

5.2.6 <<use>> and <<may use>>

The <<use>> and <<may use>> are unidirectional associations. The target must be an <<Interface>> or <<Notification>>.

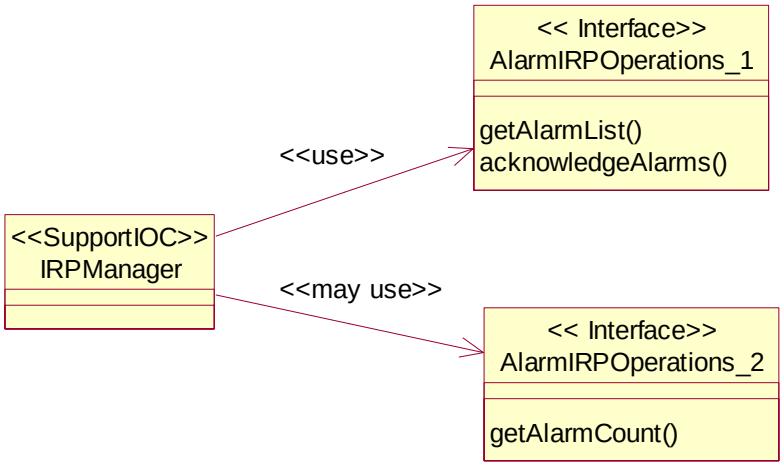
In the case where the target is <<Interface>>, the <<use>> states that the source class must have the capability to use the target <<Interface>> in that it can invoke the operations defined by the <<Interface>>. Support of the capability by

the source entity is mandatory. The <<may use>> states that the source class may have the capability to use the target <<Interface>> in that it may invoke the operations defined by the <<Interface>>. Support of the capability by the source entity is optional.

In the case the target is <<Notification>>, the <<use>> states that the source class must be the originator of the notifications defined by the target <<Notification>>. Support of the capability by the source entity is mandatory. The <<may use>> states that the source class may be the originator of the notifications defined by the target <<Notification>>. Support of the capability by the source entity is optional.

5.2.6.1 Sample for target <<Interface>>

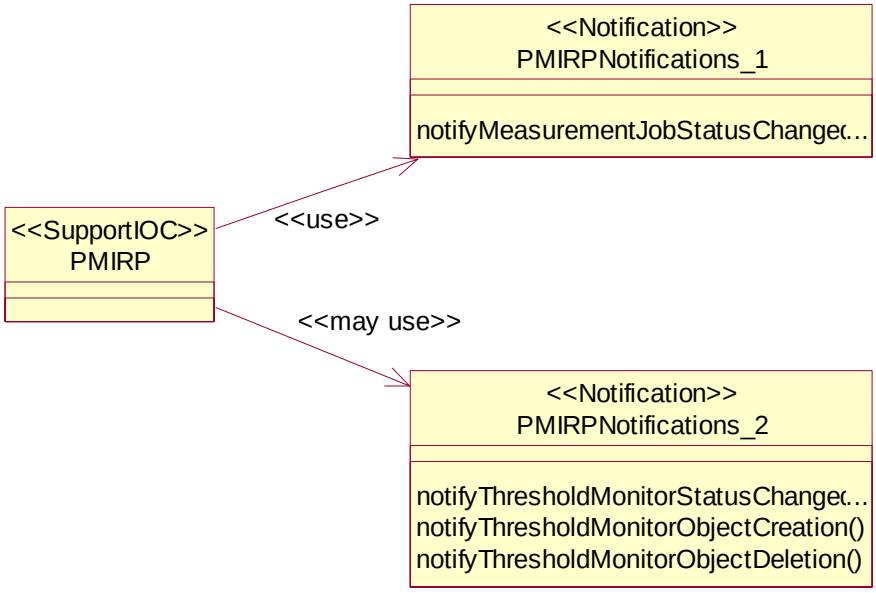
This shows that the `IRPManager` shall use the operations defined by `AlarmIRPOperations_1` and may use the operations defined by `AlarmIRPOperations_2`.



<<use>> and <<may use>> Notation for target <<Interface>>

5.2.6.2 Sample for target <<Notification>>

This shows that the `PMIRP` shall have the capability to emit or originate notifications defined by `PMIRPNotifications_1` and may have the capability to emit or originate notifications defined by `PMIRPNotifications_2`.



<<use>> and <<may use>> Notation for target <<Notification>>

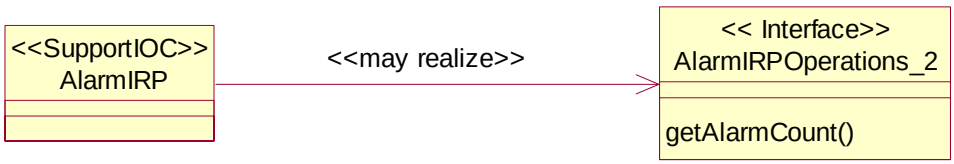
5.2.7 <<may realize>>

The <<may realize>> is an unidirectional association. The target must be an <<Interface>>. The <<may realize>> shows that the source entity may realize the operations defined by the target <<Interface>>.

Note that the UML basic element has defined the realize association (and therefore, there is no need to define a stereotype of such association). The realize association shows that the source entity must realize (or implement) the operations defined by the target <<Interface>>.

5.2.7.1 Sample

This shows that the `AlarmIRP` may realize the operation of `AlarmIRPOperations_2`.



<<may realize>> Notations

5.2.8 Void

5.2.9 <<names>>

It specifies a unidirectional composition. The target instance is uniquely identifiable, within the namespace of the source entity, among all other targeted instances of the same target classifier and among other targeted instances of other classifiers that have the same <<names>> composition with the source.

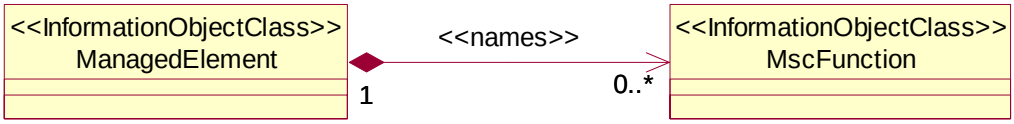
The source classifier and target classifier shall both have a naming attribute.

Composition used as the act of name containment provides a semantic of a whole-part relationship between the domain and the named elements that are contained, even if only by name. From the management perspective access to the part is through the whole. Multiplicity shall be indicated at both ends of the relationship.

A target instance can not have multiple <<names>> with multiple sources, i.e. a target instance can not participate in or belong to multiple namespaces.

5.2.9.1 Sample

This shows that all instances of `MscFunction` are uniquely identifiable within a `ManagedElement` instance's namespace.



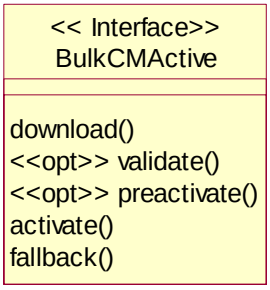
<<names>> Notation

5.2.10 <<opt>>

The <<opt>> (alternatively <<optional>>) enables the indication of optionality of attributes, parameters and operations (respectively) within the UML diagrams. The semantics of optionality are defined in TS 32.150 [3].

In the absence of the stereotype, the attribute, parameter, or operation in question is mandatory.

5.2.10.1 Sample



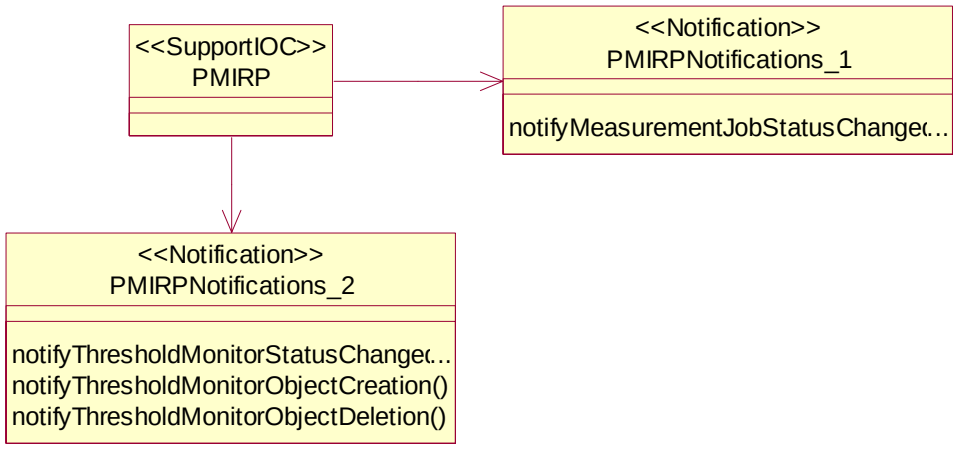
<<opt>> Notation for operations

5.2.11 <<Notification>>

<<Notification>> is a named set of notifications.

5.2.11.1 Sample

This sample shows a <<Notification>> named "`PMIRPNotifications_1`" and another <<Notification>> named "`PMIRPNotifications_2`". Both of them have notification(s). An example of a notification can be notifyMeasurementJobStatusChanged().



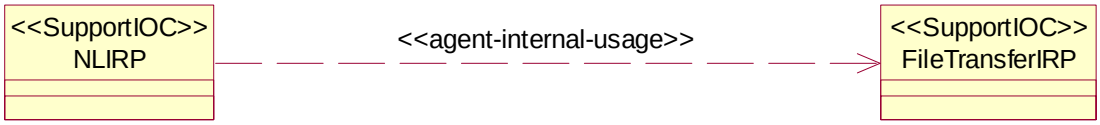
<<Notification>> Notation

5.2.12 <<agent-internal-usage>>

This is a unidirectional association. The source passes network management information to target. The source and target are entities or processes running in different IRP instances such as `AlarmIRP`, `PMIRP`. The instances may be name-contained by the same `IRPAgent` or different `IRPAgent` instances. The precise network management information passed and the information transfer mechanism are not standardized and are vendor-specific.

5.2.12.1 Sample

This shows that NLIRP (NotificationLog IRP) can pass some network management information to FTIRP.



<<agent-internal-usage>> Notation

5.2.13 <<SupportIOC>>

It is the descriptor for a set of management capabilities.

The <<`SupportIOC`>> is an extension of UML *class*. See Annex B for the differences between <<`InformationObjectClass`>> and <<`SupportIOC`>>.

Subclause 3.22.1 of [5]: "A *class* represents a concept within the system being modelled. Classes have data structure and behaviour and relationships to other elements."

5.2.13.1 Sample

This sample shows an `AlarmList` <<SupportIOC>>.

<<SupportIOC>> Notation

5.3 Void

5.4 Association classes

Subclause 3.46 of OMG: "Unified Modelling Language Specification defines an association class as

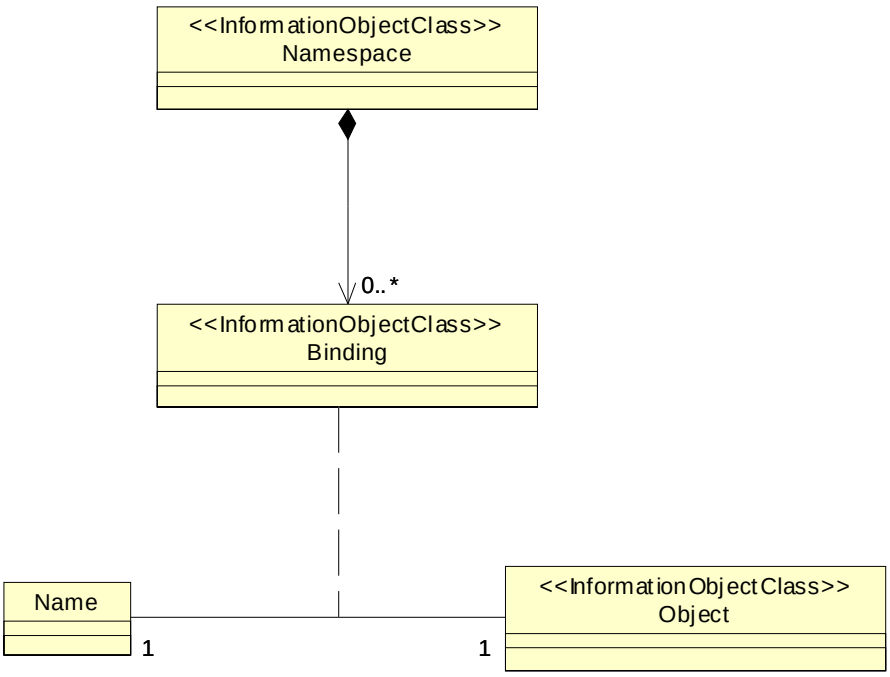
> "An association class is an association that also has class properties (or a class that has association properties). Even though it is drawn as an association and a class, it is really just a single model element.".

5.4 Association classes

Subclause 3.46 of OMG: "Unified Modelling Language Specification, Version 1.51" [5] http://www.omg.org/technology/documents/formal/uml.htmd defines an association class as

> "An association class is an association that also has class properties (or a class that has association properties). Even though it is drawn as an association and a class, it is really just a single model element.".

Association classes are appropriate for use when an «InformationObjectClass» needs to maintain associations to several other «InformationObjectClass»'s and there are relationships between the members of the associations within the scope of the "containing" «InformationObjectClass». For example, a namespace maintains a set of bindings, a binding ties a name to an object. A Binding «IOC» can be modelled as an Association Class that provides the binding semantics to the relationship between a name and some other «InformationObjectClass». This is depicted in the following figure (exemplary only, not taken from another 3GPP specification).



Example of an Association Class

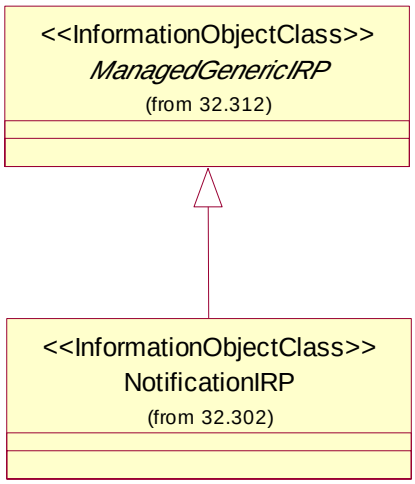
5.5 Abstract Class

It specifies a <<InformationObjectClass>> as a base class to be inherited by subclasses. An abstract class can not be.

Abstract class notation is the use of italics in the class name of the corresponding <<InformationObjectClass>> in the diagram.

5.5.1 Sample

This shows that ManagedGenericIRP is an abstract <<InformationObjectClass>>.



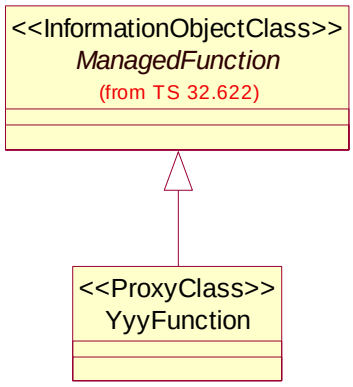
Abstract Class Notation

Annex A (informative): Samples of using <<ProxyClass>> to model Link related IOCs

A.1 First Sample

This shows a <<ProxyClass>> named YyyFunction. It represents all IOCs listed in the Note under the UML diagram. All the listed IOCs, in the context of this sample, inherit from ManagedFunction IOC.

The use of <<ProxyClass>> eliminates the need to draw multiple UML <<InformationObjectClass>> boxes, i.e. those whose names are listed in the Note, in the UML diagram.



Note: The YyyFunction <<ProxyClass>> represents `AsFunction, AucFunction, BgFunction, …`

<<ProxyClass>> Notation Sample A.1

A.2 Second Sample

This shows a <<ProxyClass>> named YyyFunction. It represents all IOCs listed in the Note right under the UML diagram. All the listed IOCs, in the context of this sample, have link (internal and external) relations.

The actual names of the IOC represented by InternalYyyFunction <<ProxyClass>> and by the ExternalYyyFunction <<ProxyClass>> are listed under the subsection of X.Y of the associated YyyFunction. For example, under X.Y.1 for AsFunction, two paragraphs are added to list all peer internal entities and external entities that are linked with AsFunction. See sample in quotation below that is using AsFunction as a sample for YyyFunction.

The actual names of the IOC represented by Link_a_z <<ProxyClass>> and by ExternalLink_a_z <<ProxyClass>> are listed under the subsection of X.Y of the associated YyyFunction. For example, under X.Y.1 for AsFunction, two paragraphs are added to list the names of the IOCs represented by Link_a_z and by ExternalLink_a_z. See the quoted text below that is using AsFunction as a sample for YyyFunction.

“

X.Y.1 AsFunction

X.Y.1.1 Definition

This IOC represents As functionality. For more information about the As, see 3GPP TS 23.002 [15].

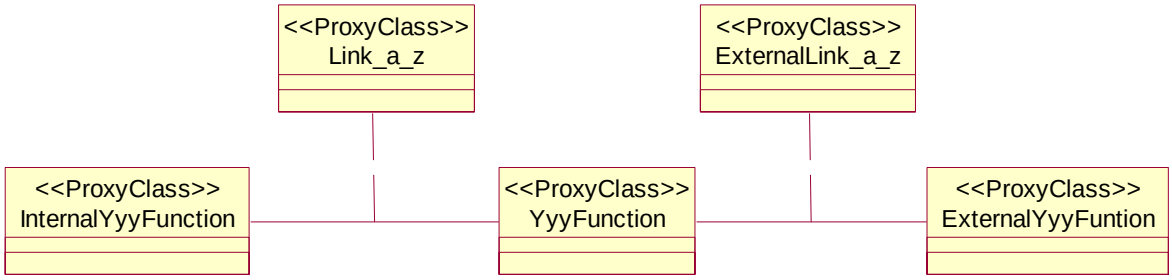
The linked InternalYyyFunction <<ProxyClass>> represents SlsFunction, CscfFunction, HlrFunction ...

The linked ExternalYyyFunction <<ProxyClass>> represents …

The Link_a_z <<ProxyClass>> represents Link_As_Scscf, Link_Bgcf_Scscf …

The ExternalLink_a_z <<ProxyClass>> represents …

“



NOTE: The ‘Yyy’ of YyyFunction <<ProxyClass>> represents AsFunction, AucFunction, …

<<ProxyClass>> Notation Sample A.2

Annex B (normative): Application of <<InformationObjectClass>> and <SupportIOC>>

The <<InformationObjectClass>> and <<SupportIOC>> are stereotypes. These two stereotypes serve similar purpose in that each is a named set of network resource properties. However, their applications, in the context of supporting network management over Itf-N, can be different. This Annex highlights their similarities and differences of such application.

	<<InformationObjectClass>>	**<<SupportIOC>>**
Can it be an abstract class?	Yes	Yes
Can it be a concrete class?	Yes	Yes
Can it inherit from <<InformationObjectClass>>?	Yes	No, except for <<IOC>> Top.
Can it inherit from <<SupportIOC>>?	No	Yes
Can it be name-contained by <<InformationObjectClass>>?	Yes	Yes
Can it be name-contained by <<SupportIOC>>?	No	Yes
Can an instance have a DN?	<<InformationObjectClass>> must be a class of a naming-tree meaning all its instances must have a DN.	<<SupportIOC>> may be used by specification author for a class within a naming-tree. If so, it means that all its instances will have a DN.
Can IRPManager use operations of Basic CM IRP [6] and Bulk CM IRP [7] to access the information in an instance?	IRPManager can use the Basic CM IRP and Bulk CM IRP operations to access information of all <<InformationObjectClass>> defined in all NRM IRPs, in accordance to the qualifier values of the <<InformationObjectClass>>.	IRPManager can use the Basic CM IRP and Bulk CM IRP operations to access information of instances of <<SupportIOC>> defined in their respective Interface IRP (i.e. Basic CM IRP or Bulk CM IRP), in accordance to the qualifier values of the <<SupportIOC>>. IRPManager can not use the Basic CM IRP and Bulk CM IRP operations to access information of instances of <<SupportIOC>> not defined in their respective Interface IRP (i.e. Basic CM IRP or Bulk CM IRP).
Can IRPManager use operations of Interface IRP, except Basic CM IRP [6] and Bulk CM IRP [7] (e.g. Alarm IRP), to access the information?	No	IRPManager can use the Interface IRP operations to access information of <<SupportIOC>> defined in their respective Interface IRP, in accordance to qualifier values of the <<SupportIOC>>. IRPManager can not use the Interface IRP operations to access information of <<SupportIOC>> not defined in their respective Interface IRP.
Can IRPManager receive information via Notification [8] whose objectClass and objectInstance parameters carry the instance DN?	Yes. The types of notification emitted are shown by the Notification Table associated with the class definition.	Yes if <<SupportIOC>> is a class of a naming-tree. The types of notification emitted are shown by the Notification Table associated with the class definition. No if <<SupportIOC>> is not a class of a naming-tree.
Measurement [9]	Measurements can be associated with <<InformationObjectClass>> instances.	Measurements can be associated with <<SupportIOC>> instances if <<SupportIOC>> class is used in a naming-tree.

Annex C (informative): Change history

Change history								
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Cat**	**Old**	**New**
Dec 2003	SA_22	SP-030615	--	--	Submitted to TSG SA#22 for Information	--	1.0.0	
Mar 2004	SA_23	SP-040115	--	--	Submitted to TSG SA#23 for Approval	--	2.0.0	6.0.0
Oct 2006	--	--	--	--	Corrected TS ver. and Rel #; added GSM logo in line with 41.101	--	6.0.0	6.0.1
Jun 2004	SA_24	SP-040242	0001	--	UML Repertoire Updates (Associations)	C	6.0.1	6.1.0
Mar 2005	SA_27	SP-050023	0002	--	Add “abstract” adornment	F	6.1.0	6.2.0
Jun 2005	SA_28	SP-050284	0003	--	Add the stereotype “Notification” to UML repertoire. Align with TS 32.412	F	6.2.0	6.3.0
Jun 2005	SA_28	SP-050284	0004	--	Correction of stereotype name capitalization and visibility qualifiers	F	6.2.0	6.3.0
Dec 2005	SA_30	SP-050729	0005	--	Apply IS template - Align with 32.151	F	6.3.0	7.0.0
Mar 2007	SA_35	SP-070045	0006	--	Add samples to illustrate the use of ProxyClass to model Link related objects	F	7.0.1	7.1.0
Mar 2007	SA_35	SP-070045	0007	--	Correct the references of IRPAgent and IRPManager	F	7.0.1	7.1.0
Mar 2007	--	--	--	--	Deleted reference to CMIP SS, discontinued from R7 onwards	--	7.0.1	7.1.0
Sep 2007	SA_37	SP-070614	0008	--	Modify usage of cardinality zero for IOC	C	7.1.0	8.0.0
Sep 2007	SA_37	SP-070614	0009	--	Remove visibility qualifier stereotype	C	7.1.0	8.0.0
Sep 2007	SA_37	SP-070614	0010	--	Add "agent internal usage" stereotype description	C	7.1.0	8.0.0
Sep 2009	SA_45	SP-090627	0011	-	Introduce SupportIOC Stereotype	F	8.0.0	9.0.0
Sep 2009	SA_45	SP-090534	0012	-	Align Repertoire with practice	F	8.0.0	9.0.0
Dec 2009	SA_46	SP-090719	0013	--	Add a new Annex B on Application of Stereotypes SupportIOC and IOC	F	9.0.0	9.1.0
Dec 2009	SA_46	SP-090719	0014	--	Clarify rules for <<names>> stereotype	F	9.0.0	9.1.0
Mar 2010	SA_47	SP-100035	0015	--	Clarify class support of notification re 32.152	F	9.1.0	9.2.0
Sep 2010	SA_49	SP-100489	0016	--	Introduction of additional basic model element {xor}	F	9.1.0	10.0.0
2012-09	-	-	-	-	-	Update to Rel-11 version (MCC)	10.0.0	**11.0.0**
2012-12	SA_58	SP-120797	0017	1	CR R11 32152 on SupportIOC v3	F	11.0.0	**11.1.0**
2013-12	SA_62	SP-130615	0018	2	Modification on the application of SupportIOC	F	11.1.0	**12.0.0**

History

Document history		
V12.0.0	October 2014	Publication