TSGS#28(05)0284

Source: **SA5 (Telecom Management)**

Title: **CR 32152 IRP Information Service (IS) Unified Modelling Language (UML) repertoire**

Document for: **Approval**

Agenda Item: **7.5.3**

Doc-1st-Level	Spec__#_	CR__#	R	Phase	Subject	Cat	Ver-Cur	Doc-2nd-Level	Workitem
SP-050284	32.152	0003	-	Rel-6	Add the stereotype “Notification” to UML repertoire. Align with TS 32.412	F	6.2.0	S5-056325	OAM-NIM
SP-050284	32.152	0004	-	Rel-6	Correction of stereotype name capitalization and visibility qualifiers	F	6.2.0	S5-056362	OAM-NIM

CR-Form-v7.1

CHANGE REQUEST

⌘ **32.152** CR **0003** ⌘rev **-** ⌘ Current version: **6.2.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network **X** Core Network **X**

Title:	⌘	Add the stereotype “Notification” to UML repertoire. Align with TS 32.412
Source:	⌘	SA5 Ericsson (ulf.hubinette@ericsson.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	13/05/2005
Category:	⌘	**F**
Release:	⌘	Rel-6

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

Use one of the following releases:
Ph2 *(GSM Phase 2)*
R96 *(Release 1996)*
R97 *(Release 1997)*
R98 *(Release 1998)*
R99 *(Release 1999)*
Rel-4 *(Release 4)*
Rel-5 *(Release 5)*
Rel-6 *(Release 6)*
Rel-7 *(Release 7)*

Reason for change:	⌘	The UML stereotype “Notification” is used in the PM IRP IS, but the corresponding stereotype definition is missing in the UML repertoire specified in TS 32.152.
Summary of change:	⌘	Add “Notification” as allowable stereotype to the UML repertoire. Replace example for the stereotype <<opt>>, which refers to obsolete PM IRP notifications and is using the symbol “#”. Correct reference and editorial errors.
Consequences if not approved:	⌘	3GPP SA5 IRP IS TSs would contain a stereotype not defined in the 32.152 UML repertoire.

Clauses affected:	⌘	2 and 5.2

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

⌘

Other comments:	⌘	This CR is based on the technically approved CR S5-056227 from SA5#41bis. It has been updated due to the TSG-SA#27 approval of version 6.2.0 of the TS.

Change in Clause 2

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~~1~~". http://www.omg.org/technology/documents/formal/uml.htm

End of change in Clause 2

Change in Clause 5.2

5.2 Stereotype

This subclause defines all allowable stereotypes that are summarized in the following table. Except <<Interface>>, <<Type>> and <<use>> (which are defined in [5]), all other stereotypes are extensions specifically designed for use in IRP IS specifications.

Table: Stereotypes

Stereotype	Base Class	Affected Metamodel Elements
Interface	Class	
Type	Class	
ProxyClass	Class	
Notification	Class	
Archtetype	Classifier (subclause 2.5.2.10 of [5])	
InformationObjectClass	Classifier	
use	Association	
may use	Association	
may realize	Association	
emits	Association	
names	Composition	--
opt (alternatively «optional»)	ModelElement	Attribute, Parameter, and Operation
%	3GPPVisibilityKind	--

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.

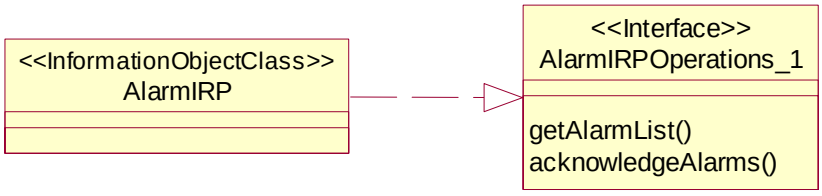
…

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. Insatiable 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 operation visibility is public (see definition of public visibility applicable to operation in subclause "visibility"). 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>>.



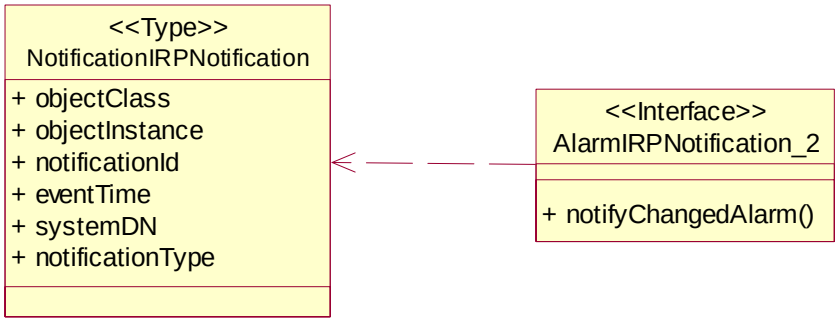
<<Interface>> Notation

5.2.2 <<Type>>

Subclause 3.28 of [5]: "[A Type is] a domain of objects together with the operations applicable to the objects, without defining the physical implementation of those objects. A Type may not contain any methods, maintain its own thread of control, or be nested. However, it may have Attributes and Associations. The Associations of a Type are defined solely for the purpose of specifying the behaviour of the Type's operations and do not represent the implementation of state data".

5.2.2.1 Sample

This sample shows the NotificationIRPNotification <<Type>> that specifies the five parameters (the notification header of Notification IRP). The AlarmIRPNotification_2 <<Interface>> depends (see the dependency relationship, a dashed open arrow line) on this <<Type>> for the construction of the notification emitted via the operation notifyChangedAlarm(). The visibility of attributes and operation in the example is public.



<<Type>> Notation

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>> are 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>>.

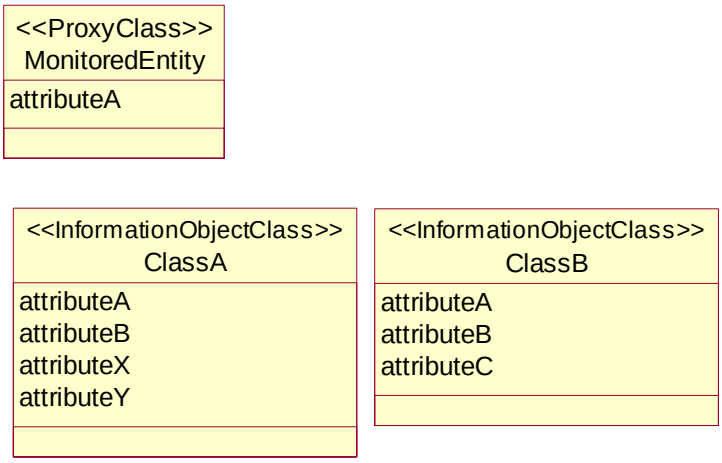
A particular <<InformationObjectClass>> can be represented by zero, one or more <<ProxyClass>> or <<Archetype>>. For example, the ManagedElement <<InformationObjectClass>> can have MonitoredEntity <<ProxyClass>> and ManagedEntity <<ProxyClass>>.

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. The MonitoredEntity plays the role of theMonitoredEntity.

Note that <<MonitoredEntity>> does not define attributeA. The attributeA is already defined by all <<InformationObjectClass>> represented by the <<MonitoredEntity>>, i.e. ClassA and ClassB.



<<ProxyClass>>

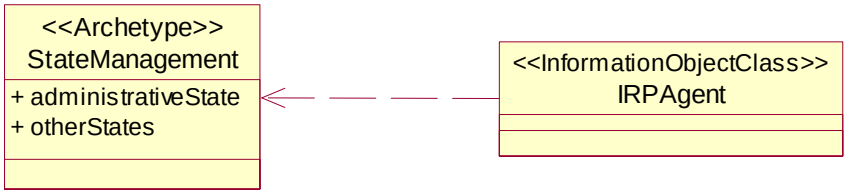
5.2.4 <<Archetype>>

It is a form or template representing a number of <<InformationObjectClass>>. It encapsulates attributes, links, operations, and interactions that are typical of the 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 a <<Archetype>> named StateManagement. It also shows a <<InformationObjectClass>> IRPAgent that depends on this StateManagement. Note that the StateManagement has defined a number of attributes, 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 the IRPAgent specification.



<<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 such as GDMO Managed Object Class for CMIP technology. The mapping of IS modelling constructs to technology specific modelling constructs are captured in the corresponding IRP Solution Set specifications.

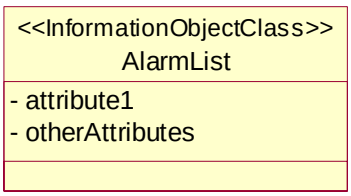
The name of a <<InformationObjectClass>> has scope within the 3GPP IRP IS document in which it is specified and the name must be unique among all <<InformationObjectClass>> names within that 3GPP IRP IS document. The IRP IS document name is considered in the similar way as the UML Package-name.

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."

5.2.5.1 Sample

This sample shows an AlarmList <<InformationObjectClass>>.



<<InformationObjectClass>>> Notation

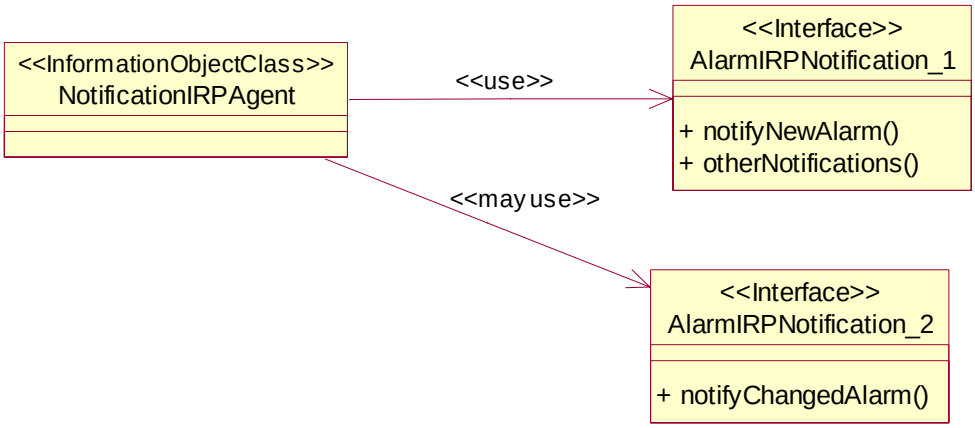
5.2.6 <<use>> and <<may use>>

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

The operations defined by the <<interface>> are visible across the itf-N.

5.2.6.1 Sample

This shows that the NotificationIRPAgent shall use the notifyNewAlarm and otherNotifications of AlarmIRPNotification_1 and may use the notifyChangedAlarm of AlarmIRPNotification_2.



<<use>> and <<may use>> Notation

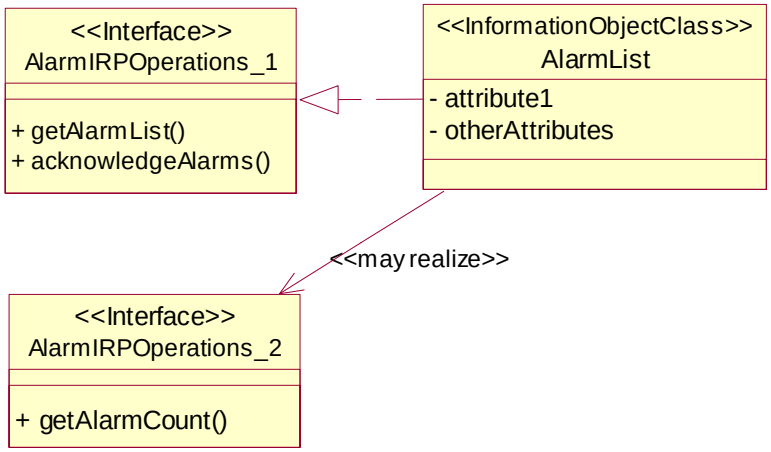
5.2.7 Relationship realize and <<may realize>>

The relationship realize and <<may realize>> are unidirectional association. The target must be an <<interface>>. The relationship "realize" shows that the source entity must realize the operations defined by the target <<interface>>. Realization of operations by the source entity is mandatory. The <<may realize>> shows the source entity may realize the operations defined by the target <<interface>>. Realization of the <<interface>> by the source entity is optional.

The operations defined by <<interface>> are visible across the itf-N.

5.2.7.1 Sample

This shows that the AlarmList shall realize (or support, implement) the two operations of AlarmIRPOperations_1 and may realize the operation of AlarmIRPOperations_2.



Relationship realize and <<may realize>> Notations

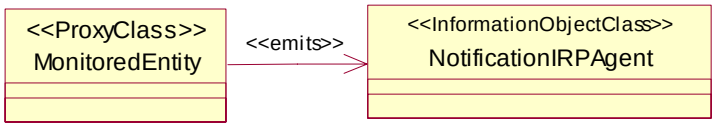
5.2.8 <<emits>>

This is a unidirectional association. The source sends information to target. In the case that the target is NotificationIRPAgent, the information will then carry the semantics of 3GPP notification (e.g. notifyObjectCreation, notifyNewAlarm) such that the target NotificationIRPAgent can construct the relevant 3GPP notification for reception by the NotificationIRPManager.

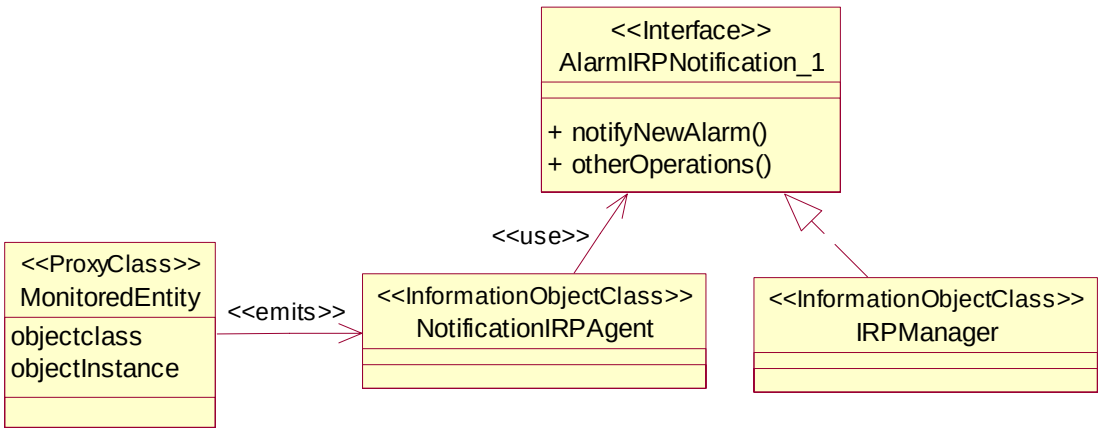
The visibility of the information passed by <<emits>> is always "IRPAgent Internal" (see subclause on "Visibility").

5.2.8.1 Sample

This shows the MonitoredEntity (e.g. a GgsnFunction instance) emits notifications that are received by the NotificationIRPAgent. The emission is not visible across the itf-N.



<<emits>> Notation



<<use>>, <<emits>> and realize relationship Notation

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.

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 on both ends of the relationship.

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

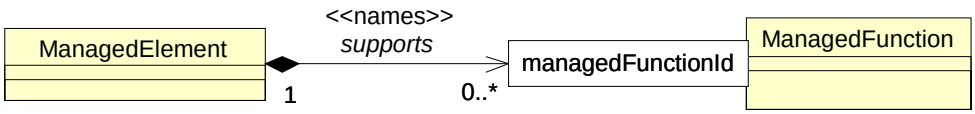
By convention, the name of the attribute in the target model element to hold part of the unique identification shall be formed by the name of the target class concatenated with "Id". There are two presentation options for the unique identification attribute of the class being named.

1) The use of the role qualifier allow the unique identification attribute to be attached to the target end of the <<names>> association (see the following figure).

2) The unique identification attribute may be indicated as a normal attribute within the class attribute compartment.

5.2.9.1 Sample

This shows that all instances of ManagedFunction are uniquely identifiable within the ManagedElement namespace. Note the use of the label supports in specifications is optional.

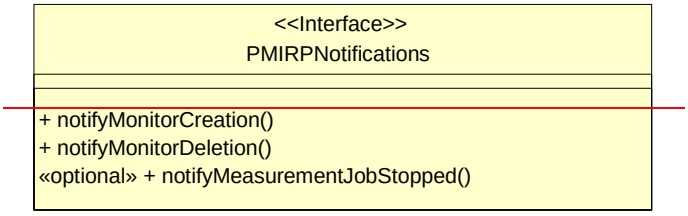


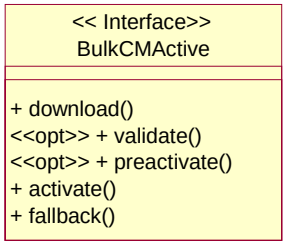
<<names>> Notation, Composition and explicit Qualifier

5.2.10 <<opt>>

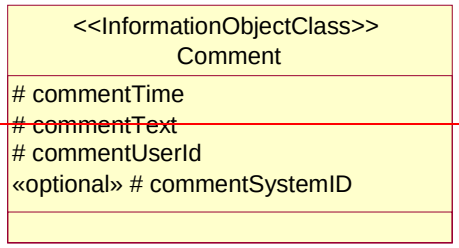
The <<opt>> (alternatively <<optional>>) enables the indication of optionality of attributes, parameters and operations (respectively) within the UML diagrams ~~within TS32 series documents~~. The semantics of optionality are clearly defined in TS 32.150 [3]. ~~within subclause 10.6.~~

In the absence of the ~~<<opt>>~~ stereotype, the attribute, parameter, or operation in question is mandatory.





~~(a)~~ Example of the use of optionality indicator for operations



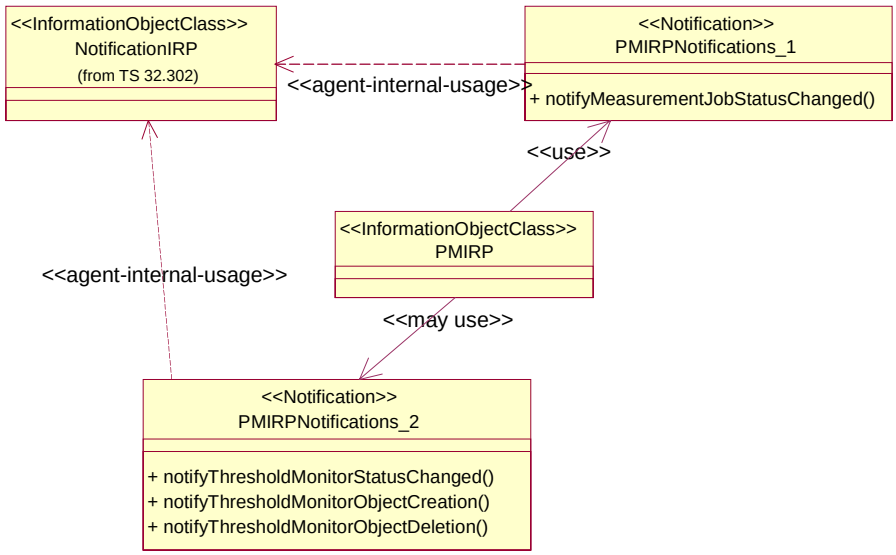
~~(b) Example of the use of optionality indicator for attributes~~

5.2.11 <<Notification>>

<<Notification>> is a named set of notifications. In the metamodel, a <<Notification>> contains a set of Notifications that together define a service offered by a Classifier realizing the <<Notification>>.

5.2.11.1 Sample

This sample shows a <<Notification>> named "PMIRPNotifications_1" that has one notification and a <<Notification>> named "PMIRPNotifications_2" that has three notifications.



End of Change in Clause 5.2
End of document

Annex A (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Jun 2004	S_24	SP-040242	001		UML Repertoire Updates (Associations)	6.0.0	6.1.0
Mar 2005	S_27	SP-050023	002	--	Add “abstract” adornment	6.1.0	6.2.0

Tdoc S5-056362

CR-Form-v7.1

CHANGE REQUEST

⌘ **32.152** CR **0004** ⌘ rev **-** ⌘ Current version: **6.2.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network **X** Core Network **X**

Title:	⌘	Correction of stereotype name capitalization and visibility qualifiers
Source:	⌘	SA5 Ericsson (ulf.hubinette@ericsson.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	13/5/2005
Category:	⌘	**F**
Release:	⌘	Rel-6

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

Use one of the following releases:
Ph2 *(GSM Phase 2)*
R96 *(Release 1996)*
R97 *(Release 1997)*
R98 *(Release 1998)*
R99 *(Release 1999)*
Rel-4 *(Release 4)*
Rel-5 *(Release 5)*
Rel-6 *(Release 6)*
Rel-7 *(Release 7)*

Reason for change:	⌘	Incorrect capitalization of some stereotypes. Inconsistent usage of visibility qualifiers. Unclear <<ProxyClass>> example. Obsolete statement about rel-5 operations.
Summary of change:	⌘	Correct capitalisation of stereotypes. Correct visibility qualifiers. Clarify <<ProxyClass>> example. Remove obsolete statement on rel-5 operations.
Consequences if not approved:	⌘	The IRP IS UML repertoire would remain incorrect, which could lead to problems with interpretation and interoperability.

Clauses affected:	⌘	5.2.1.1, 5.2.3, 5.2.3.1, 5.2.4, 5.2.5, 5.2.6, 5.2.7, 5.3

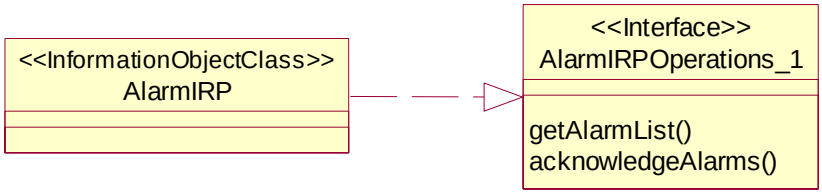
	Y	N		
Other specs affected: ⌘		**N**	Other core specifications	⌘
		N	Test specifications	
		N	O&M Specifications	

Other comments: ⌘

Change in Clause 5.2.1.1

5.2.1.1 Sample

This sample shows an AlarmIRPOperations_1 <<Interface>> that has two operations. The operation visibility is public (see definition of public visibility applicable to operation in subclause "visibility"). 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

End of Change in Clause 5.2.1.1

Change in Clause 5.2.3

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>> are 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 <<Archtype>>. 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>>.

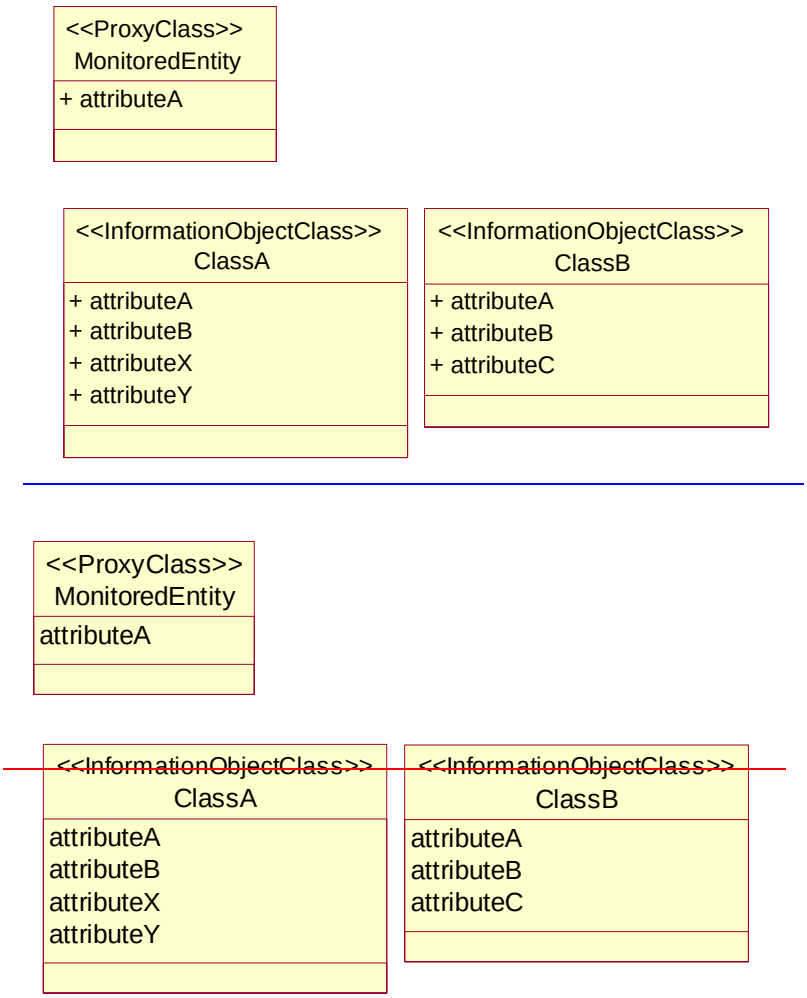
End of Change in Clause 5.2.3

Change in Clause 5.2.3.1

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. ~~The MonitoredEntity plays the role of theMonitoredEntity.~~

Note that <<MonitoredEntity>> does not define attributeA. The attributeA is already defined by all <<InformationObjectClass>> represented by the <<MonitoredEntity>>, i.e. ClassA and ClassB.



<<ProxyClass>>

End of Change in Clause 5.2.3.1

Change in Clause 5.2.4

5.2.4 <<Archetype>>

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

The semantics of an <<Aarchetype>> is that all attributes, links operations and interactions encapsulated by the <<Aarchetype>> 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.

End of Change in Clause 5.2.4

Change in Clause 5.2.5

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 <<I~~i~~nterface>> are mapped into technology specific model elements such as GDMO Managed Object Class for CMIP technology. The mapping of IS modelling constructs to technology specific modelling constructs are captured in the corresponding IRP Solution Set specifications.

The name of a <<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."

End of Change in Clause 5.2.5

Change in Clause 5.2.6

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

The <<use>> and <<may use>> are unidirectional associations. The target must be an <<I~~i~~nterface>>. The <<use>> states that the source class must have the capability to use the target <<I~~i~~nterface>> in that it can invoke the operations defined by the <<I~~i~~nterface>>. 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 <<I~~i~~nterface>> in that it may invoke the operations defined by the <<I~~i~~nterface>>. Support of the capability by the source entity is optional.

The operations defined by the <<I~~i~~nterface>> are visible across the itf-N.

End of Change in Clause 5.2.6

Change in Clause 5.2.7

5.2.7 Relationship realize and <<may realize>>

The relationship realize and <<may realize>> are unidirectional association. The target must be an <<I~~i~~nterface>>. The relationship "realize" shows that the source entity must realize the operations defined by the target <<I~~i~~nterface>>. Realization of operations by the source entity is mandatory. The <<may realize>> shows the source entity may realize the operations defined by the target <<I~~i~~nterface>>. Realization of the <<I~~i~~nterface>> by the source entity is optional.

The operations defined by <<I~~i~~nterface>> are visible across the itf-N.

Change in Clause 5.2.7

Change in Clause 5.3

5.3 Visibility

It specifies the accessibility of the operation and attribute. There are three types of visibility, i.e. private, public and IRPAgent Internal.

Private Visibility (notation "-")

Operation	NA
Attribute	It indicates that the attribute is not accessible by other entities, e.g. the IRPManager, other entities not holding the subject attribute

Public Visibility (notation "+")(default)

Operation	It indicates that the operation is visible across the itf-N, e.g. the IRPManager can invoke the operation across the itf-N interface.
Attribute	it indicates that the attribute is accessible across the itf-N, i.e. the IRPManager can invoke an operation to read the attribute and to write to this attribute if the attribute is so qualified. The read or write operation must be directly invoked against the entity holding the subject attribute or against the CM IRP Agent.

IRPAgent Internal Visibility (notation "%")

Operation	It indicates that the operation is not visible across the itf-N, i.e. the IRPManager cannot invoke the operation. However, other entities can invoke the operation (see note).
Attribute	It indicates that the attribute is not directly accessible across the itf-N, i.e. the IRPManager cannot read/write this attribute. However, other entities can read/write this attribute.
~~NOTE: No Release 5 operations are of this kind.~~	

End of Change in Clause 5.3
End of document

Annex A (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Jun 2004	S_24	SP-040242	001		UML Repertoire Updates (Associations)	6.0.0	6.1.0
Mar 2005	S_27	SP-050023	002	--	Add “abstract” adornment	6.1.0	6.2.0