Information Agreement Guidelines

Abstract

This document contains the unique design reference for the development of all Information Agreements (IA).

Table of Contents

1 Introduction

The purpose of this document is to be the unique design reference for the development of all Information Agreements (IA).

The Unified Modelling Language (UML) Version 2.1 [**Ref**] will be used as the notation for the IA. The IA will be created with the "Rational Software Modeler" (RSM) tool [**Ref**] provided by IBM. However, in the future, it is possible that other tools may be used.

These guidelines shall ensure that different people can create individual parts of the Information Agreement with the same "look and feel" which can then be combined into a Release.

1.1 Status

This document is a Guideline document.

1.2 Context

The definition of these mTOP IA guidelines is driven by the development of the MTOSI Release 2.x mTOP product. The writing of these IA guidelines has been made possible based on the experience the mTOP team gained from the previous MTOSI and MTNM releases.

1.3 Future

These IA guidelines are not a static document. The following factors will certainly influence the evolution of these guidelines sometime in the future:

- During the creation of the individual IA DDPs the editors become more and more familiar with the RSM tool which may lead to improvements.
- mTOP Methodology: The idea of generating a protocol specific UML which then allows one to automatically (tool supported) create the IIS (XML, IDL) is still the targeted goal. The guidelines will therefore have to be adapted to the process/tooling chosen.
- Harmonization/Convergence with the UML models from other standards (SID, OSS/J, 3GPP, etc) will also lead to an evolution of these guidelines.

2 General Guidelines

2.1 IA Architecture

2.1.1 Overview

The purpose of the mTOP IA is to provide a description of the **protocol neutral** mTOP Information Model based on UML.

The overall mTOP IA is structured along the mTOP DDP partitioning which is based on the eTOM level2/3 processes. All IA definitions must relate to the scope of the mTOP DDP that it is part of. Refer to the mTOP Guidelines DDP Definitions documents [**Ref**] for a rational description of the mTOP DDPs.

2.1.2 Logical View

The complete definition of the mTOP Information Model is based on the use of individual UML Models containing the mTOP DDPs.

The DDPs are grouped into two basic types:

- Data Model DDPs (DM-DDPs) containing the static data entities.
- Operations Model DDPs (OM-DDPs) containing the dynamic (service) interfaces where each service interface consists of several operations.

The following DDPs will be created for MTOSI Release 2.0:

- Framework (FMW)
- RM&O + RD&M
  - Data Model DDPs
    - Network Resource Basic (NRB)
    - Network Resource Fulfillment (NRF)
    - Network Resource Assurance (NRA)
  - Operations Model DDPs
    - Manage Resource Inventory (MRI)
    - Resource Provisioning (RP)
    - Resource Trouble Management (RTM)
    - Resource Performance Management (RPM)
- SM&O + SD&M
  - Data Model DDPs
    - Service Basic (SB)
  - Operations Model DDPs
    - Manage Service Inventory (MSI)
    - Service Activation (SA)

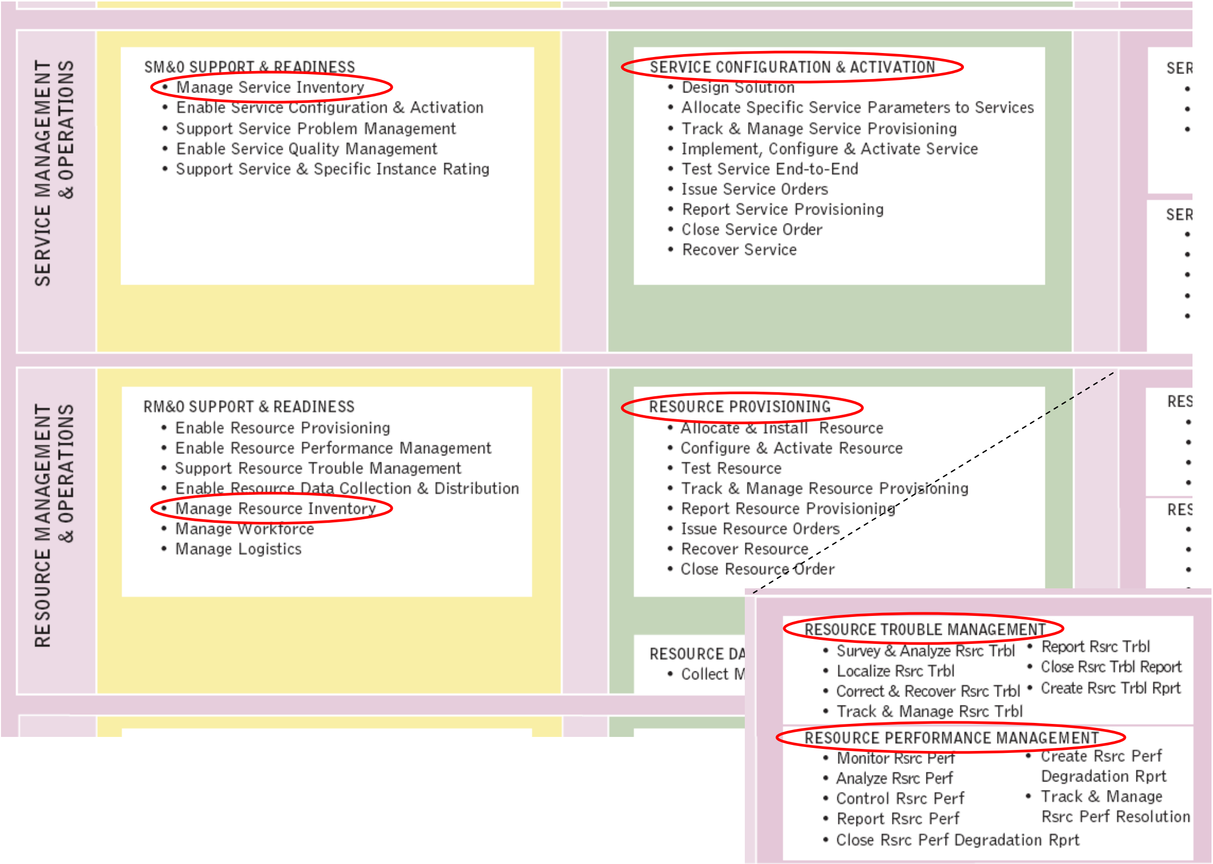

SERVICE MANAGEMENT & OPERATIONS
SM&O SUPPORT & READINESS
• Manage Service Inventory
• Enable Service Configuration & Activation
• Support Service Problem Management
• Enable Service Quality Management
• Support Service & Specific Instance Rating
SERVICE CONFIGURATION & ACTIVATION
• Design Solution
• Allocate Specific Service Parameters to Services
• Track & Manage Service Provisioning
• Implement, Configure & Activate Service
• Test Service End-to-End
• Issue Service Orders
• Report Service Provisioning
• Close Service Order
• Recover Service
SER
SER
RESOURCE MANAGEMENT & OPERATIONS
RM&O SUPPORT & READINESS
• Enable Resource Provisioning
• Enable Resource Performance Management
• Support Resource Trouble Management
• Enable Resource Data Collection & Distribution
• Manage Resource Inventory
• Manage Workforce
• Manage Logistics
RESOURCE PROVISIONING
• Allocate & Install Resource
• Configure & Activate Resource
• Test Resource
• Track & Manage Resource Provisioning
• Report Resource Provisioning
• Issue Resource Orders
• Recover Resource
• Close Resource Order
RES
RES
RESOURCE DA
• Collect M
RESOURCE TROUBLE MANAGEMENT
• Survey & Analyze Rsrc Trbl
• Localize Rsrc Trbl
• Correct & Recover Rsrc Trbl
• Track & Manage Rsrc Trbl
• Report Rsrc Trbl
• Close Rsrc Trbl Report
• Create Rsrc Trbl Rprt
RESOURCE PERFORMANCE MANAGEMENT
• Monitor Rsrc Perf
• Analyze Rsrc Perf
• Control Rsrc Perf
• Report Rsrc Perf
• Close Rsrc Perf Degradation Rprt
• Create Rsrc Perf Degradation Rprt
• Track & Manage Rsrc Perf Resolution

Figure : "Location" of the MTOSI Release 2.0 DDPs within eTOM

Note: The development of the Framework DDP will have the highest priority because this DDP is used in all other DDPs.

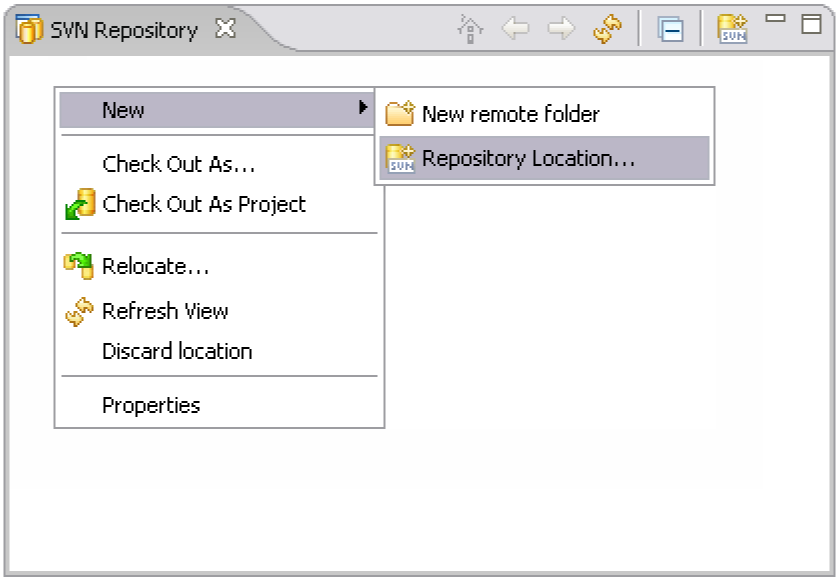
2.1.3 Development View

The development view of the mTOP IA is maintained and managed through the TMF SourceForge Subversion repository.

The following steps describe how to add the IA (in fact it adds the complete specification) from the GForge Subversion repository as a new project into RSM.

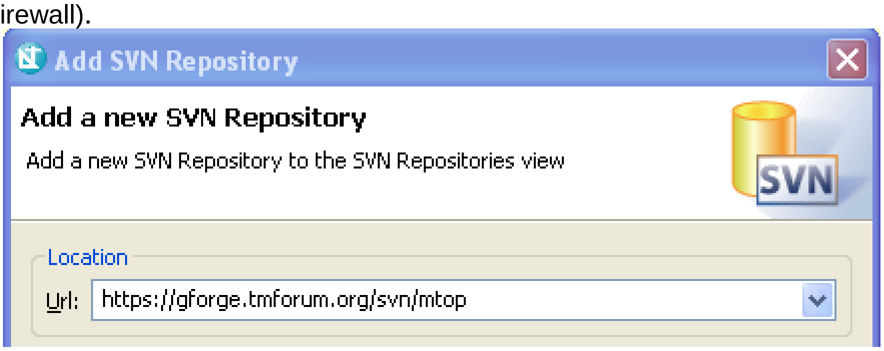
1. A prerequisite is that you have installed the "Subclipse" (an Eclipse Plug-in) Subversion tool into RSM.

2. Creating a new Repository Location (*customised based on the "Subclipse" help documentation*): You create and work with your repository locations from the ***SVN Repository View***, which is part of the ***SVN Repository Exploring Perspective***. To create a new repository location, click the **Add SVN Repository** button or right clicking inside the view and selecting **New > Repository Location...**



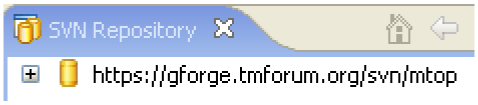
The ***Add SVN Repository*** dialog will prompt you for a repository location URL. Enter the URL of the mTOP repository

- svn+ssh://<username>@svn.gforge.tmforum.org/svn/mtop (<username> is your TMF GForge registered username) or
- https://gforge.tmforum.org/svn/mtop/ (**Note**: https should only be used if svn+ssh is blocked by the firewall).



Url: https://gforge.tmforum.org/svn/mtop

Once you click finish, the wizard will create the location:



https://gforge.tmforum.org/svn/mtop

Expand the location to see the contents for the given URL.

Note: If your PC is connected to the Internet via a firewall you have to provide the firewall as a proxy. This is done by updating the “servers” file located at ~/.subversion on Linux and OS X and at C:\Documents and Settings\userName\Application Data\Subversion on Windows:

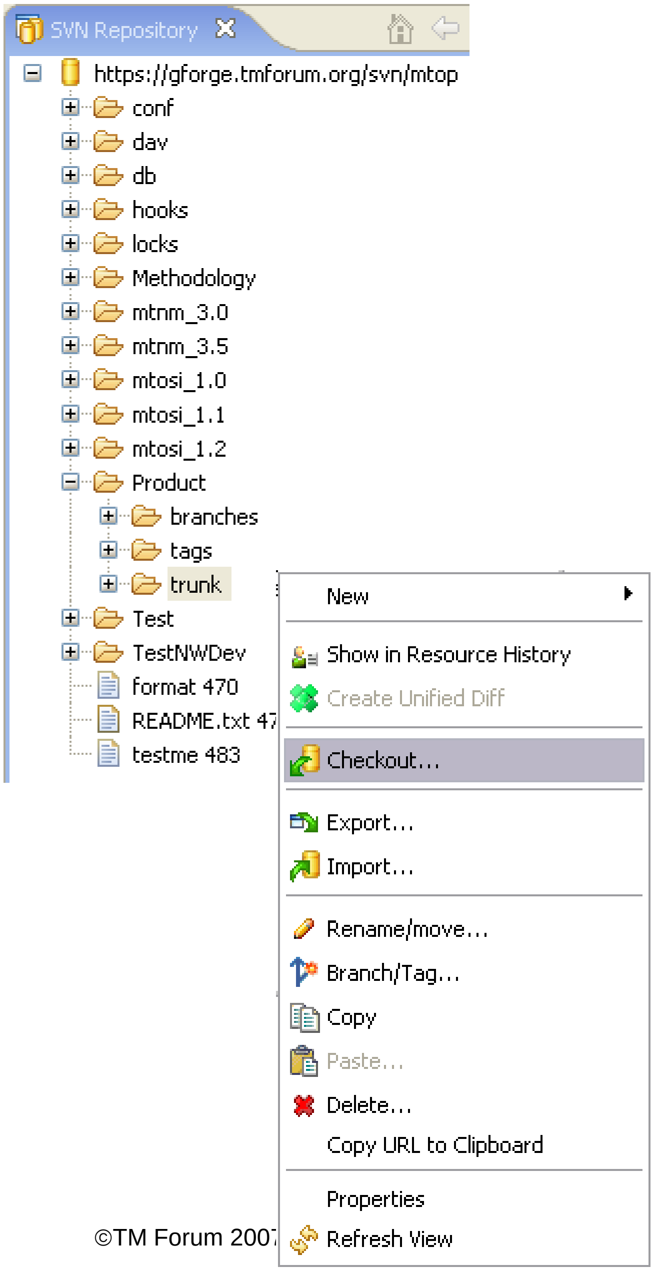
```
[groups]
group1 = gforge.tmforum.org

### Information for the first group:
[group1]
http-proxy-host = <yourProxy.some-domain-name.com>
```

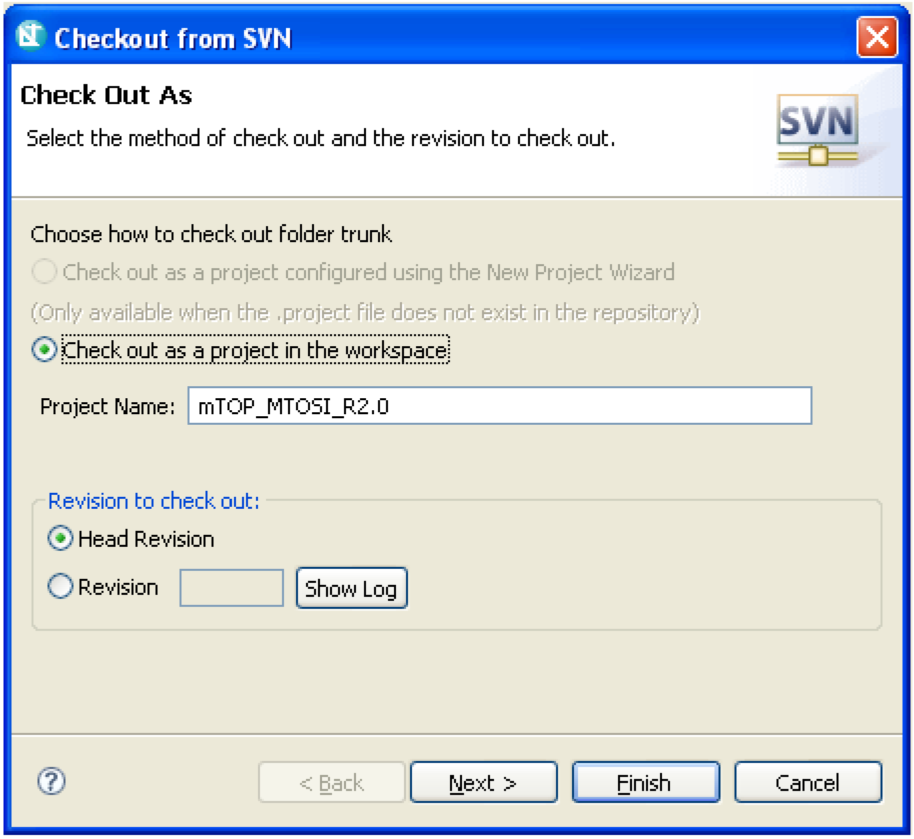
```
http-proxy-port = <yourProxyPort>
http-proxy-username = blah
http-proxy-password = doubleblah
http-timeout = 60
neon-debug-mask = 130
```

3. Checking out the draft MTOSI Release 2.0 specification from the GForge repository:
   Excerpt from "Subclipse" help documentation: *Check out is the term used to describe the process of making a copy of a project from a repository into your local workspace. In SVN, the check out process creates a working copy. A working copy is a specially formatted folder structure which contains additional .svn folders that store SVN information, as well as a pristine copy of each item that is checked out.*

   In the SVN Repository window, select the **Product / trunk** repository directory and choose **Checkout** from the context menu:



Choose **Check out as a project in the workspace** and click **Finish**:



Checkout from SVN

Check Out As

Select the method of check out and the revision to check out.

Choose how to check out folder trunk

○ Check out as a project configured using the New Project Wizard

(Only available when the .project file does not exist in the repository)

● Check out as a project in the workspace

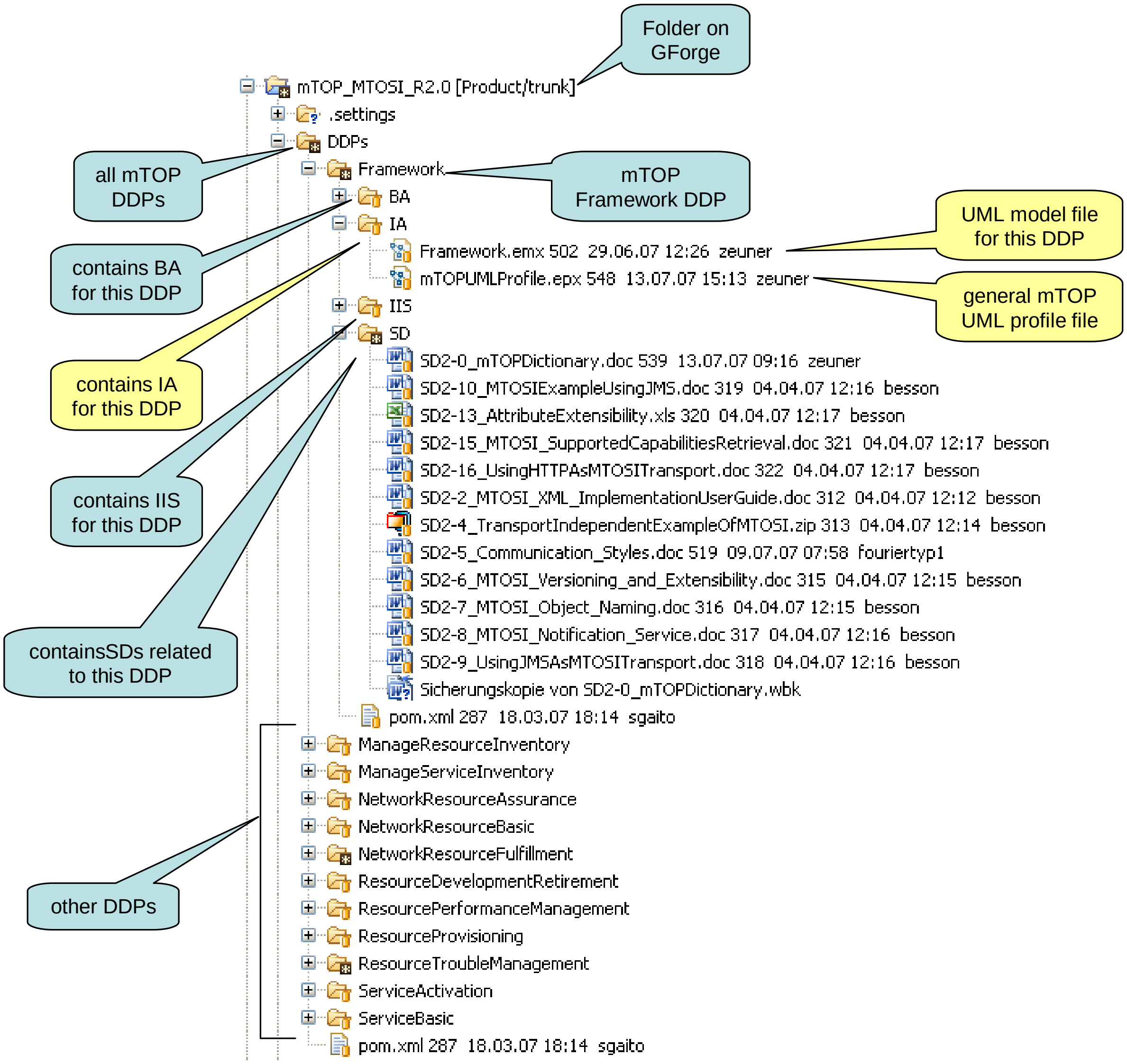
Project Name: mTOP_MTOSI_R2.0

Revision to check out:

● Head Revision

○ Revision

Finally, move to the ***Modeling view*** in RSM and you will find the new project **mTOP_MTOSI_R2.0** in the Project Explorer:



```
mTOP_MTOSI_R2.0 [Product/trunk]
  .settings
  DDPs
    Framework
      BA
      IA
        Framework.emx 502  29.06.07 12:26  zeuner
        mTOPUMLProfile.epx 548  13.07.07 15:13  zeuner
      IIS
      SD
        SD2-0_mTOPDictionary.doc 539  13.07.07 09:16  zeuner
        SD2-10_MTOSIExampleUsingJMS.doc 319  04.04.07 12:16  besson
        SD2-13_AttributeExtensibility.xls 320  04.04.07 12:17  besson
        SD2-15_MTOSI_SupportedCapabilitiesRetrieval.doc 321  04.04.07 12:17  besson
        SD2-16_UsingHTTPAsMTOSITransport.doc 322  04.04.07 12:17  besson
        SD2-2_MTOSI_XML_ImplementationUserGuide.doc 312  04.04.07 12:12  besson
        SD2-4_TransportIndependentExampleOfMTOSI.zip 313  04.04.07 12:14  besson
        SD2-5_Communication_Styles.doc 519  09.07.07 07:58  fouriertyp1
        SD2-6_MTOSI_Versioning_and_Extensibility.doc 315  04.04.07 12:15  besson
        SD2-7_MTOSI_Object_Naming.doc 316  04.04.07 12:15  besson
        SD2-8_MTOSI_Notification_Service.doc 317  04.04.07 12:16  besson
        SD2-9_UsingJMSAsMTOSITransport.doc 318  04.04.07 12:16  besson
        Sicherungskopie von SD2-0_mTOPDictionary.wbk
      pom.xml 287  18.03.07 18:14  sgaito
    ManageResourceInventory
    ManageServiceInventory
    NetworkResourceAssurance
    NetworkResourceBasic
    NetworkResourceFulfillment
    ResourceDevelopmentRetirement
    ResourcePerformanceManagement
    ResourceProvisioning
    ResourceTroubleManagement
    ServiceActivation
    ServiceBasic
    pom.xml 287  18.03.07 18:14  sgaito
```

Folder on GForge
all mTOP DDPs
mTOP Framework DDP
UML model file for this DDP
contains BA for this DDP
general mTOP UML profile file
contains IA for this DDP
contains IIS for this DDP
containsSDs related to this DDP
other DDPs

Figure : Structure of the IA Model files on GForge during development

2.2 Naming Conventions

Refer to the mTOP Naming Convention Guidelines in SD0-1_Dictionary [Ref **].**

mTOP artifacts are the Business Agreement (BA), the Information Agreement (IA) and the Interface Implementation Specification (IIS).

UML artifacts are classes, attributes, operations, notifications, parameters and relationships.

In this document the following tags are used as variables:

- <DDP_Full> for the full DDP name
- <DDP> for the short DDP name (acronym)

2.3 Versioning

The IA is a protocol neutral specification. Backward compatibility with previous IA versions is therefore not required.

- It is allowed to add new attributes to classes and new operations to interfaces.
- IA artefacts (e.g., classes, interfaces, attributes, operations) which become obsolete in a new version must not be deleted. They shall be identified as "obsolete" using the <<stereotype>> field; i.e., <<obsolete>>.
- IA artefacts which have been enhanced in a new version shall be identified as "enhanced" using the <<stereotype>> field; i.e., <<enhanced>>.

2.4 Documentation

The documentation of the mTOP UML IA will be done with the tool "Rational Software Modeler" from IBM. HTML-based documentation of the UML models will also be delivered.

3 IA Modelling Guidelines

3.1 Introduction

The information modeling comprises the "analysis" part of the specification. The requirements and use cases defined in the BA are the basis for the analysis.

The purpose of this section is to provide a set of UML design recommendations to promote all mTOP IA editors to specify the IA in a standardized mTOP UML style. The benefits are:

- Readability of the UML specifications as they all have the same style.
- Providing the necessary information to distribute the workload between many editors.

3.2 General IA Requirements

- The UML 2.1 (Unified Modeling Language) is used as a notation for the model.
- The IA shall be protocol-neutral, i.e., not reflect any middleware protocol-specific characteristics (like CORBA, HTTP, JMS).
- The IA shall be map-able to various protocol-specific interfaces.
  Until this mapping is automated, it will be necessary to (manually) create an IA to IIS mapping document.
- The requirements and use cases from the BA are converted into the model as object classes (containing attributes or operations), relationships between the objects and notifications.
- For attributes with defined value transitions, state transition diagrams shall be created.
- For operations with high complexity, activity diagrams shall be created.
- Traceability from each modeling construct back to the requirements and use cases has to be provided.

- The information agreement has to provide guidelines on how the UML notation (e.g., relationship stereotypes) is used to model specific requirements.

3.3 New General IA Requirements

This section provides the requirements which are new wrt. the last mTOP IA version (i.e., TMF608 v3.4).

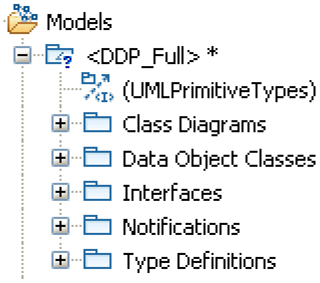
- The IA shall be documented in UML 2.1 using the "Eclipse dialect".
- The IA shall be divided into:
  - Data entities – represent managed entities specified in the DM-DDPs.
    These entities only have attributes and associations.
    Notifications are also treated as data entities.
  - Service interfaces[1] – specified in the OM-DDPs.
    These interfaces only have operations.
- The IA shall be sectionalized based on the DDP separations.
- This will allow for:
  - Multiple editors of the IA
  - Release of portions of the IA.
- Overlap of information between the BA and IA is allowed, but is strongly recommended that the BA be considered the source and the other document be populated automatically with the redundant information.
- Eventually, the intent is to have UML models for the various protocol-specific interfaces (IDL, XML, WSDL). These UML models will be considered part of the IIS and not the IA.

3.4 General UML Guidelines

This section provides the general guidelines (i.e., which are not specific to the Data Model or the Service Interfaces Model).

Each DDP shall be stored in a separate Model file (xxx.emx file). Each Model shall be structured in the following way:

[1] The term Service Interface comes from ETSI TS 188 001. "NGN OSS Service Interface (NGN OSS SI): A well defined grouping of related NGN OSS Operations and constant data which are necessary to deliver coherent business or system functionality."

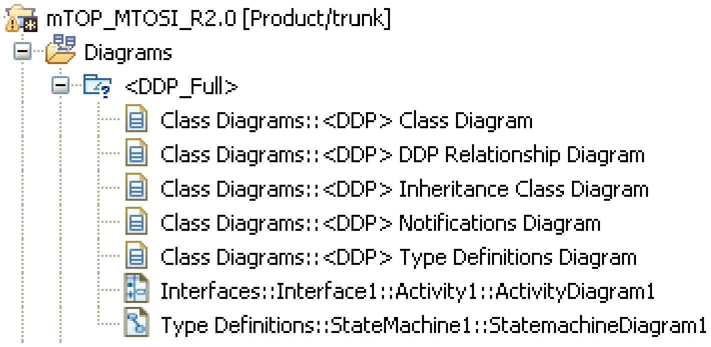


```
Models
  <DDP_Full> *
    (UMLPrimitiveTypes)
    Class Diagrams
    Data Object Classes
    Interfaces
    Notifications
    Type Definitions
```

Figure : UML Model structure

The Data Model DDPs only have "Data Object Classes" and the Service Interface Model DDPs only have "Interfaces". The only exception which has both types of classes is the Framework DDP.

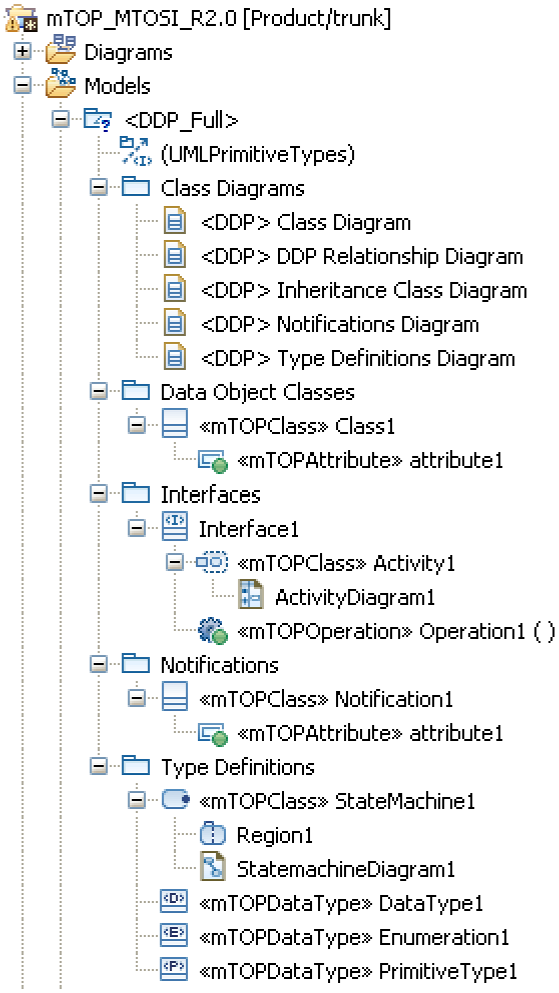
Note: RSM creates automatically an additional "Diagrams" folder that contains all the diagrams which are created in the specific Model.



```
mTOP_MTOSI_R2.0 [Product/trunk]
  Diagrams
    <DDP_Full>
      Class Diagrams::<DDP> Class Diagram
      Class Diagrams::<DDP> DDP Relationship Diagram
      Class Diagrams::<DDP> Inheritance Class Diagram
      Class Diagrams::<DDP> Notifications Diagram
      Class Diagrams::<DDP> Type Definitions Diagram
      Interfaces::Interface1::Activity1::ActivityDiagram1
      Type Definitions::StateMachine1::StatemachineDiagram1
```

Figure : UML Model diagrams structure

The following figure shows the location of the UML artifacts:



```
mTOP_MTOSI_R2.0 [Product/trunk]
  Diagrams
  Models
    <DDP_Full>
      (UMLPrimitiveTypes)
      Class Diagrams
        <DDP> Class Diagram
        <DDP> DDP Relationship Diagram
        <DDP> Inheritance Class Diagram
        <DDP> Notifications Diagram
        <DDP> Type Definitions Diagram
      Data Object Classes
        «mTOPClass» Class1
          «mTOPAttribute» attribute1
      Interfaces
        Interface1
          «mTOPClass» Activity1
            ActivityDiagram1
          «mTOPOperation» Operation1 ( )
      Notifications
        «mTOPClass» Notification1
          «mTOPAttribute» attribute1
      Type Definitions
        «mTOPClass» StateMachine1
          Region1
          StatemachineDiagram1
        «mTOPDataType» DataType1
        «mTOPDataType» Enumeration1
        «mTOPDataType» PrimitiveType1
```

Figure : UML artifact structure

3.4.1 mTOP Profile

The RSM tool allows extending the UML core metamodel (see Figure) by using profiles. An mTOP UML Profile has been defined to extend the "class", "structuralFeature" and "operation" metaclasses of the core metamodel.

The "**class**" metaclass has been extended with the capability to specify:

- the creation notifications of the resource/service object class (NO | YES | NOT_APPLICABLE), default value = NOT_APPLICABLE
- the deletion notifications of the resource/service object class (NO | YES | NOT_APPLICABLE), default value = NOT_APPLICABLE
- the discovery notifications of the resource/service object class (NO | YES | NOT_APPLICABLE), default value = NOT_APPLICABLE.

The "**structuralFeature**" metaclass (i.e., the superclass of the attributes) has been extended with the capability to specify:

- the kind of notification that has to be send when an attribute changes its value (ATTRIBUTE_VALUE_CHANGE | STATE_CHANGE | NO_NOTIFICATION | NOT_APPLICABLE), default value = NOT_APPLICABLE
- if the attribute is invariant after it is created (Boolean), default value = false
- if the attribute can be set/changed by an OS (NONE_OS | TARGET_OS_ONLY | REQUESTING_OS_ONLY | TARGET_AND_REQUESTING_OS), default value = REQUESTING_OS_ONLY

is invariant (after attribute creation)	**send notification**	**writeable by target OS**	**writeable by requesting OS**	**read only (is dependent on writeable by requesting OS)**	**meaning**
X	X	-	-	-	not allowed; because if invariant it will never send a notification
X		X	X		can be set by target OS only during creation when requesting OS does not provide a value; can be set by requesting OS only during creation
X		X		X	can be set by target OS only during creation; can **not** be set by requesting OS
X			X		can **not** be set by target OS; can be set by requesting OS only during creation
X				X	can **not** be set by target OS; can **not** be set by requesting OS; value is set by the network only during creation
	-			X	can **not** be set by target OS; can **not** be set by requesting OS; value can be set by the network at anytime
	-	X	X		can be set by target OS at anytime; can be set by requesting OS at anytime
	-	X		X	can be set by target OS at anytime; can **not** be set by requesting OS
	-		X		can **not** be set by target OS; can be set by requesting OS at anytime

- → don't care

The "**operation**" metaclass has been extended with the capability to specify:

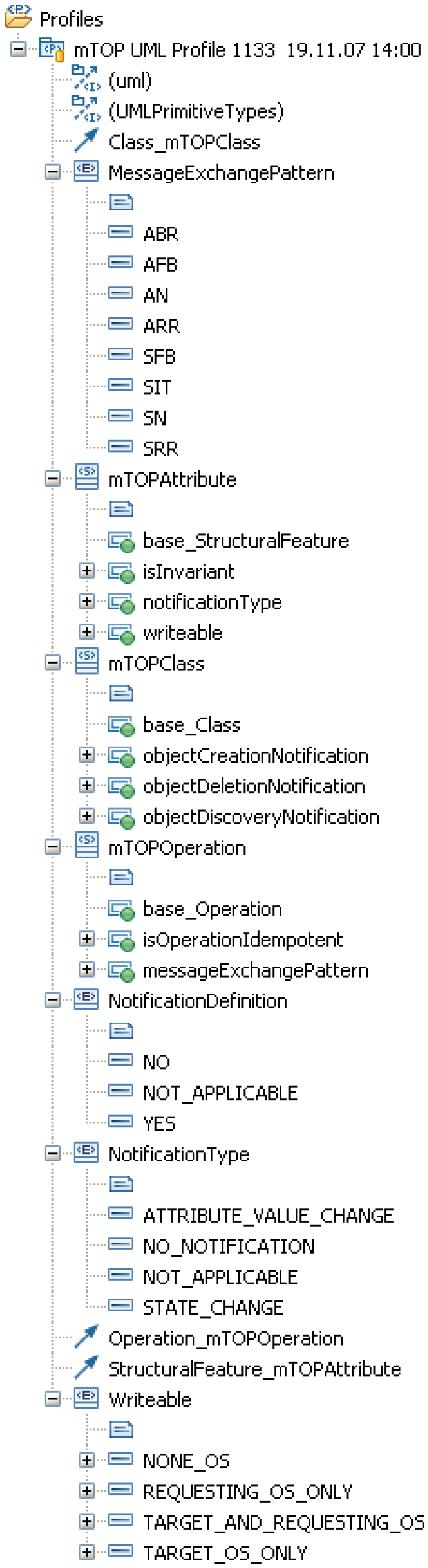
- Message Exchange Patterns (MEP) for the operations (SRR | ARR | SIT | ABR | SFB | AFB | SN | AN), default = empty list, see SD2-5 Communication Styles [Ref].
- if the operation is idempotent (Boolean), default value = false

All extensions have been defined as **mandatory**. For the cases where an additional property makes no sense, the value "NOT_APPLICABLE" has been defined.

A description of the defined stereotypes is contained in Appendix C: Requirements for additional properties in the mTOP IA UML model.

Figure explains how the Ecore metamodel is enhanced with the Profile by showing the relationships between the Ecore metamodel, the mTOP UML Profile and the enhanced mTOP model.

Figure shows how the mTOP Profile is defined in the RSM tool.

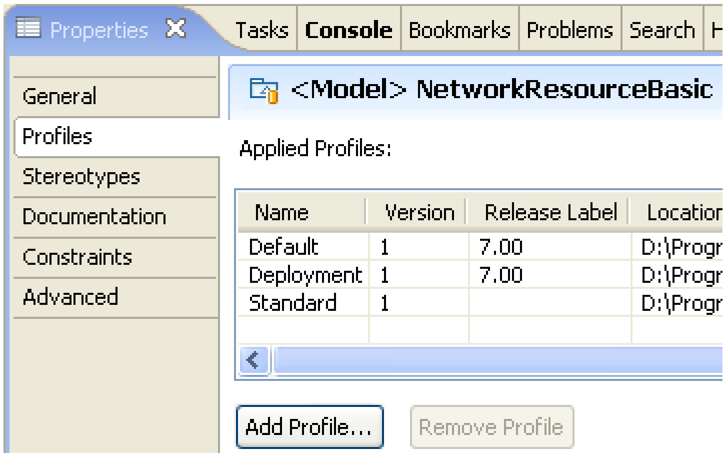


```
Profiles
  mTOP UML Profile 1133  19.11.07 14:00
    (uml)
    (UMLPrimitiveTypes)
    Class_mTOPClass
    MessageExchangePattern
      ABR
      AFB
      AN
      ARR
      SFB
      SIT
      SN
      SRR
    mTOPAttribute
      base_StructuralFeature
      isInvariant
      notificationType
      writeable
    mTOPClass
      base_Class
      objectCreationNotification
      objectDeletionNotification
      objectDiscoveryNotification
    mTOPOperation
      base_Operation
      isOperationIdempotent
      messageExchangePattern
    NotificationDefinition
      NO
      NOT_APPLICABLE
      YES
    NotificationType
      ATTRIBUTE_VALUE_CHANGE
      NO_NOTIFICATION
      NOT_APPLICABLE
      STATE_CHANGE
    Operation_mTOPOperation
    StructuralFeature_mTOPAttribute
    Writeable
      NONE_OS
      REQUESTING_OS_ONLY
      TARGET_AND_REQUESTING_OS
      TARGET_OS_ONLY
```

Figure : mTOP Profile structure

The mTOP UML Profile has to be added to all UML model files where these additional properties are necessary.

Left click the UML model file that you would like to add the profile to in the **Project Explorer** (in our example the NetworkResourceBasic model. Go to the **Profiles** tab within the **Properties** and click on **Add Profile …**

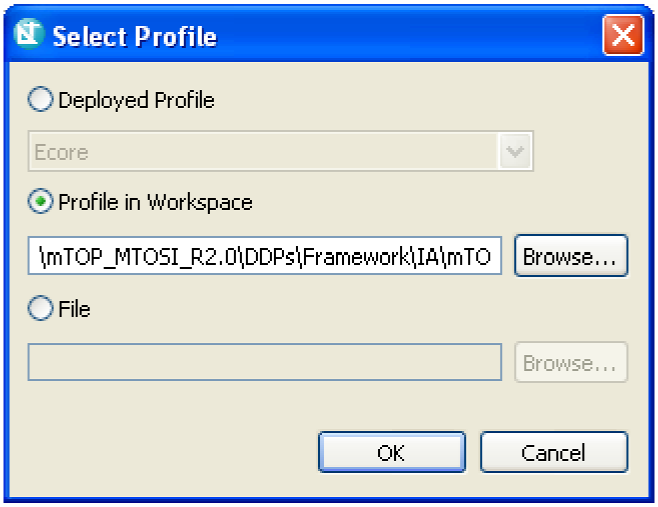


<Model> NetworkResourceBasic

Applied Profiles:

Name	Version	Release Label	Locatior
Default	1	7.00	D:\Progr
Deployment	1	7.00	D:\Progr
Standard	1		D:\Progr

Choose the **mTOPUMLProfile.epx** file from your RSM workspace in folder: \mTOP_MTOSI_R2.0\DDPs\Framework\IA\.



○ Deployed Profile

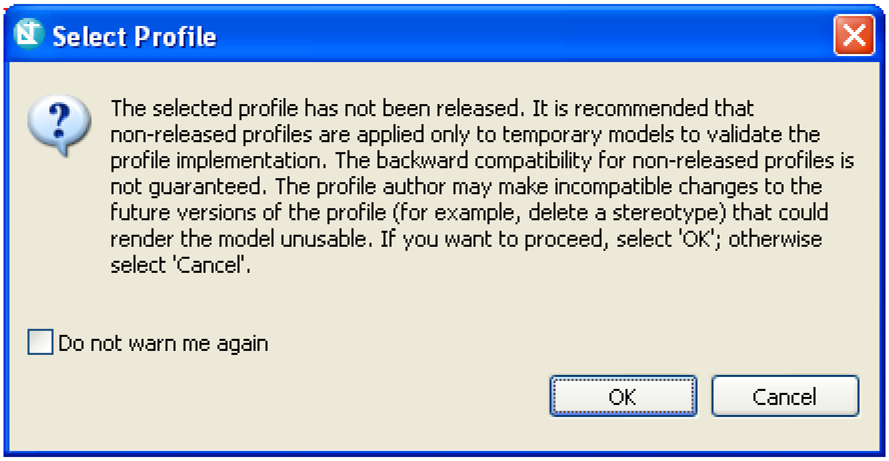
Ecore

● Profile in Workspace

\mTOP_MTOSI_R2.0\DDPs\Framework\IA\mTO

○ File

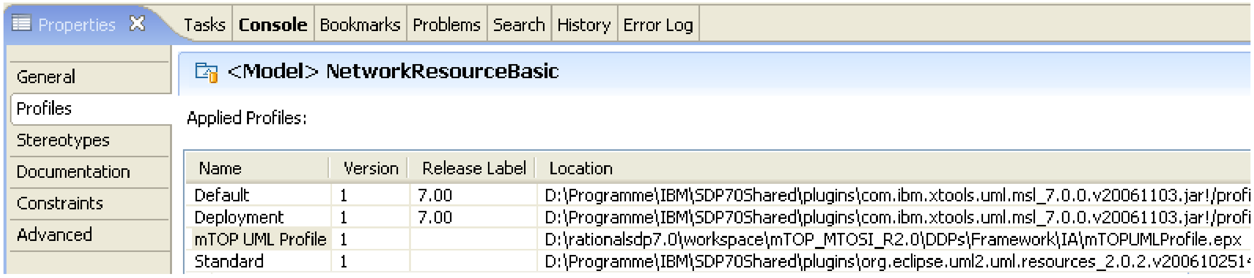
The current Profile has not been released so far because it is not mature enough. Therefore click **OK** on the next window:



The selected profile has not been released. It is recommended that non-released profiles are applied only to temporary models to validate the profile implementation. The backward compatibility for non-released profiles is not guaranteed. The profile author may make incompatible changes to the future versions of the profile (for example, delete a stereotype) that could render the model unusable. If you want to proceed, select 'OK'; otherwise select 'Cancel'.

□ Do not warn me again

The mTOP UML Profile is now added to the list of profiles:

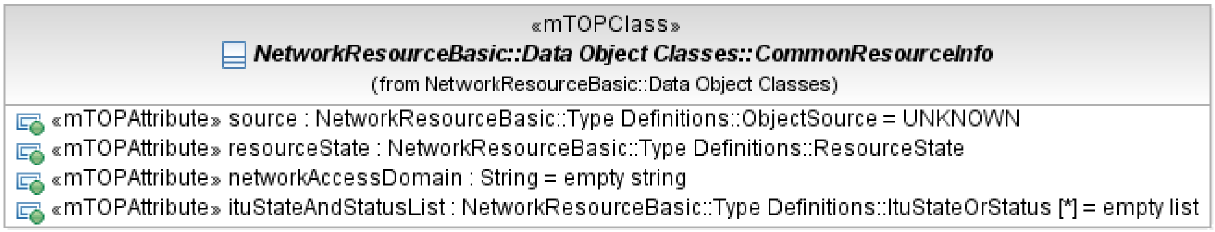


<Model> NetworkResourceBasic

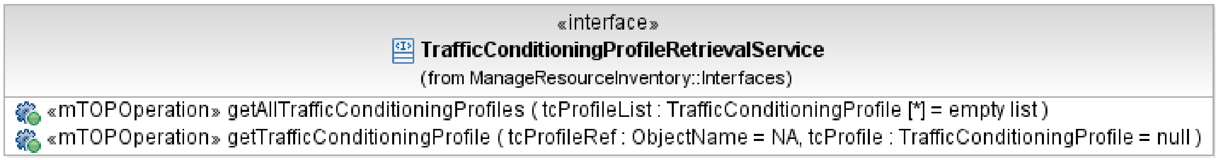
Applied Profiles:

Name	Version	Release Label	Location
Default	1	7.00	D:\Programme\IBM\SDP70Shared\plugins\com.ibm.xtools.uml.msl_7.0.0.v20061103.jar!/profi
Deployment	1	7.00	D:\Programme\IBM\SDP70Shared\plugins\com.ibm.xtools.uml.msl_7.0.0.v20061103.jar!/profi
mTOP UML Profile	1		D:\rationalsdp7.0\workspace\mTOP_MTOSI_R2.0\DDPs\Framework\IA\mTOPUMLProfile.epx
Standard	1		D:\Programme\IBM\SDP70Shared\plugins\org.eclipse.uml2.uml.resources_2.0.2.v20061025

All object classes, attributes, operations and associations contain now the additional properties. This is identified by the stereotypes <<mTOPClass>> for the object classes, <<mTOPAttribute>> for the attributes and <<mTOPOperation>> for the operations:

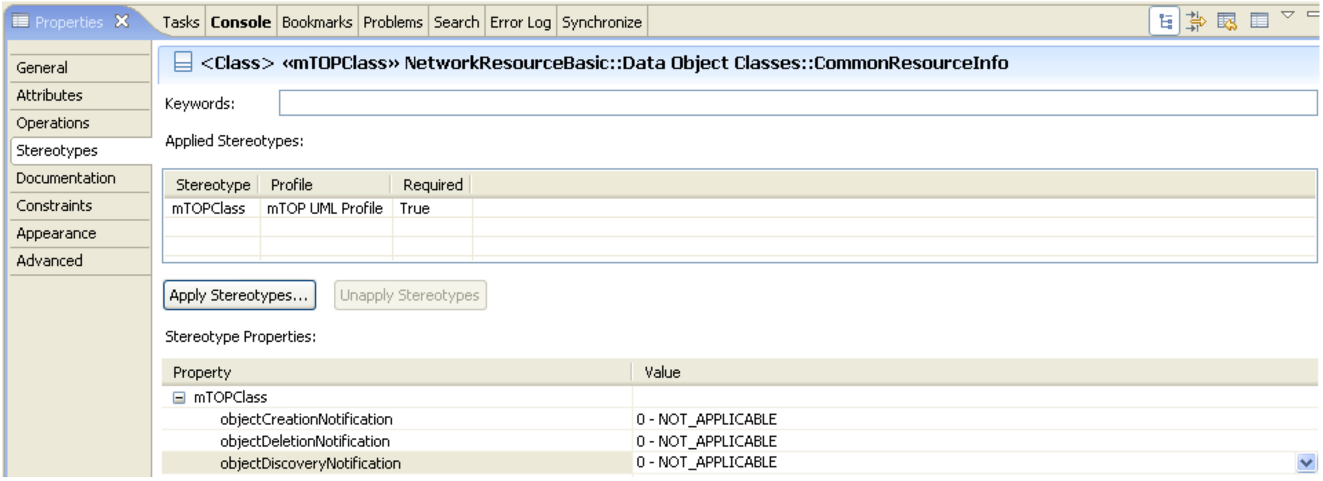

«mTOPClass»
NetworkResourceBasic::Data Object Classes::CommonResourceInfo
(from NetworkResourceBasic::Data Object Classes)

«mTOPAttribute» source : NetworkResourceBasic::Type Definitions::ObjectSource = UNKNOWN
«mTOPAttribute» resourceState : NetworkResourceBasic::Type Definitions::ResourceState
«mTOPAttribute» networkAccessDomain : String = empty string
«mTOPAttribute» ituStateAndStatusList : NetworkResourceBasic::Type Definitions::ItuStateOrStatus [*] = empty list


«interface»
TrafficConditioningProfileRetrievalService
(from ManageResourceInventory::Interfaces)

«mTOPOperation» getAllTrafficConditioningProfiles (tcProfileList : TrafficConditioningProfile [*] = empty list)
«mTOPOperation» getTrafficConditioningProfile (tcProfileRef : ObjectName = NA, tcProfile : TrafficConditioningProfile = null)

The additional properties are defined as stereotypes as depicted in the following figures:



<Class> «mTOPClass» NetworkResourceBasic::Data Object Classes::CommonResourceInfo

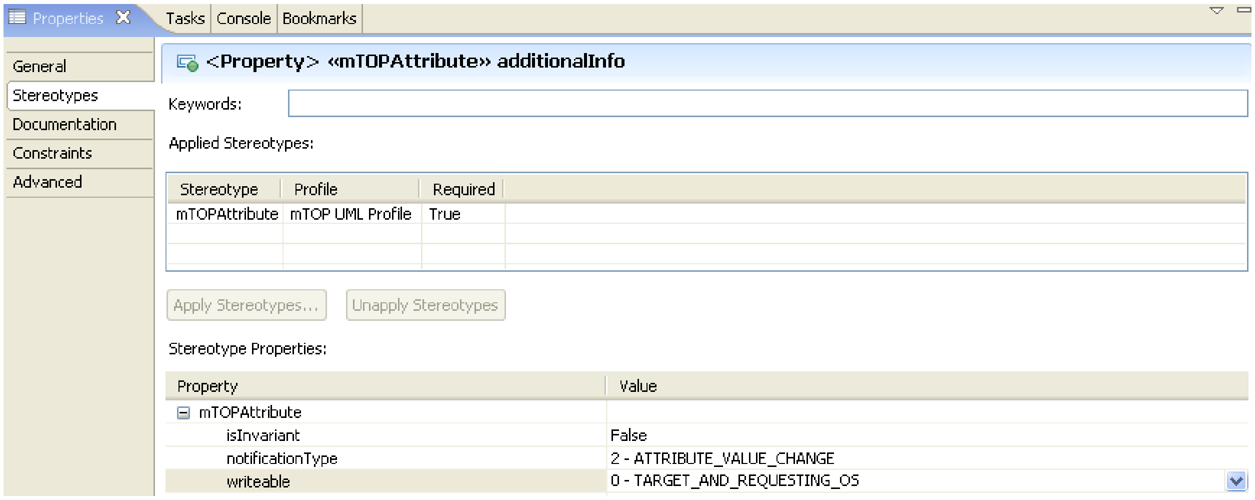
Keywords:

Applied Stereotypes:

Stereotype	Profile	Required
mTOPClass	mTOP UML Profile	True

Stereotype Properties:

Property	Value
⊟ mTOPClass	
objectCreationNotification	0 - NOT_APPLICABLE
objectDeletionNotification	0 - NOT_APPLICABLE
objectDiscoveryNotification	0 - NOT_APPLICABLE



<Property> «mTOPAttribute» additionalInfo

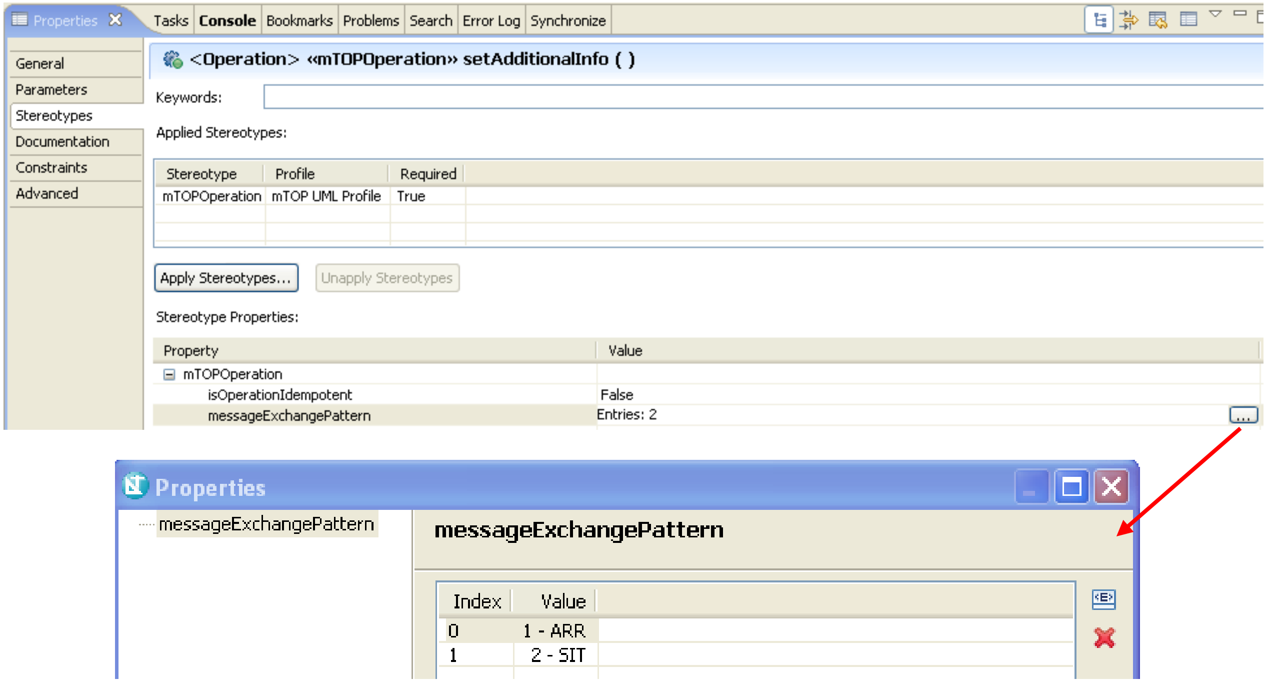
Keywords:

Applied Stereotypes:

Stereotype	Profile	Required
mTOPAttribute	mTOP UML Profile	True

Stereotype Properties:

Property	Value
⊟ mTOPAttribute	
isInvariant	False
notificationType	2 - ATTRIBUTE_VALUE_CHANGE
writeable	0 - TARGET_AND_REQUESTING_OS



<Operation> «mTOPOperation» setAdditionalInfo ( )

Keywords:

Applied Stereotypes:

Stereotype	Profile	Required
mTOPOperation	mTOP UML Profile	True

Stereotype Properties:

Property	Value
⊟ mTOPOperation	
isOperationIdempotent	False
messageExchangePattern	Entries: 2

messageExchangePattern

messageExchangePattern

Index	Value
0	1 - ARR
1	2 - SIT

3.4.2 Data Type Guidelines

We distinguish between "basic" and "complex" data types.

The **basic data types** do not need additional specification. They are provided by the RSM tool as (UMLPrimitiveTypes).

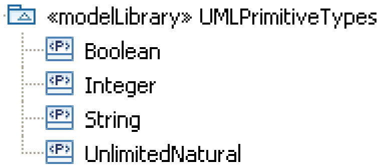

«modelLibrary» UMLPrimitiveTypes
- Boolean
- Integer
- String
- UnlimitedNatural

Figure : Basic data types

In addition four more basic data types are defined:

- ItuTTime
- ObjectName
- Real
- UnsignedInteger.

Basic data types are Boolean, Integer, ItuTTime, ObjectName; Real, String, UnsignedInteger.

The ItuTTime is defined as:
"yyyyMMddhhmmss.s[Z|{+|-}HHMm]" where:

yyyy	"0000".."9999"	year
MM	"01".."12"	month
dd	"01".."31"	day
hh	"00".."23"	hour

mm	"00".."59"	minute
ss	"00".."59"	second
s	".0"..".9"	tenth of second (set to ".0" if EMS or ME cannot support this granularity)
Z	"Z"	indicates UTC (rather than local time)
{+|-}	"+" or "-"	delta from UTC
HH	"00".."23"	time zone difference in hours
Mm	"00".."59"	time zone difference in minutes.

The ObjectName has to be used for the unique, read-only name of an object. The exact type is defined in the protocol specific IIS.

Complex data types are specified for attribute and parameter types when basic data types cannot be used. They are defined as object classes with attributes and are contained in the Type Definitions folder within each UML Model. Complex data types use basic and/or other complex data types for their Type definitions.

General data type Guidelines:

- The category of the data type is identified by a little icon in front of the type name. The RSM meta-model already provides the dataType = <D>, enumeration = <E> and primitive <P> data types.

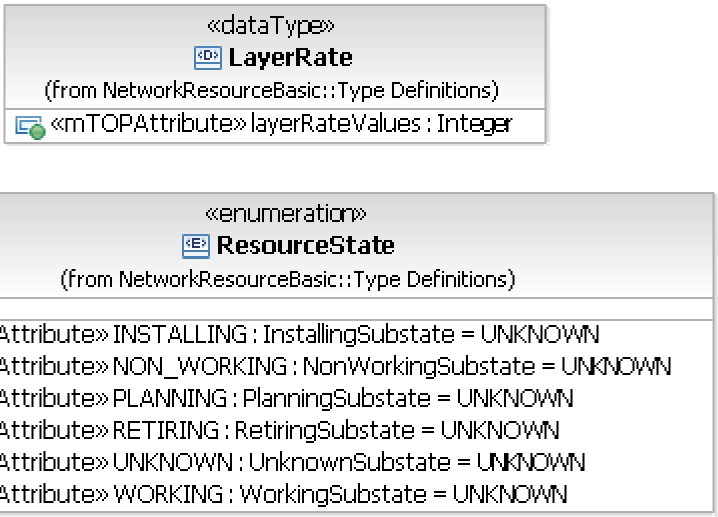

«dataType»
LayerRate
(from NetworkResourceBasic::Type Definitions)
«mTOPAttribute» layerRateValues : Integer
«enumeration»
ResourceState
(from NetworkResourceBasic::Type Definitions)
«mTOPAttribute» INSTALLING : InstallingSubstate = UNKNOWN
«mTOPAttribute» NON_WORKING : NonWorkingSubstate = UNKNOWN
«mTOPAttribute» PLANNING : PlanningSubstate = UNKNOWN
«mTOPAttribute» RETIRING : RetiringSubstate = UNKNOWN
«mTOPAttribute» UNKNOWN : UnknownSubstate = UNKNOWN
«mTOPAttribute» WORKING : WorkingSubstate = UNKNOWN

Figure : Complex data types (example)

- Parameter lists for creation and modification of resources shall be specified as <D> (dataTypes) in the Type Definitions folder.

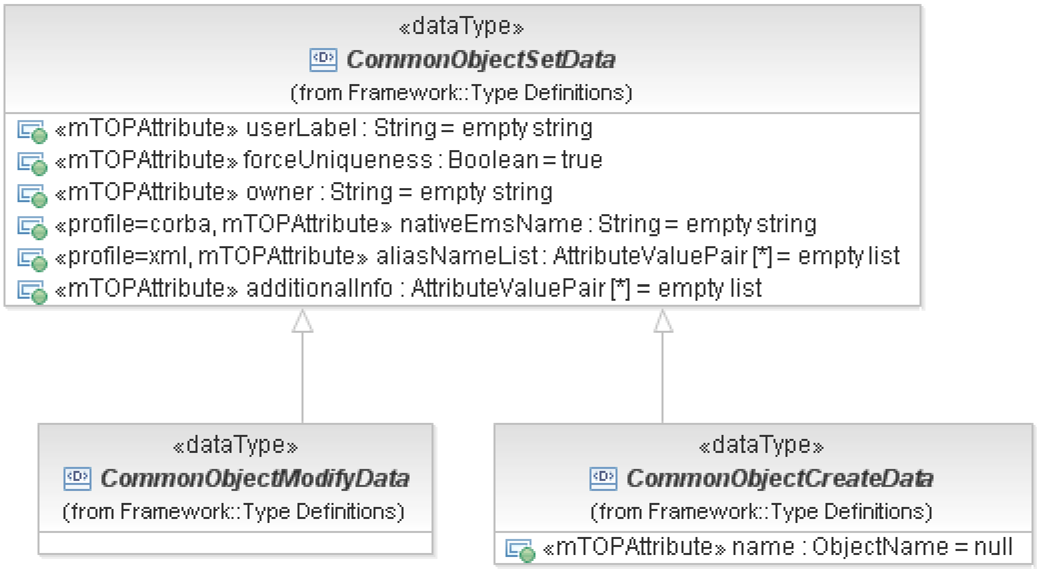
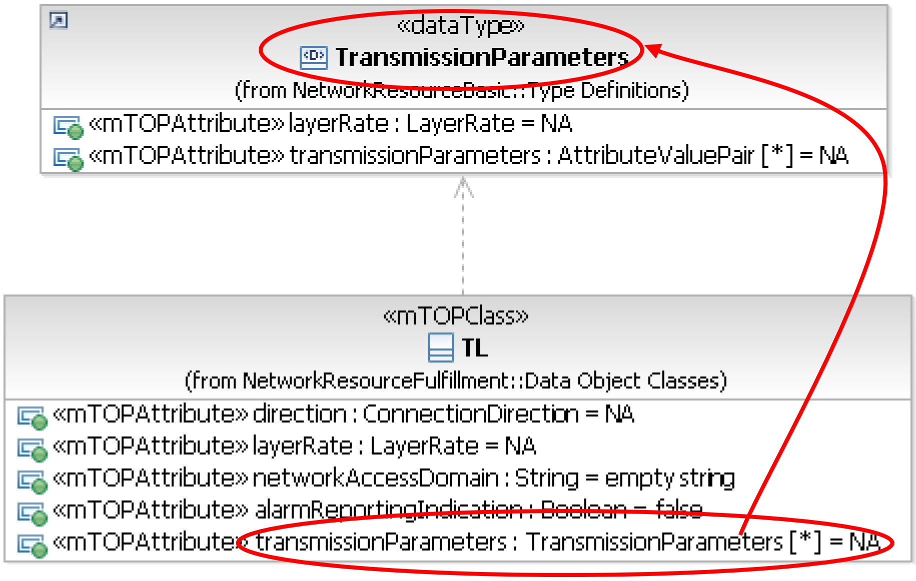
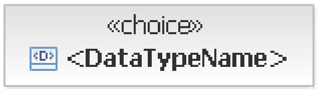
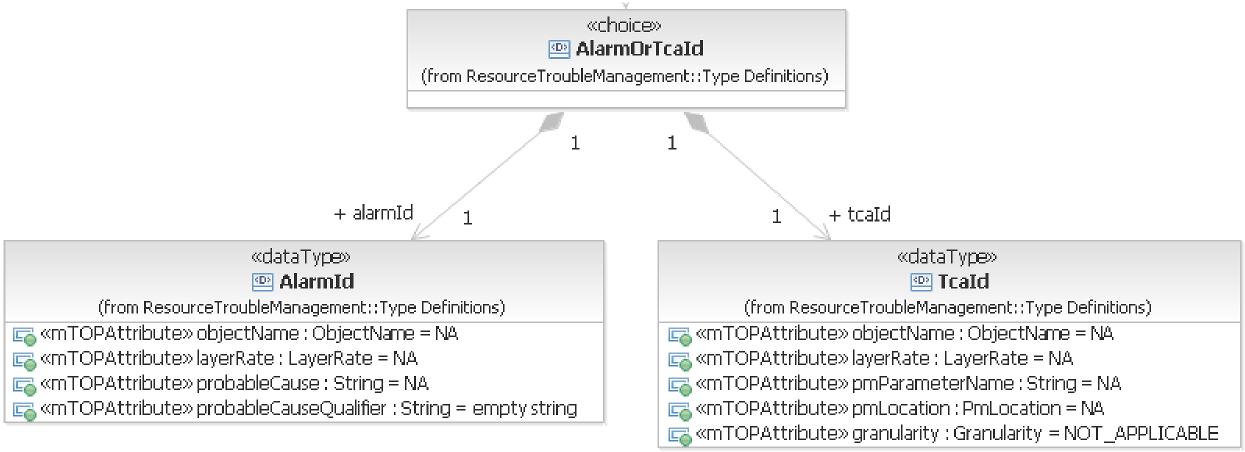

«dataType»
CommonObjectSetData
(from Framework::Type Definitions)
«mTOPAttribute» userLabel : String = empty string
«mTOPAttribute» forceUniqueness : Boolean = true
«mTOPAttribute» owner : String = empty string
«profile=corba, mTOPAttribute» nativeEmsName : String = empty string
«profile=xml, mTOPAttribute» aliasNameList : AttributeValuePair [*] = empty list
«mTOPAttribute» additionalInfo : AttributeValuePair [*] = empty list
«dataType»
CommonObjectModifyData
(from Framework::Type Definitions)
«dataType»
CommonObjectCreateData
(from Framework::Type Definitions)
«mTOPAttribute» name : ObjectName = null

Figure : Common Object Creation/Modification parameter data types

- Groups of attributes which can be used in object classes shall be specified as (dataTypes). Example:


«dataType»
TransmissionParameters
(from NetworkResourceBasic::Type Definitions)
«mTOPAttribute» layerRate : LayerRate = NA
«mTOPAttribute» transmissionParameters : AttributeValuePair [*] = NA
«mTOPClass»
TL
(from NetworkResourceFulfillment::Data Object Classes)
«mTOPAttribute» direction : ConnectionDirection = NA
«mTOPAttribute» layerRate : LayerRate = NA
«mTOPAttribute» networkAccessDomain : String = empty string
«mTOPAttribute» alarmReportingIndication : Boolean = false
«mTOPAttribute» transmissionParameters : TransmissionParameters [*] = NA

- Data types containing a group of attributes where only one of which can be active at a time shall be specified as  (dataTypes with choice stereotype). Example:


«choice»
AlarmOrTcaId
(from ResourceTroubleManagement::Type Definitions)
1
1
+ alarmId
1
1
+ tcaId
«dataType»
AlarmId
(from ResourceTroubleManagement::Type Definitions)
«mTOPAttribute» objectName : ObjectName = NA
«mTOPAttribute» layerRate : LayerRate = NA
«mTOPAttribute» probableCause : String = NA
«mTOPAttribute» probableCauseQualifier : String = empty string
«dataType»
TcaId
(from ResourceTroubleManagement::Type Definitions)
«mTOPAttribute» objectName : ObjectName = NA
«mTOPAttribute» layerRate : LayerRate = NA
«mTOPAttribute» pmParameterName : String = NA
«mTOPAttribute» pmLocation : PmLocation = NA
«mTOPAttribute» granularity : Granularity = NOT_APPLICABLE

Note: These data types can be implemented in IDL via a "union" construct or in XML via an "<xsd:choice>".

- Data type names
  - follow Upper CamelCase (UCC)
  - must be unique across all data type names defined in the whole mTOP IA
  - special definitions are in the mTOP Naming Convention Guidelines in SD0-1_mTOPDictionary [Ref].
- Attributes within data types
  - Name
    - follow Lower CamelCase (LCC)
    - must be unique across all attribute names defined in the whole mTOP IA
    - special definitions are in the mTOP Naming Convention Guidelines in SD0-1_mTOPDictionary [Ref].
  - Visibility
    - this property can be ignored
    - fixed to "Public".
  - Qualifiers
    - this property can be ignored
    - fixed to "Unique".
  - Type
    - refers to a basic or complex data type.
  - Default Value
    - provides the value that the attribute has to start with in case the value is not provided.
      In the case of a single-value property, the default value shall be specified as an <Opaque Expression>, except for the case in which the type of the property is an Enumeration type, in which case it shall be an Enumeration Literal selected from the Enumeration. (Note that an Enumeration Literal must be selected from the appropriate browser in RSM and must not simply be typed as a string.)

In the case of a multiple-valued property, the default shall be specified as a comma-separated list of <Opaque Expression> enclosed in braces. (Note that it is not possible to have a multiple-value property of an Enumeration type.)
If a default value makes no sense (e.g., if representing state of the system) = "<Opaque Expression> NA".
No default value for object name = "<Literal Null>".
No default value for String = "<Opaque Expression> empty string".
No default value for Sequence = "<Opaque Expression> empty list".this property can be ignored

  - Multiplicity
    - ○ defines the number of values the attribute can simultaneously have
  - Aggregation
    - ○ this property can be ignored
    - ○ fixed to "None".
  - Invariant (contained in mTOPAttribute stereotype)
    - ○ this property can be ignored
    - ○ fixed to "True"; see also section 3.4.1.
  - Notification Type (contained in mTOPAttribute stereotype)
    - ○ this property can be ignored
    - ○ fixed to "NOT_APPLICABLE"; see also section 3.4.1.
  - Writeable (contained in mTOPAttribute stereotype)
    - ○ this property can be ignored
    - ○ fixed to "TARGET_AND_REQUESTING_OS"; see also section 3.4.1.

- Enumeration "value" names of data types
  - have only upper case characters
  - words are separated by "_"
  - examples are in the mTOP Naming Convention Guidelines in SD0-1_mTOPDictionary [Ref].
- Enumeration "values" are specified as Enumeration Literals in the enumeration object classes.

Importing Data Types from other DDPs (e.g., Framework DDP)

DDPs can use the general Data Types defined in other DDPs by importing the DDP package which contains the common Data Types defined:

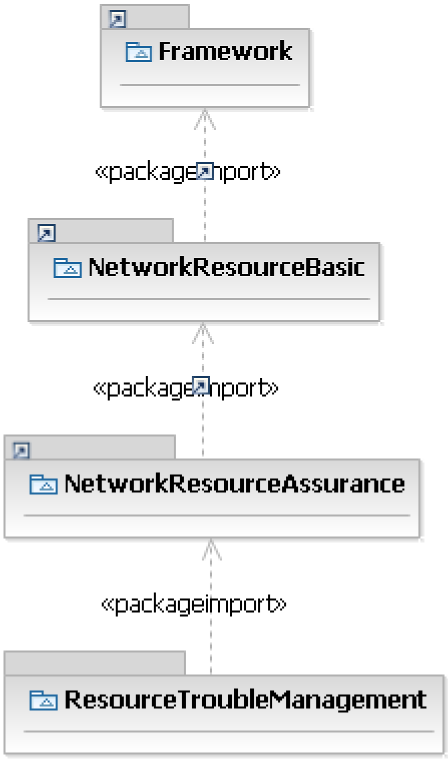


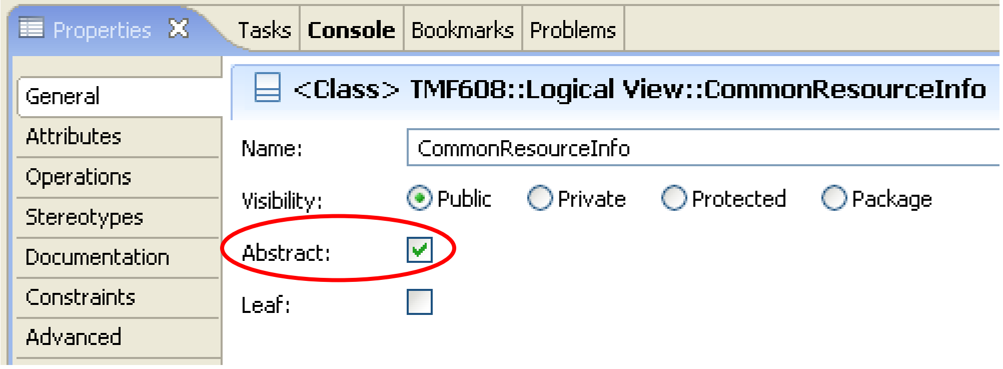
Figure : Data type import using "DDP inheritance" (example)

3.4.3 Object Class Guidelines

Object classes are used to model data entities in the DM model or (service) interfaces in the OM model.

Each object class ▤ comprises:

- Object Class Name
  - follow Upper CamelCase (UCC)
  - must be unique across all class names defined in the whole mTOP IA
  - special definitions are in the mTOP Naming Convention Guidelines in SD0-1_mTOPDictionary [Ref].
- Object Class Documentation ▤
  - contains a short summary of usage
  - refers to the appropriate BA requirement.
- Superclass(es)
  - inheritance and multiple inheritance may be used.
- Abstract Object Class
  - indicates if the object class can be instantiated or is just used for inheritance.



<Class> TMF608::Logical View::CommonResourceInfo

Name: CommonResourceInfo

Visibility: ● Public ○ Private ○ Protected ○ Package

Abstract: ☑

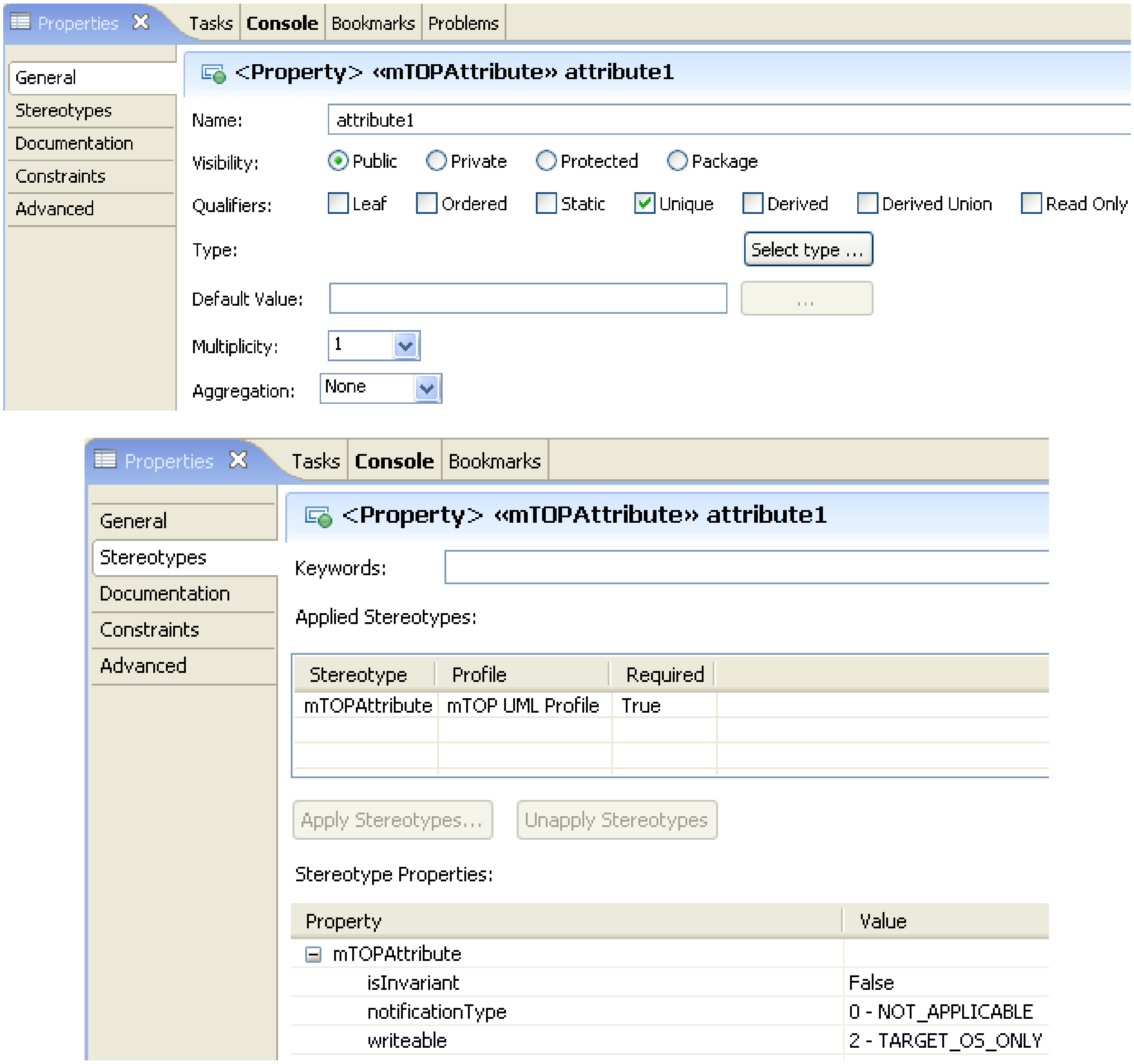
Leaf: □

Figure : Defining an object class as "abstract" (example)

- Object Notifications (contained in mTOPClass stereotype)
  - identifies if creation/deletion notifications have to be send
    "objectCreationNotification" <NO | YES | NOT_APPLICABLE>
    "objectDeletionNotification" <NO | YES | NOT_APPLICABLE>
    "objectDiscoveryNotification" <NO | YES | NOT_APPLICABLE>; see also section 3.4.1.

3.4.4 Object Class Attribute Guidelines

Each **attribute** in an object class comprises:



<Property> «mTOPAttribute» attribute1

Name: attribute1

Visibility: ● Public ○ Private ○ Protected ○ Package

Qualifiers: □ Leaf □ Ordered □ Static ☑ Unique □ Derived □ Derived Union □ Read Only

Type:

Default Value:

Multiplicity: 1

Aggregation: None

<Property> «mTOPAttribute» attribute1

Keywords:

Applied Stereotypes:

Stereotype	Profile	Required
mTOPAttribute	mTOP UML Profile	True

Stereotype Properties:

Property	Value
⊟ mTOPAttribute	
isInvariant	False
notificationType	0 - NOT_APPLICABLE
writeable	2 - TARGET_OS_ONLY

Figure : Defining attribute properties (example)

- Attribute Name
  - follow Lower CamelCase (LCC)
  - must be unique across all attribute names defined in the whole mTOP IA
  - special definitions are in the mTOP Naming Convention Guidelines in SD0-1_mTOPDictionary [Ref]
  - Boolean typed attribute names always start with a verb like ‘is’, 'must', etc. (e.g., ‘isAbstract’) and the whole attribute name must be composed in a way that it is possible to answer it by "true" or "false".
  - Enumeration typed attributes always end with “Kind” (e.g., ‘aggregationKind’).

- Attribute Documentation
    - contains a short summary of usage
    - refers to the specific BA requirement if defined.
- Visibility
    - Public: if the attribute shall be inherited to subclasses; this should be used in general within the IA.
    - Private: if the attribute shall not be inherited to subclasses.
- Qualifiers
    - Leaf: defines if it is allowed to specialize or redefine the attribute in a subclass; default value is false.
    - Ordered: for a multi-valued multiplicity; this specifies whether the values in an instantiation of this attribute are sequentially ordered; default is false
    - Static: specifies whether this feature characterizes individual instances classified by the classifier (false) or the classifier itself (true); default value is false
    - Unique: for a multi-valued multiplicity, this specifies whether the values in an instantiation of this attribute are unique (i.e., no duplicate attribute values are allowed); default is true.
      Excerpt from UML Superstructure Specification, v2.1.1: *When isUnique is true (the default) the collection of values may not contain duplicates. When isOrdered is true (false being the default) the collection of values is ordered. In combination these two allow the type of a property to represent a collection in the following way:*

Ordered	**Unique**	**Collection type**
false	true	Set
true	true	OrderedSet
false	false	Bag
true	false	Sequence

Table : Table 7.1 - Collection types for properties
from UML Superstructure Specification, v2.1.1

-
    - Derived: defines whether the attributes value or values can be computed from other information; default value is false
    - Derived Union: defines whether the attributes value or values are derived as the union of all of the values that are constrained to subset it; default value is false
    - Read Only: if true, the attribute may only be read, and not written by the Requesting OS. The default value is false. This state is dependent on the additional mTOP UML attribute property *writeable*. I.e., if the attribute can not be set/changed by the Requesting OS, it is read only.
- Type
    - refers to a basic or complex data type.
- Default Value
  provides the value that the attribute has to start with in case the value is not provided during creation or already defined because of a system state.
  In the case of a single-value property, the default value shall be specified as an <Opaque Expression>, except for the case in which the type of the property is an Enumeration type, in which

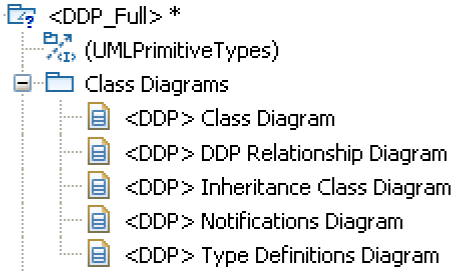
case it shall be an Enumeration Literal selected from the Enumeration. (Note that an Enumeration Literal must be selected from the appropriate browser in RSM and must not simply be typed as a string.)
In the case of a multiple-valued property, the default shall be specified as a comma-separated list of <Opaque Expression> enclosed in braces. (Note that it is not possible to have a multiple-value property of an Enumeration type.)
If a default value makes no sense (e.g., if representing state of the system) = "<Opaque Expression> NA".
No default value for object name = "<Literal Null>".
No default value for String = "<Opaque Expression> empty string".
No default value for Sequence = "<Opaque Expression> empty list".

- Multiplicity
  - defines the number of values the attribute can simultaneously have.
- Aggregation
  An association may represent a composite aggregation (i.e., a whole/part relationship). Only binary associations can be aggregations. Composite aggregation is a strong form of aggregation that requires a part instance be included in at most one composite at a time. If a composite is deleted, all of its parts are normally deleted with it. Note that a part can (where allowed) be removed from a composite before the composite is deleted, and thus not be deleted as part of the composite. Compositions may be linked in a directed acyclic graph with transitive deletion characteristics; that is, deleting an element in one part of the graph will also result in the deletion of all elements of the subgraph below that element. Composition is represented by the isComposite attribute on the part end of the association being set to true.
  - None: the affect on the attribute is unspecified when the parent is deleted
  - Shared: the attribute is not deleted when its parent is deleted
  - Composite: the attribute is deleted when its parent is deleted.
- Invariant (contained in mTOPAttribute stereotype)
  - identifies if the value of the attribute can be changed after it has been created
  - default value "False"; see also section 3.4.1.
- Attribute Notifications (contained in mTOPAttribute stereotype)
  - identifies which kind of notification has to be sent in case of a value change:
    "Changes in the value of the attribute lead to an <AVC | SC> notification." or
    "Changes in the value of the attribute will not be notified automatically."
    See also section 3.4.1.
- Writeable (contained in mTOPAttribute stereotype)
  - identifies who can set/change the value of the attribute: either none OS, or the target OS only, or the requesting OS only or both, the target and requesting OSs.
  - default value "requesting OS only"; see also section 3.4.1.
  - note that the Read Only qualifier has to be true if the requesting OS is not allowed to set/change the value of the attribute.

3.4.5 UML Diagram Guidelines

Objects and their relationships are added to class diagrams. It is recommended to create

- an overview class diagram containing all object classes from the DDP (<DDP> Class Diagram)
- an overview interface diagram containing all interfaces defined in the DDP (<DDP> Interface Diagram)
- a relationship class diagram showing the associated DDPs (<DDP> DDP Relationship Diagram)
- a separate inheritance class diagram in case the overview diagram would be overloaded when showing the inheritance structure (<DDP> Inheritance Class Diagram)
- a class diagram containing the defined notifications (<DDP> Notifications Diagram)
- a class diagram containing the defined types (<DDP> Type Definitions Diagram).



<DDP_Full> *
(UMLPrimitiveTypes)
Class Diagrams
<DDP> Class Diagram
<DDP> DDP Relationship Diagram
<DDP> Inheritance Class Diagram
<DDP> Notifications Diagram
<DDP> Type Definitions Diagram

Figure : Recommended Class Diagrams

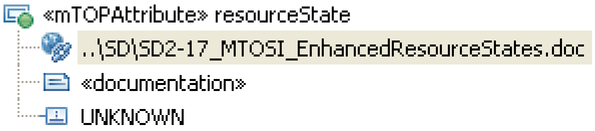
- Additional class diagrams may be established to show specific parts of the DDP in detail.
- State diagrams shall be created for complex state attributes.
- Activity diagrams\Sequence Diagrams (possibly use cases) shall be created for complex operations.
- The class name compartment shall contain the "Qualified Name"
- The class attributes and operation shall show the "Signature".

3.4.6 Use Case Guidelines

Use Cases are only specified in the BA. The use case specification template is contained in section "Specifications for Use Cases" of supporting guideline document SD0-2_mTOPGuidelines_BA.

3.4.7 Reference Guidelines

Where appropriate, a UML artifact should reference external (wrt. UML) documentations (e.g., supporting documents) by a URL .



«mTOPAttribute» resourceState
..\SD\SD2-17_MTOSI_EnhancedResourceStates.doc
«documentation»
UNKNOWN

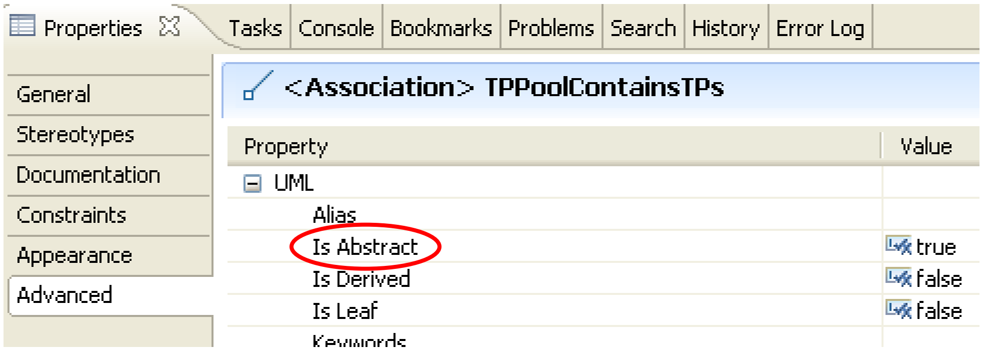
Figure : URL example

3.4.8 Relationship Guidelines

Relationships between the object classes are derived from the dynamic and static requirements in the BA. Each relationship comprises:

- Relationship Name
  - follow Upper CamelCase (UCC)
  - format: "<ClassName><VerbPhrase><ClassName>" where the verb phrase creates a sequence that is readable and meaningful
  - must be unique across all relationship names defined in the whole mTOP IA.
- Stereotype
  - e.g., <<naming>> shall be used if the relationship defines the object naming tree; see 3.5.4.1.
- Relationship Type
  - e.g., inheritance, association (composition, aggregation, association class), dependency, realisation.
- Role Names
  - identifies the role that the object plays at this end of the relationship
  - follow Lower CamelCase (LCC)
  - the plural form shall be "<name>List"
  - special definitions are in the mTOP Naming Convention Guidelines in SD0-1_mTOPDictionary [Ref]
  - only navigable relationships have role names.
- Cardinality
  - identifies the number of objects that can have this relationship.
- Constraint(s)
  - list the constraint(s) under which the relationship can exist.
- Abstract
  - It is recommended to create relationships which are just for explanation to the reader of the model. These relationships should be defined as "abstract", they are not navigable and have no role names. They will therefore not be taken into account in the protocol specific specification. This can for example be used to show the relationship to the object which is retrieved by a get-operation in another DDP.

  Example:



<Association> TPPoolContainsTPs

Property	Value
UML	
Alias	
Is Abstract	true
Is Derived	false
Is Leaf	false
Keywords	

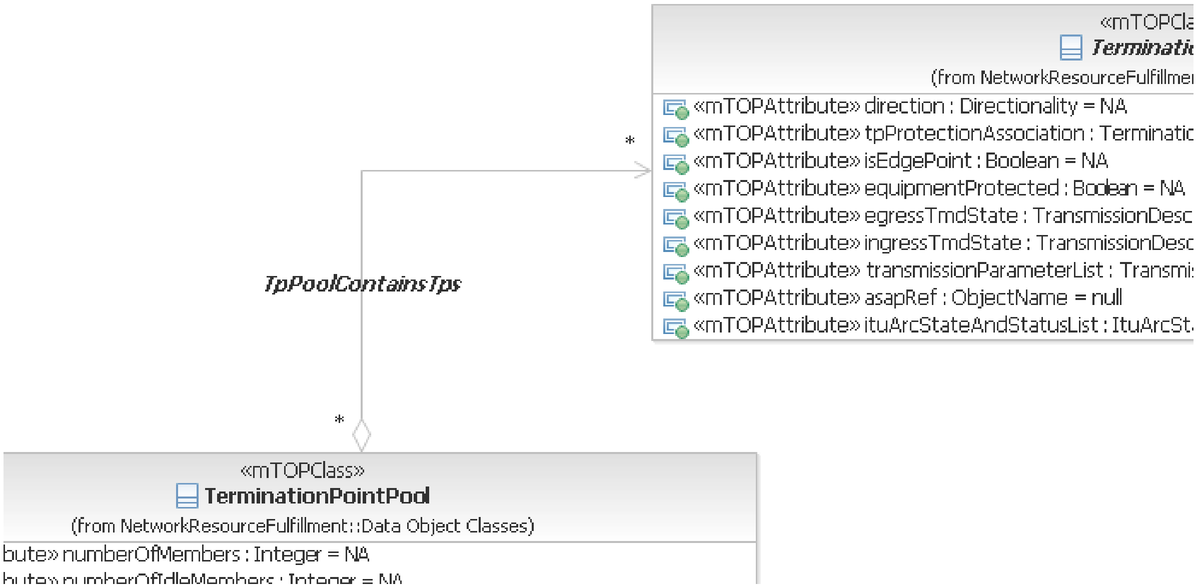

«mTOPCla
Terminati
(from NetworkResourceFulfillme
«mTOPAttribute» direction : Directionality = NA
«mTOPAttribute» tpProtectionAssociation : Terminatic
«mTOPAttribute» isEdgePoint : Boolean = NA
«mTOPAttribute» equipmentProtected : Boolean = NA
«mTOPAttribute» egressTmdState : TransmissionDesc
«mTOPAttribute» ingressTmdState : TransmissionDesc
«mTOPAttribute» transmissionParameterList : Transmi
«mTOPAttribute» asapRef : ObjectName = null
«mTOPAttribute» ituArcStateAndStatusList : ItuArcSt
*
TpPoolContainsTps
*
«mTOPClass»
TerminationPointPool
(from NetworkResourceFulfillment::Data Object Classes)
bute» numberOfMembers : Integer = NA
bute» numberOfIdleMembers : Integer = NA

Figure : "Abstract" relationship (example)

3.4.9 Traceability Guidelines

Traceability from IA back to BA shall be provided in 2 ways:

1. Where appropriate, a UML artifact should reference the corresponding requirement and/or use case identifier in the documentation field.
2. Traceability matrices have to be provided as defined in 2. Traceability Guidelines in section Appendix E: known issues with the IA Guidelines.

3.4.10 Product Identification Guidelines

As a general guideline, the UML artifacts are not product specific. If, in an exceptional case, it is necessary to define product specific artefacts, the product shall be identified in the stereotype of the artefact.

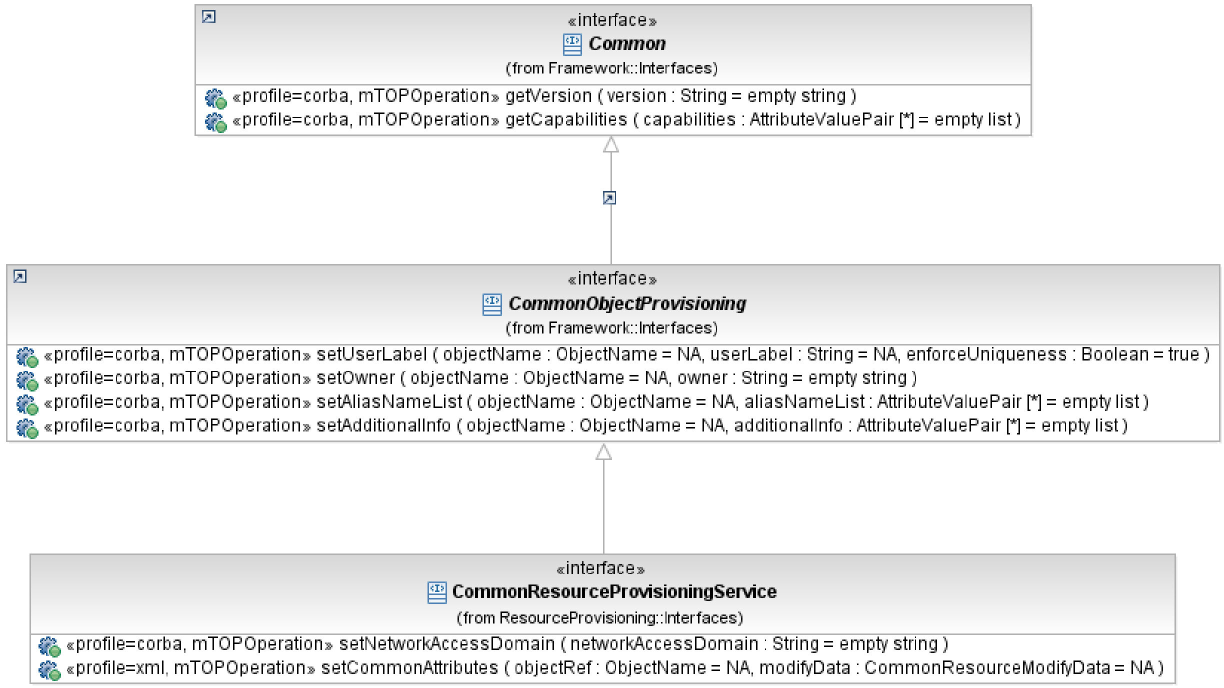

«interface»
Common
(from Framework::Interfaces)
«profile=corba, mTOPOperation» getVersion (version : String = empty string)
«profile=corba, mTOPOperation» getCapabilities (capabilities : AttributeValuePair [*] = empty list)
«interface»
CommonObjectProvisioning
(from Framework::Interfaces)
«profile=corba, mTOPOperation» setUserLabel (objectName : ObjectName = NA, userLabel : String = NA, enforceUniqueness : Boolean = true)
«profile=corba, mTOPOperation» setOwner (objectName : ObjectName = NA, owner : String = empty string)
«profile=corba, mTOPOperation» setAliasNameList (objectName : ObjectName = NA, aliasNameList : AttributeValuePair [*] = empty list)
«profile=corba, mTOPOperation» setAdditionalInfo (objectName : ObjectName = NA, additionalInfo : AttributeValuePair [*] = empty list)
«interface»
CommonResourceProvisioningService
(from ResourceProvisioning::Interfaces)
«profile=corba, mTOPOperation» setNetworkAccessDomain (networkAccessDomain : String = empty string)
«profile=xml, mTOPOperation» setCommonAttributes (objectRef : ObjectName = NA, modifyData : CommonResourceModifyData = NA)

Figure : Product specific artefacts (example)

3.5 Data Model (DM) DDP Guidelines

3.5.1 Overview

There is an mTOP UML Model for each mTOP DM DDP; see section 2.1.2.

3.5.2 Location

An mTOP IA UML data model file corresponds to an mTOP DM DDP and is contained in the ***IA*** folder of this DDP; see section 2.1.3.

3.5.3 Object Class Guidelines

DM Model object classes are derived from the static requirements (i.e., category I requirements) in the BA.

3.5.4 Relationship Guidelines

3.5.4.1 Naming relationship

Naming relationships are defined using the composition association with:

- a private role name at the subordinate class

- stereotype = <<naming>>
- abstract = true
- no relationship name.

The attribute which is used for naming is identified by an additional stereotype <<objectRefKey>>.

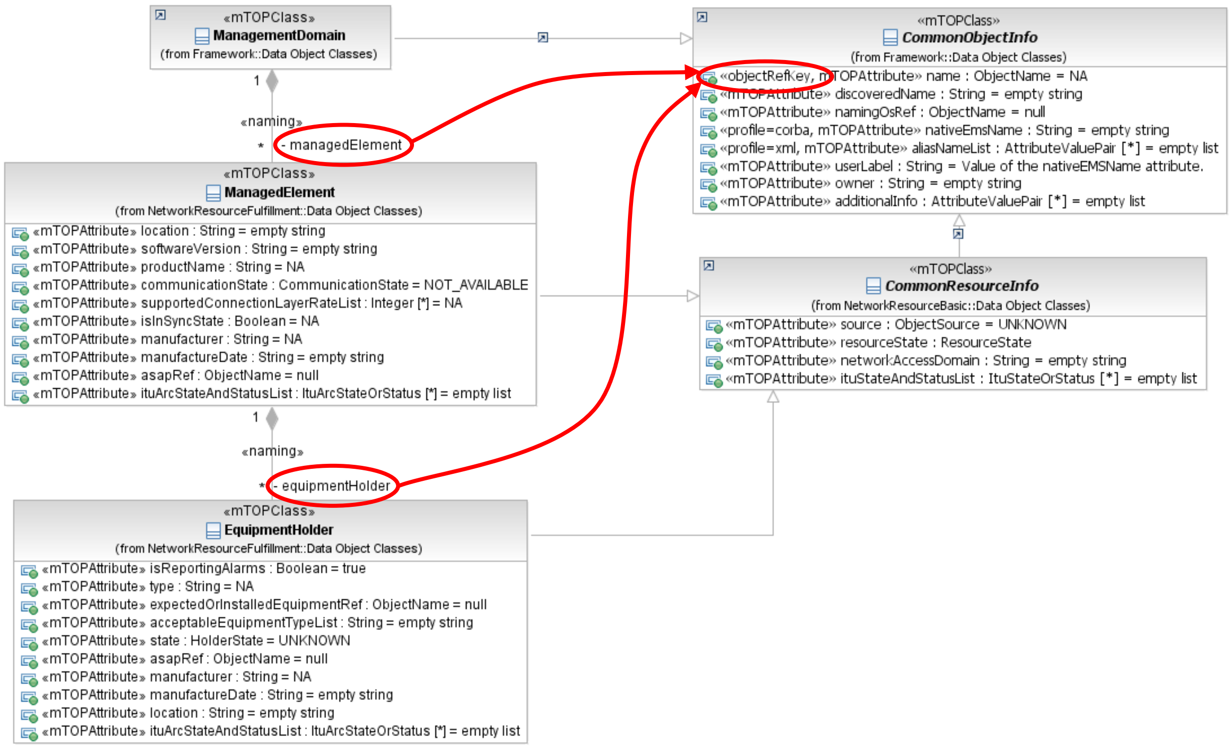

«mTOPClass»
ManagementDomain
(from Framework::Data Object Classes)
1
«naming»
*
- managedElement
«mTOPClass»
ManagedElement
(from NetworkResourceFulfillment::Data Object Classes)
«mTOPAttribute» location : String = empty string
«mTOPAttribute» softwareVersion : String = empty string
«mTOPAttribute» productName : String = NA
«mTOPAttribute» communicationState : CommunicationState = NOT_AVAILABLE
«mTOPAttribute» supportedConnectionLayerRateList : Integer [*] = NA
«mTOPAttribute» isInSyncState : Boolean = NA
«mTOPAttribute» manufacturer : String = NA
«mTOPAttribute» manufactureDate : String = empty string
«mTOPAttribute» asapRef : ObjectName = null
«mTOPAttribute» ituArcStateAndStatusList : ItuArcStateOrStatus [*] = empty list
1
«naming»
*
- equipmentHolder
«mTOPClass»
EquipmentHolder
(from NetworkResourceFulfillment::Data Object Classes)
«mTOPAttribute» isReportingAlarms : Boolean = true
«mTOPAttribute» type : String = NA
«mTOPAttribute» expectedOrInstalledEquipmentRef : ObjectName = null
«mTOPAttribute» acceptableEquipmentTypeList : String = empty string
«mTOPAttribute» state : HolderState = UNKNOWN
«mTOPAttribute» asapRef : ObjectName = null
«mTOPAttribute» manufacturer : String = NA
«mTOPAttribute» manufactureDate : String = empty string
«mTOPAttribute» location : String = empty string
«mTOPAttribute» ituArcStateAndStatusList : ItuArcStateOrStatus [*] = empty list
«mTOPClass»
CommonObjectInfo
(from Framework::Data Object Classes)
«objectRefKey, mTOPAttribute» name : ObjectName = NA
«mTOPAttribute» discoveredName : String = empty string
«mTOPAttribute» namingOsRef : ObjectName = null
«profile=corba, mTOPAttribute» nativeEmsName : String = empty string
«profile=xml, mTOPAttribute» aliasNameList : AttributeValuePair [*] = empty list
«mTOPAttribute» userLabel : String = Value of the nativeEMSName attribute.
«mTOPAttribute» owner : String = empty string
«mTOPAttribute» additionalInfo : AttributeValuePair [*] = empty list
«mTOPClass»
CommonResourceInfo
(from NetworkResourceBasic::Data Object Classes)
«mTOPAttribute» source : ObjectSource = UNKNOWN
«mTOPAttribute» resourceState : ResourceState
«mTOPAttribute» networkAccessDomain : String = empty string
«mTOPAttribute» ituStateAndStatusList : ItuStateOrStatus [*] = empty list

Figure : Naming relationship (example)

3.5.4.2 Instance Reference relationship

An object instance (name) is referenced in another object by either using the **aggregation** association or the **simple** association with:

- a relationship name
  - format: "<ClassName><VerbPhrase><ClassName>" or "<ClassName><VerbPhrase>" where the verb phrase creates a sequence that is readable and meaningful
  - must be unique across all relationship names defined in the whole mTOP IA
- a public role name at the referenced class side
- default value = NA
- notificationType = ATTRIBUTE_VALUE_CHANGE or NO_NOTIFICATION.

Example:

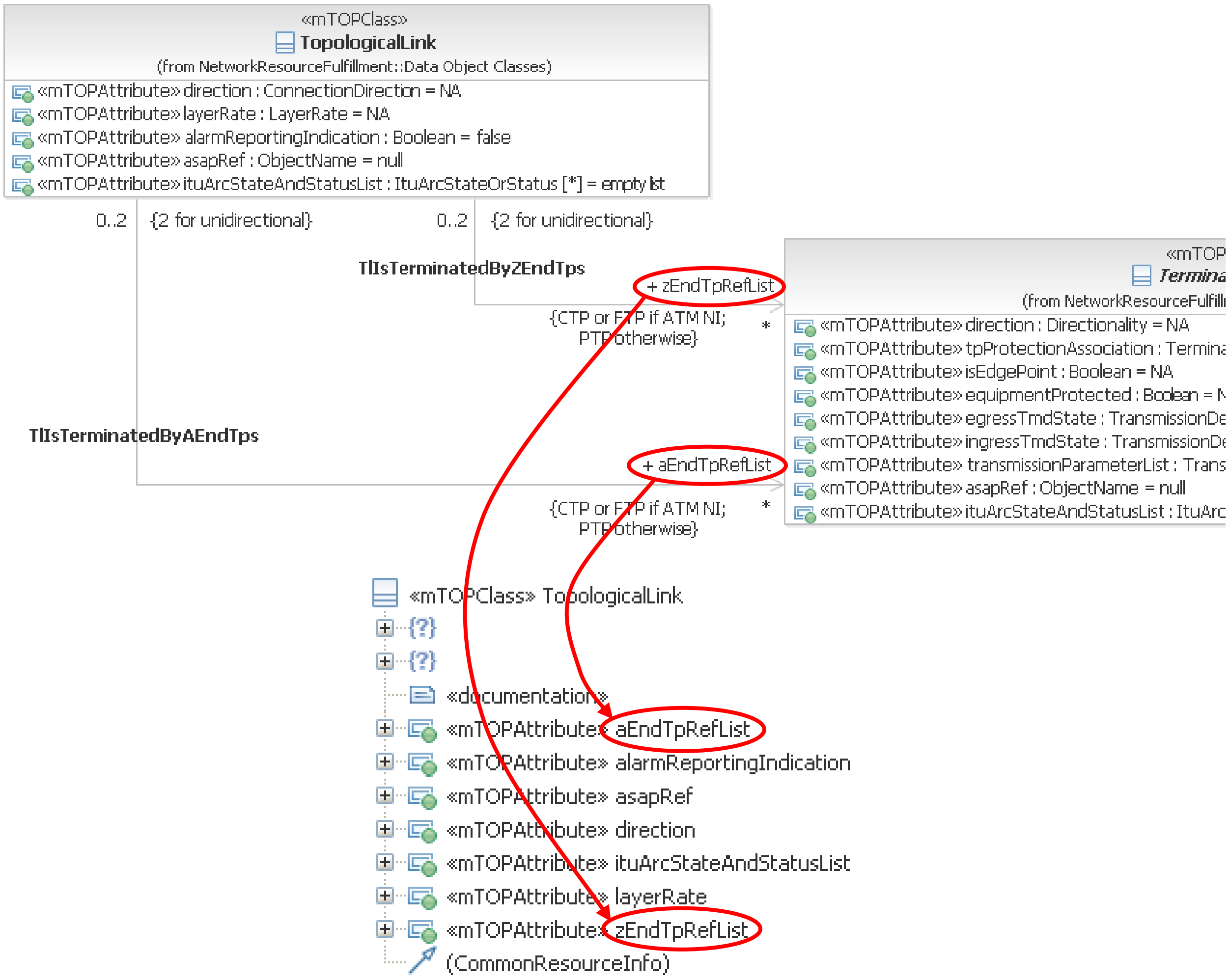
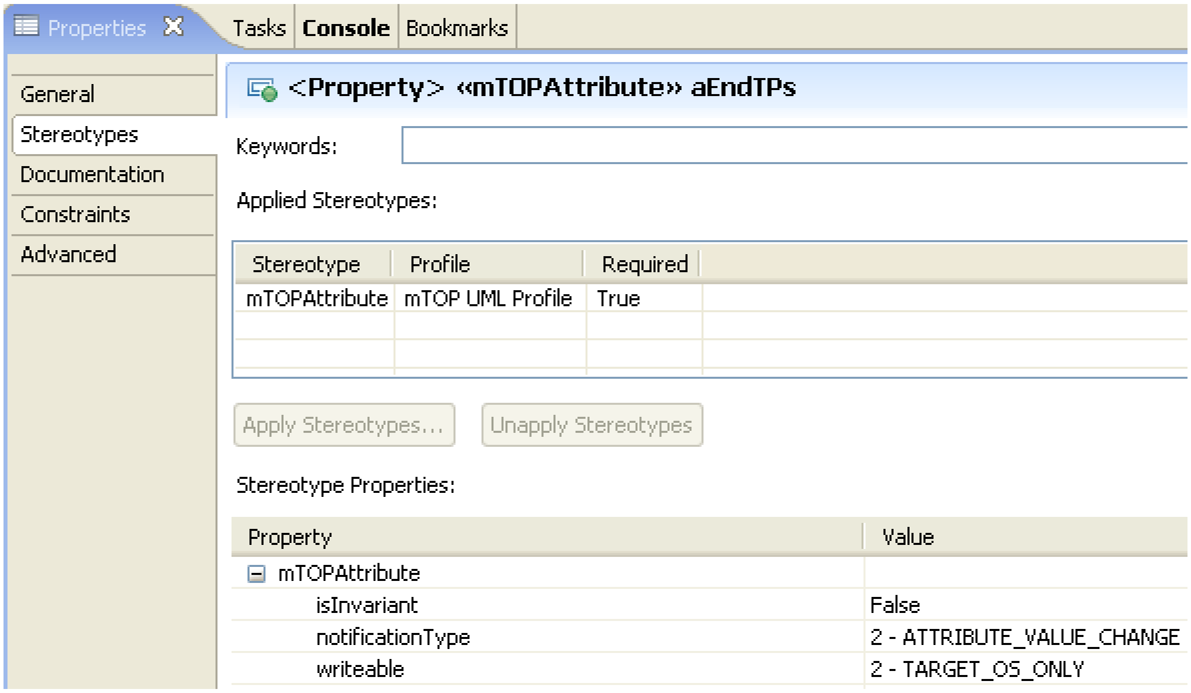

«mTOPClass»
TopologicalLink
(from NetworkResourceFulfillment::Data Object Classes)
«mTOPAttribute» direction : ConnectionDirection = NA
«mTOPAttribute» layerRate : LayerRate = NA
«mTOPAttribute» alarmReportingIndication : Boolean = false
«mTOPAttribute» asapRef : ObjectName = null
«mTOPAttribute» ituArcStateAndStatusList : ItuArcStateOrStatus [*] = empty list
0..2
{2 for unidirectional}
0..2
{2 for unidirectional}
TlIsTerminatedByZEndTps
+ zEndTpRefList
{CTP or FTP if ATM NI; PTP otherwise}
*
TlIsTerminatedByAEndTps
+ aEndTpRefList
{CTP or FTP if ATM NI; PTP otherwise}
*
«mTOP
Termina
(from NetworkResourceFulfill
«mTOPAttribute» direction : Directionality = NA
«mTOPAttribute» tpProtectionAssociation : Termina
«mTOPAttribute» isEdgePoint : Boolean = NA
«mTOPAttribute» equipmentProtected : Boolean = N
«mTOPAttribute» egressTmdState : TransmissionDe
«mTOPAttribute» ingressTmdState : TransmissionDe
«mTOPAttribute» transmissionParameterList : Trans
«mTOPAttribute» asapRef : ObjectName = null
«mTOPAttribute» ituArcStateAndStatusList : ItuArc
«mTOPClass» TopologicalLink
{?}
{?}
«documentation»
«mTOPAttribute» aEndTpRefList
«mTOPAttribute» alarmReportingIndication
«mTOPAttribute» asapRef
«mTOPAttribute» direction
«mTOPAttribute» ituArcStateAndStatusList
«mTOPAttribute» layerRate
«mTOPAttribute» zEndTpRefList
(CommonResourceInfo)

Figure : Instance Reference relationship using simple association (example)



<Property> «mTOPAttribute» aEndTPs

Keywords:

Applied Stereotypes:

Stereotype	Profile	Required
mTOPAttribute	mTOP UML Profile	True

Stereotype Properties:

Property	Value
⊟ mTOPAttribute	
isInvariant	False
notificationType	2 - ATTRIBUTE_VALUE_CHANGE
writeable	2 - TARGET_OS_ONLY

Figure : Additional properties of role attribute in example

In the example (Figure and Figure), the names of the A-End Termination Points (TP) shall be part of the information that is provided when the Topological Link (TL) is retrieved via the interface using the role name as the identifier (e.g., aEndTPs shall be implemented as an attribute of TL).

3.6 Service Interface Guidelines

Service Interfaces are specified in the OM-DDPs.

3.6.1 Overview

There is an mTOP UML Model for each mTOP OM DDP; see section 2.1.2.

3.6.2 Location

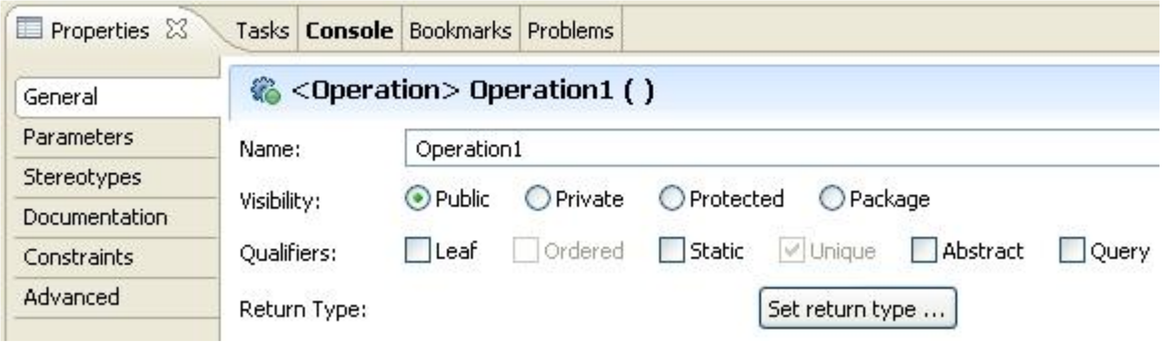
An mTOP IA UML operations model file corresponds to an mTOP OM DDP and is contained in the ***IA*** folder of this DDP; see section 2.1.3.

3.6.3 Interface Guidelines

OM Model Interfaces are derived from the dynamic requirements (i.e., category II requirements) in the BA. The Interfaces are modelled as object classes (i.e., the service interfaces).

3.6.4 Operation Guidelines

Operations (within the Interfaces) are derived from the dynamic requirements and use cases in the BA. Each operation comprises:



<Operation> Operation1 ( )

Name: Operation1

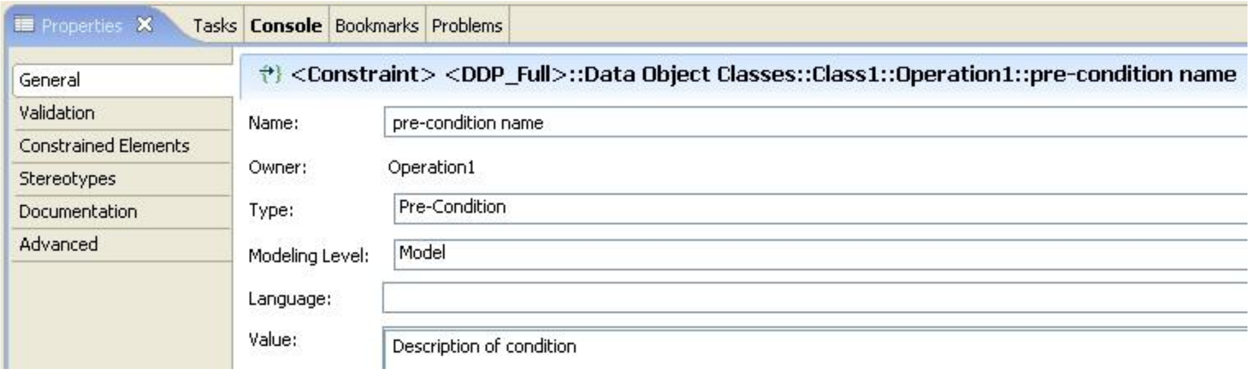
Visibility: ● Public ○ Private ○ Protected ○ Package

Qualifiers: □ Leaf □ Ordered □ Static ☑ Unique □ Abstract □ Query

Return Type: Set return type ...

Figure : Defining operation properties (example)

- Operation Name
  - follow Lower CamelCase (LCC)
  - must be unique across all operation names defined in the whole mTOP IA
  - special definitions are in the mTOP Naming Convention Guidelines in SD0-1_mTOPDictionary [Ref].
- Visibility
  - Public: if the operation shall be inherited to subclasses
  - Private: if the operation shall not be inherited to subclasses = default value.
- Qualifiers
  - Leaf: defines if it is allowed to specialize or redefine the operation in a subclass; default value is false.
  - Static: specifies whether the operation has class or instance scope.
  - Abstract: if true, then the behavioral feature does not have an implementation, and one must be supplied by a more specific element. If false, the behavioral feature must have an implementation in the classifier or one must be inherited from a more general element. Default value is false.
  - Query: defines whether an execution of the BehavioralFeature leaves the state of the system unchanged (= true) or whether side effects may occur (= false); default value is false.
  - Return Type : always "void".
- Operation Documentation
  - contains a short summary of usage.
- Pre-condition(s)
  Pre-condition(s) are modeled as Constraints. They list the conditions that have to be true before the operation can be started (i.e., if not true, the operation will not start at all).



<Constraint> <DDP_Full>::Data Object Classes::Class1::Operation1::pre-condition name

Name: pre-condition name

Owner: Operation1

Type: Pre-Condition

Modeling Level: Model

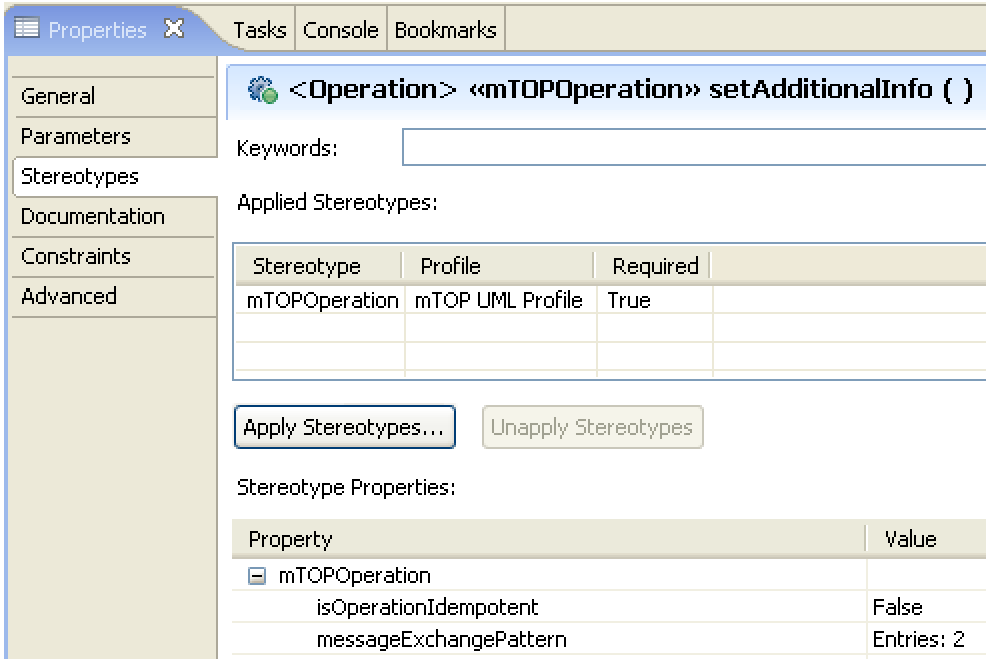
Language:

Value: Description of condition

Note: It is also possible to define the pre-condition in OCL.

Figure : Defining operation pre-conditions

- Parameter(s)
  see below.
- Post-condition(s)
  Post-condition(s) are modeled as Constraints. They describe the state of the system after the operation has been successfully executed.
- Idempotency (contained in mTOPOperation stereotype)
  - this additional property defines if the operation is idempotent or not; see also section 3.4.1.
- Message Exchange Pattern (MEP) (contained in mTOPOperation stereotype)
  - the MEP fully identifies the messages and the choreography (sequencing and cardinality) of the messages independently from a business activity (SRR | ARR | SIT | ABR | SFB | AFB | SN | AN) ; see also section 3.4.1 and SD2-5 Communication Styles [**Ref**].



<Operation> «mTOPOperation» setAdditionalInfo ( )

Keywords:

Applied Stereotypes:

Stereotype	Profile	Required
mTOPOperation	mTOP UML Profile	True

Stereotype Properties:

Property	Value
⊟ mTOPOperation	
isOperationIdempotent	False
messageExchangePattern	Entries: 2

Figure : mTOPOperation additional properties (example)

- Operation Exceptions
  List the allowed exceptions (together with a failure reason) for the operation.
  - ACCESS_DENIED - Raised when the requesting OS is not permitted to perform the operation.
  - CAPACITY_EXCEEDED - Raised when the request will result in resources being created or activated beyond the capacity supported by the NE or target OS.
  - COMM_LOSS - Raised when the target OS (which is a top-level OS) is unable to communicate with the subordinate OS and communication is required to complete the request.
  - ENTITY_NOT_FOUND - Raised when the specified object instance does not exist.
  - INTERNAL_ERROR - Raised when the request has resulted in an OS internal error.
  - INVALID_INPUT - Raised when the operation contains an input parameter that is syntactically incorrect or identifies an object of the wrong type or is out of range.
  - NE_COMM_LOSS - Raised when communication to the managed element is lost.
  - NOT_IMPLEMENTED - Raised when the target OS does not support this operation.
  - NOT_IN_VALID_STATE - Raised when the state of the specified object is such that the target OS cannot perform the operation.
  - OBJECT_IN_USE - Raised when the object identified in the request is currently in use.
  - PROTECTION_EFFORT_NOT_MET - Raised when the level of protection effort in the request cannot be met by the target OS.
  - TIMESLOT_IN_USE - Raised when a requested timeslot is already in use.
  - TP_INVALID_ENDPOINT - Raised when the specified TP does not exist or cannot be created.
  - UNABLE_TO_COMPLY - Raised when the target OS cannot respond to the request.

- UNSUPPORTED_ROUTING_CONSTRAINTS - Raised when the target OS is unable to satisfy the requested routing constraints.
- USERLABEL_IN_USE - Raised when the user label uniqueness constraint is not met; the specified user label is currently being used.

The exceptions are contained in the folder "Operation Exceptions" in the Framework DDP.

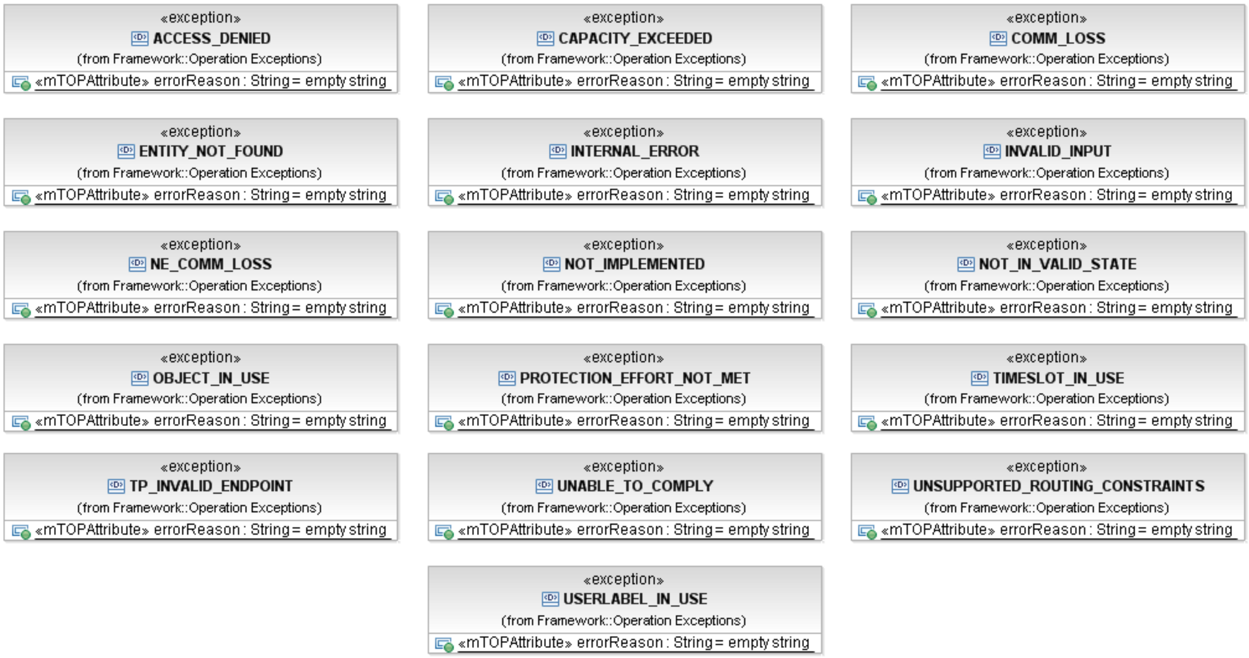
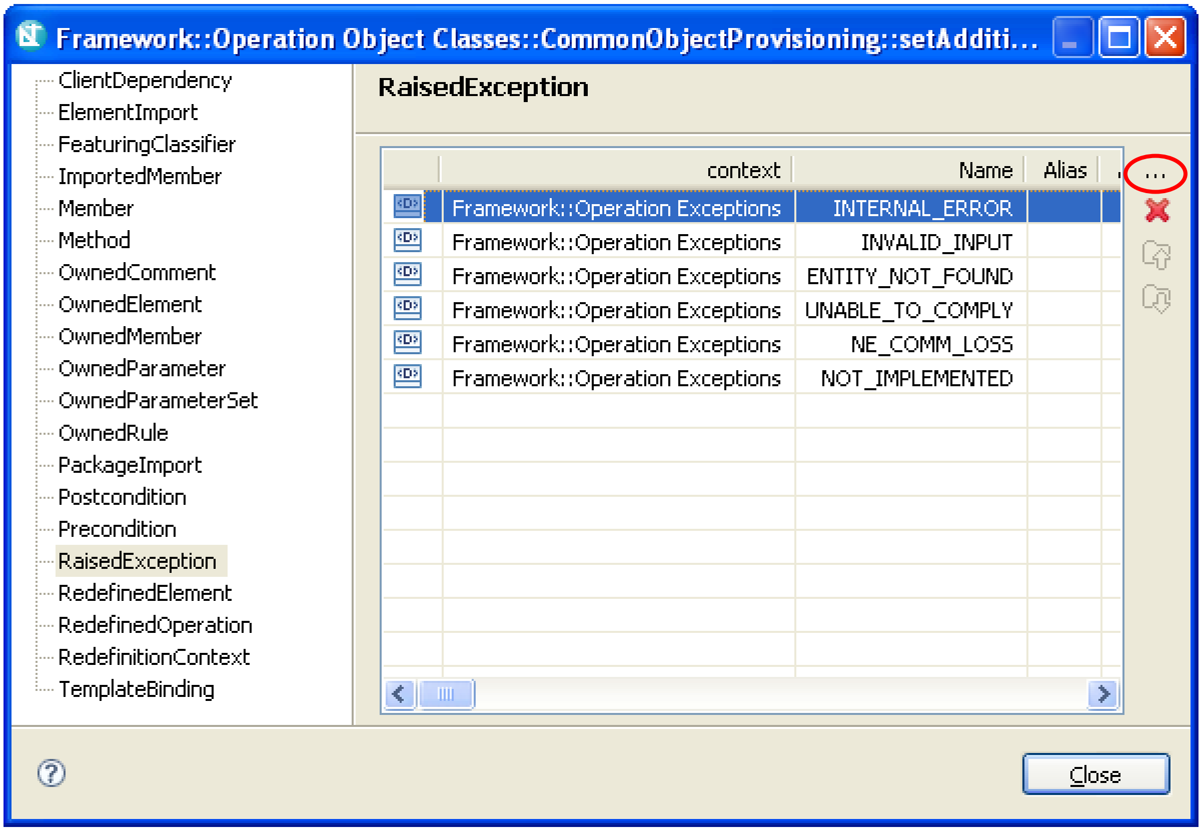

«exception»
ACCESS_DENIED
(from Framework::Operation Exceptions)
«mTOPAttribute» errorReason : String = empty string
«exception»
CAPACITY_EXCEEDED
(from Framework::Operation Exceptions)
«mTOPAttribute» errorReason : String = empty string
«exception»
COMM_LOSS
(from Framework::Operation Exceptions)
«mTOPAttribute» errorReason : String = empty string
«exception»
ENTITY_NOT_FOUND
(from Framework::Operation Exceptions)
«mTOPAttribute» errorReason : String = empty string
«exception»
INTERNAL_ERROR
(from Framework::Operation Exceptions)
«mTOPAttribute» errorReason : String = empty string
«exception»
INVALID_INPUT
(from Framework::Operation Exceptions)
«mTOPAttribute» errorReason : String = empty string
«exception»
NE_COMM_LOSS
(from Framework::Operation Exceptions)
«mTOPAttribute» errorReason : String = empty string
«exception»
NOT_IMPLEMENTED
(from Framework::Operation Exceptions)
«mTOPAttribute» errorReason : String = empty string
«exception»
NOT_IN_VALID_STATE
(from Framework::Operation Exceptions)
«mTOPAttribute» errorReason : String = empty string
«exception»
OBJECT_IN_USE
(from Framework::Operation Exceptions)
«mTOPAttribute» errorReason : String = empty string
«exception»
PROTECTION_EFFORT_NOT_MET
(from Framework::Operation Exceptions)
«mTOPAttribute» errorReason : String = empty string
«exception»
TIMESLOT_IN_USE
(from Framework::Operation Exceptions)
«mTOPAttribute» errorReason : String = empty string
«exception»
TP_INVALID_ENDPOINT
(from Framework::Operation Exceptions)
«mTOPAttribute» errorReason : String = empty string
«exception»
UNABLE_TO_COMPLY
(from Framework::Operation Exceptions)
«mTOPAttribute» errorReason : String = empty string
«exception»
UNSUPPORTED_ROUTING_CONSTRAINTS
(from Framework::Operation Exceptions)
«mTOPAttribute» errorReason : String = empty string
«exception»
USERLABEL_IN_USE
(from Framework::Operation Exceptions)
«mTOPAttribute» errorReason : String = empty string

Figure : Operation Exceptions Class Diagram

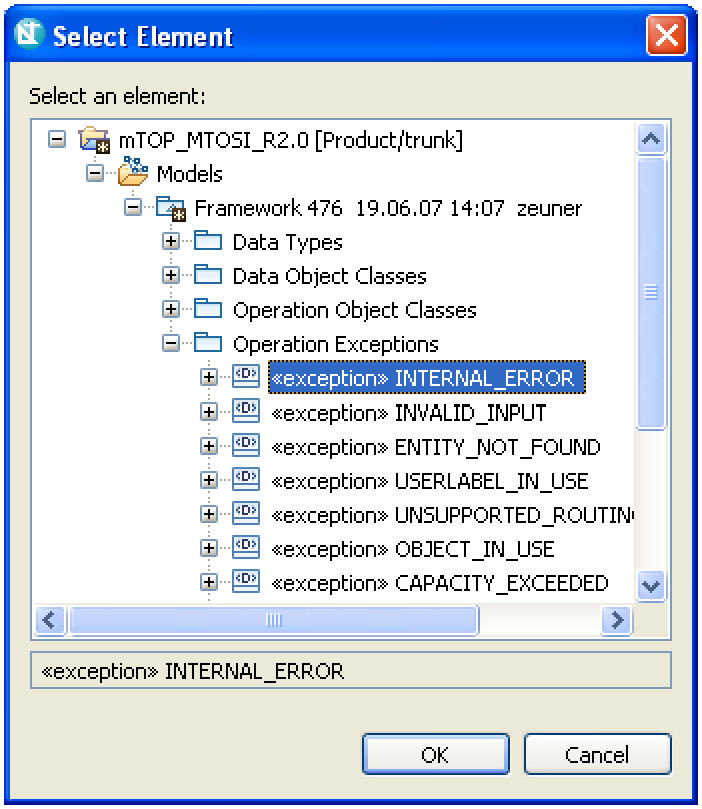
The relevant exceptions can be added to an operation by right click on the operation, choose **UML Properties …**, choose **RaisedException** and browse to the exception class via the "**…**" button on the right:



RaisedException

	context	Name	Alias
	Framework::Operation Exceptions	INTERNAL_ERROR	
	Framework::Operation Exceptions	INVALID_INPUT	
	Framework::Operation Exceptions	ENTITY_NOT_FOUND	
	Framework::Operation Exceptions	UNABLE_TO_COMPLY	
	Framework::Operation Exceptions	NE_COMM_LOSS	
	Framework::Operation Exceptions	NOT_IMPLEMENTED	

Figure : Raised exception window

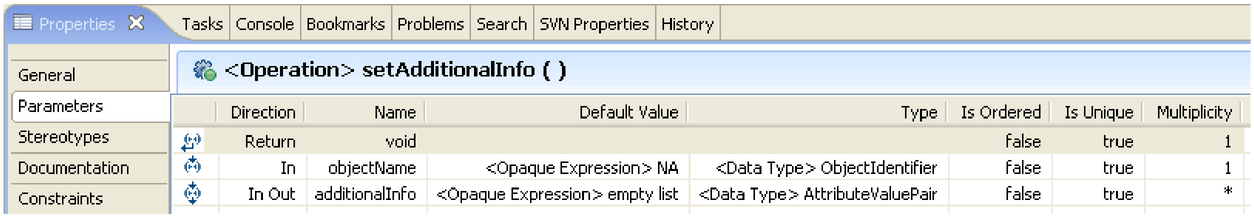

Select Element
Select an element:
mTOP_MTOSI_R2.0 [Product/trunk]
Models
Framework 476 19.06.07 14:07 zeuner
Data Types
Data Object Classes
Operation Object Classes
Operation Exceptions
«exception» INTERNAL_ERROR
«exception» INVALID_INPUT
«exception» ENTITY_NOT_FOUND
«exception» USERLABEL_IN_USE
«exception» UNSUPPORTED_ROUTIN
«exception» OBJECT_IN_USE
«exception» CAPACITY_EXCEEDED
OK
Cancel

«exception» INTERNAL_ERROR

Figure : Select exception window

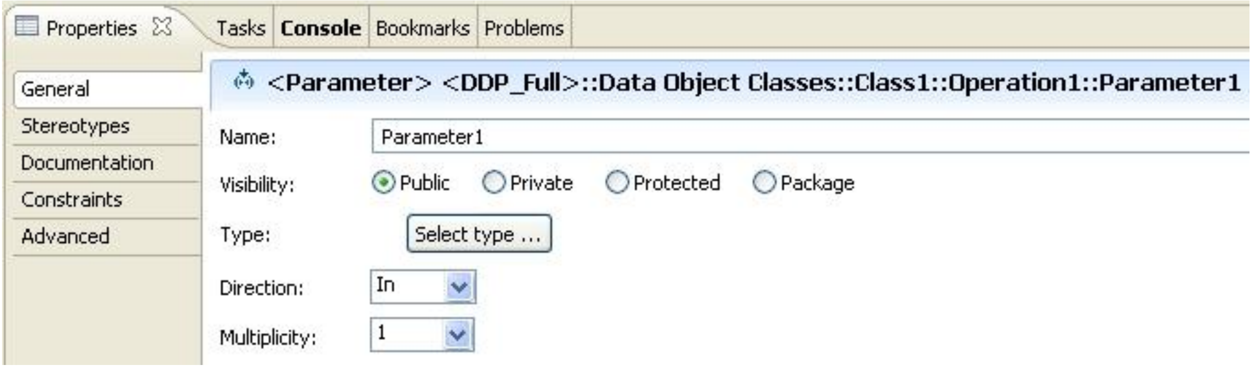
3.6.5 Operation Parameter Guidelines

Each **parameter** in an operation comprises:



<Operation> setAdditionalInfo ( )

	Direction	Name	Default Value	Type	Is Ordered	Is Unique	Multiplicity
	Return	void			false	true	1
	In	objectName	<Opaque Expression> NA	<Data Type> ObjectIdentifier	false	true	1
	In Out	additionalInfo	<Opaque Expression> empty list	<Data Type> AttributeValuePair	false	true	*



<Parameter> <DDP_Full>::Data Object Classes::Class1::Operation1::Parameter1

Name: Parameter1

Visibility: ● Public ○ Private ○ Protected ○ Package

Type: Select type ...

Direction: In

Multiplicity: 1

Figure : Defining operation parameters

- Name
  - follow Lower CamelCase (LCC)
  - special definitions are in the mTOP Naming Convention Guidelines in SD0-1_mTOPDictionary [Ref].
- Visibility
  - Public: if the attribute shall be inherited to subclasses
  - Private: if the attribute shall not be inherited to subclasses = default value.
- Type
  - refers to a basic or complex data type
  - a list of input (in a few cases also output) parameters could also be combined in a data type
  - if possible, the following common set of resource create and modify input parameters shall be used:

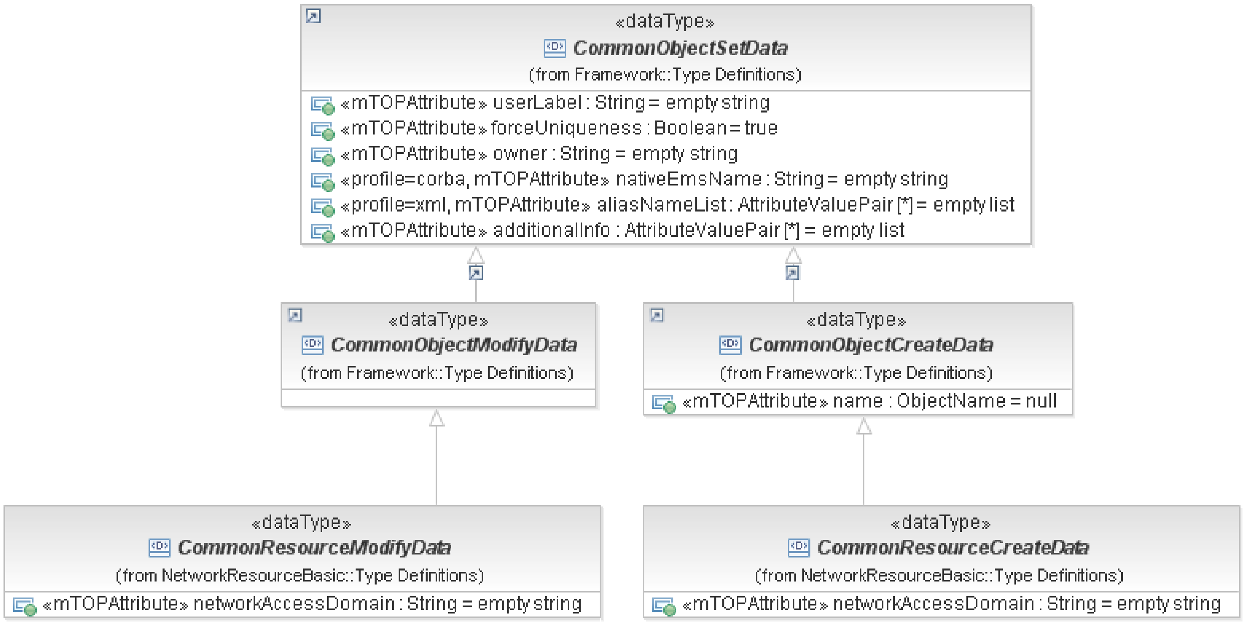

«dataType»
CommonObjectSetData
(from Framework::Type Definitions)
«mTOPAttribute» userLabel : String = empty string
«mTOPAttribute» forceUniqueness : Boolean = true
«mTOPAttribute» owner : String = empty string
«profile=corba, mTOPAttribute» nativeEmsName : String = empty string
«profile=xml, mTOPAttribute» aliasNameList : AttributeValuePair [*] = empty list
«mTOPAttribute» additionalInfo : AttributeValuePair [*] = empty list
«dataType»
CommonObjectModifyData
(from Framework::Type Definitions)
«dataType»
CommonObjectCreateData
(from Framework::Type Definitions)
«mTOPAttribute» name : ObjectName = null
«dataType»
CommonResourceModifyData
(from NetworkResourceBasic::Type Definitions)
«mTOPAttribute» networkAccessDomain : String = empty string
«dataType»
CommonResourceCreateData
(from NetworkResourceBasic::Type Definitions)
«mTOPAttribute» networkAccessDomain : String = empty string

Figure : Common Resource Create and Modify parameters

- Direction
  - In | In Out | Out | Return .
- Default Value
  provides the value that the parameter has to start with in case the value is not provided.
  In the case of a single-value property, the default value shall be specified as an <Opaque Expression>, except for the case in which the type of the property is an Enumeration type, in which case it shall be an Enumeration Literal selected from the Enumeration. (Note that an Enumeration Literal must be selected from the appropriate browser in RSM and must not simply be typed as a string.)
  In the case of a multiple-valued property, the default shall be specified as a comma-separated list of <Opaque Expression> enclosed in braces. (Note that it is not possible to have a multiple-value property of an Enumeration type.)
  If a default value makes no sense (e.g., the requesting OS has to provide a value for this parameter) = "<Opaque Expression> NA".
  No default value for object identifier = "<Literal Null>".
  No default value for String = "<Opaque Expression> empty string".
  No default value for Sequence = "<Opaque Expression> empty list".
- Is Ordered
  for a multivalued parameter; the order of the values is important.
- Is Unique
  for a multivalued parameter, no duplicate values are allowed.
- Multiplicity
  defines the number of values the parameter can simultaneously have.
- parameter Documentation
  contains a short summary of usage.

For attributes in Data Types which are used as a parameter list (e.g., CommonResourceCreateModifyData; see Figure), the additional properties contained in the mTOPAttribute stereotype are not meaningful. The properties shall be fixed to:

- isInvariant fixed to "True"
- notificationType fixed to "NOT_APPLICABLE"
- writeable fixed to "REQUESTING_OS_ONLY"

3.6.6 Relationship Guidelines

It is recommended to use the dependency relationship to associate input or output parameter lists to the operations in an object. These relationships are just for explanation to the reader of the model. No special care has to be taken in the protocol specific part; they are usually modelled in operations of the dependent class.

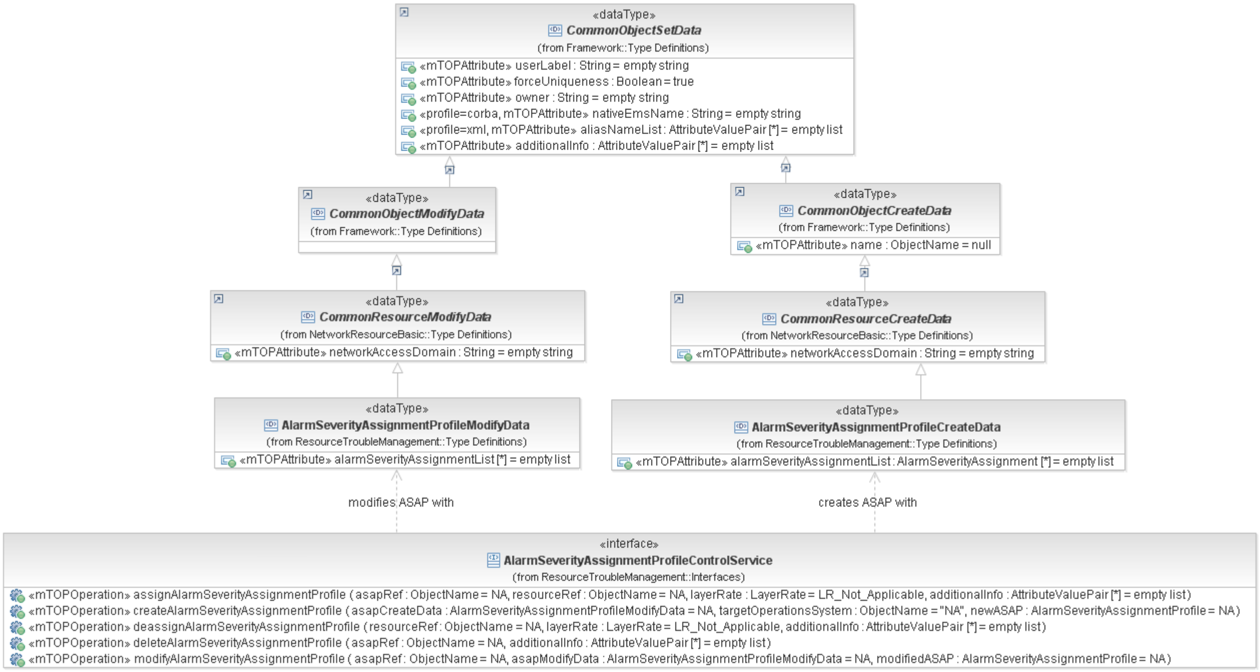

«dataType»
CommonObjectSetData
(from Framework::Type Definitions)
«mTOPAttribute» userLabel : String = empty string
«mTOPAttribute» forceUniqueness : Boolean = true
«mTOPAttribute» owner : String = empty string
«profile=corba, mTOPAttribute» nativeEmsName : String = empty string
«profile=xml, mTOPAttribute» aliasNameList : AttributeValuePair [*] = empty list
«mTOPAttribute» additionalInfo : AttributeValuePair [*] = empty list
«dataType»
CommonObjectModifyData
(from Framework::Type Definitions)
«dataType»
CommonObjectCreateData
(from Framework::Type Definitions)
«mTOPAttribute» name : ObjectName = null
«dataType»
CommonResourceModifyData
(from NetworkResourceBasic::Type Definitions)
«mTOPAttribute» networkAccessDomain : String = empty string
«dataType»
CommonResourceCreateData
(from NetworkResourceBasic::Type Definitions)
«mTOPAttribute» networkAccessDomain : String = empty string
«dataType»
AlarmSeverityAssignmentProfileModifyData
(from ResourceTroubleManagement::Type Definitions)
«mTOPAttribute» alarmSeverityAssignmentList [*] = empty list
«dataType»
AlarmSeverityAssignmentProfileCreateData
(from ResourceTroubleManagement::Type Definitions)
«mTOPAttribute» alarmSeverityAssignmentList : AlarmSeverityAssignment [*] = empty list
modifies ASAP with
creates ASAP with
«interface»
AlarmSeverityAssignmentProfileControlService
(from ResourceTroubleManagement::Interfaces)
«mTOPOperation» assignAlarmSeverityAssignmentProfile (asapRef : ObjectName = NA, resourceRef : ObjectName = NA, layerRate : LayerRate = LR_Not_Applicable, additionalInfo : AttributeValuePair [*] = empty list)
«mTOPOperation» createAlarmSeverityAssignmentProfile (asapCreateData : AlarmSeverityAssignmentProfileModifyData = NA, targetOperationsSystem : ObjectName = "NA", newASAP : AlarmSeverityAssignmentProfile = NA)
«mTOPOperation» deassignAlarmSeverityAssignmentProfile (resourceRef : ObjectName = NA, layerRate : LayerRate = LR_Not_Applicable, additionalInfo : AttributeValuePair [*] = empty list)
«mTOPOperation» deleteAlarmSeverityAssignmentProfile (asapRef : ObjectName = NA, additionalInfo : AttributeValuePair [*] = empty list)
«mTOPOperation» modifyAlarmSeverityAssignmentProfile (asapRef : ObjectName = NA, asapModifyData : AlarmSeverityAssignmentProfileModifyData = NA, modifiedASAP : AlarmSeverityAssignmentProfile = NA)

Figure : Depiction of operation parameter classes (example)

3.7 Notification Guidelines

3.7.1 Overview

The IA specifies the information which has to be provided by a notification; i.e., the notification parameters.

This section explains how to model this information. The trigger events for the notifications are described in other sections; i.e., at the place where the artifact which causes the notification is defined (e.g., object creation notification in section 3.4.3, attribute value change notification in section 3.4.4.

3.7.2 Location

Notifications are contained in a separate "Notifications" package within the Data Model DDPs.

A notification class diagram is added in the "Class Diagrams" package.

3.7.3 Guidelines

Object classes are used to model the notifications in the DM model.

Each notification comprises:

- Notification Name
  - follow Upper CamelCase (UCC)
  - ends with the word "Notification" (e.g., EquipmentProtectionSwitchNotification)
  - must be unique across all notification names defined in the whole mTOP IA
  - special definitions are in the mTOP Naming Convention Guidelines in SD0-1_mTOPDictionary [**Ref**].
- Notification Documentation
  - contains a short summary of usage
refers to the appropriate BA requirement.
- Superclass(es)
  - inheritance and multiple inheritance may be used
  - two common lists of notification parameters are defined in the Framework DDP which shall be used whenever possible:

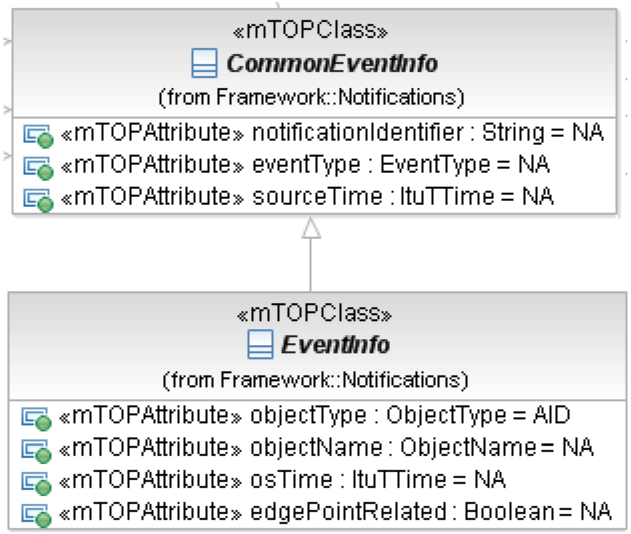

«mTOPClass»
CommonEventInfo
(from Framework::Notifications)
«mTOPAttribute» notificationIdentifier : String = NA
«mTOPAttribute» eventType : EventType = NA
«mTOPAttribute» sourceTime : ItuTTime = NA
«mTOPClass»
EventInfo
(from Framework::Notifications)
«mTOPAttribute» objectType : ObjectType = AID
«mTOPAttribute» objectName : ObjectName = NA
«mTOPAttribute» osTime : ItuTTime = NA
«mTOPAttribute» edgePointRelated : Boolean = NA

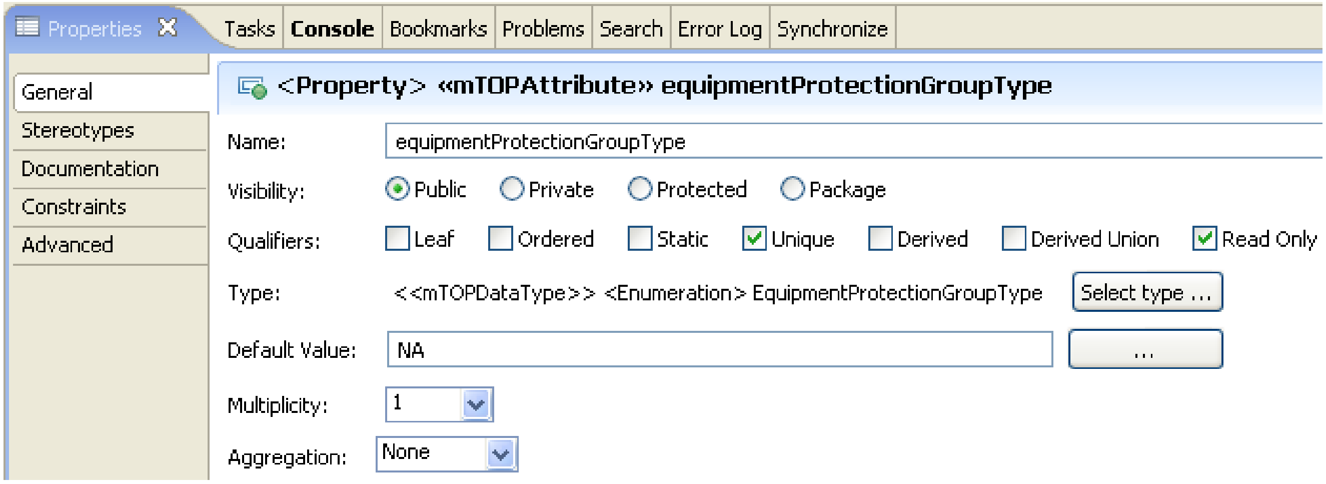
Figure : Common Notification parameters

- Abstract Object Class
    - indicates if the notification can be instantiated or is just used for inheritance (like the previous common parameter lists).
- Object Notifications (contained in mTOPClass stereotype)
    - this property can be ignored for Notifications
    - "objectCreationNotification" fixed to "NOT_APPLICABLE"
      "objectDeletionNotification" fixed to "NOT_APPLICABLE"
      "objectDiscoveryNotification" fixed to "NOT_APPLICABLE"; see also section 3.4.1.

Notification Parameter Guidelines

Attributes are used to model the parameters in the Notifications.

Each **parameter** (attribute**)** in a Notification comprises:



<Property> «mTOPAttribute» equipmentProtectionGroupType

Name: equipmentProtectionGroupType

Visibility: ● Public ○ Private ○ Protected ○ Package

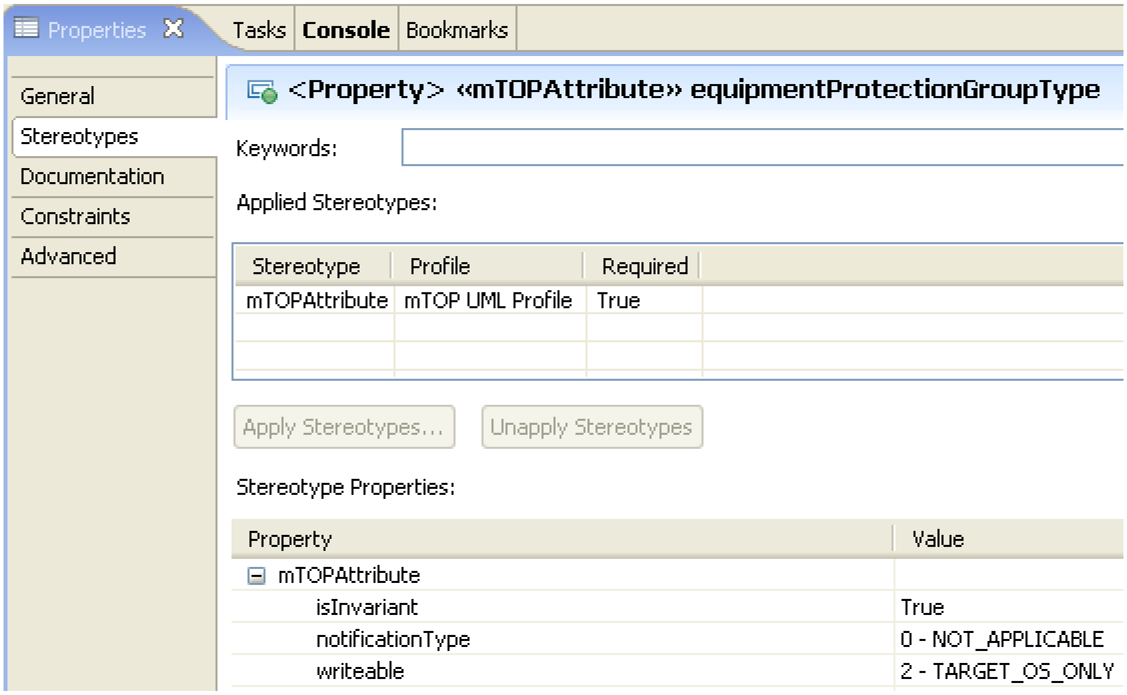
Qualifiers: □ Leaf □ Ordered □ Static ☑ Unique □ Derived □ Derived Union ☑ Read Only

Type: <<mTOPDataType>> <Enumeration> EquipmentProtectionGroupType

Default Value: NA

Multiplicity: 1

Aggregation: None



<Property> «mTOPAttribute» equipmentProtectionGroupType

Keywords:

Applied Stereotypes:

Stereotype	Profile	Required
mTOPAttribute	mTOP UML Profile	True

Stereotype Properties:

Property	Value
⊟ mTOPAttribute	
isInvariant	True
notificationType	0 - NOT_APPLICABLE
writeable	2 - TARGET_OS_ONLY

Figure : Notification Parameter properties (example)

- Parameter Name
  - follow Lower CamelCase (LCC)
  - must be unique across all notification parameter names defined in the whole mTOP IA
  - special definitions are in the mTOP Naming Convention Guidelines in SD0-1_mTOPDictionary [**Ref**]
  - Boolean typed parameters always start with ‘is’ (e.g., ‘isClearable’)
  - Enumeration typed attributes always end with “Kind” (e.g., ‘aggregationKind’).
- Parameter Documentation

    - contains a short summary of usage
    - refers to the specific BA requirement if defined.
- Visibility
    - this property can be ignored for Notification parameters
    - fixed to "Public".
- Qualifiers
    - this property can be ignored for Notification parameters
    - fixed to "Unique" and "Read Only".
- Type
    - refers to a basic or complex data type.
- Default Value
    - this property can be ignored for Notification parameters
    - fixed to "NA" because it has to reflect the current situation.
- Multiplicity
    - this property can be ignored for Notification parameters
    - fixed to "1".
- Aggregation
    - this property can be ignored for Notification parameters
    - fixed to "None".
- Invariant (contained in mTOPAttribute stereotype)
    - this property can be ignored for Notification parameters
    - fixed to "True"; see also section 3.4.1.
- Notification Type (contained in mTOPAttribute stereotype)
    - this property can be ignored for Notification parameters
    - fixed to "NOT_APPLICABLE"; see also section 3.4.1.
- Writeable (contained in mTOPAttribute stereotype)
    - this property can be ignored for Notification parameters
    - fixed to "TARGET_OS_ONLY"; see also section 3.4.1.

4 Changes History

The purpose of this section is to summarize the evolution/changes of mTOP IA guidelines / best-practices between product releases.

4.1 IA v3.4 (used in MTNM R3.5) to IA v3.5 (used in MTOSI R2.0)

The following list includes all the important new or updated features introduced into the development of MTOSI R2.0 IA:

- IA for MTOSI R2.0 is based on the new mTOP DDPs.
- Tool change from "Rational Rose" to "Rational Software Modeler".

5 References

5.1 External References

Ref OMG Unified Modeling Language (UML) – **http://www.uml.org/**

Ref IBM Rational Software Modeler (RSM) – **http://www-306.ibm.com/software/awdtools/modeler/swmodeler/index.html**

5.2 Internal References

Ref mTOP Wiki (GForge) – **http://gforge.tmforum.org/plugins/wiki/index.php?id=12&type=g**

Ref mTOP DDP Guidelines – **MTOSI_DDP_Maps.xls**

Ref mTOP Naming Convention Guidelines – **SD0-1_Dictionary**

Ref SD2-5 Communication Styles – **SD2-5_Communication_Styles**

Ref SD2-6 MTOSI Versioning and Extensibility supporting document – **SD2-6_VersioningAndExtensibility**

Appendix A: mTOP Parameters Supporting Documents

The latest versions of these SDs are available on GForge.

Supporting Document	Version	Comments
SD1-8_encodingX731M3100		
SD1-16_Layered Parameters		See Appendix B for all layered parameters applicable domains
SD1-17_LayerRates		
SD1-28_PerformanceParameters		
SD1-29_PGPParameters		
SD1-33_ProbableCauses		

Table : Supporting Documents containing Parameters

Appendix B: mTOP Layered Parameters Domains

Generally Applicable Parameters

Connection-oriented Technology Parameters

General parameters for Inverse Multiplexing

SDH/SONET/PDH specific parameters

ATM specific parameters

ATM parameters relevant only to ATM Network Interfaces supporting PNNI

IMA (Inverse Multiplexing for ATM) specific parameters

ATM specific Traffic Descriptor parameters

WDM specific parameters

Protection specific parameters

TCM related parameters for SDH/SONET

TCM related parameters for DWDM/OTN

Radio specific parameters

Frame Relay specific parameters

DSL (Digital Subscriber Line/Loop) specific parameters

Connectionless Technology Parameters

Ethernet specific Parameters

Spanning Tree Protocol (STP) related parameters

VLAN related parameters

Traffic conditioning parameters

Resilient Packed Ring (RPR) related parameters

Class of Service parameters

Ethernet MAU parameters

Link Aggregation Parameters

Appendix C: Requirements for additional properties in the mTOP IA UML model

This section contains the requirements for the additional properties grouped by Stereotypes.

The following tables contain the stereotypes that have already been defined for mTOP:

Stereotype: mTOPClass

name of property	type	allowed values	default value	associated to metaclass	mandatory/optional	comments
objectCreationNotification	enumeration	NO, YES, NOT_APPLICABLE	NOT_APPLICABLE	EClass	mandatory	
objectDeletionNotification	enumeration	NO, YES, NOT_APPLICABLE	NOT_APPLICABLE			
objectDiscoveryNotification	enumeration	NO, YES, NOT_APPLICABLE	NOT_APPLICABLE			

Table : Stereotype: mTOPClass

Stereotype: mTOPAttribute

name of property	type	allowed values	default value	associated to metaclass	mandatory /optional	comments
notificationType	enumeration	ATTRIBUTE_VALUE_CHANGE, STATE_CHANGE, NO_NOTIFICATION, NOT_APPLICABLE	NOT_APPLICABLE	EStructuralFeature	mandatory	
writeable	enumeration	NONE_OS REQUESTING_OS_ONLY, TARGET_OS_ONLY, TARGET_AND_REQUESTING_OS	REQUESTING_OS_ONLY			An indication of which OS can set the attribute.
isInvariant	boolean	true/false	false			

Table : Stereotype: mTOPAttribute

Stereotype: mTOPOperation

name of property	type	allowed values	default value	associated to metaclass	mandatory/optional	comments
messageExchangePattern	enumeration list	SRR, ARR, SIT, ABR, SFB, AFB, SN, AN	empty list	EOperation	mandatory	A description of the values can be found in SD2-5_Communication_Styles.pdf.
isOperationIdempotent	boolean	True/False	false			

Table : Stereotype: mTOPOperation

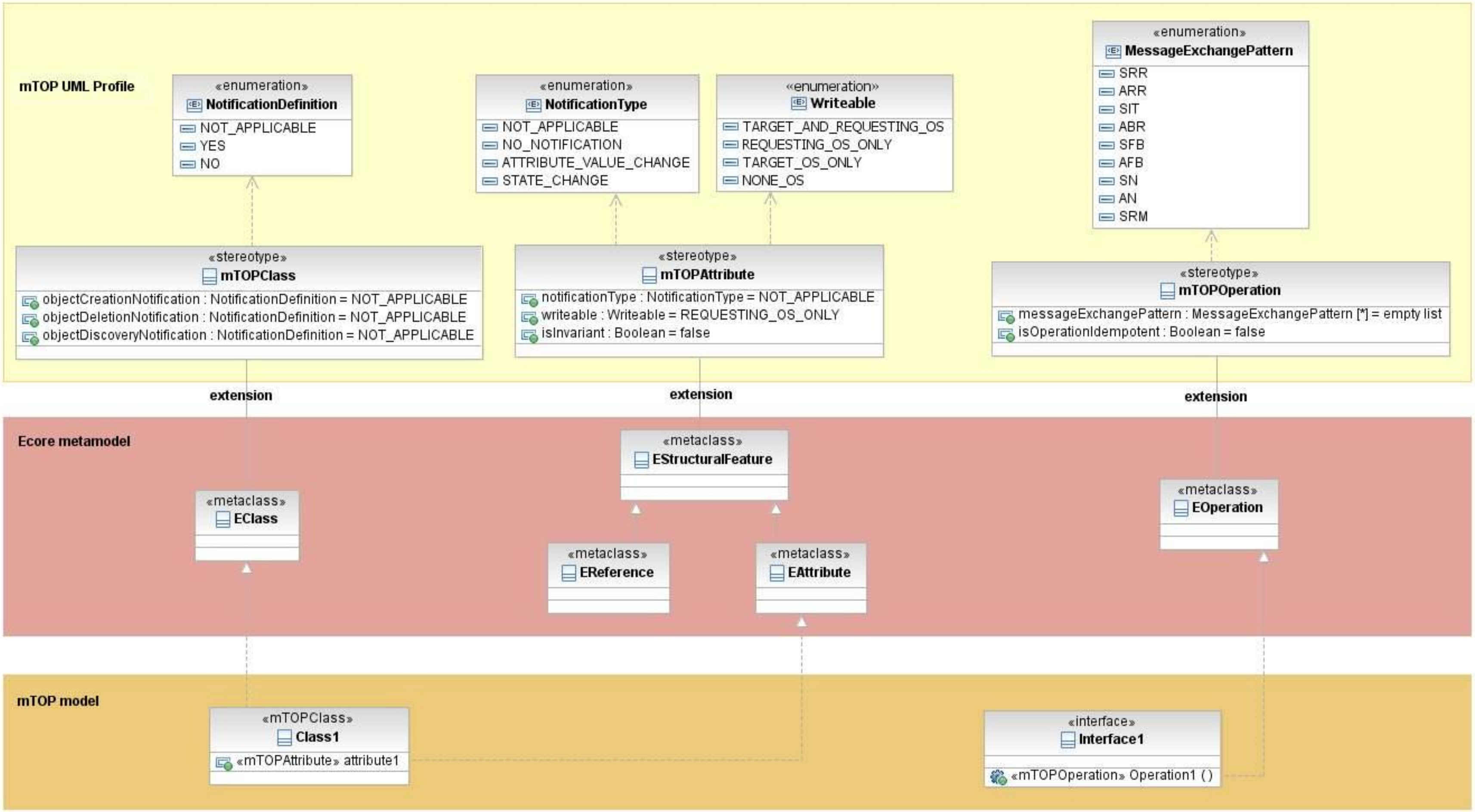


Figure : RSM model structure

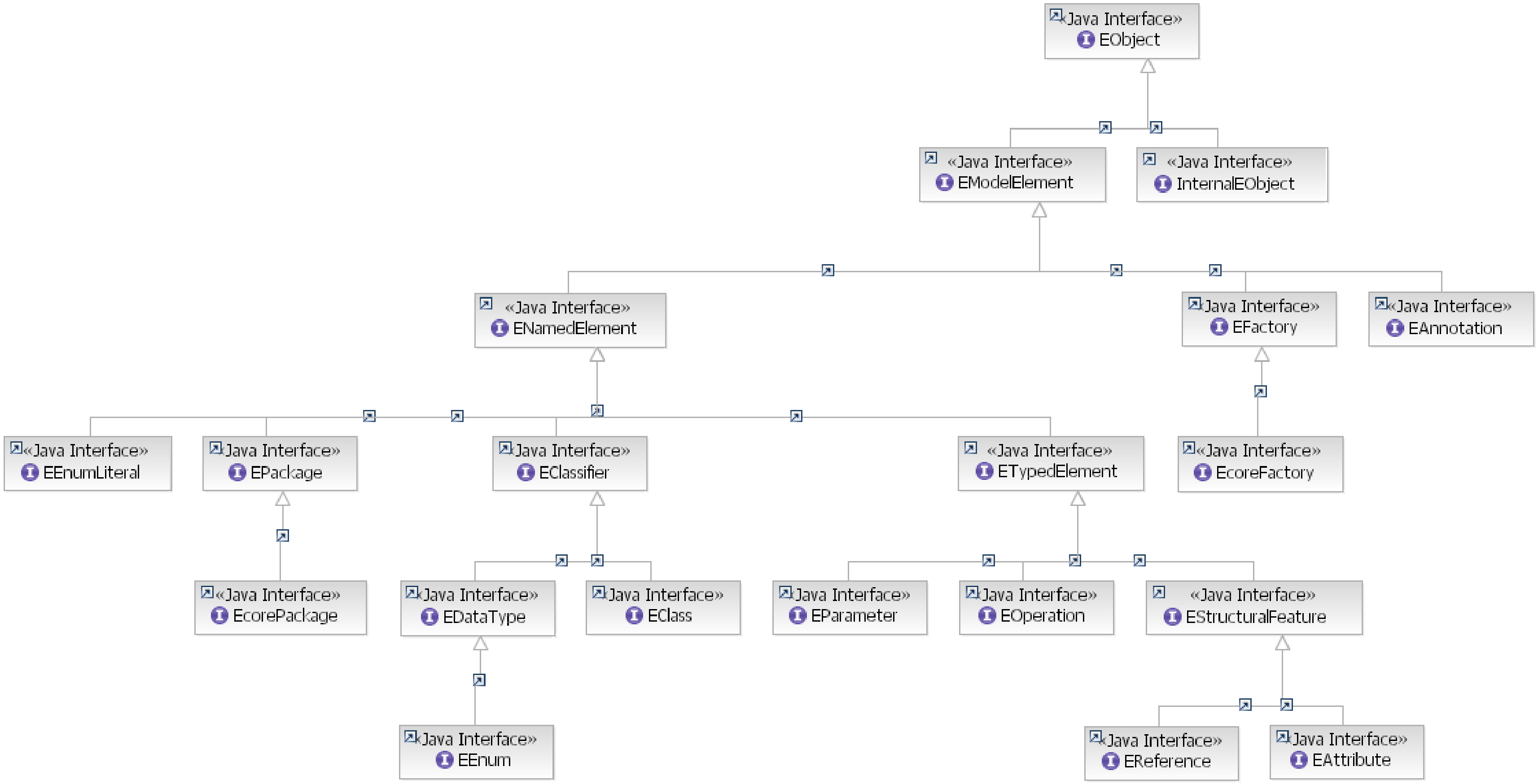

«Java Interface» EObject
«Java Interface» EModelElement
«Java Interface» InternalEObject
«Java Interface» ENamedElement
«Java Interface» EFactory
«Java Interface» EAnnotation
«Java Interface» EEnumLiteral
«Java Interface» EPackage
«Java Interface» EClassifier
«Java Interface» ETypedElement
«Java Interface» EcoreFactory
«Java Interface» EcorePackage
«Java Interface» EDataType
«Java Interface» EClass
«Java Interface» EParameter
«Java Interface» EOperation
«Java Interface» EStructuralFeature
«Java Interface» EEnum
«Java Interface» EReference
«Java Interface» EAttribute

Figure : The RSM Ecore metamodel

Appendix D: known issues with RSM/RSA

This section contains all known open issues with regard to the RSM/RSA tool.

Note: Finally this section will be (hopefully) deleted.

1. HTML Export is incomplete.

Appendix E: known issues with the IA Guidelines

This section contains all known open/postponed IA Guideline issues. This list has to be revisited after completion of MTOSI Rel. 2.0.

1. It may be necessary to enhance the list of agreed protocol neutral basic data types to accommodate harmonization with other models.
   Copy from Chameleon (Cisco's internal information model for management) section 9.2:

   Most of the datatypes in Chameleon are taken from the XML XSD specification [XSD] and have the same semantics. The differences are:

   - ***anyURI** : Can also be used to encode a reference, that could be an Entity Class reference. Use the <<TypeOfUri>> stereotype to specify a specific Entity Type.*
   - ***objectid** : An opaque identity.*
   - ***octetstring** : ASN Octetstring presentation.*
   - ***void** : Used on Operation return to specify that there is no return parameter.*

   These primitive types are defined in the MODEL-TYPES model library.

Name	***Description***
anyURI	*Used to encode a reference, that could be an Entity Class reference. Use TypeOfUri to specify a specific type.*
boolean	*As XSD*
byte	*As XSD*
datetime	*As XSD*
double	*As XSD*
duration	*As XSD*
float	*As XSD*
hexBinary	*As XSD*
int	*As XSD*
long	*As XSD*
objectId	*An opaque identity.*
octetstring	*As XSD*
short	*As XSD*
string	*As XSD*
unsignedByte	*As XSD*
unsignedInt	*As XSD*
unsignedLong	*As XSD*
unsignedShort	*As XSD*
void	*Used on Operation return to specify that there is no*

Name	*Description*
	return parameter.
blob	*Used to encode a loosely typed object. Typically in java this will be mapped to java.lang.Object*

2. Traceability Guidelines
Not required for MTOSI 2.0; will be added once they can be generated automatically.
The following traceability matrices shall be provided:

- mapping from object classes to requirements
- mapping from object class attributes to requirements
- mapping from object class operations to requirements
- mapping from object classes to use cases.

3. OM Service description structure
An OM Service description structure has to be defined. The "SOA Reference Model" developed in OASIS (http://wiki.oasis-open.org/soa-rm/TheArchitecture/ServiceView/ServiceDescription) shall be used as a basis.

However we should start with a simplified version.
Key attributes are:

- Business activities (operations)
- Message Exchange Patterns

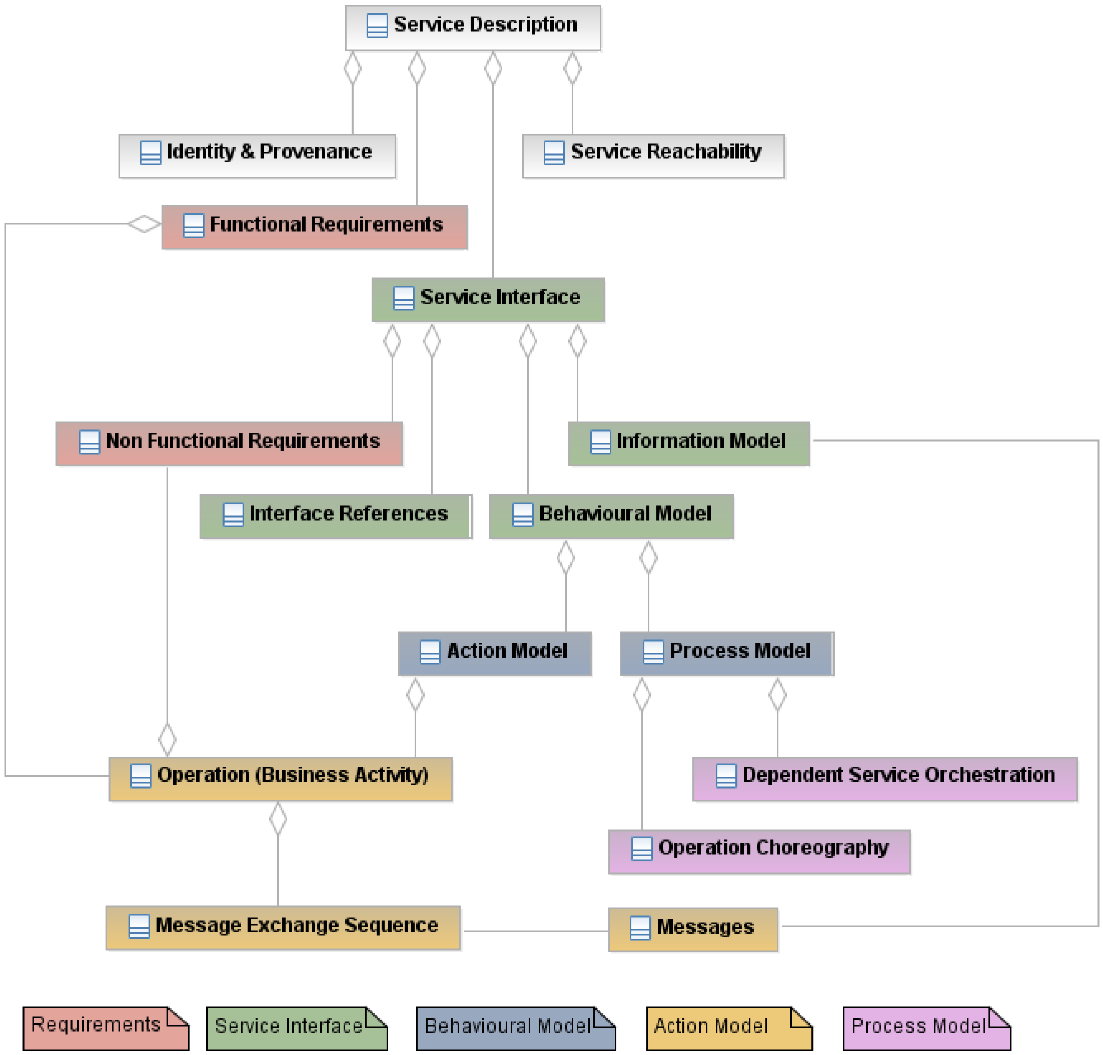


Figure - OM Service description structure

6 Administrative Appendix

6.1 Document History

Version Number	Date Modified	Modified by:	Description of changes
0.1	20 July 2007	Bernd Zeuner	Initial version based on mTOP_Guidelines_IA.doc in Methodology folder (20 July 2007; revision 585).
0.2	24 July 2007	Bernd Zeuner	Updated based on discussion during TAW Boston.
0.3	16 August 2007	Bernd Zeuner	Comments from Chris Hartley, Steve Jerman and Hector Trevino added and partly answered.
0.4	27 August 2007	Bernd Zeuner	Comments from Steve Fratini added and partly answered. Guidelines for Notifications added.
0.5	07 Sept. 2007	Bernd Zeuner	Comments from Cisco (Chris, Steve, Hector) added and partly answered. Appendix C: "Requirements for additional properties in the mTOP IA UML model" added.
0.6	14 Sept. 2007	Bernd Zeuner	Definitions from Chameleon (Cisco) added. Stereotype <<objectRefKey>> added. mTOP UML Profile updated: 1. Property "Idempotent" added to <<mTOPOperation>> 2. Properties "Invariant" and "WriteableByRequestingOS" added to <<mTOPAttribute>> 3. Property "AttributeOrderSignificant" removed from <<mTOPClass>>.
0.7 Revision 876	05 Oct. 2007	Bernd Zeuner	Converted to new SD template (Revision 850). Data Type extensibility deleted.

Version Number	Date Modified	Modified by:	Description of changes
0.8	12 Nov. 2007	Bernd Zeuner	mTOPAttribute stereotype updated: - writeable defined, - passedBy deleted. UnsignedInteger added as basic type. COMM_LOSS exception added. Attribute List relationship deleted.
0.9	20 Nov. 2007	Bernd Zeuner	ObjectName added as basic data type. Literal NONE_OS added to writeable attribute property. Postponed functions are summarized in the new Appendix E: known issues with the IA Guidelines.
0.10	11 Dez. 2007	Bernd Zeuner	BA reference requirements added. Remaining comments deleted as accepted during conference call on 29 November. Firewall/proxy configuration added for Subclipse. Common Object Provisioning interface added.
0.11	30 April. 2008	Bernd Zeuner	Guidelines for data type attributes added. "Choice" data type added. Product Identification Guidelines added.
0.12	08 May. 2008	Bernd Zeuner	Reference to SD0-1_mTOPGuidelines_NamingConventions.doc replaced by reference to SD0-1_mTOPDictionary.doc.

6.2 How to comment on this document

Comments and requests for information must be in written form and addressed to the contact identified below:

Bernd	Zeuner	Deutsche Telekom AG
Phone:	+49 6151 628 2086 (Phone)	
Fax:		
e-mail:	b.zeuner@telekom.de	

Please be specific, since your comments will be dealt with by the team evaluating numerous inputs and trying to produce a single text. Thus we appreciate significant specific input. We are looking for more input than wordsmith" items, however editing and structural help are greatly appreciated where better clarity is the result.