magicdraw™
Architecture Made Simple

xODP PLUGIN

user guide

version 1.0.2

Atenea Systems Modeling Group
July 2011

xODP PLUGIN FOR MAGICDRAW

1.- Introduction

The Reference Model of Open Distributed Processing (RM-ODP) is a joint effort by ISO/IEC and ITU-T, which provides a co-ordinating framework for the standardization of open distributed processing (ODP) which supports distribution, interworking, platform and technology independence, and portability, together with an Enterprise Architectural Framework for the specification of ODP systems (see www.rm-odp.net).

The fact that ODP languages are notation-independent motivated ISO/IEC and ITU-T to start a joint project in 2004 whose end result is the international standard "ITU-T Rec. X.906 | ISO/IEC 19793: Information technology - Open distributed processing - Use of UML for ODP system specifications", also known as **UML4ODP**.

UML4ODP defines a set of UML profiles, one for each viewpoint language and one to express the correspondences between viewpoints, and an approach for structuring them according to the RM-ODP principles. The purpose of UML4ODP is

(a) to allow ODP modelers to use the UML notation for expressing their ODP specifications in a standard graphical way;

(b) to allow UML modelers to use the RM-ODP concepts and mechanisms to structure their large UML system specifications according to a mature and standard proposal; and

(c) to allow UML tools to be used to process viewpoint specifications, thus facilitating the software design process and the enterprise architecture specification of large software systems.

The availability of tools for UML4ODP enables the integration of the RM-ODP enterprise architecture framework with other proposals, such as the OMG's Model-Driven Architecture (MDA) initiative, or with the Service-Oriented Architecture (SOA).

xODP plugin for MagicDraw enables the development of ODP systems specifications according to the guidelines provided by the ISO and ITU-T UML4ODP standard. It has been built using MagicDraw's OpenAPI, and it is fully integrated into MagicDraw modeling environment.

More precisely, this plugin increases MagicDraw's modeling capabilities by providing the following features:

- It serves to UML modelers as a guide to the use of the UML4ODP profile, since it customizes the MagicDraw environment and facilitates the identification and usage of the different ODP-specific elements.
- It allows UML modelers to specify all the different kinds of elements defined in the five ODP viewpoints, using the sets of UML Profiles for ODP systems

defined in ISO/IEC 19793 | ITU-T X.906. This includes the ability to create and design models of ODP systems using the UML diagrams defined in UML4ODP:

- o the ODP_SystemSpec model;
- o the Enterprise Viewpoint model, and its related diagrams (e.g., behaviour);
- o the Information Viewpoint model, and its related diagrams;
- o the Computational Viewpoint model, and its related diagrams (e.g., sequence diagrams);
- o the Engineering Viewpoint model, and its related diagrams;
- o the Technology Viewpoint model, and its related diagrams; and
- o the Correspondence model.

- It enables the validation of all user viewpoint models, by checking that they fulfill all constraints prescribed by UML4ODP and RM-ODP. Errors or warnings found in the models during the validation process are clearly reported to the user, showing the precise elements where the conflicts were found. Endpoints at correspondence links can also be validated.

2. Installation

1. Download the most recent version (as ZIP file) from `http://www.jrromero.net/tools.html`.
2. Exit the MagicDraw UML application, if currently running.
3. Extract the contents of the ZIP file. Check if you have extracted three new folders:
   a. `Perspective`
   b. `Profiles`
   c. `Plugins`
4. Copy the UML4ODP perspective file, '`RM-ODP.umd`', located in '**Perspective**' folder to:

`<MagicDraw installation directory>\data\defaults\data\perspectives`

5. Copy the UML4ODP profile file, '`ODP_Profile.mdzip`', located in '**Profiles**' folder to:

`<MagicDraw installation directory>\profiles`

6. Copy the UML4ODP plugin directory, '`es.uco.mdplugin.RM-ODP`', located in '**Plugins**' folder to:

`<MagicDraw installation directory>\plugins`

7. Restart the MagicDraw UML application.

You can also use the Resource Manager (> **Help** > **Resource Manager**) to install the UML4ODP Plugin. There are two ways to do this: Local or Remote installation.

a) <u>Local Installation</u>

1. Download the most recent standalone installation version (as a ZIP file) from `http://www.jrromero.net/tools.html`
2. From the **Help** main menu, select **Resource/Plugin Manager**. The **Resource/Plugin Manager** appears and prompts to check for available updates and new resources. Click **Import**.
3. Select the xODP plugin ZIP file from your local disk. Click **Open.**
4. Click **Yes** to proceed with the installation.
5. Restart the MagicDraw UML application.

b) <u>Remote Installation</u>

1. From the **Help** main menu, select **Resource/Plugin Manager**. The **Resource/Plugin Manager** appears and prompts to check for available updates and new resources. Click **Check**.

NOTE: Specify HTTP Proxy Settings for connection to start MagicDraw UML updates and resources.

8

1. Select the **xODP Plugin** check box and click **Download/Install**.
2. Restart the MagicDraw UML application.

3. The RM-ODP Perspective

The MagicDraw features for UML4ODP support are accessible when this perspective is selected.

To switch from any other perspective to the RM-ODP perspective:
1. From the **Options** main menu, select **Perspective -> Perspective**.
2. Select **RM-ODP** from the **Perspectives** dialog box and click **Apply.**
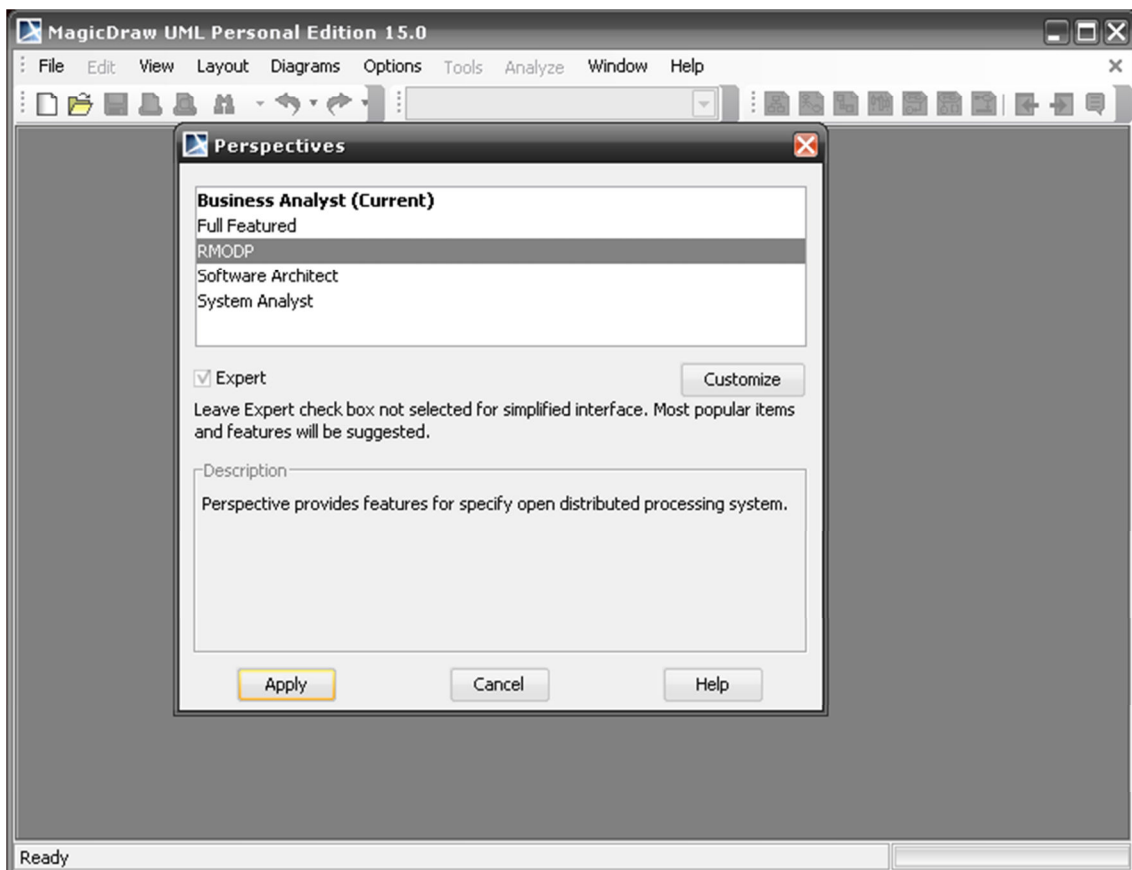
Figure 1 - Perspectives dialog

For more information about working with perspectives, see *Perspectives selection and customization* in the "Getting Started" section of the MagicDraw User Manual.

WORKING WITH RM-ODP PROJECTS

4. Working with RM-ODP Projects

To create a new workspace for a blank RM-ODP project
1. Do any of the following.
   - From the **File** menu, select **New Project**.
   - On the main toolbar, click the **New Project** button.
   - Press shortcut key **CTRL+N**.
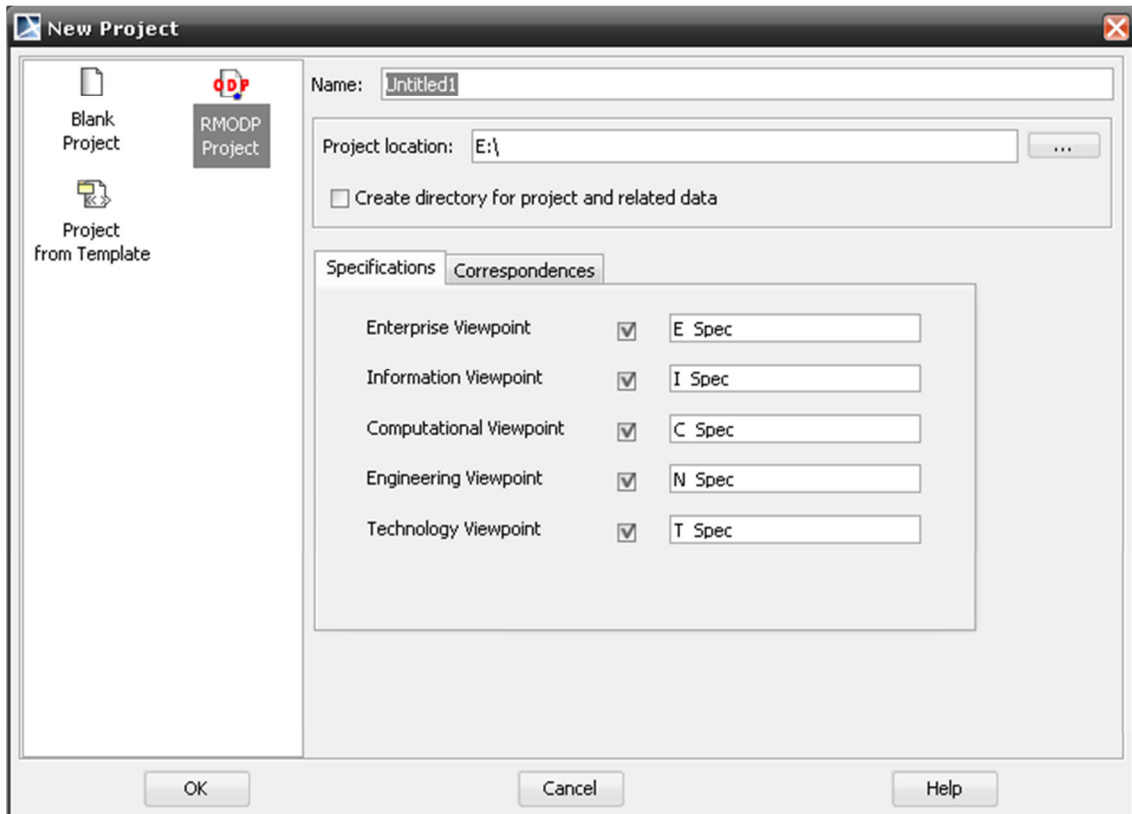
Figure 2 - New Project dialog-Specifications

2. The **New Project** dialog opens. Select the **RM-ODP Project** icon.
3. Enter the file name in the **Name** text box.
4. Click the "**...**" button to locate where to store the newly created project in your computer.
5. From the tabbed panel select the **Specifications** and **Correspondences** models that you want to create initially. Enter the name for these elements in the proper text box. (See figure 3).
6. Click **OK**.
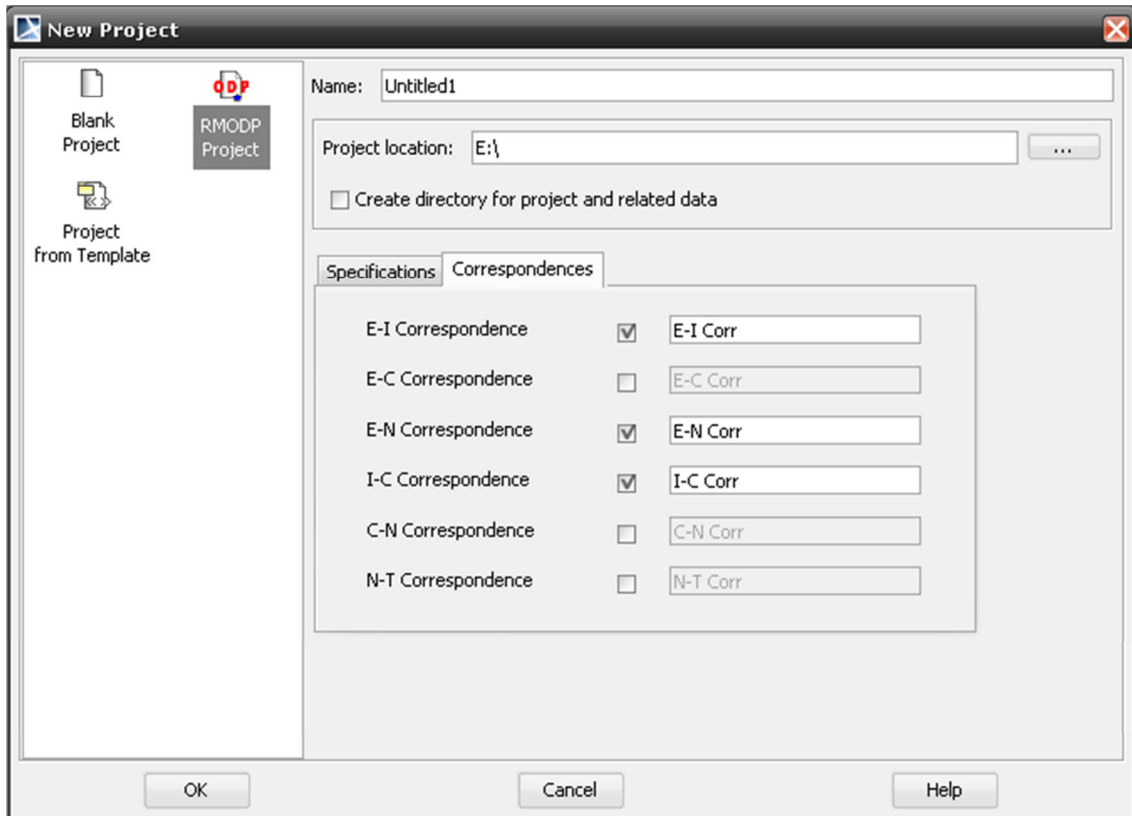
Figure 3 - The New Project dialog-Correspondences

For more information about working with projects, see the "Working with Projects" section in the *MagicDraw User Manual[1]*.

---

[1] Visit http://www.magicdraw.com for further information.

5. RM-ODP Diagrams

The RM-ODP plugin supports the following RM-ODP diagrams:

- o ODP_SystemSpec diagram [UML: Class diagram]
- o Enterprise Viewpoint (EV):
  - ▪ EV diagram [UML: Class diagram]
  - ▪ EV_Process (activity) [UML: Activity diagram]
  - ▪ EV_Behaviour (behaviour) [UML: State machine diagram]
- o Information Viewpoint (IV):
  - ▪ IV diagram [UML: Class diagram]
  - ▪ IV_DinamicSchema (behaviour) [UML: State machine diagram]
- o Computational Viewpoint:
  - ▪ CV diagram [UML: Class diagram]
  - ▪ CV_Sequence (sequence) [UML: Interaction diagram]
- o Engineering Viewpoint:
  - ▪ NV diagram [UML: Class diagram]
 - ▪ NV_Deactivation (activity) [UML: Activity diagram]
  - ▪ NV_Checkpointing (activity) [UML: Activity diagram]
  - ▪ NV_Reactivation (activity) [UML: Activity diagram]
  - ▪ NV_Deletion (activity) [UML: Activity diagram]
  - ▪ NV_Cloning (activity) [UML: Activity diagram]
  - ▪ NV_Recovery (activity) [UML: Activity diagram]
  - ▪ NV_Instantiation (activity) [UML: Activity diagram]
  - ▪ NV_Migration (activity) [UML: Activity diagram]
- o Technology Viewpoint diagram, and its related diagrams:
  - ▪ TV diagram [UML: Class diagram]
  - ▪ TV_Implementation (activity) [UML: Activity diagram]
- o Correspondence diagram [UML: Class diagram].

For more information about working with diagrams, see the "Working with Diagrams" section in the *MagicDraw User Manual*.

6. RM-ODP Interface

The RM-ODP plugin adds several interface elements in MagicDraw to gain easy access to these elements and diagrams.

6.1 The main RM-ODP toolbar

Figure 4 - Main RM-ODP toolbar

From **RM-ODP toolbar** we can create a new diagram by clicking on the **proper icon**.

- Open a new diagram *'System Spec Diagram'*.

- Open a new diagram *'Enterprise Viewpoint'*.

- Open a new diagram *'Information Viewpoint'*.

- Open a new diagram *'Computational Viewpoint'*.

- Open a new diagram *'Engineering Viewpoint'*.

- Open a new diagram *'Technology Viewpoint'*.

- Open a new diagram *'Correspondence'*.

6.2 The RM-ODP main menu

From the RM-ODP main menu we can perform two differents actions:

1. **Create a new diagram**. Click on the proper item to do this.

2. **Validate the complete ODP specification model**. (See Section 7: Validation). Click on '**Validate ODP system**' to perform it.
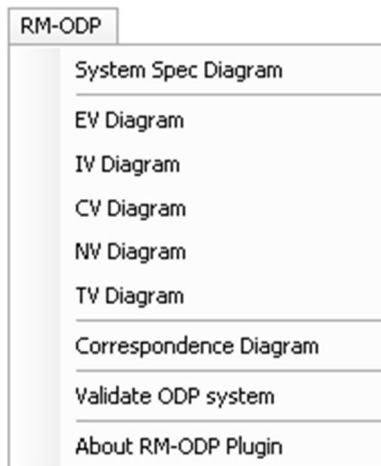
Figure 5 – RM-ODP main menu

6.3 Diagrams menu

From the **Diagrams menu** we can create a **new RM-ODP diagram** by clicking on the appropriate icon**.**
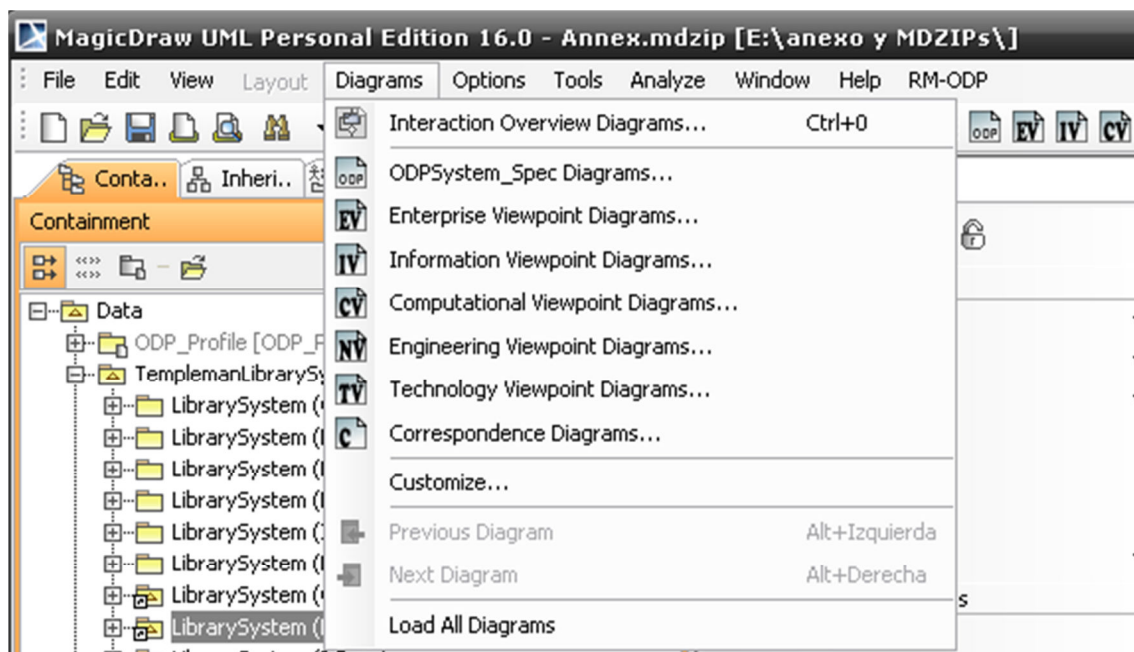
Figure 6 – Diagrams menu

6.4 The Browser context menu

The xODP plugin adds a new action in the browser context menu called 'Validate viewpoint specification'. (See section 7.1).

6.5 ODPSystem_Spec diagram Interface

From the ODP_SystemSpec toolbar you can create any of the following elements:

- Enterprise_Spec
- Information_Spec
- Computational_Spec
- Engineering_Spec

- Technology_Spec
- CorrespondenceSpecification
- CorrespondingSpecification
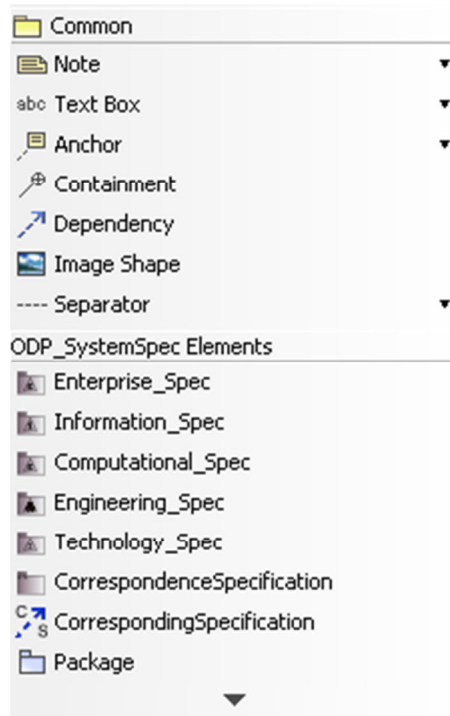
Figure 7 – ODPSystem_Spec toolbar

6.6 Enterprise Viewpoint diagram Interface

From the **EV toolbar** you can create any of the following elements:

- EV_ODPSystem
- EV_Community
- EV_Object
- EV_TypeObject
- EV_TemplateObject
- EV_CommunityObject
- EV_Objective

- EV_CommunityContract
- EV_Process
- EV_Interaction [signal]
- EV_Interaction [class]
- EV_Role
- EV_Artefact
- EV_PolicyEnvelope

- EV_PolicyValue
- EV_Party
- EV_ObjectiveOf
- EV_FulfilsRole
- EV_InteractionResponder
- EV_InteractionInitiator
- EV_Delegation

- EV_Accountable
- EV_ArtefactRole
- EV_ControllingBehaviour
- EV_ArtefactReference
- EV_RefinesAsCommunity
- EV_AffectedBehaviour
- EV_CommunityBehaviour
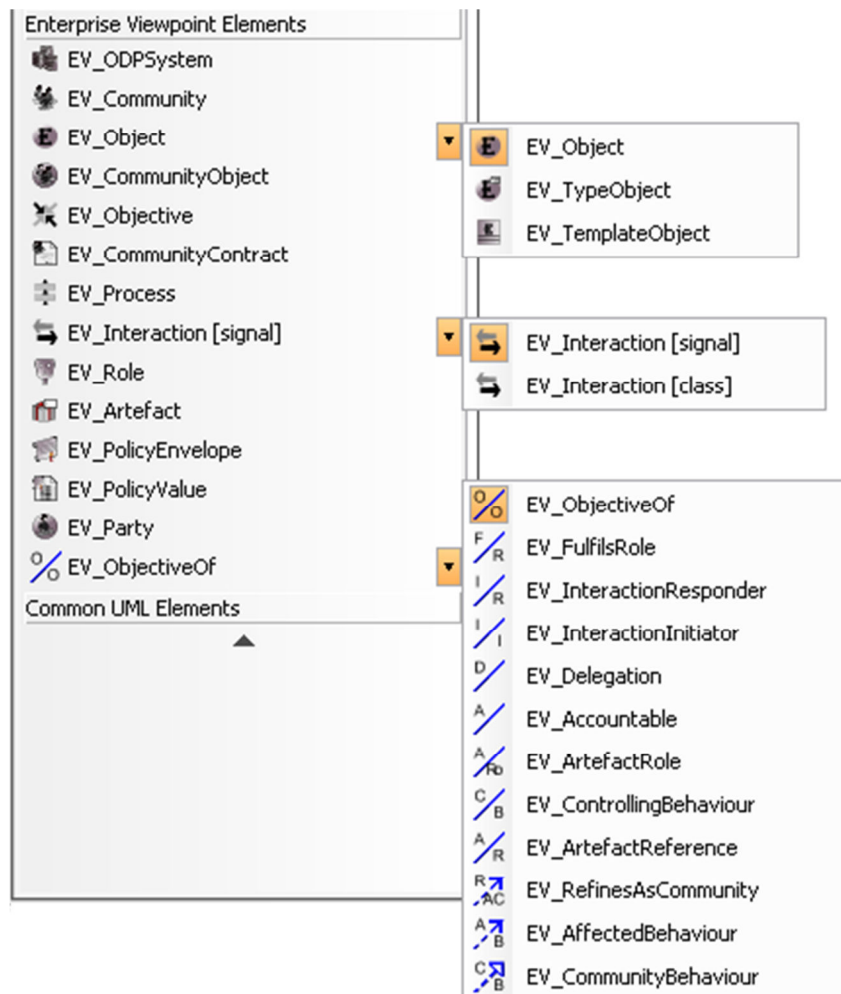
Figure 8 - ODPSystem_Spec toolbar

From the context toolbar you can also select a relationship to draw.
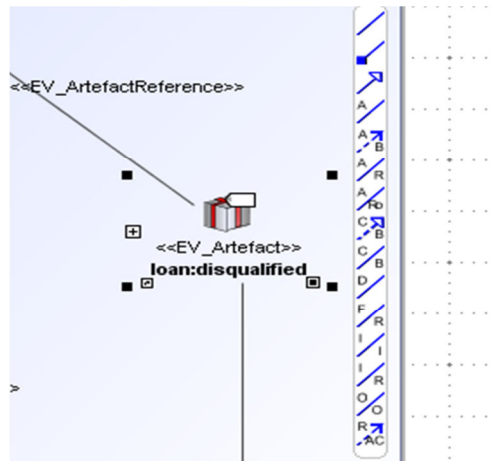
Figure 9 - ODPSystem_Spec context toolbar

6.7 Information Viewpoint diagram Interface

From the **IV toolbar** you can create any of the following elements:

- IV_Object
- IV_TypeObject
- IV_TemplateObject

- IV_Action
- IV_InvariantSchema
- IV_StaticSchema
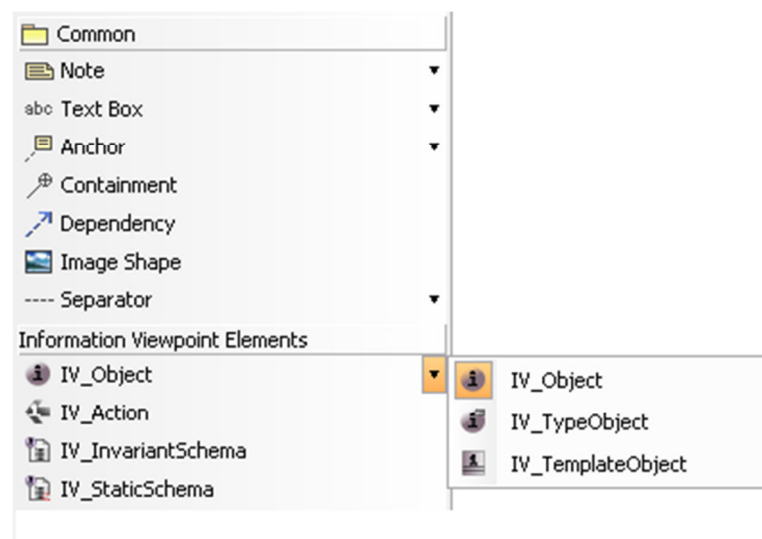
Figure 10 – IV toolbar

From context toolbar you can select a relationship to draw.
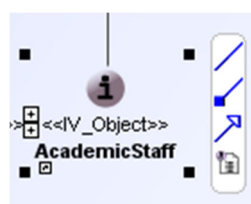
Figure 11 – IV context toolbar

6.8 Computational Viewpoint diagram Interface

From the **CV toolbar** you can create any of the following elements:

- CV_Object
- CV_TypeObject
- CV_TemplateObject
- CV_BindingObject
- CV_SignalInterface
- CV_SignalInterfaceSignature
- CV_OperationInterface

- CV_OperationInterfaceSignature
- CV_StreamInterface
- CV_StreamInterfaceSignature
- CV_FlowSignature
- CV_PrimitiveBinding [Dependency]
- CV_PrimitiveBinding [Connector]

Figure 12 – CV toolbar

From context toolbar you can select a relationship to draw.
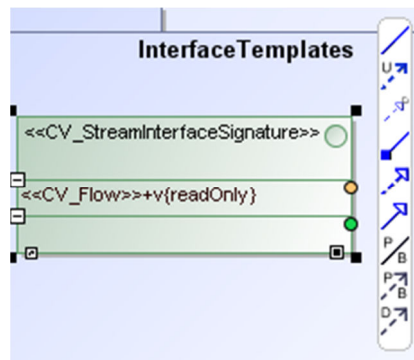
Figure 13 – CV context toolbar

6.9 Engineering Viewpoint diagram Interface

From **NV toolbar** you can create any of the following elements:

- NV_Object
- NV_BEO
- NV_TypeObject
- NV_TemplateObject
- NV_Node
- NV_Nucleus
- NV_Capsule
- NV_CapsuleManager
- NV_Cluster
- NV_ClusterManager
- NV_CommunicationAuthority
- NV_Channel
- NV_Stub
- NV_Binder
- NV_Interceptor
- NV_ProtocolObject
- NV_CommunicationDomain
- NV_CommunicationInterface
- NV_InterfaceReferenceManagementDomain
- NV_InterfaceReference
- NV_Checkpoint
- NV_ClusterCheckpoint
- NV_Checkpointing [UML Activity]
- NV_Deactivation [UML Activity]
- NV_Cloning [UML Activity]
- NV_Recovery [UML Activity]
- NV_Reactivation [UML Activity]

- NV_Migration [UML Activity]
- NV_Instantiation [UML Activity]
- NV_Deletion [UML Activity]
- NV_ObjectManagement
- NV_NodeManagement
- NV_ClusterManagement
- NV_CapsuleManagement
- NV_EventNotification
- NV_CheckpointingAndRecovery
- NV_DeactivationAndReactivation
- NV_Group
- NV_Replication
- NV_Migration
- NV_InterfaceReferenceTracking
- NV_ACIDTransaction
- NV_Transaction
- NV_Storage
- NV_InformationOrganization
- NV_Relocation
- NV_TypeRepository
- NV_Trading
- NV_AccessControl
- NV_SecurityAudit
- NV_Authentication
- NV_Integrity
- NV_Confidentiality
- NV_NonRepudiation
- NV_KeyManagement
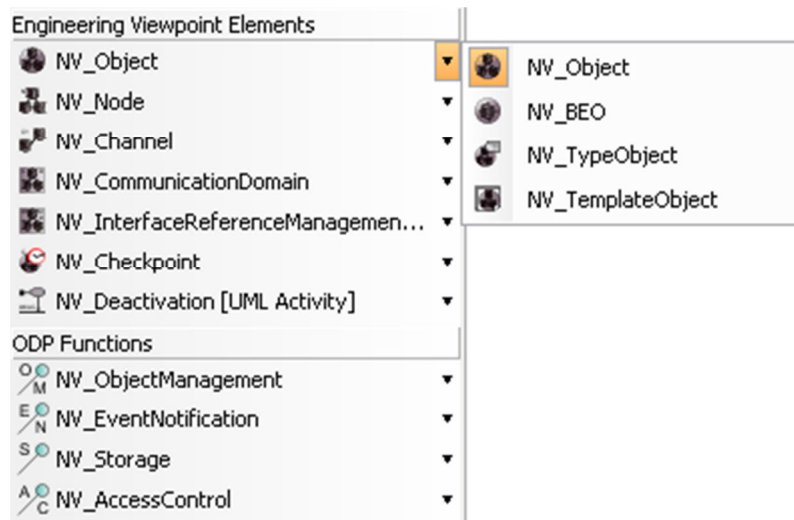
Figure 14 –The NV toolbar

From the context toolbar you can select a relationship to draw.
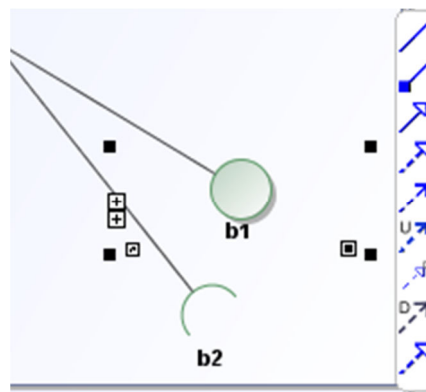
Figure 15 – NV context toolbar

6.10 Technology Viewpoint diagram Interface

From **TV toolbar** you can create any of the following elements:

- TV_Object [Node]
- TV_Object [Artifact]
- TV_TypeObject [Node]
- TV_TypeObject [Artifact]
- TV_TemplateObject [Node]
- TV_TemplateObject [Artifact]
- TV_ImplementableStandard
- TV_Implementation

- TV_IXIT
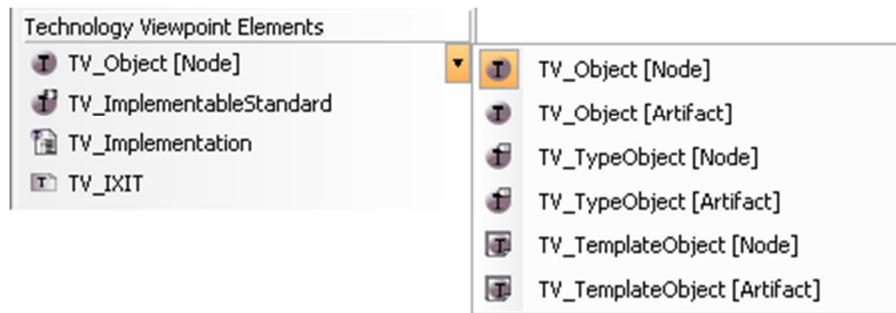
Figure 16 – The TV toolbar

From the context toolbar you can select a relationship to draw.

Figure 17 – TV context toolbar

6.11 Correspondence diagram Interface

From the **Correspondence toolbar** you can create any of the following elements:

- CorrespondenceLink

- Package

- Composition
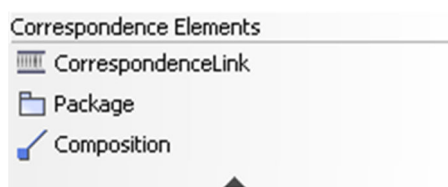
Figure 18 – Correspondence toolbar

6.12 Report window[2]

The results of the validation process are shown in a **report window** (Fig. 19). This report window shows the following information.
- The model on which the validation has been performed [Top]. In this case, the 'Enterprise Viewpoint'.
- Number of warnings and errors found [Bottom] (see section 7.3).
- The 'Save as...' button [Right]. This action saves the report information in a file. This information can be filtered using the 'Items' panel, by selecting-unselecting the appropriate checkboxes.

---

[2] This feature is only available in the Full Edition.

- The information about the report is shown in a tabular way in the central part of the window. Rows indicate the individual constraints checked, and the results for that constraint. The information shown in each column contains four fields:
  - o **Std.:** Section of the ISO/IEC standard | ITU-T Recommendation (i.e., the UML4ODP document) that describes the validation constraint.
  - o **Description:** a brief description of the constraint.
  - o **Result:** the result obtained after the validation process for the constraint.
  - o **Elements:** Model elements where the warnings/conflicts/errors were found.
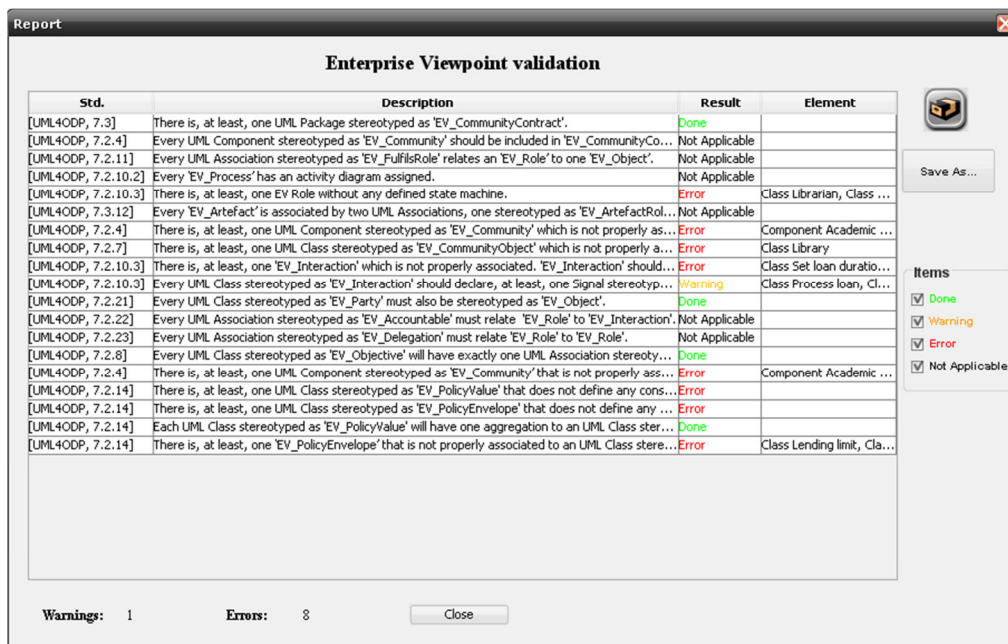
Figure 19 – Result screen after the "Validate viewpoint specification" action

To see further details about one particular validation constraint, click on the corresponding row. A new window will open with the detailed information about this individual constraint (see Fig. 20 for an example).
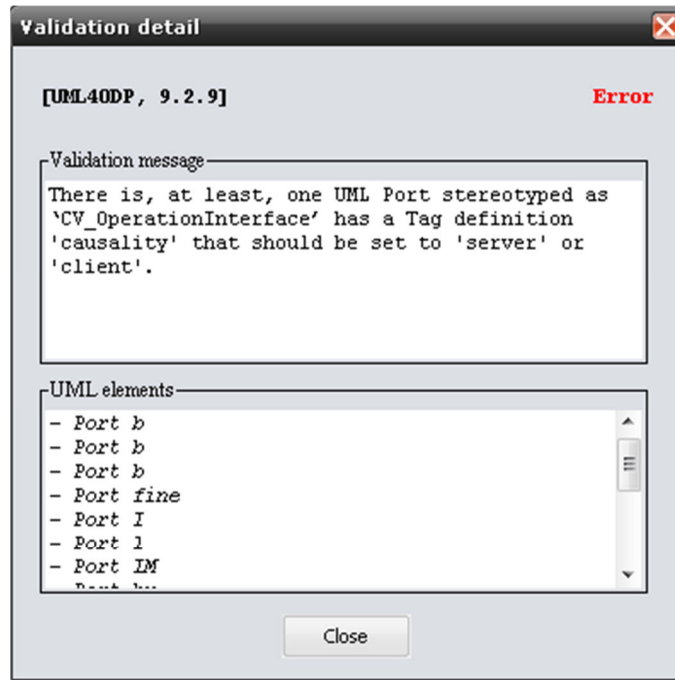
Figure 20 – Detailed validation window

7. Validation

The xODP plugin provides a validation suite[3] for validating UML4ODP models. It allows UML modellers to check that their models are conformant to the UML4ODP standard (both the individual viewpoint models and the model with the complete ODP specification). The validation process generates reports with information about the checks performed and the results of these checks (in terms of warnings and errors found).

7.1 Individual model validation

As shown in Fig. 21, there is a new action called 'Validate viewpoint specification'. This action will be executed only if the selected element is a viewpoint specification, a correspondence specification, or the complete system specification, i.e., it is a model stereotyped with any of the following :

- «ODP_SystemSpec»
- «EV_Spec»
- «IV_Spec»
- «CV_Spec»
- «NV_Spec»
- «TV_Spec»
- «CorrespondenceSpecification»
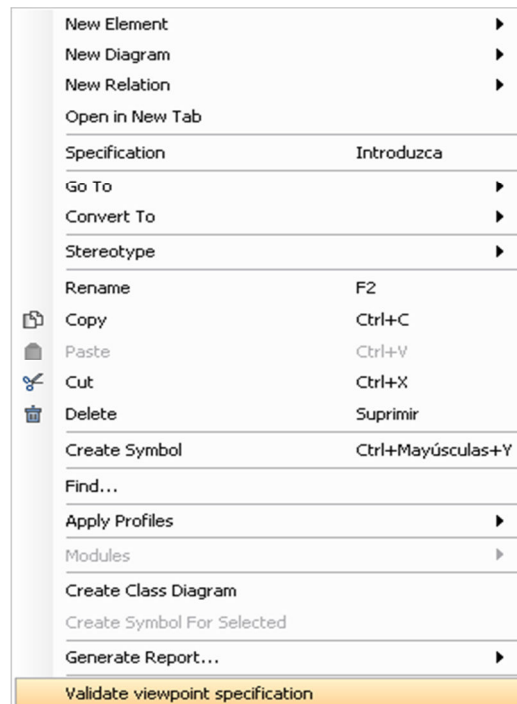
Figure 21 – Validate viewpoint specification action

---

[3] Only available in full edition.

To validate one individual viewpoint or diagram, right-click on the appropriate element. For example, if you want to validate the Enterprise viewpoint, you should display the context menu of the UML model stereotyped «EV_Spec» and then select the *Validate viewpoint specification* action. Then the validation process is performed on the elements of that model, and the results are shown using the Report window described above in Section 6.12.

7.2 Complete ODP specification validation

For validating the complete ODP specification model you have to click on 'Validate ODP System' action as explained in section 6.2. This process validates each individual viewpoint model, each individual correspondence model defined in the system specifications, and that the overall consistency of this set of models.

7.3 Validation constraints

The following tables describe in detail all the constraints that are checked by the validation process. They are organized by viewpoint or diagram. Each row describes one constraint. For each constraint the tables shows the following information:

- **ID:** identifier of the constraint.
- **Reference:** the section in the UML4ODP document where this constraint is specified.
- **Description:** a brief description of the constraint.
- **Case:** different cases that are considered in the validation process.
- **Result:** the results obtained. A result can be any of the following:
  - o **Ok:** Constraint is satisfactorily fulfilled.
  - o **Warning:** Constraint is not properly satisfied, but some minor improvement is required.
  - o **Not Applicable (N/A):** No elements were found in the model to validate the constraint.
  - o **Error:** Constraint is not satisfied and major changes are required.
- **Message:** the message displayed in each case.

7.3.1 ODP diagram

ID	Reference	Description	Case	Result	Message
ODP001	[UML4ODP, 6.6]	There is only one model stereoyped as «Enterprise_Spec».	N = 1	Ok	[DESC]⁴
			N = 0	Warning	No «Enterprise_Spec» model found in the specification. There should be one.
			N > 1	Error	More than one «Enterprise_Spec» model found in the specification. There should be just one Enterprise specification of the system.
ODP002	[UML4ODP, 6.6]	There is only one model stereoyped as «Information_Spec».	N = 1	Ok	[DESC]
			N = 0	Warning	No «Information_Spec» model found in the specification. There should be one Information specification of the system.
			N > 1	Error	More than one «Information_Spec» model found in the specification. There should be just one Information specification of the system.
ODP003	[UML4ODP, 6.6]	There is only one model stereoyped as «Computational_Spec».	N = 1	Ok	[DESC]
			N = 0	Warning	No «Computational_Spec» model found in the specification. There should be one Computational specification of the system.
			N > 1	Error	More than one «Computational_Spec» model found in the specification. There should be just one Computational specification of the system.
ODP004	[UML4ODP, 6.6]	There is only one model stereoyped as «Engineering_Spec».	N = 1	Ok	[DESC]
			N = 0	Warning	No «Engineering_Spec» model found in the specification. There should be one Engineering specification of the system.
			N > 1	Error	More than one «Engineering_Spec» model found in the specification. There should be just one Engineering specification of the system.
ODP005	[UML4ODP, 6.6]	There is only one model stereoyped as «Technology_Spec».	N = 1	Ok	[DESC]
			N = 0	Warning	No «Technology_Spec» model found in the specification. There should be one Technology specification of the system.
			N > 1	Error	More than one «Technology_Spec» model found in the specification. There should be just one Technology specification of the system.
ODP006	[UML4ODP, 12.1]	There exist, at least, one UML Package stereotyped «CorrespondenceSpecification» that defines the correspondence between EV and IV elements.	N > 0	Ok	[DESC]
			N = 0	Warning	The ODP system specification recommends a minimum set of correspondence specifications between EV and IV elements.
ODP007	[UML4ODP, 12.1]	There exist, at least, one UML	N > 0	Ok	[DESC]

---

⁴ [DESC] refers to the text comprised by the "Description" column.

ID	Reference	Constraint	Condition	Severity	Message
		Package «CorrespondenceSpecification» that defines the correspondence between EV and CV elements.	N = 0	Warning	**The ODP system specification recommends a minimum set of correspondence specifications between EV and CV elements.**
				N/A	
				Error	[DESC]
ODP008	[UML4ODP, 12.1]	There exist, at least, one UML Package «CorrespondenceSpecification» that defines the correspondence between EV and NV elements.	N > 0	Ok	[DESC]
			N = 0	Warning	**The ODP system specification recommends a minimum set of correspondence specifications between EV and NV elements.**
ODP009	[UML4ODP, 12.1]	There exist, at least, one UML Package «CorrespondenceSpecification» that defines the correspondence between CV and IV elements.	N > 0	Ok	[DESC]
			N = 0	Warning	**The ODP system specification recommends a minimum set of correspondence specifications between CV and IV elements.**
ODP010	[UML4ODP, 12.1]	There exist, at least, one UML Package «CorrespondenceSpecification» that defines the correspondence between CV and NV elements.	N > 0	Ok	[DESC]
			N = 0	Warning	**The ODP system specification recommends a minimum set of correspondence specifications between CV and NV elements.**
ODP011	[UML4ODP, 12.1]	There exist, at least, one UML Package «CorrespondenceSpecification» that defines the correspondence between NV and TV elements.	N > 0	Ok	[DESC]
			N = 0	Warning	**The ODP system specification recommends a minimum set of correspondence specifications between NV and TV elements.**
ODP012	[UML4ODP, 12.2.1]	Every UML Package stereotyped «CorrespondenceSpecification» has exactly two UML Usages stereotyped «CorrespondingSpecification» to UML Models that represents the specification of the viewpoints. Both models must be different.	(N=2) and (Model_1 ≠ Model_2)	Ok	[DESC]
			No UML Package stereotyped «CorrespondenceSpecificatio n» found.	N/A	[DESC]
			(N≠2) or [(N=2) and (Model_1 = Model_2)]	Error	**There is, at least, one UML Package stereotyped «CorrespondenceSpecification» which is not properly associated. A «CorrespondenceSpecification» should have exactly two UML Usage dependencies stereotyped «CorrespondingSpecification». Both models should be different.**

Table 1 – ODP Constraints

7.3.2 Enterprise Viewpoint

ID	Reference	Description	Case	Result	Message
EV001	[UML4ODP, 7.3]	There is, at least, one UML Package stereotyped «EV_CommunityContract».	N>0	Ok	[DESC]
			N=0	Error	No «EV_CommunityContract» found in the specification. You need to specify a UML Package stereotyped «EV_CommunityContract».
EV002	[UML4ODP, 7.2.4]	Every UML Component stereotyped «EV_Community» should be included in «EV_CommunityContract».	OK	Ok	[DESC]
			«EV_Community» not found.	N/A	[DESC]
			At least one UML Component stereotyped «EV_Community» is not owned by a UML Package stereotyped «EV_CommunityContract»	Error	There is, at least, one UML Component stereotyped «EV_Community» which is not owned by a UML Package stereotyped «EV_CommunityContract».
EV003	[UML4ODP, 7.2.11]	Every UML Association stereotyped «EV_FulfilsRole» relates an «EV_Role» to one «EV_Object».	OK	Ok	[DESC]
			«EV_FulfilsRole» not found.	N/A	[DESC]
			There is, at least, one related element which is not «EV_Role» or «EV_Object», or both elements have the same stereotype.	Error	An «EV_FulfilsRole» association should relate an «EV_Role» to one «EV_Object».
EV004	[UML4ODP, 7.2.10.2]	Every «EV_Process» has an activity diagram assigned.	OK	Ok	[DESC]
			No «EV_Process» found.	N/A	[DESC]
			There is, at least, one «EV_Process» which has not any activity diagram assigned.	Error	There is, at least, one EV process without any activity diagram assigned.
EV005	[UML4ODP, 7.2.10.3]	Every «EV_Role» defines, at least, one UML StateMachine stereotyped «EV_Behaviour».	OK	Ok	[DESC]
			«EV_Role» not found	N/A	[DESC]
			There is, at least, one «EV_Role» without any defined state machine.	Error	There is, at least, one EV Role without any associated state machine.
EV006	[UML4ODP, 7.3.12]	Every «EV_Artefact» is associated by two UML Associations, one stereotyped «EV_ArtefactRole» and the other «EV_ArtefactReference».	OK	Ok	[DESC]
			«EV_Artefact» not found.	N/A	[DESC]
			There is, at least, one «EV_Artefact» which is not associated by two UML	Error	There is, at least, one «EV_Artefact» which is not associated by two UML Associations «EV_ArtefactRole» and «EV_ArtefactReference».

ID	Reference	Rule	Condition	Result	[DESC]
			Associations, «EV_ArtefactRole» and «EV_ArtefactReference».		
EV007	[UML4ODP, 7.2.4]	Every UML Component stereotyped «EV_Community» will have exactly one association stereotyped «EV_ObjectiveOf» to a class stereotyped «EV_Objective».	N=1	Ok	[DESC]
			«EV_Community» not found.	N/A	[DESC]
			N≠1	Error	[DESC] **There is, at least, one UML Component stereotyped «EV_Community» which is not properly associated. Every «EV_Community» should have exactly one association stereotyped «EV_ObjectiveOf» with a class stereotyped «EV objective».**
EV008	[UML4ODP, 7.2.7]	Every UML Class stereotyped «EV_CommunityObject» should have exactly one UML Dependency, stereotyped «EV_RefinesAsCommunity» to a component stereotyped «EV_Community».	N=1	Ok	[DESC]
			«EV_CommunityObject» not found.	N/A	[DESC]
			N≠1	Error	[DESC] **There is, at least, one UML Class stereotyped «EV_CommunityObject» which is not properly associated. Every «EV_CommunityObject» should have exactly one dependency stereotyped «EV_RefinesAsCommunity» to an «EV_Community».**
EV009	[UML4ODP, 7.2.10.3]	Every UML Class stereotyped «EV_Interaction» will have, at least, one Association stereotyped «EV_InteractionInitiator» or «EV_InteractionResponder» to a Class stereotyped «EV_Role».	N>0	Ok	[DESC]
			«EV_Interaction» not found.	N/A	[DESC]
			N=0	Error	[DESC] **There is, at least, one «EV_Interaction» which is not properly associated. Every «EV_Interaction» should have, at least, one Association stereotyped «EV_InteractionInitiator» or «EV_InteractionResponder» to a Class stereotyped «EV_Role».**
EV010	[UML4ODP, 7.2.10.3]	Every UML Class stereotyped «EV_Interaction» should declare, at least, one Signal stereotyped «EV_Interaction».	N>0	Ok	[DESC]
			N=0	Warning	[DESC] **Every UML Class stereotyped «EV_Interaction» should declare, at, least, one Signal stereotyped «EV_Interaction».**
			«EV_Interaction» not found.	N/A	[DESC]
EV011	[UML4ODP, 7.2.21]	Every UML Class stereotyped «EV_Party» must also be stereotyped «EV_Object».	OK	Ok	[DESC]
			«EV_Party» not found.	N/A	[DESC]
			There is, at least, one «EV_Party» not properly stereotyped.	Error	[DESC] **There is, at least, one «EV_Party» which is not properly stereotyped. «EV_Party» needs to be stereotyped «EV_Object» too.**
EV012	[UML4ODP, 7.2.22]	Every UML Association stereotyped «EV_Accountable» must relate «EV_Role» to «EV_Interaction».	OK	Ok	[DESC]
			«EV_Accountable» not found.	N/A	[DESC]
			There is, at least, one UML Association stereotyped «EV_Accountable» that is not properly defined.	Error	[DESC] **There is, at least, one UML Association stereotyped «EV_Accountable» that is not properly defined. It should relate «EV_Role» and «EV_Interaction» elements.**
EV013	[UML4ODP, 7.2.23]	Every UML Association stereotyped «EV_Delegation» must relate «EV_Role» to «EV_Role».	OK	Ok	[DESC]
			«EV_Delegation» not found.	N/A	[DESC]
			There is, at least, one UML Association stereotyped	Error	[DESC] **There is, at least, one UML Association stereotyped «EV_Delegation» that is not properly defined. It can only relate «EV_Role» elements.**

ID	Reference	Constraint	Condition	Result		Description
			«EV_Delegation» that is not properly defined.			
EV014	[UML4ODP, 7.2.8]	Every UML Class stereotyped «EV_Objective» will have exactly one UML_ Association stereotyped «EV_ObjectiveOf» to a component stereotyped «EV_Community».	N=1	Ok	[DESC]	
			«EV_Objective» not found.	N/A	[DESC]	
			N≠1	Error	[DESC]	**There is, at least, one UML Class stereotyped «EV_Objective» that is not properly associated to a component stereotyped «EV_Community».**
EV015	[UML4ODP, 7.2.4]	Every UML Component stereotyped «EV_Community» will have a set of realizations, each stereotyped «EV_CommunityBehaviour», to a UML Class stereotyped «EV_Role».	N>0	Ok	[DESC]	
			«EV_Community» not found.	N/A	[DESC]	
			N=0	Error	[DESC]	**There is, at least, one UML Component stereotyped «EV_Community» that is not properly associated to a UML Class stereotyped «EV_Role».**
EV016	[UML4ODP, 7.2.14]	Every UML Class stereotyped «EV_PolicyValue» will have a set of constraints stereotyped «EV_PolicyValueRule».	N>0	Ok	[DESC]	
			«EV_PolicyValue» not found.	N/A	[DESC]	
			N=0	Error	[DESC]	**There is, at least, one UML Class stereotyped «EV_PolicyValue» that does not define any constraint stereotyped «EV_PolicyValueRule».**
EV017	[UML4ODP, 7.2.14]	Every UML Class stereotyped «EV_PolicyEnvelope» will have a set of constraints stereotyped «EV_PolicyEnvelopeRule».	N>0	Ok	[DESC]	
			«EV_PolicyEnvelope» not found.	N/A	[DESC]	
			N=0	Error	[DESC]	**There is, at least, one UML Class stereotyped «EV_PolicyEnvelope» that does not define any constraints stereotyped «EV_PolicyEnvelopeRule».**
EV018	[UML4ODP, 7.2.14]	Each UML Class stereotyped «EV_PolicyValue» will have one aggregation to a UML Class stereotyped «EV_PolicyEnvelope».	N=1	Ok	[DESC]	
			«EV_PolicyValue» not found.	N/A	[DESC]	
			N≠1	Error	[DESC]	**There is, at least, one «EV_PolicyValue» that is not properly associated to a UML Class stereotyped «EV_PolicyEnvelope».**
EV019	[UML4ODP, 7.2.14]	Each UML Class stereotyped «EV_PolicyValue» will have an association to a class stereotyped «EV_PolicyEnvelope».	N=1	Ok	[DESC]	
			«EV_PolicyEnvelope» not found.	N/A	[DESC]	
			N≠1	Error	[DESC]	**There is, at least, one «EV_PolicyEnvelope» that is not properly associated to a UML Class stereotyped «EV_PolicyEnvelope».**

Table 2 – EV Constraints

7.3.3 Information Viewpoint

ID	Reference	Description	Case	Result	Message
IV001	[UML4ODP, 8.3]	There exists at least one UML Package stereotyped «IV_InvariantSchema».	N > 0	Ok	[DESC]
			N = 0	Warning	**There is no UML Package stereotyped «IV_InvariantSchema». At least, one invariant schema should be included in the specification.**
IV002	[UML4ODP, 8.3]	There exists at least one UML Package stereotyped «IV_StaticSchema».	N > 0	Ok	[DESC]
			N = 0	Warning	**There is no UML Package stereotyped «IV_StaticSchema». At least, one static schema should be specified.**
IV003	[UML4ODP, 8.3]	Every IV invariant schema compromises at least one IV object.	N > 0	Ok	[DESC]
			N = 0	Warning	**One or more information objects should be specified in the invariant schema.**
			«IV_InvariantSchema» not found.	N/A	[DESC]
IV004	[UML4ODP, 8.3]	Every instantiated IV object in the static schema should be also declared in the invariant schema.	OK	Ok	[DESC]
			«IV_Object» not found.	N/A	[DESC]
			There is, at least, one Instance of IV_Object into IV_StaticSchema that its classifier is not declared in IV_InvariantSchema	Error	**There is, at least, one UML Instance of an information object which is not declared in the invariant schema despite being specified in the static schema.**

Table 3 – IV Constraints

7.3.4 Computational Viewpoint

ID	Reference	Description	Case	Result	Message
CV001	[UML4ODP, 9.2.1]	For Every Computational Object the value of the attribute 'isIndirectlyInstantiated' is set to true.	OK	Ok	[DESC]
			CV_Object no found	N/A	[DESC]
			At least one CV_Object has the value of this attribute set to false	Error	There is, at least, one CV_Object whose attribute 'isIndirectlyInstantiated' is set to false.
CV002	[UML4ODP, 9.2.2]	Every UML Component stereotyped «CV_TemplateObject» will have the value of the attribute 'isAbstract' set to false.	OK	Ok	[DESC]
			CV_Template Object not found	N/A	[DESC]
			At least one CV_Template Object has the value of this attribute set to true	Error	There is, at least, one CV_TemplateObject whose attribute 'isAbstract' is set to true.
CV003	[UML4ODP, 9.2.3]	Every UML Component stereotyped «CV_BindingObject» should bind two or more computational objects through the same type of interface.	OK	Ok	[DESC]
			CV_Binding Object not found	N/A	[DESC]
			One or more CV_Binding Object doesn't bind, at least, two Computational Object through the same type of interface.	Error	One or more CV_BindingObject own interfaces (UML Ports) that do not bind two or more different CV_Objects.
CV004	[UML4ODP, 9.2.9]	Every UML Port comprised by a UML Component (stereotyped «CV_Objects») should be stereotyped as «CV_OperationInterface», «CV_SignalInterface» or «CV_StreamInterface» depending on its type.	OK	Ok	[DESC]
			CV_Object not found	N/A	[DESC]
			At least one Port of a Component is not correctly stereotyped.	Error	Any UML Port comprised by a UML Component stereotyped «CV_Object» should be stereotyped «CV_SignalInterface», «CV_OperationInterface» or «CV_StreamInterface».
CV005	[UML4ODP, 9.2.9]	Every UML Port stereotyped «CV_SignalInterface», «CV_OperationInterface» or «CV_StreamInterface» should be owned by a UML Component stereotyped «CV_Object» or «CV_BindingObject».	OK	Ok	[DESC]
			UML Port stereotyped «CV_SignalInterface», «CV_OperationInterface» or «CV_StreamInterface» not found.	N/A	[DESC]
			At least one UML Port stereotyped «CV_SignalInterface»	Error	At least one UML Port stereotyped «CV_SignalInterface» «CV_OperationInterface» or «CV_StreamInterface» is not owned by a UML Component stereotyped «CV_Object» or «CV_BindingObject».

ID	Reference	Rule	Condition	Status	Type	Message
CV006	[UML4ODP, 9.2.9]	Every UML Port stereotyped «CV_OperationInterface» has a Tag definition 'causality', which should be set to 'server' or 'client'.	«CV_OperationInterface» or «CV_StreamInterface» is not owned by a UML Component stereotyped «CV_Object» or «CV_BindingObject». OK	Ok	[DESC]	
			UML Port stereotyped «CV_OperationInterface» not found.	N/A	[DESC]	
			At least one UML Port stereotyped «CV_OperationInterface» has a Tag definition 'causality' that should be set to 'server' or 'client'.	Error	[DESC]	**There is, at least, one UML Port stereotyped «CV_OperationInterface» whose Tag definition 'causality' is not set to either 'server' or 'client'.**
CV007	[UML4ODP, 9.2.9]	Every UML Port stereotyped «CV_SignalInterface» has a Tag definition 'causality', which should be set to 'initiator' or 'responder'.	OK	Ok	[DESC]	
			UML Port stereotyped «CV_SignalInterface» not found.	N/A	[DESC]	
			At least one UML Port stereotyped «CV_SignalInterface» has a Tag definition 'causality' that should be set to 'initiator' or 'responder'.	Error	[DESC]	**There is, at least, one UML Port stereotyped «CV_SignalInterface» whose Tag definition 'causality' is not set to either 'initiator' or 'responder'.**
CV008	[UML4ODP, 9.2.15]	Every UML Port stereotyped «CV_TerminationSignature» has a Tag definition 'invocation'.	OK	Ok	[DESC]	
			UML Port stereotyped «CV_TerminationSignature» not found.	N/A	[DESC]	
			At least one UML Port stereotyped «CV_TerminationSignature» has a Tag definition 'invocation' which is not properly initialized.	Error	[DESC]	**There is, at least, one UML Port stereotyped «CV_TerminationSignature» whose Tag definition 'invocation' is not properly initialized.**
CV009	[UML4ODP, 9.2.17]	For every UML Assembly Connector stereotyped «CV_PrimitiveBinding» connecting interfaces, these interfaces should be stereotyped «CV_OperationInterfaceSignature».	OK	Ok	[DESC]	
			UML Assembly Connector stereotyped «CV_PrimitiveBinding» not	N/A	[DESC]	

ID	Reference	Rule	Result message	Status	Detail
			found.		
			At least one, UML Assembly Connector stereotyped «CV_PrimitiveBinding» that connects interfaces, this interfaces is not correctly stereotyped.	Error	**There is, at least, one UML Assembly Connector stereotyped «CV_PrimitiveBinding» connecting interfaces which are not stereotyped «CV_OperationInterfaceSignature».**
CV010	[UML4ODP, 9.2.17]	For every UML Assembly Connector stereotyped «CV_PrimitiveBinding» connecting UML Ports, these ports should be stereotyped «CV_SignalInterface», «CV_OperationInterface» or «CV_StreamInterface». Note that both stereotypes should match each other, according to the primitive binding rules.	OK	Ok	[DESC]
			UML Assembly Connector stereotyped «CV_PrimitiveBinding» not found.	N/A	[DESC]
			There is, at least, one UML Assembly Connector stereotyped «CV_PrimitiveBinding» connecting UML Ports that are not properly stereotyped. These stereotypes of this ports should be «CV_SignalInterface»,«CV_OperationInterface» or «CV_StreamInterface».	Error	**There is, at least, one UML Assembly Connector stereotyped «CV_PrimitiveBinding» connecting UML Ports which are not stereotyped «CV_SignalInterface», «CV_OperationInterface» or «CV_StreamInterface».**
CV011	[UML4ODP, 9.2.17]	For every UML Assembly Connector stereotyped «CV_PrimitiveBinding» connecting UML Ports, these ports should be stereotyped «CV_SignalInterface», «CV_OperationInterface» or «CV_StreamInterface». Note that both stereotypes should match each other, according to the primitive binding rules.	OK	Ok	[DESC]
			UML Assembly Connector stereotyped «CV_PrimitiveBinding» not found.	N/A	[DESC]
			There is, at least, one UML Assembly Connector stereotyped «CV_PrimitiveBinding» connecting UML Ports that are not properly stereotyped. Note that both ports in the relationship should be equally stereotyped	Error	**There is, at least, one UML Assembly Connector stereotyped «CV_PrimitiveBinding» connecting UML Ports which are nor stereotyped with the same stereotyped.**
CV012	[UML4ODP, 9.2.17]	For every UML Assembly Connector stereotyped «CV_PrimitiveBinding» connecting UML Ports, these ports	OK	Ok	[DESC]
			UML Assembly Connector	N/A	[DESC]

Code	Reference	Rule	Result message	Status	Description
		should be stereotyped «CV_SignalInterface», «CV_OperationInterface» or «CV_StreamInterface». Note that both stereotypes should match each other, according to the primitive binding rules.	stereotyped «CV_PrimitiveBinding» not found.		
			There is, at least, one UML Assembly Connector stereotyped «CV_PrimitiveBinding» connecting UML Ports, where the client port is not compatible with the interface represented by the server port, according to the primitive binding rules.	Error	**There is, at least, one UML Assembly Connector stereotyped «CV_PrimitiveBinding» connecting UML Ports, where the interface represented by the client port is not compatible with the interface represented by the server port, according to the primitive binding rules.**
CV013	[UML4ODP, 9.2.18]	Every UML Property stereotyped «CV_Flow» is comprised by an interface stereotyped «CV_StreamInterfaceSignature»	OK.	Ok	[DESC]
			UML Property stereotyped «CV_Flow» not found.	N/A	[DESC]
			There is, at least, one UML Property stereotyped «CV_Flow» which is not comprised by an interface stereotyped «CV_StreamInterfaceSignature».	Error	**There is, at least, one UML Property stereotyped «CV_Flow» which is not owned by an interface stereotyped «CV_StreamInterfaceSignature».**
CV014	[UML4ODP, 9.2.18]	Every flow signature is expressed by the type of a UML property, which should be declared by a UML Interface stereotyped «CV_FlowSignature».	OK.	Ok	[DESC]
			UML Property stereotyped «CV_Flow» not found.	N/A	[DESC]
			There is, at least, one flow signature that is expressed by the type of a UML property, which was declared by a UML Interface not stereotyped «CV_FlowSignature».	Error	**There is, at least, one flow signature that is expressed by the type of a UML property, which was declared by a UML Interface not stereotyped «CV_FlowSignature».**
CV015	[UML4ODP, 9.2.18]	Every UML Property stereotyped «CV_Flow» has a tag definition 'causality'.	OK.	Ok	[DESC]
			UML Property stereotyped «CV_Flow» not found.	N/A	[DESC]
			There is, at least, one UML Property stereotyped «CV_Flow» which has not a tag definition 'causality'.	Error	**There is, at least, one UML Property stereotyped «CV_Flow» which has not a tag definition named 'causality'.**
CV016	[UML4ODP, 9.2.18]	For any flow, its causality is expressed	OK.	Ok	[DESC]

			N/A	[DESC]
by the tag definition, 'causality', of stereotype «CV_Flow», whose value should be set to one of the following literals: 'consumer' or 'producer'.	UML Property stereotyped «CV_Flow» not found.	There is, at least, one flow whose causality (expressed by Tag Definition, 'causality' of stereotype «CV_Flow») is neither 'consumer' nor 'producer'.	Error	**There is, at least, one flow whose causality (expressed by Tag Definition, 'causality' of stereotype «CV_Flow») is neither 'consumer' nor 'producer'.**

Table 4 – CV Constraints

7.3.5 Engineering Viewpoint

ID	Reference	Description	Case	Result	Message
NV001	[UML4ODP, 10.1.6.2]	Every UML Component stereotyped «NV_Nucleus» is related to a UML Component stereotyped «NV_Node».	OK	Ok	[DESC]
			UML Component stereotyped «NV_Nucleus» not found.	N/A	[DESC]
			At least one NV_Nucleus is not related to NV_Node.	Error	**There is, at least, one NV_Nucleus which is not related to a NV_Node.**
NV002	[UML4ODP, 10.1.6.2]	Every UML Component stereotyped «NV_Node» is related to exactly one UML Component stereotyped «NV_Nucleus».	N = 1	Ok	[DESC]
			UML Component stereotyped «NV_Node» not found.	N/A	[DESC]
			N≠1	Error	**There is, at least, one UML Component stereotyped «NV_Node» which is not related to exactly one UML Component stereotyped «NV_Nucleus».**
NV003	[UML4ODP, 10.1.6.2]	Every UML Component stereotyped «NV_Capsule» is related to a UML Component stereotyped «NV_Node».	OK	Ok	[DESC]
			UML Component stereotyped «NV_Capsule» not found.	N/A	[DESC]
			At least one NV_Capsule is not related to a NV_Node.	Error	**There is, at least, one NV_Capsule which is not related to a NV_Node.**
NV004	[UML4ODP, 10.1.6.2]	Every UML Component stereotyped «NV_Nucleus» is related to exactly one UML Component stereotyped «NV_NodeManagement».	N=1	Ok	[DESC]
			Component stereotyped «NV_Nucleus» not found.	N/A	[DESC]
			N≠1	Error	**There is, at least, one UML Component stereotyped «NV_Nucleus» which is not related to exactly one UML Component stereotyped «NV_NodeManagement».**
NV005	[UML4ODP, 10.1.6.2]	Every UML Interface stereotyped «NV_NodeManagement» is related to a UML Component stereotyped «NV_Nucleus».	OK	Ok	[DESC]
			UML Interface stereotyped «NV_NodeManagement» not found.	N/A	[DESC]
			At least one «NV_NodeManagement» is not related to a NV_Nucleus.	Error	**There is, at least, one «NV_NodeManagement» which is not related to a «NV_Nucleus».**
NV006	[UML4ODP, 10.1.6.2]	Every UML Component stereotyped «NV_CapsuleManager» is related to a UML Component stereotyped «NV_Capsule».	OK	Ok	[DESC]
			UML Component stereotyped «NV_CapsuleManager» not found.	N/A	[DESC]
			At least one «NV_CapsuleManager» is not related to a NV_Capsule.	Error	**There is, at least, one «NV_CapsuleManager» which is not related to a «NV_Capsule».**
NV007	[UML4ODP, 10.1.6.2]	Every UML Component stereotyped «NV_Capsule» is related to exactly one UML Component stereotyped	N=1	Ok	[DESC]
			UML Component stereotyped «NV_Capsule» not found.	N/A	[DESC]

ID	Reference	Rule	Condition	Status	Description
NV008	[UML4ODP, 10.1.6.2]	«NV_CapsuleManager».	N≠1	Error	There is, at least, one UML Component stereotyped «NV_Capsule» which is not related to exactly one UML Component stereotyped «NV_CapsuleManager».
		Every UML Component stereotyped «NV_ClusterManager» is related to a UML Component stereotyped «NV_Capsule».	OK	Ok	[DESC]
			UML Component stereotyped «NV_ClusterManager» not found	N/A	[DESC]
			At least one «NV_ClusterManager» is not related to a NV_Capsule.	Error	There is, at least, one «NV_ClusterManager» which is not related to a NV_Capsule.
NV009	[UML4ODP, 10.1.6.2]	Every UML Component stereotyped «NV_Cluster» is related to a UML Component stereotyped «NV_Capsule».	OK	Ok	[DESC]
			UML Component stereotyped «NV_Cluster» not found.	N/A	[DESC]
			At least one «NV_Cluster» is not related to a «NV_Capsule».	Error	There is, at least, one «NV_Cluster» which is not related to a NV_Capsule.
NV010	[UML4ODP, 10.1.6.2]	Every UML Interface stereotyped «NV_CapsuleManagement» is related to a UML Component stereotyped «NV_CapsuleManager».	OK	Ok	[DESC]
			UML Interface stereotyped «NV_CapsuleManagement» not found.	N/A	[DESC]
			At least one «NV_CapsuleManagement» is not related to a NV_CapsuleManager.	Error	There is, at least, one «NV_CapsuleManagement» which is not related to a «NV_CapsuleManagement».
NV011	[UML4ODP, 10.1.6.2]	Every UML Component stereotyped «NV_CapsuleManager» is related to exactly one UML Component stereotyped «NV_CapsuleManagement».	N=1	Ok	[DESC]
			UML Component stereotyped «NV_CapsuleManager» not found.	N/A	[DESC]
			N≠1	Error	There is, at least, one UML Component stereotyped «NV_CapsuleManager» which is not related to exactly one UML Component stereotyped «NV_CapsuleManagement».
NV012	[UML4ODP, 10.1.6.2]	Every UML Component stereotyped «NV_BEO» is related to a UML Component stereotyped «NV_Cluster».	OK	Ok	[DESC]
			UML Component stereotyped «NV_BEO» not found.	N/A	[DESC]
			At least one «NV_BEO» is not related to a «NV_Cluster».	Error	There is, at least, one «NV_BEO» which is not related to a «NV_Cluster».
NV013	[UML4ODP, 10.1.6.2]	Every UML Interface stereotyped «NV_ClusterManagement» is related to a UML Component stereotyped «NV_ClusterManager».	OK	Ok	[DESC]
			UML Interface stereotyped «NV_ClusterManagement» not found.	N/A	[DESC]
			At least one «NV_ClusterManagement» is not related to a UML Component stereotyped	Error	There is, at least, one NV_ClusterManagement which is not related to a UML Component stereotyped «NV_ClusterManager».

ID	Reference	Rule	Condition	Status	Message
NV014	[UML4ODP, 10.1.6.2]	Every UML Component stereotyped «NV_ClusterManager» is related to exactly one UML Component stereotyped «NV_ClusterManagement».	«NV_ClusterManager».		
			N=1	Ok	[DESC]
			UML Component stereotyped «NV_ClusterManager» not found.	N/A	[DESC]
			N≠1	Error	**There is, at least, one UML Component stereotyped «NV_ClusterManager» which is not related to exactly one UML Component stereotyped «NV_ClusterManagement».**
NV015	[UML4ODP, 10.1.6.3]	Every UML Component stereotyped «NV_Channel» is related to, at least, two UML Component stereotyped «NV_BEO».	N>1	Ok	[DESC]
			UML Component stereotyped «NV_Channel» not found.	N/A	[DESC]
			N<2	Error	**There is, at least, one UML Component stereotyped «NV_Channel» which is not related to two or more UML Component stereotyped «NV_BEO».**
NV016	[UML4ODP, 10.1.6.3]	Every UML Component stereotyped «NV_Stub», «NV_Binder» or «NV_ProtocolObject» is related to a UML Component stereotyped «NV_Channel».	OK	Ok	[DESC]
			UML Component stereotyped «NV_Channel» not found.	N/A	[DESC]
			At least one «NV_Stub», «NV_Binder» or «NV_ProtocolObject» is not related to a UML Component stereotyped «NV_Channel».	Error	**There is, at least, one «NV_Stub», «NV_Binder» or «NV_ProtocolObject» which is not related to a UML Component stereotyped «NV_Channel».**
NV017	[UML4ODP, 10.1.6.3]	Every UML Component stereotyped «NV_Channel» is related to, at least, two UML Component stereotyped «NV_Stub».	N>1	Ok	[DESC]
			UML Component stereotyped «NV_Channel» not found.	N/A	[DESC]
			N<2	Error	**There is, at least, one UML Component stereotyped «NV_Channel» which is not properly related to two or more UML Components stereotyped «NV_Stub».**
NV018	[UML4ODP, 10.1.6.3]	Every UML Component stereotyped «NV_Channel» is related to, at least, two UML Component stereotyped «NV_Binder».	N>1	Ok	[DESC]
			UML Component stereotyped «NV_Channel» not found.	N/A	[DESC]
			N<2	Error	**There is, at least, one UML Component stereotyped «NV_Channel» which is not properly related to two or more UML Components stereotyped «NV_Binder».**
NV019	[UML4ODP, 10.1.6.3]	Every UML Component stereotyped «NV_Channel» is related to, at least, two UML Component stereotyped «NV_ProtocolObject».	N>1	Ok	[DESC]
			UML Component stereotyped «NV_Channel» not found.	N/A	[DESC]
			N<2	Error	**There is, at least, one UML Component stereotyped «NV_Channel» which is not properly related to two or more UML Components stereotyped «NV_ProtocolObject».**
NV020	[UML4ODP, 10.1.6.3]	Every UML Component stereotyped «NV_Stub» is associated with exactly one UML Component stereotyped «NV_Binder».	N=1	Ok	[DESC]
			UML Component stereotyped «NV_Stub» not found.	N/A	[DESC]
			N≠1	Error	**There is, at least, one UML Component stereotyped «NV_Stub» which is not associated with exactly one UML Component stereotyped «NV_Binder».**
NV021	[UML4ODP, 10.1.6.3]	Every UML Component stereotyped	N>0	Ok	[DESC]

		«NV_Binder» is associated with at least one UML Component stereotyped «NV_ProtocolObject».	UML Component stereotyped «NV_Binder» not found.	N/A	[DESC]
			N=0	Error	**There is, at least, one UML Component stereotyped «NV_Binder» which is not associated with any UML Component stereotyped «NV_ProtocolObject».**
NV022	[UML4ODP, 10.2.1]	For Every UML Component stereotyped «NV_Object» the value of the attribute 'isIndirectlyInstantiated' is set to false.	OK	Ok	[DESC]
			UML Component stereotyped «NV_Object» not found.	N/A	[DESC]
			At least one «NV_Object» has the value of this attribute set to true.	Error	**There is, at least, one «NV_Object» whose attribute 'isIndirectlyInstantiated' is set to true.**
NV023	[UML4ODP, 10.2.1]	Every UML Component stereotyped «NV_Object» has a Tag definition 'deployedNode', 'securityDomain' and 'managementDomain'.	OK	Ok	[DESC]
			UML Component stereotyped «NV_Object» not found.	N/A	[DESC]
			Some Tagged value was not initialized.	Error	**There is, at least, one «NV_Object» with any of its tagged values 'deployedNode', 'securityDomain' and 'managementDomain' defined.**

Table 5– NV Constraints

7.3.6 Technology Viewpoint

ID	Reference	Description	Case	Result	Message
TV001	[UML4ODP, 11.2.6]	Every UML Node or Artifact stereotyped «TV_Object» is related to at least one UML Component stereotyped «TV_ImplementableStandard».	UML Component stereotyped «TV_Object» not found. N>0	Ok	[DESC]
				N/A	[DESC]
			N≠0	Error	There is, at least, one UML Node or Artifact stereotyped «TV_Object» which is not related to any UML Component stereotyped «TV_ImplementableStandard».
TV002	[UML4ODP, 11.2.6]	Every UML Component stereotyped «TV_ImplementableStandard» is related to, at least, one UML Node or Artifact stereotyped «TV_Object».	UML Component stereotyped «TV_ImplementableStandard» not found. N>0	Ok	[DESC]
				N/A	[DESC]
			N≠0	Error	There is, at least, one UML Component stereotyped «TV_ImplementableStandard» that is not properly associated to any «TV_Object».
TV003	[UML4ODP, 11.2.6]	Every UML Activity stereotyped «TV_Implementation» is associated with one or more UML Component stereotyped «TV_Object».	UML Component stereotyped «TV_Implementation» not found. OK	Ok	[DESC]
				N/A	[DESC]
			At least one «TV_Implementable» is not associated with any «TV_Object».	Error	There is, at least, one UML Activity stereotyped «TV_Implementation» which is not associated with any UML Component stereotyped «TV_Object».

Table 6 – TV Constraints

7.3.7 Correspondence Diagram

ID	Reference	Description	Case	Result	Message
CR001	[UML4ODP, 12.2.3]	Every UML Class stereotyped «CorrespondenceLink» has two tag definitions (named endPoint1 and endPoint2) which specify the two correspondence endpoints of the correspondence link.	OK	Ok	[DESC]
			UML Class stereotyped «CorrespondenceLink» not found	N/A	[DESC]
			Some endPoint was not initialized	Error	There is, at least, one UML Class stereotyped «CorrespondenceLink» whose endPoints are not properly defined.
CR002	[UML4ODP, 12.2.3]	Every UML Class stereotyped «CorrespondenceLink» has two tag definitions (named endPoint1 and endPoint2) which specify the two correspondence endpoints of the correspondence link.	OK	Ok	[DESC]
			UML Class stereotyped «CorrespondenceLink» not found	N/A	[DESC]
			endPoint1 and endPoint2 reference to the same viewpoint	Error	There is, at least, one UML Class stereotyped «CorrespondenceLink» whose endpoints refer to elements in the same viewpoint.
CR003	[UML4ODP, 12.2.3]	Every UML Class stereotyped «CorrespondenceLink» has two tag definitions (named endPoint1 and endPoint2) which specify the two correspondence endpoints of the correspondence link.	OK	Ok	[DESC]
			UML Class stereotyped «CorrespondenceLink» not found	N/A	[DESC]
			endPoint1 or endPoint2 do not reference to valid elements in the correspondence	Error	There is, at least, one UML Class stereotyped «CorrespondenceLink» in a correspondence model with an endpoint which does not refer to an element of a valid viewpoint for that correspondence.

Table 7 – Correspondence Constraints

7.4 How to create and manage custom constraints

The following sections explain how the interested user can code his/her own validations and add them to the repository, so they can be enabled and automatically executed during the validation process.

7.4.1 Preparing the development environment

To illustrate the development task of a new validation, a sample Eclipse project has been created. This project contains all the necessary files (including libraries required by MagicDraw and other dependencies) to develop a new validation.

Please follow the instructions below:

1. Download the Eclipse IDE (no specific version is required, just the regular one for Java programmers, at least) from the URL:
   `http://www.eclipse.org/downloads/`
2. Download the example project from the xODP website at
   `http://www.jrromero.net/tools/mdplugin/`
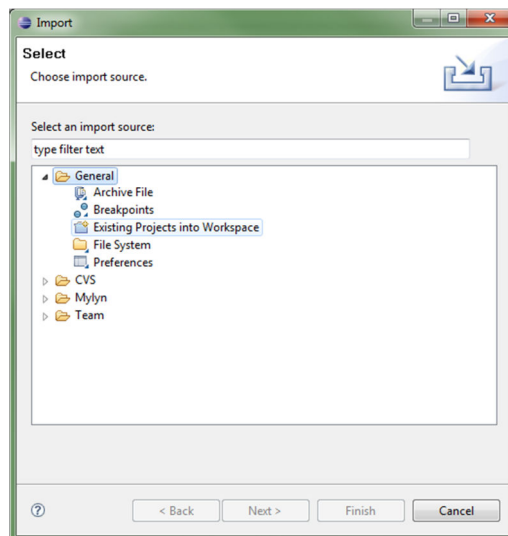3. Import the project into the eclipse workspace as shown in the following figure.

Figure 22 – Importing the example project

7.4.2 Coding a validation class

Every validation test is implemented by its corresponding Java class, which should be defined in a package called `customized`, and should also implement a predefined Java interface, `IValidate`. Such an interface declares one public method, called `validate()`, being it return type of class `ValidationResult`. Instances of this class provide information about the result of the validation test performed (i.e., error, warning or any other information), a reference to the standard statement checked and

the model elements that caused the non-compliance. The following is an excerpt code which is included for illustrative purposes.

```
package customized;
public class AValidation implements IValidate {
        private es.uco.md.rmodp.view.Check.ValidationResult
    result;

        public AValidation(){
        this.result = new
    es.uco.md.rmodp.view.Check.ValidationResult();
        }

    public ValidationResult validate (Element e){

        if (hasIV_Object(e)){
            result.setCode(ValidationResultCode.INFORMATION);
            result.setDescription("Yes, there is at least one IV
            Object!");
        } else {
            result.setCode(ValidationResultCode.WARNING);
            result.setDescription("No IV Object found in the
            specification.");
            result.addElements(e.getName());
        }

        result.setStandard("A simple example.");
        return result;
        }
        …
}
```

The complete piece of code corresponds to the class `AValidation`, which is included in the Eclipse project available for download.

7.4.3 Enabling and disabling custom validations

You can also manage your own validations from the tool menu. To do that, please click on the 'Constraint Management' menu. Then, as you can see in Figure 23, you will be able to enable or disable your validations by selecting the appropriate action.
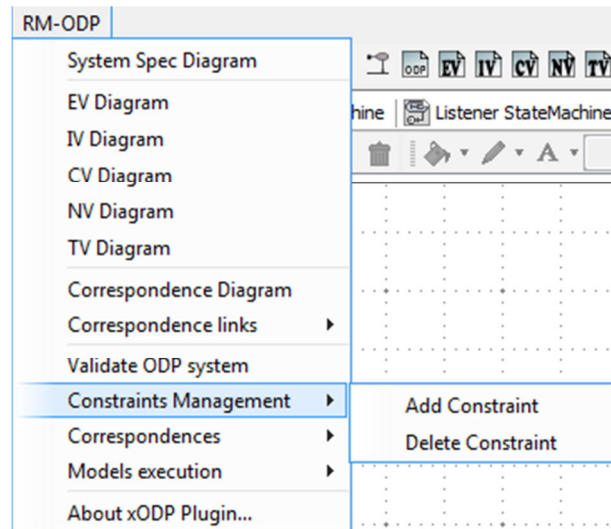
Figure 23 – Constraint management menu

For adding a new validation into the repository:

1. Select 'Add Constraint' action.

2. Fill the form in the 'Add Constraint' window (See Fig. 24).

   a. Viewpoint: Select the viewpoint where the new validation will be applied.

   b. Class Signature: This field is not editable by the user, since it will be automatically extracted from the 'Class File' textbox.

   c. Class File: Use the file browser to find the corresponding validation class (as .class file).

3. Click on 'Accept' button.
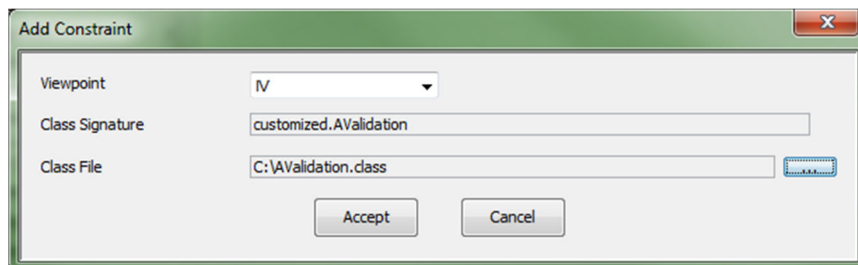
4. Reboot MagicDraw.

Figure 24 – Add constraint window

For removing a constraint from the validation repository:

1. Select the option 'Delete Constraint'.
2. Fill the form in the 'Remove Customized Constraints' window (See Fig. 25).
   a. Viewpoint: Select the viewpoint currently affected by the validation.
   b. Click on the 'Load' button. (custom validations available in this repository will be loaded next).
   c. Select the validation(s) you want to remove (See Fig.26).
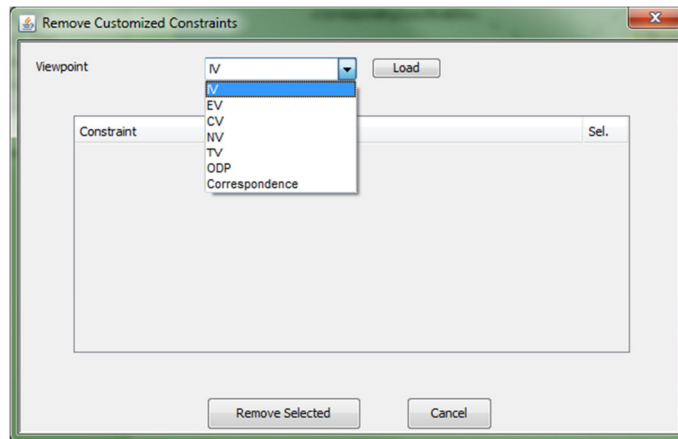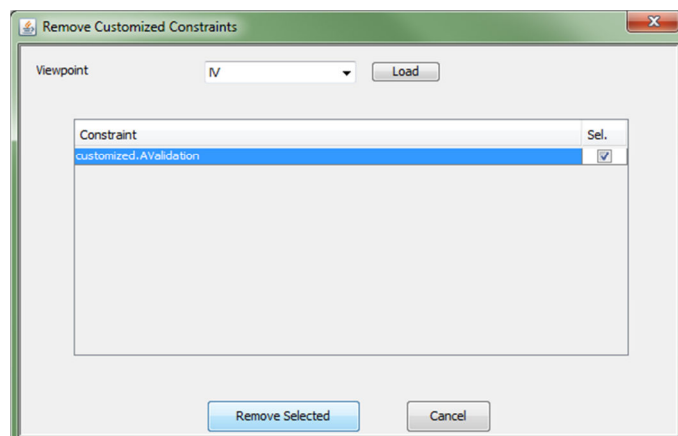3. Click on the 'Remove Selected' button.
4. Reboot MagicDraw.

Figure 25 – Remove customized window(1)

Figure 26 – Remove customized window(2)