Viewpoint DSL

Target Application

Version 1.0.0

Agenda

1 Introduction

2 User Perspective

3 Developer Perspective

Purpose

Context

- The viewpoint DSL, for the definition of viewpoint and generation of viewpoint artefacts, has a default and standard configuration (i.e., EMF)

Need

- Extending the viewpoint editor to integrate in its scope metamodels and representations of existing MBE workbenches (e.g., Capella)
- Adapting the generators to the MBE workbenches

Objective

- Introducing the notion of Target Application to enrich the viewpoint DSL with the specifies of MBE workbench environments

Agenda

1. Introduction
2. User Perspective
3. Developer Perspective

Defining a new Target Application

Stage 1. Configuration of Kitalpha

Main Use Case

1. The Software Architect[1] creates a new Target Application
2. The Software Architect sets Target Application default data
3. The Software Architect binds the Target Application with:
   a. MBE Ecore models
   b. MBE Sirius diagrams specifications
4. The Software Architect binds the Target Application with customs generations *(If there are generations provided for the corresponding MBE Workbench)*
5. The Software Architect packages and installs the Target Application in the Viewpoint Development Environment [VDE]

   Example*: Kitalpha, Capella-Studio*, etc.

(1) The person able to configure Kitalpha

Viewpoint DSL project creation

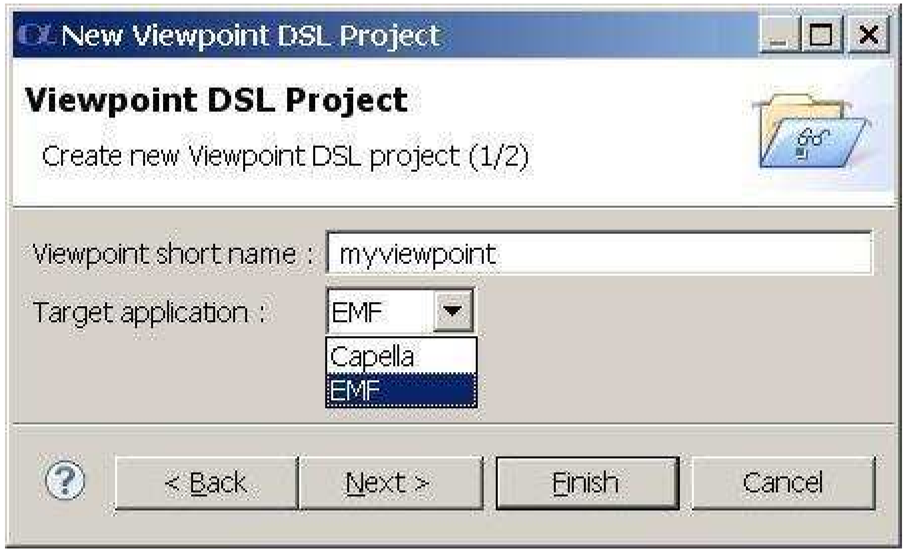
Stage 1. Configuration of Kitalpha → Stage 2. Viewpoint Development → Viewpoint Definition

Use Case 1: Viewpoint DSL project creation

1. The User creates a new Viewpoint DSL project
2. The User selects one Target Application from the available ones
3. The User validates project creation

Viewpoint DSL project creation

Viewpoint DSL Project creation wizard



New Viewpoint DSL Project

Viewpoint DSL Project

Create new Viewpoint DSL project (1/2)

Viewpoint short name : myviewpoint

Target application : EMF

Objective	Defining the targeted MBE Workbench at creation time of the viewpoint
Actions	In the Viewpoint DSL project creation wizard, the user selects one Target Application among those available

Setting the viewpoint Target Application

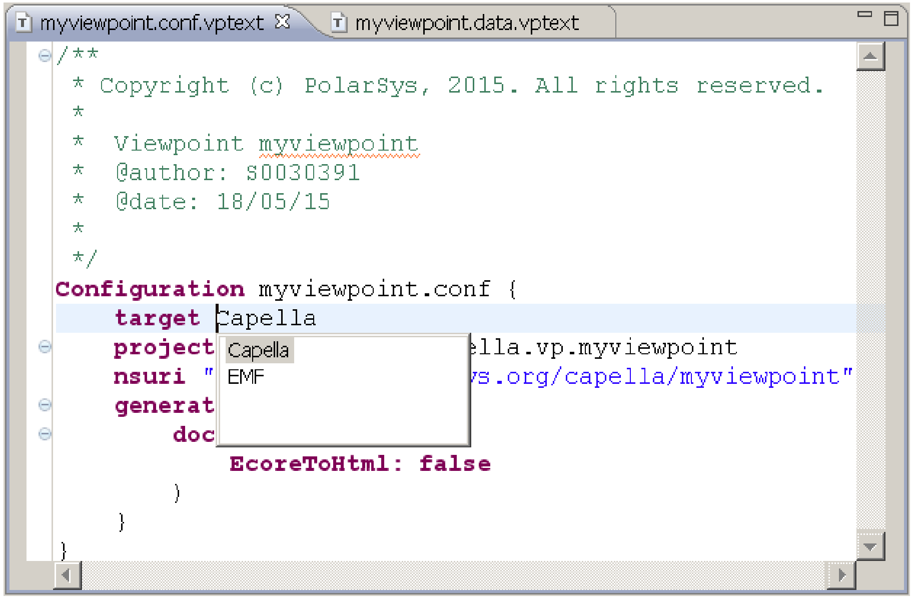
Stage 1. Configuration of Kitalpha → Stage 2. Viewpoint Development → Viewpoint Definition in DSL

Use Case 2: Setting the viewpoint Target Application

1. When the viewpoint description is created, the User opens the configuration editor of the Viewpoint DSL
2. The User sets the Target Application

Setting the viewpoint Target Application

Viewpoint configuration editor



```
/**
 * Copyright (c) PolarSys, 2015. All rights reserved.
 *
 *  Viewpoint myviewpoint
 *  @author: S0030391
 *  @date: 18/05/15
 *
 */
Configuration myviewpoint.conf {
    target Capella
    project          ella.vp.myviewpoint
    nsuri "          s.org/capella/myviewpoint"
    generat
        doc
            EcoreToHtml: false
        }
    }
}
```

Objective	Defining the targeted MBE during the viewpoint description
Actions	In the configuration editor of the Viewpoint DSL, the user selects one Target Application among those available

Using Target Application elements – Data Aspect

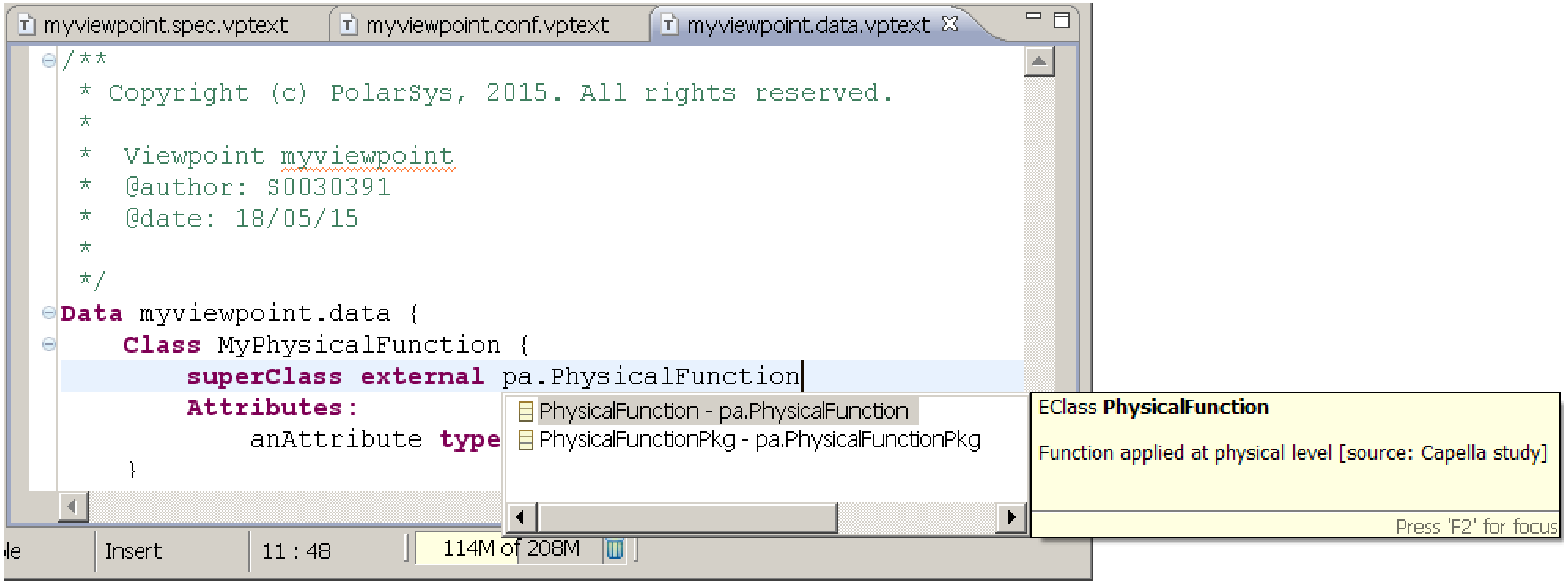
Stage 1. Configuration of Kitalpha | Stage 2. Viewpoint Development | Viewpoint Definition in DSL

Use Case 3: Using Target Application elements – Data Aspect

1. The User opens the Data editor of the Viewpoint DSL
2. The User creates a Class
3. For the specification of the super-class, class extension, association & destination class of an association, or enumeration, the User references an element from a metamodel defined in the Target Application

THALES

Using Target Application elements – Data Aspect



```
/**
 * Copyright (c) PolarSys, 2015. All rights reserved.
 *
 *  Viewpoint myviewpoint
 *  @author: S0030391
 *  @date: 18/05/15
 *
 */
Data myviewpoint.data {
    Class MyPhysicalFunction {
        superClass external pa.PhysicalFunction
        Attributes:
            anAttribute type
    }
```

```
PhysicalFunction - pa.PhysicalFunction
PhysicalFunctionPkg - pa.PhysicalFunctionPkg
```

```
EClass PhysicalFunction

Function applied at physical level [source: Capella study]

Press 'F2' for focus
```

Objective	Giving the ability to the viewpoint developer to link its specification to Target Application elements
Actions	In data.vptext or diagram.vptext, the user can use: - **external** and select one EClass, EReference, EAttribute, EEnum, etc.

Using Target Application elements – Diagram Aspect

Stage 1. Configuration of Kitalpha | Stage 2. Viewpoint Development | Viewpoint Definition in DSL

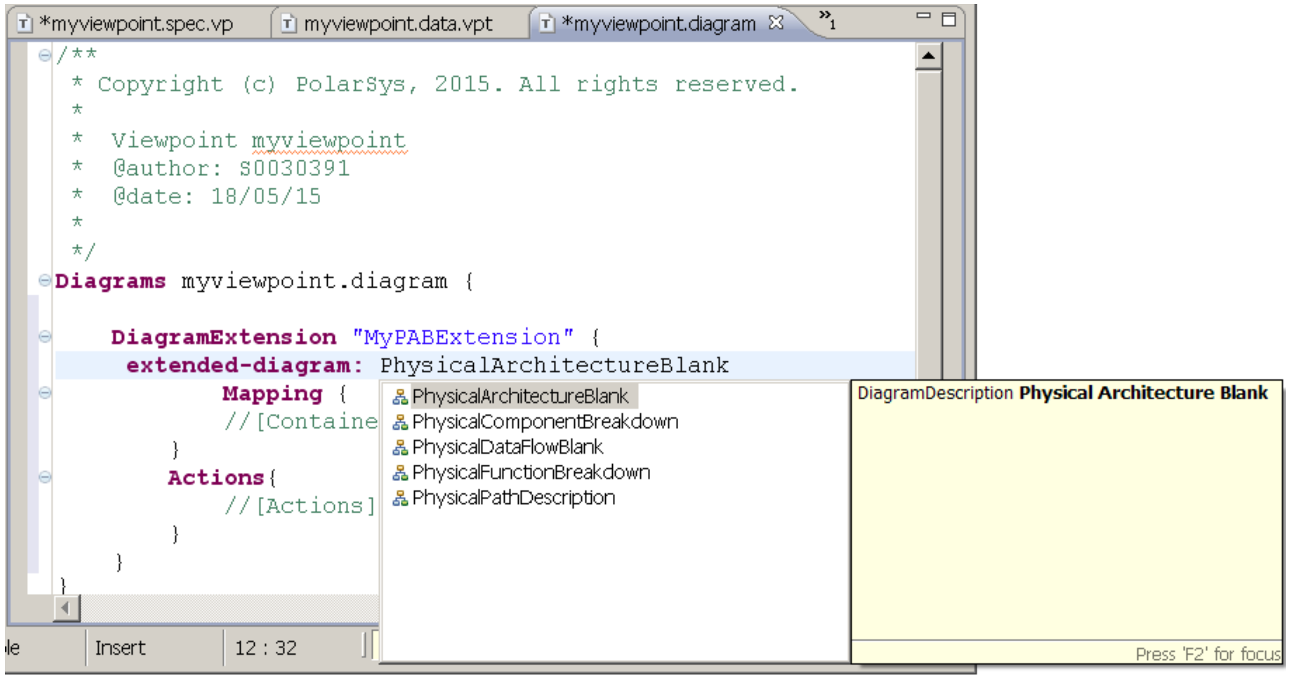
Use Case 4: Using Target Application elements – Diagram Aspect

1.The User opens the Diagram editor of the Viewpoint DSL

2.The User creates a diagram

3.For the specification of the diagram to be extended, class, or association, the User references an element from a diagram or metamodel defined in the Target Application

Topic: Using Target Application Sirius diagrams descriptions



```
/**
 * Copyright (c) PolarSys, 2015. All rights reserved.
 *
 *  Viewpoint myviewpoint
 *  @author: S0030391
 *  @date: 18/05/15
 *
 */
Diagrams myviewpoint.diagram {

    DiagramExtension "MyPABExtension" {
     extended-diagram: PhysicalArchitectureBlank
           Mapping {
           //[Containe
        }
        Actions{
           //[Actions]
        }
    }
}
```

```
PhysicalArchitectureBlank
PhysicalComponentBreakdown
PhysicalDataFlowBlank
PhysicalFunctionBreakdown
PhysicalPathDescription
```

DiagramDescription **Physical Architecture Blank**

Objective	Giving the ability to the viewpoint developer to extends, reuse or modify the targeted MBE diagrams representations
Actions	In diagram.vptext, the user can use: - **extended-diagram** and select one Diagram description - **import** and select one Mapping (container, node, edge).

THALES

Using Target Application elements – Diagram Aspect

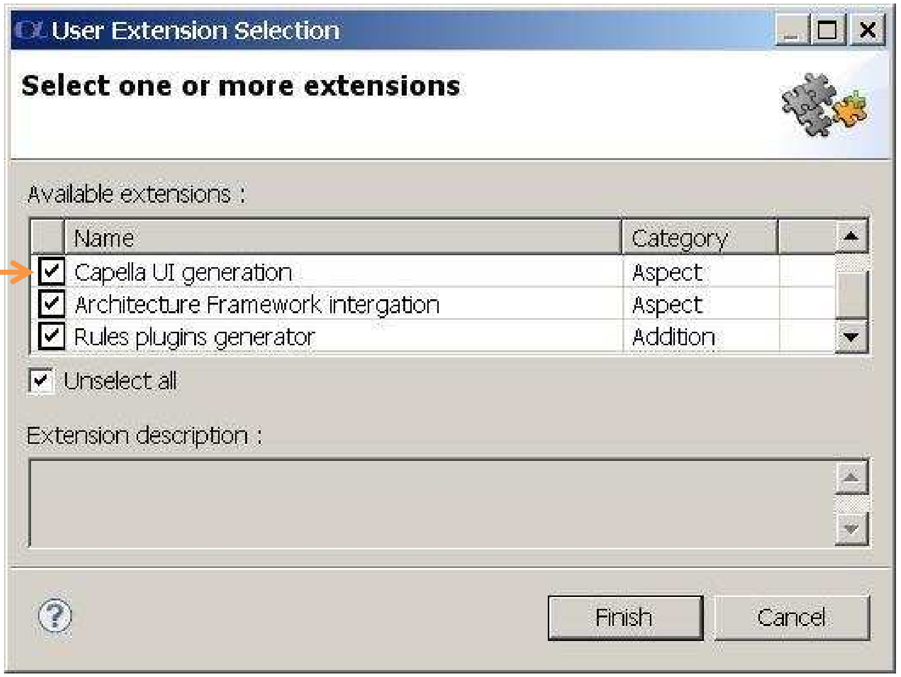
Stage 1. Configuration of Kitalpha | Stage 2. Viewpoint Development | Generation

Use Case 4: Selection of generation topic defined in a Target Application

1. The User launches a viewpoint generation with selection
2. The User selects/unselects generation topics defined in the Target Application

Selection of generation topic

Capella UI generation is available only for Capella viewpoints



Select one or more extensions

Available extensions :

Name	Category
☑ Capella UI generation	Aspect
☑ Architecture Framework intergation	Aspect
☑ Rules plugins generator	Addition

☑ Unselect all

Extension description :

Objective	Selection of generation topics from the used target application
Actions	• The user launches the generation with the "Generate viewpoint" or "Generate viewpoint with selection" command in the main viewpoint editor • The user selects/unselects generation topics

THALES

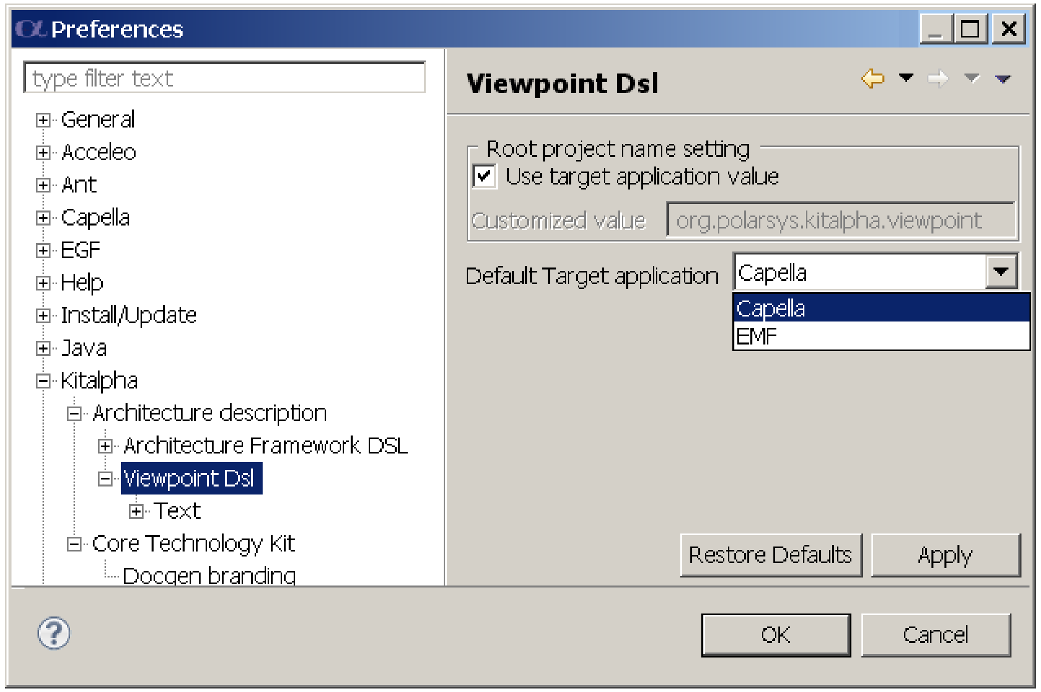
Setting viewpoint Target Application

Stage 1. Configuration of Kitalpha | Stage 2. Viewpoint Development | Viewpoint Configuration

Use Case 5: Assignment of the default Target Application

1. The User opens the configuration text file
2. Next of **target** keyword, the User press *Ctrl*+*Space* keyboard keys
3. The User selects one Target Application from the available ones
4. The User saves modifications

Assignment of the default Target Application



Root project name setting

- [✓] Use target application value

Customized value: org.polarsys.kitalpha.viewpoint

Default Target application: Capella

- Capella
- EMF

Objective	Defining the default Target Application to use in the Viewpoint DSL project creation wizard
Actions	In Eclipse preferences, the user selects one Target Application from the available ones

THALES

Agenda

1 Introduction

2 User Perspective

3 Developer Perspective

Foundations

Defining new Target Application

Phases

1

Creating new Target Application

Creating a new Target Application and setting its default data

2

Binding MBE resources

Binding MBE resources with the Target Application in order to use them in viewpoint specification

3

Declaring MBE technical specificities

Additional MBE information and generations

Extension points

Definition

Defining the Target Application wherein the viewpoint will be deployed. All information provided by extension to this point will be used during project creation, viewpoint specification and viewpoint generation.

External data loader

Binding MBE ecore resource with a Target Application.

Diagram filter

Defining the filter used to identify MBE Sirius diagram representations.

Diagram expression language

Defining the expression language to use when generating the viewpoint diagram representations

UI Properties Contributors

All MBE contributors IDs used to define a new properties view.

Customs generations

Binding custom generation to the corresponding Target Application

Phase 1. Creating new Target Application (1/2)

Extension point(s)

Name	Plugin	Schema
definition	org.polarsys.kitalpha.ad.ta	definition.exsd

Extension(s)

Description	Contributions must define a **Definition** element. It allows to provide all information the viewpoint DSL editors and generations need for the viewpoint specification and generation.	
Definition	**Name**	Setting the name of a Target Application. In most cases, this attribute contains the name of the targeted MBE.
	Root Project Name	Providing a project name namespace to initialize the viewpoint root project name configuration attribute. End-users can customize this value in the viewpoint specification files (*.conf.vptext or *.vpdesc). This attribute value must respect all rules related to the Eclipse platform project naming. The Viewpoint generation uses the root project name to define each generated project name. For example, the project wherein the Sirius diagrams are generated is named as: [root project name].design.
	Class	Providing an implementation of ITargetApplication Java interface (see the next page). This class provides to the viewpoint generation the following information: - A List of EClasses to use as super classes for each generated EClass. - Defaults generations used each time a viewpoint is generated (exp: EMF Generation)
	Root NsUri	Providing an NsUri namespace to use in the generated Ecore models. The format of NsUris of any generated ecore is: [Root NsUri].[Viewpoint short name]
	Description	Providing a short description of the Target Application.

THALES

Phase 1. Creating new Target Application (2/2)

Implementation

Plugin name	org.polarsys.kitalpha.ad.ta
Java Package	org.polarsys.kitalpha.ad.ta.extension
Class name	ITargetApplication

Available extension(s)

Description	EMF Target Application
Plugin name	org.polarsys.kitalpha.ad.ta.emf
Java Package	org.polarsys.kitalpha.ad.ta.emf.definition
Class name	EMFTargetApplication

Phase 2. Binding MBE resources

Step 1. External data loader

Implementation

Plugin name	org.polarsys.kitalpha.ad.viewpoint.dsl.cs.text.common
Java Package	org.polarsys.kitalpha.ad.viewpoint.dsl.cs.text.resources
Class name	ExternalDataHelper

Extension point(s)

Name	**Plugin**	**Schema**
externaldataloader	org.polarsys.kitalpha.ad.viewpoint.dsl.cs.text.common	externaldataloader.exsd

Extension(s)

Description	Contributions must bind the Target Application with the NsURIs of the targeted MBE Ecore model.	
ExternalDataLoader	**ID**	Providing an id of the External Data Loader
	Target	Selecting the concerned Target Application
NameSpacePattern	**ID**	Providing an id of the Name Space Pattern
	Value	A Pattern representing all or a subset of Targeted MBE ecore NsURIs

THALES

Phase 2. Binding MBE resources

Step 2. Diagram filter

Implementation

Plugin name	org.polarsys.kitalpha.ad.viewpoint.dsl.as.diagram.helper
Java Package	org.polarsys.kitalpha.ad.viewpoint.dsl.as.diagram.helper.extension
Class name	ExtensionManager

Extension point(s)

Name	**Plugin**	**Schema**
diagramfilter	org.polarsys.kitalpha.ad.viewpoint.dsl.as.diagram.helper	diagramfilter.exsd

Extension(s)

Description	Contributions must define a **Filter** element that allow to bind a Target Application with the targeted MBE model file extension. Many filters can be defined.	
Filter	**Target Application**	Selecting the concerned Target Application
	Model File Extension	Providing the model file extension

THALES

Phase 2. Binding MBE resources

Step 3. Diagram expression language

Implementation

Plugin name	org.polarsys.kitalpha.ad.viewpoint.dsl.as.diagram.expression.helper
Java Package	org.polarsys.kitalpha.ad.viewpoint.dsl.as.diagram.expression.helper.extension
Class name	ExtensionManager

Extension point(s)

Name	**Plugin**	**Schema**
diagramExpressionLanguage	org.polarsys.kitalpha.ad.viewpoint.dsl.as.diagram.expression.helper	diagramExpressionLanguage.exsd

Extension(s)

Description	Contributions must define an **Expression** element that allow to bind a Target Application with the expression language supported by the targeted MBE.	
Filter	**Target Application**	Selecting the concerned Target Application
	Language	Choosing one of the following languages: - Acceleo 3 - Query Legacy

Phase 3. Declaring MBE technical specificities

Step 1. Case of Capella Target application and derived

Implementation

Plugin name	org.polarsys.kitalpha.ad.viewpoint.dsl.generation.ui
Java Package	org.polarsys.kitalpha.ad.viewpoint.dsl.generation.ui.extensions
Class name	UIPropertiesExtensionManager

Extension point(s)

Name	**Plugin**	**Schema**
UIPropertiesContributors	org.polarsys.kitalpha.ad.viewpoint.dsl.generation.ui	UIPropertiesContributors.exsd

Extension(s)

Description	Contributions provide properties contributors information to use in property view generation. A PropertyContributor is defined by extension point org.eclipse.properties.tabbed.propertyContributor	
PropertiesContributors	**Target Application**	Selecting the concerned Target Application
Contributor	**Id**	Providing a Contributor ID
	Name	Providing a name of this contributor
	Property tab category	Providing a Property Category.

THALES

Phase 3. Declaring MBE technical specificities

Step 2. Customs generations (1/2)

Implementation

Plugin name	org.polarsys.kitalpha.ad.viewpoint.dsl.generation.core
Java Package	org.polarsys.kitalpha.ad.viewpoint.dsl.generation.core.extension
Class name	IExtensionFilter, AbstractAspectExtensionFilter

Extension point(s)

Name	**Plugin**	**Schema**
ExtensionLauncher	org.polarsys.kitalpha.ad.viewpoint.dsl.generation.core	ExtensionLauncher.exsd

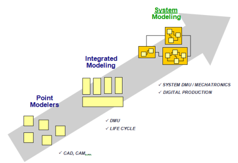
Phase 3. Declaring MBE technical specificities

Step 2. Customs generations (1/2)

Extension(s)

Description	Contributions provide a generation that is available only for one Target Application by: 1. Declaring the generation as un extension to the Viewpoint Core generation (using **Launcher** element) 2. Binding the generation with an existing Target Application (using a **Filter** element)	
Launcher	**Name**	Setting a name for the generation, it will be displayed in the “Generation with selection” wizard.
	Fcore	Referencing the EGF activity that represent the root element of the generation. The Fcore attribute value must follows the following format: platform:/plugin/[Plugin ID]/[Folder]/[File name].fcore#[Activity ID]
	Category	Generation categories are used to differentiate generations. Two categories are defined: **1. Aspect**: generations are sensitive to the viewpoint specification. In general, they are related to a viewpoint Aspect. The result of these generations vary according to viewpoints. **2. Addition**: generations aren’t sensitive to the viewpoint specification. In the most cases, this kind of generation can uses some viewpoint data like configuration attributes or viewpoint name/short name but not more. The result doesn’t vary from a viewpoint to another.
	Description	Setting a short description for a generation, it will be displayed in the “Generation with selection” wizard.
Filter	**Target Application**	Selecting the Target Application with which the generation is associated. The current generation will not be visible for the other generations.
	Java	Providing a Java implementation of a filter that activate/deactivate a generation depending on the generation context. A Java filter must implements IExtensionFilter java interface (see previous page). It can inherits from AbstractAspectExtensionFilter Java abstract class (see previous page).

THALES



CRYSTAL

Clarity

Kitalpha is supported by **Sys2Soft**, **Crystal**, and **Clarity**, French and European projects

Kitalpha

http://polarsys.org/kitalpha/

THALES