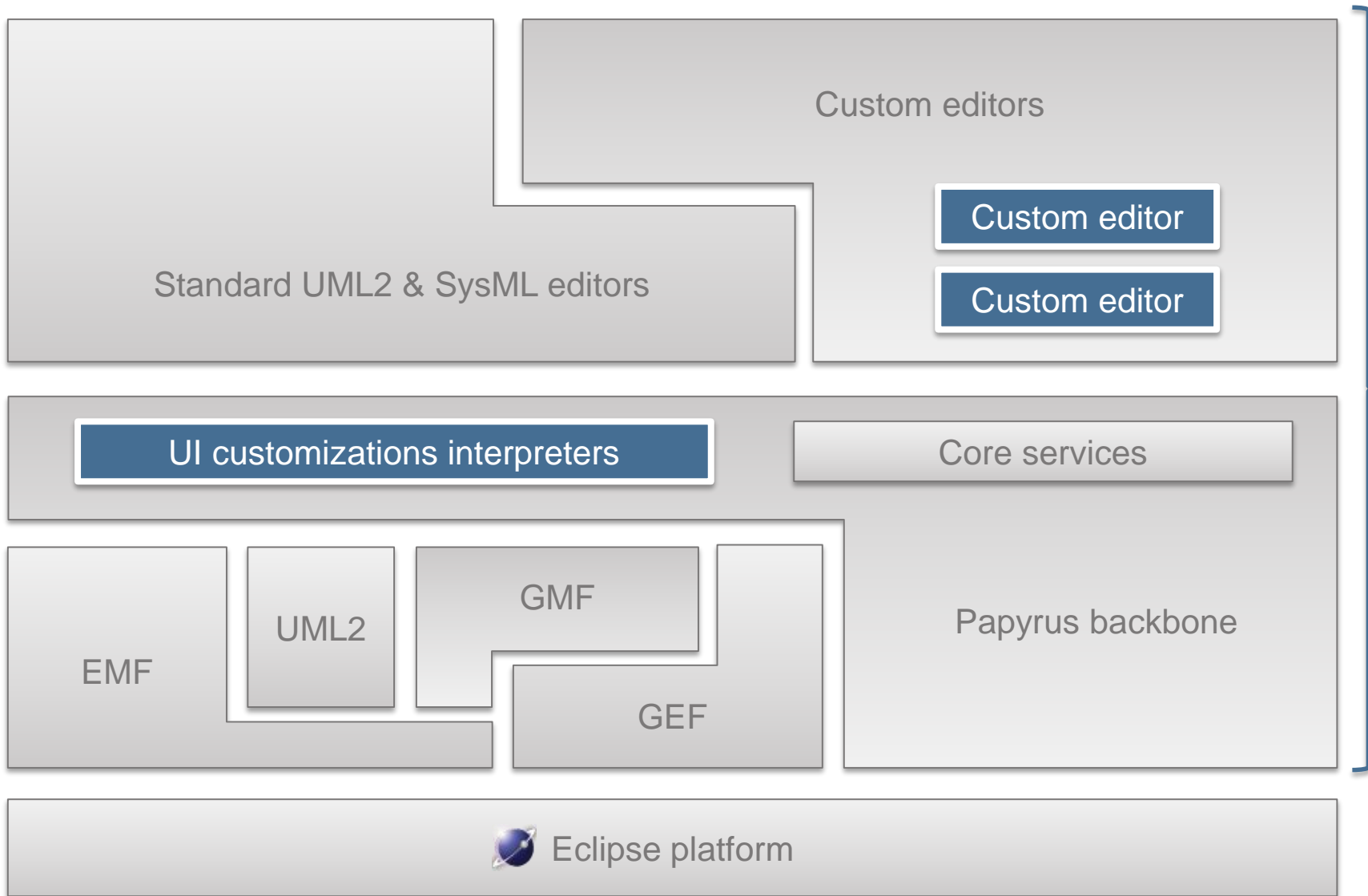


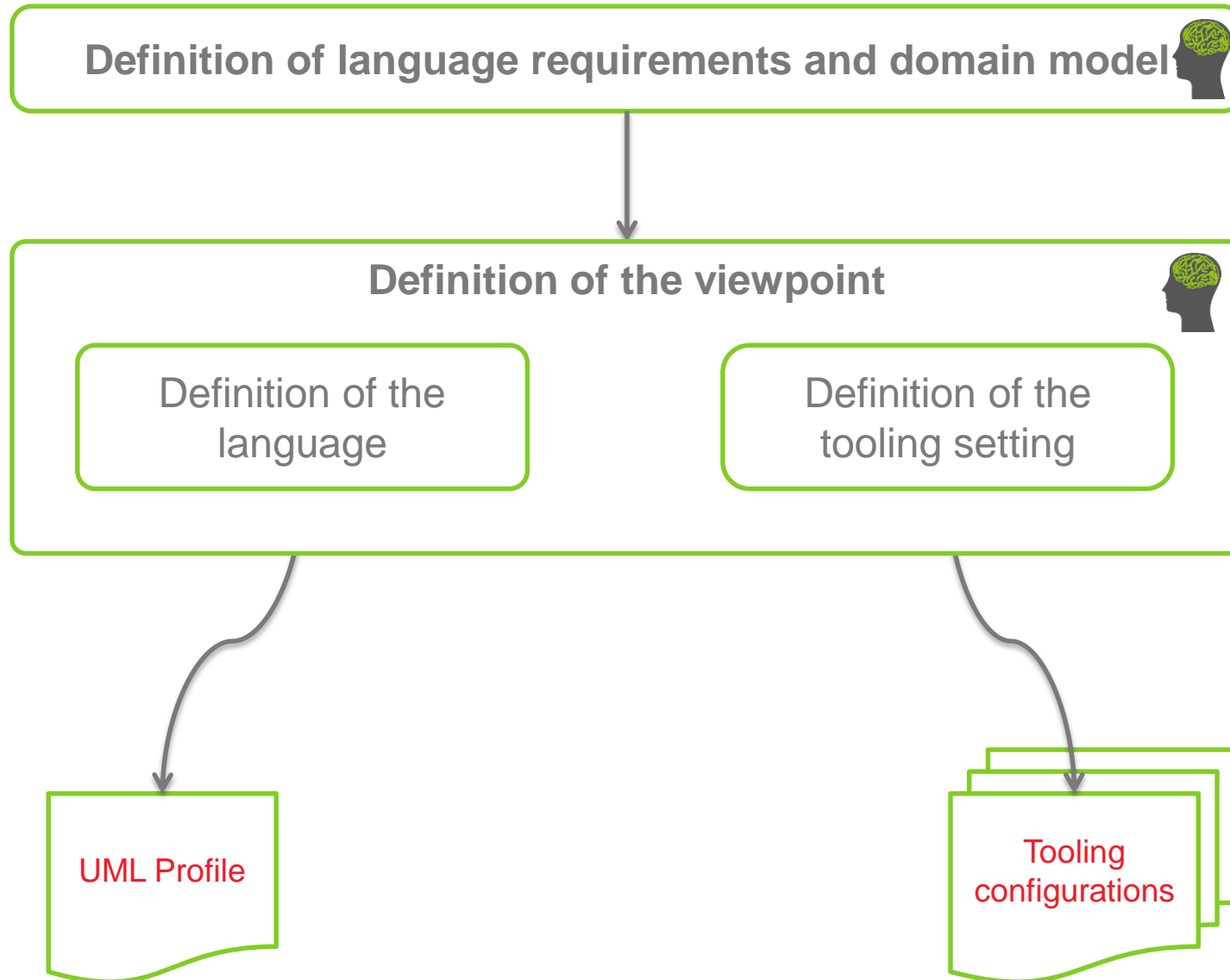
# ADVANCED DEFINITION AND DESIGN OF DSML WITHIN PAPYRUS



**list**

Papyrus Training | Papyrus Team





- **Definition of language requirements and domain model**
- **Definition of the language**
- **Definition of the tooling setting**
  - Palette
  - Properties views
  - Model explorer (tree-based view)
  - Dedicated creation menus
  - Customize Diagram Notation - Symbols and Styles
  - Textual notation
  - Tabular notation
  - Validation of abstract syntax
  - Viewpoints
- **Synthesis of customization facilities in Papyrus**

# DEFINITION OF LANGUAGE REQUIREMENTS AND DOMAIN MODEL

Definition of language requirements  
and domain model 



Definition of the viewpoint 

Definition of the language

Definition of the tooling setting

SysML Requirement for authoring the user's requirements of SimplifiedReqML.

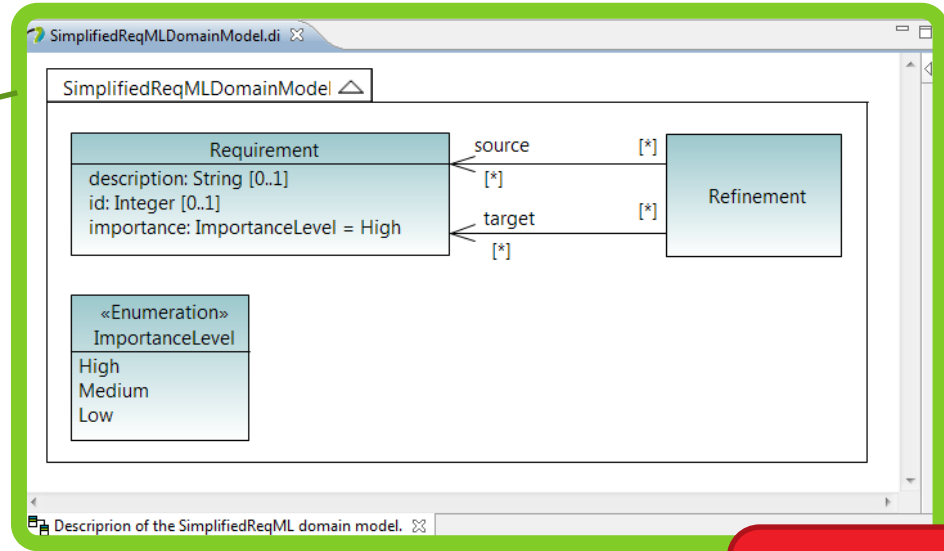
The screenshot shows a SysML diagram editor window titled 'SimplifiedReqMLRequirements.di'. It contains five requirement elements, each with a header and a body. The requirements are:

- SimplifiedReqML\_Req1:** «Requirement» text=SimplifiedReqML shall enable to author requirements diagrams. id=SRML\_Req1
- SimplifiedReqML\_Req2:** «Requirement» text=SimplifiedRqML shall enable to attach a description to a requirement. id=SRML\_Req2
- SimplifiedReqML\_Req3:** «Requirement» text=SimplifiedRqML shall enable to attach an importance (High, Medium, or Low) to a requirement. A requirement must have an importance specification, and its default value is High. id=SRML\_Req3
- SimplifiedReqML\_Req4:** «Requirement» text=SimplifiedRqML shall enable to attach an interger id to a requirement. id=SRML\_Req4
- SimplifiedReqML\_Req5:** «Requirement» text=SimplifiedRqML shall enable to define a requirement as a refinement of at least one an other requirement. id=SRML\_Req5

The diagram editor interface includes a title bar, a toolbar, and a status bar at the bottom showing 'Requirements Description of the SimplifiedReqML'.

Domain model of SimplifiedReqML

Requirements traceability table



All requirements expressed are satisfied

Domain concept	Satisfied Req. Id
Requirement	SRML_req1: « SimplifiedReqML shall enable to author requirements diagrams »
Requirement::description	SRML_req2: « SimplifiedReqML shall enable to attach a description to a requirement. »
Requirement::id	SRML_req4: « SimplifiedReqML shall enable to attach an interger id to a requirement. »
Requirement::importance	SRML_req3: « SimplifiedReqML shall enable to attach an importance (High, Medium, or Low) to a requirement. A requirement must have an importance specification, and its default value is High. »
ImportanceLevel	SRML_req3: « SimplifiedReqML shall enable to attach an importance (High, Medium, or Low) to a requirement. A requirement must have an importance specification, and its default value is High. »
Refinement	SRML_req5: « SimplifiedReqML shall enable to define a requirement as a refinement of at least one an other requirement. »

# DEFINITION OF THE LANGUAGE

Definition of language requirements  
and domain model 



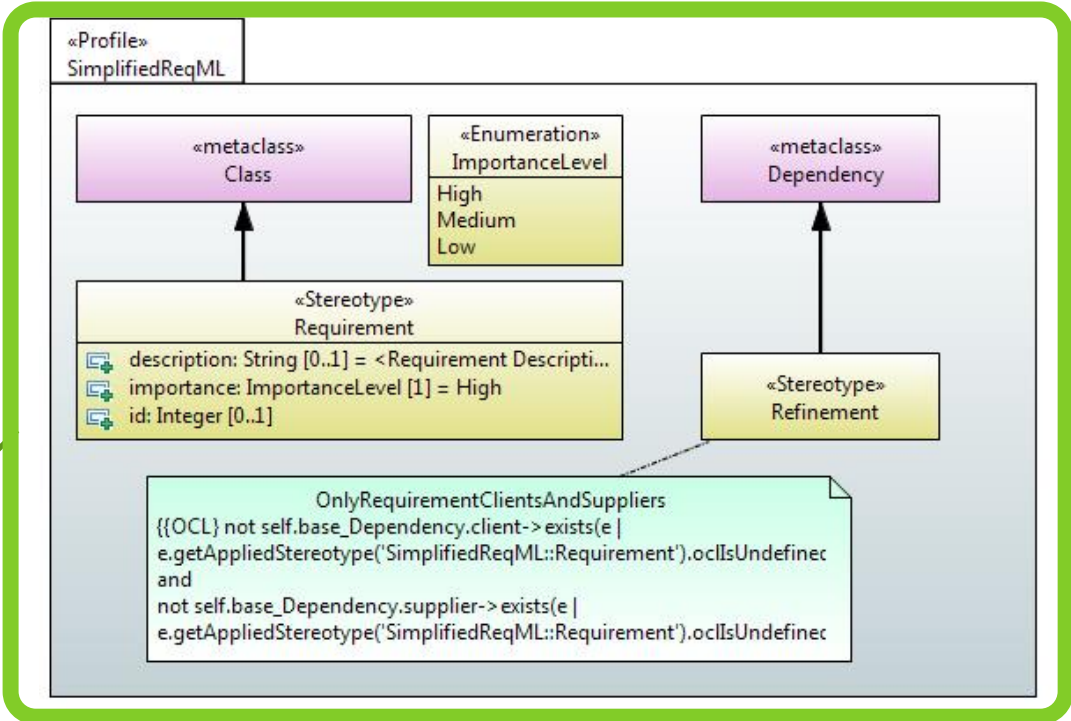
**Definition of the viewpoint** 

Definition of the language

Definition of the tooling setting



**UML Profile for SimplifiedReqML**



**Domain model to Profile traceability table**

Profile element	Domain concept
Requirement	Requirement
Requirement::description	Requirement::description
Requirement::id	Requirement::id
Requirement::importance	Requirement::importance
ImportanceLevel	ImportanceLevel
Refinement	Refinement

- **Define your domain specific syntax with the standard UML profile mechanism**
  - Stereotype definition
  - OCL in UML profiles
- **Define Profile to make them usable in Eclipse**
  - Handle multiple versions of your profiles

# *Demo*

# DEFINITION OF THE TOOLING SETTING

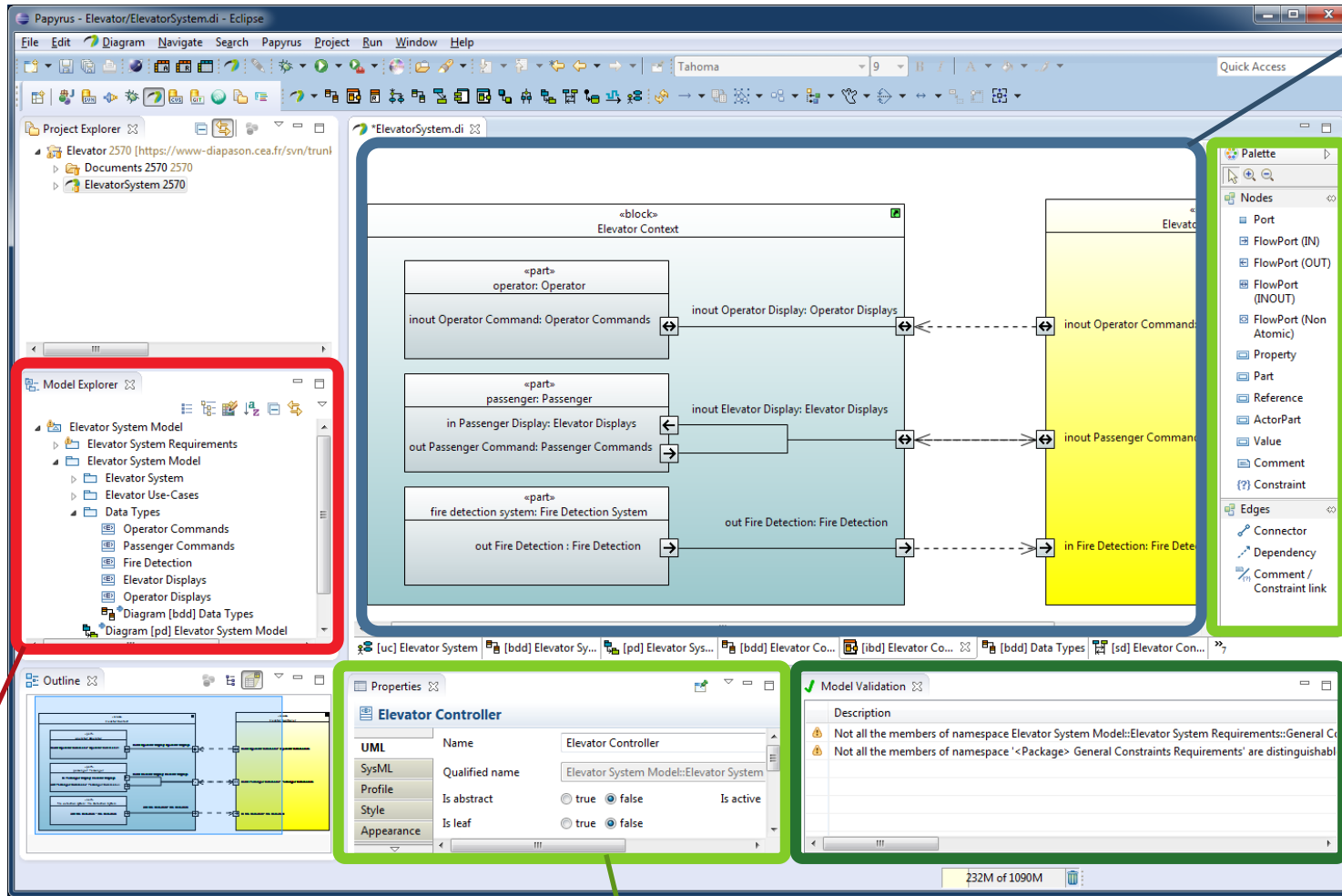
Definition of language requirements  
and domain model 



**Definition of the viewpoint** 

Definition of the language

Definition of the tooling setting



**Notation & Style**

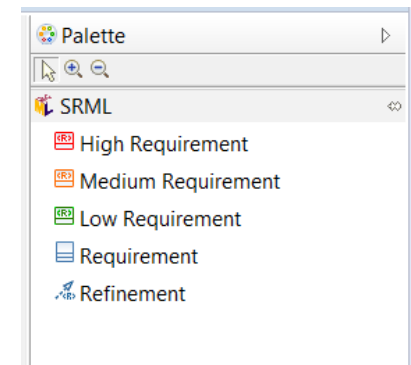
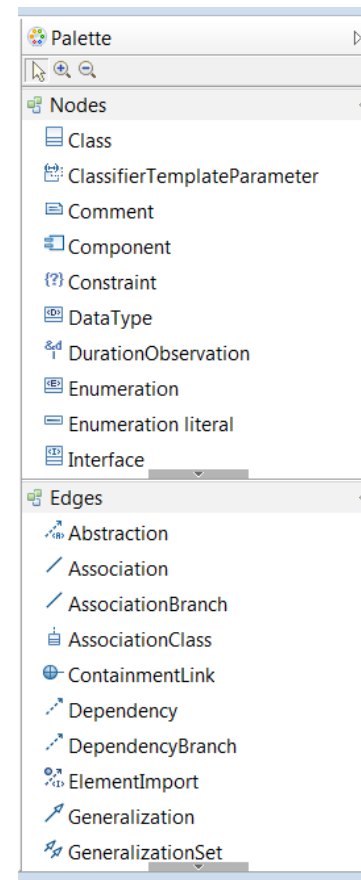
**Palette**

**Specific syntax validation**

**Model explorer**

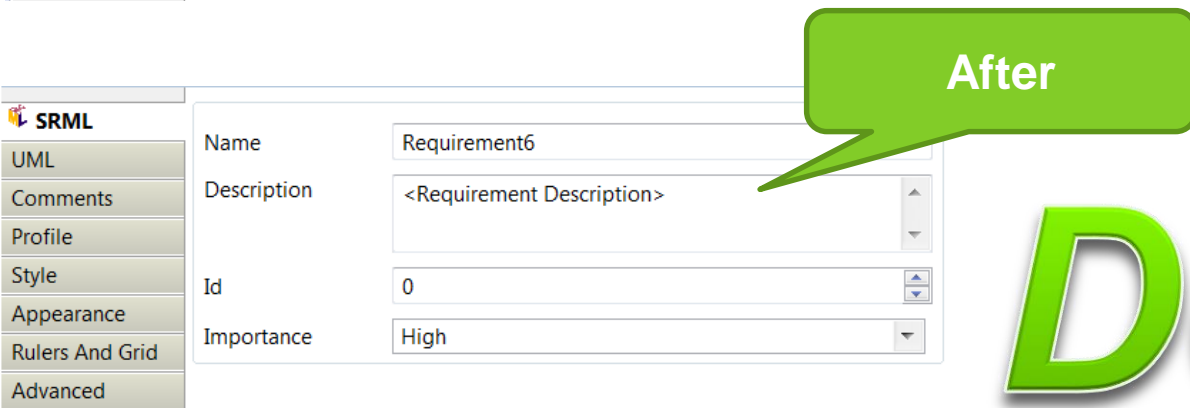
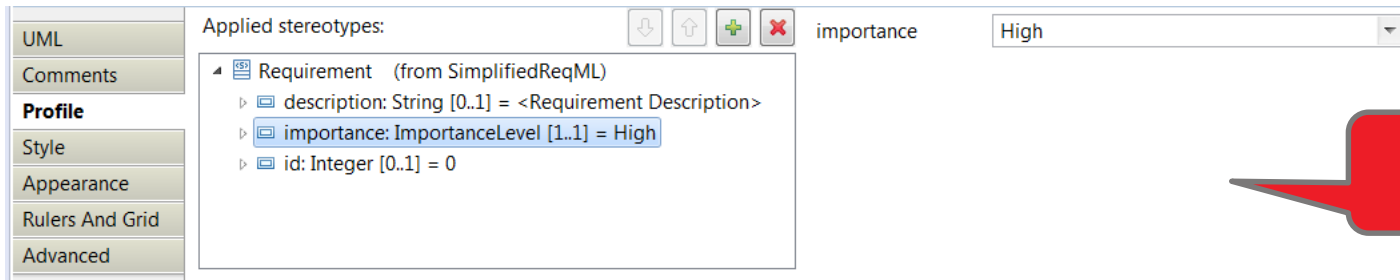
**Property view**

- Define specific palette to create element with one click
- Element creation can be arbitrarily complex:
  - Create element, set initial properties' values, apply stereotypes, set appearance...
- **Model-based definition of views**
  - Live-interpretation
  - Zero lines of code



**Demo**

- Define specific properties views to display & edit model element attributes
- Independent from represented data
- Model-based definition of views
  - Live-interpretation and preview
  - Zero lines of code



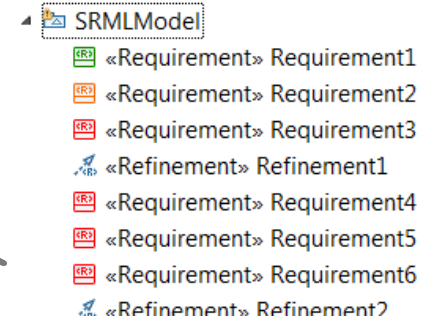
**Demo**

## Tree-based visualization of the Model

### Structure customization

- Hide details
- Simplify the structure
- Add shortcuts (“Graph-oriented” representation)
- Add custom content (e.g. Related files)
- Add groups (“Folders”)

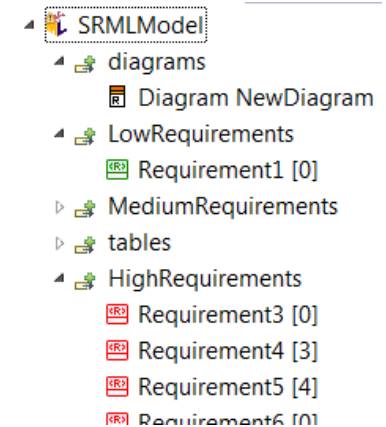
Before



### Appearance customization

- Customize image, overlay
- Customize label

After



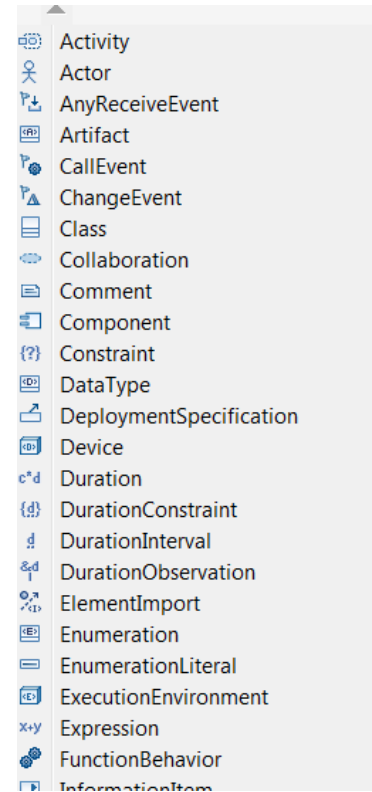
### Model-based definition of views

- Live-interpretation and preview
- Zero lines of code

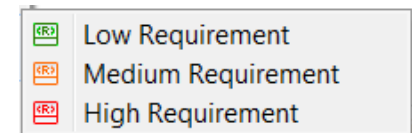
# Demo

- Add custom menus to Papyrus to ease the creation of your domain specific elements
- Model-based definition of new child menus
  - Live-interpretation
  - Zero lines of code

Before



After

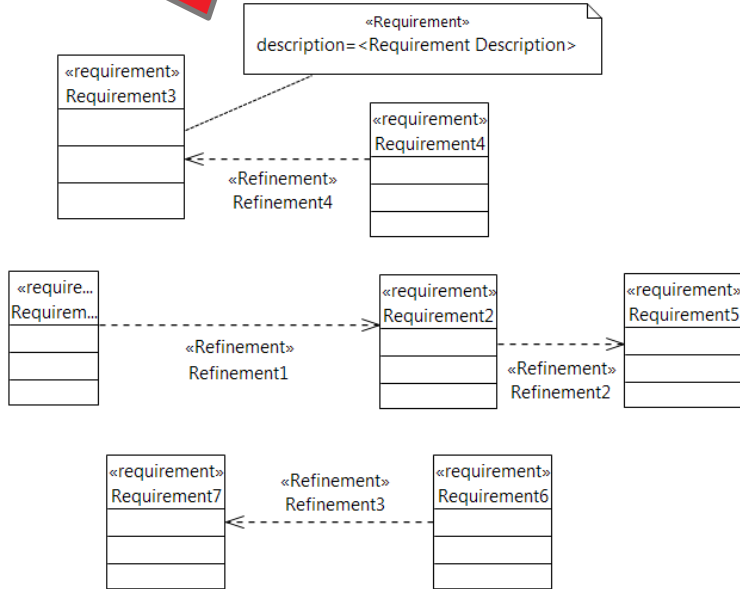


**Demo**

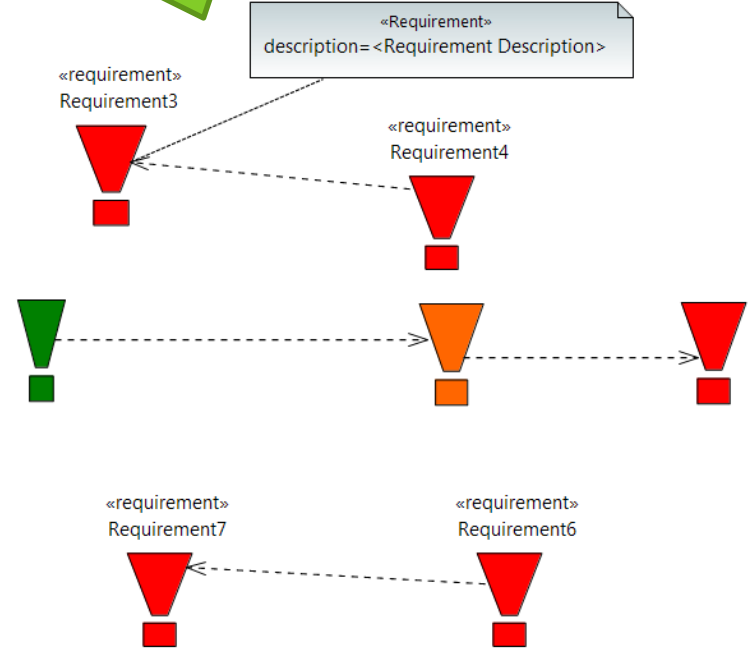


- **Standard UML notation may be inadequate to a specific domain/concern and quote notation to mark specific concepts in UML (e.g. «Block») is not always sufficient**
  - Replace the entire notation of element with a custom SVG
  - Decorate standard UML notation
  - Links Follow borders of the shape.
- **Styles can be used to have a uniform appearance**
- **Styles can be used to define notation for specific semantics**
  - Works also on SVG
- **Model-based definition of symbols and styles**
  - Live-interpretation
  - Zero lines of code

**Before**




**After**



*Demo*









- Define custom or domain specific textual notation
  - Xtext-based text editor
- Embed the textual editor for direct edition
- Embed the textual editor in the properties view
- Embed the textual editor in the Model Explorer

4  LowRequirements

```
 id = 0 description = "<Requirement Description>" importance = Low
```

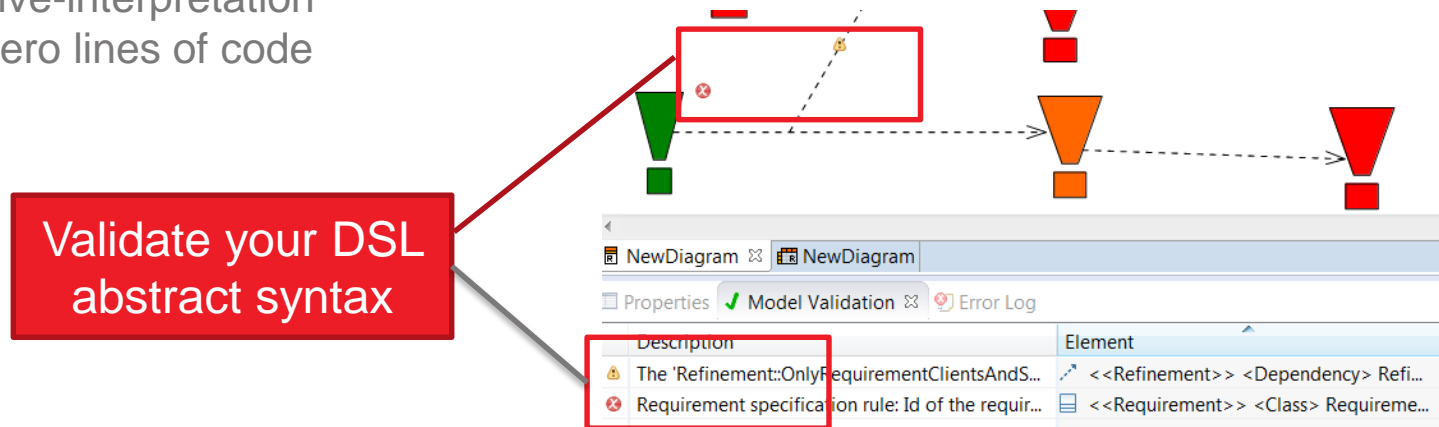
# Demo

- Copy and Paste from MS Excel
- Export to .xls
- Model-based definition of tabular notation
  - Live-interpretation
  - Zero lines of code (except for synchronized table)

	◦ id : Integer	◦ name : String	◦ importance : ImportanceLevel	◦ description : String
 Requirement1	0	Requirement1	Low	<Requirement Description>
 Requirement2	1	Requirement2	Medium	<Requirement Description>
 Requirement4	3	Requirement4	High	<Requirement Description>
 Requirement5	4	Requirement5	High	<Requirement Description>
 Requirement6	0	Requirement6	High	<Requirement Description>
 Requirement7	0	<i>Requirement7</i>	High	<Requirement Description>
 Requirement3	0	Requirement3	High	<Requirement Description>
 Requirement8	0	Requirement8	High	<Requirement Description>

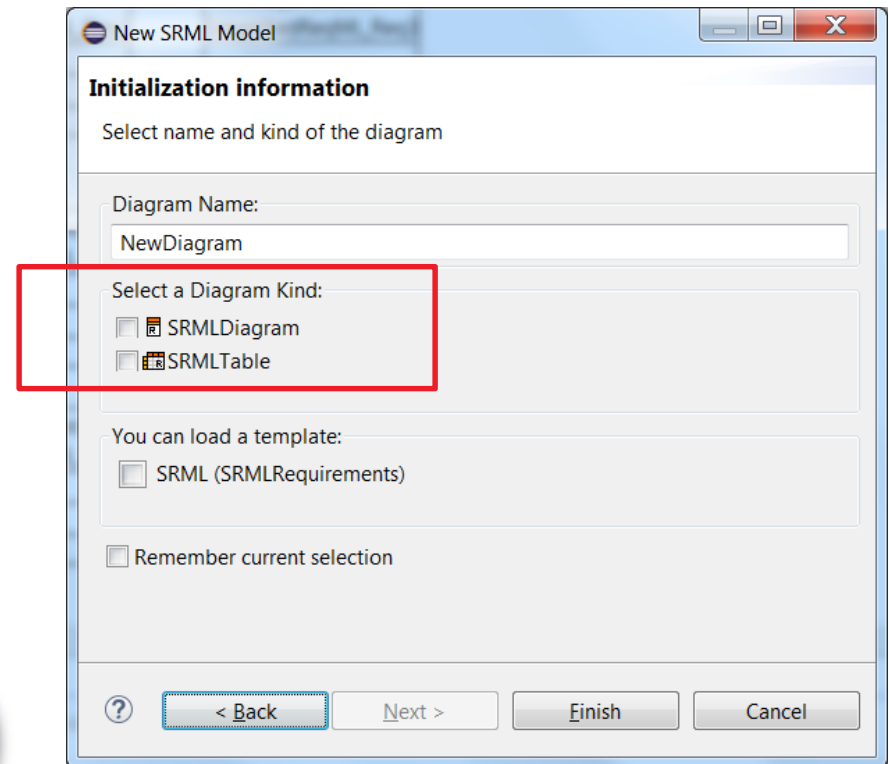
# Demo

- Do you remember the OCL constraint in the UML profile?
  - You can adapt UML abstract syntax for your specific domain
- Relies on Papyrus Validation framework
- Model-based definition of domain specific abstract syntax
  - Live-interpretation
  - Zero lines of code



# Demo

- **Specialization of the user experiences by constraining what can be seen and interacted with in models through views.**
  - Define ISO42010 compliant viewpoints: stakeholders, views, viewpoints...
- **Aggregate the various Papyrus customizations**
- **Model-based definition of viewpoints**
  - Live-interpretation
  - Zero lines of code



**Demo**

**SYNTHESIS OF  
CUSTOMIZATION FACILITIES  
IN PAPYRUS**

## Define a viewpoint

### Define the abstract syntax

Define syntactic constraints (OCL or java)

Define/provide interaction means between models and users.

### Define an editor tailored to the concrete syntax

#### Define a graphical editor from scratch

Define properties views

Define a style

Define a palette

Define creation menus

Define a tree-based visualization

Define a wizard

#### Define a tabular editor

Define properties views

#### Define a textual editor

### Override an existing Papyrus diagram

Define properties views

Define a style

Define a palette

Define creation menus

### Override the UML notation with Domain Specific Notation

Define properties views

Define a style

Define a palette

Define creation menus



# THANK YOU FOR YOUR ATTENTION

---

Commissariat à l'énergie atomique et aux énergies alternatives  
Centre de Saclay | 91191 Gif-sur-Yvette Cedex

DRT/LIST  
DILS  
LISE