CESAR - Cost-**e**fficient methods and processes for **sa**fety **r**elevant embedded systems

CESAR – Papyrus MDT Training
Sept 12th, 2011

Papyrus MDT : Advances on Papyrus technology

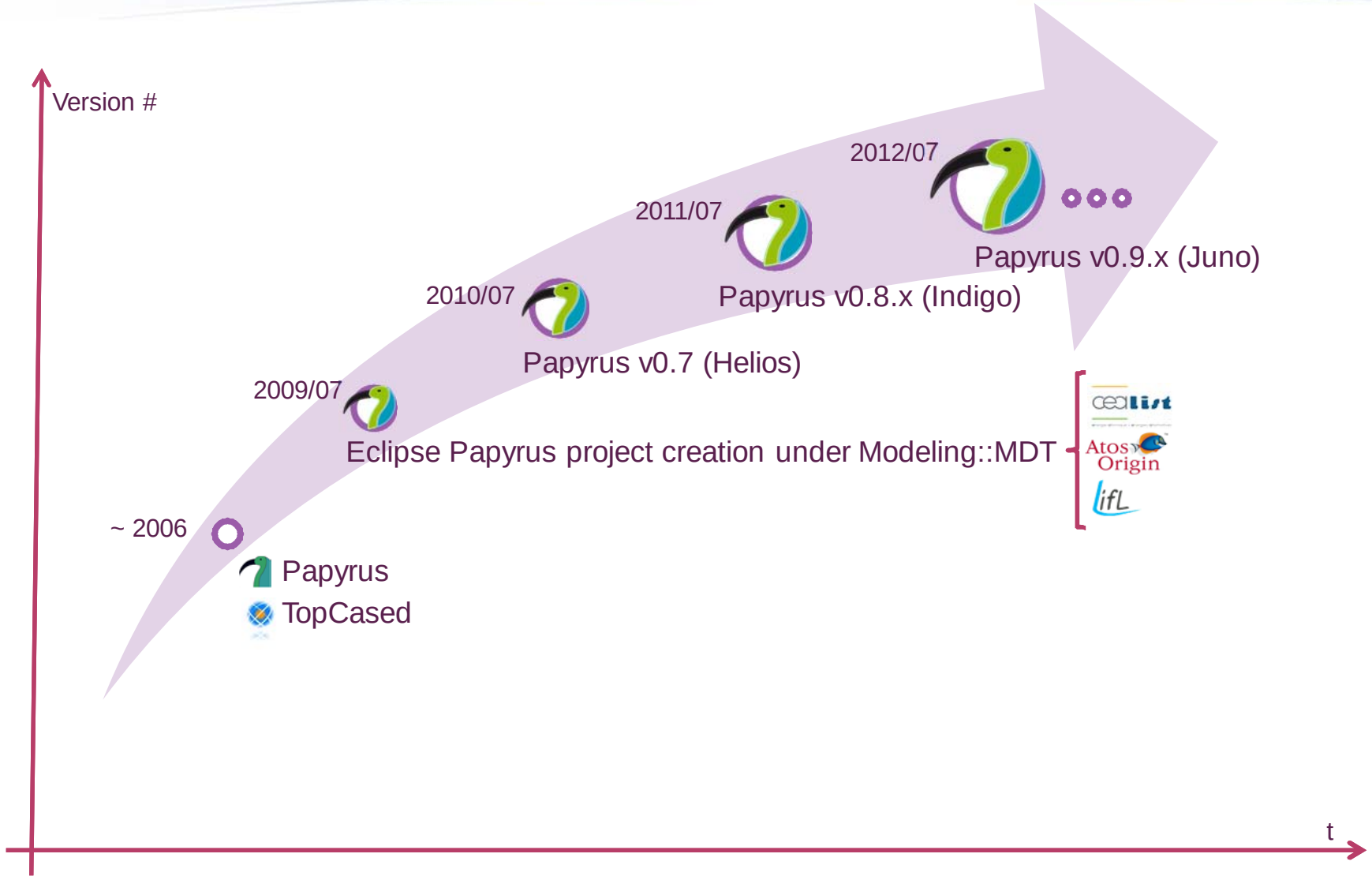
Sébastien Gérard, CEA LIST/LISE, Sebastien.Gerard@cea.fr

Francois-Xavier Dormoy, Esterel Technologies, Francois-Xavier.Dormoy@esterel-technologies.com

Agenda

1. MDT Papyrus versus Papyrus 1.x
2. MDT Papyrus for DSML
3. MDT Papyrus support of MARTE and scheduling analysis
4. MDT Papyrus integration in SCADE
5. Q&A.

Papyrus History



Papyrus 1.x versus new Papyrus

- **Papyrus v1.x**
  - CEA initiative started in
    - In collaboration with Cedric Dumoulin from LIFL
  - Scope: UML modeler and DSML based on UML
    - Includes also both SysML and MARTE standards.
  - Technology
    - Handmade Java-programming based on both GEF and EMF frameworks
- **Papyrus new generation**
  - New Papyrus is an official Eclipse project for Modelling::MDT
    - www.eclipse.org/papyrus
  - Why version number = 0.x ?
    - Due to Eclipse rules: 0.7.x = usual first version # for incubation project
  - Scope: UML2, SysML and DSML
    - Details of initial proposal here:
  - Technology
    - Model transformation / code generation and handmade Java programming based on GMF framework

Papyrus Committers & Supporters

- **CEA LIST**
  - Arnaud Cuccuru, Sebastien Gerard, Camille Letavernier, Vincent Lorenzo, Ansgar Radermacher, Rémi Schneckenburger, David Servat, Yann Tanguy and Patrick Tessier.
- **ATOS**
  - Raphael Faudou, Tristan FAURE, Vincent Hemery, Thibault Landre, Emilien Perico and Mathieu Velten.
- **LIFL**
  - Cedric Dumoulin.

- **Main Current Industrial Supporters (in alphabatical order):**
  - AIRBUS, ATOS, CEA, Ericsson and Esterel Technologies (http://www.listerel.org/)

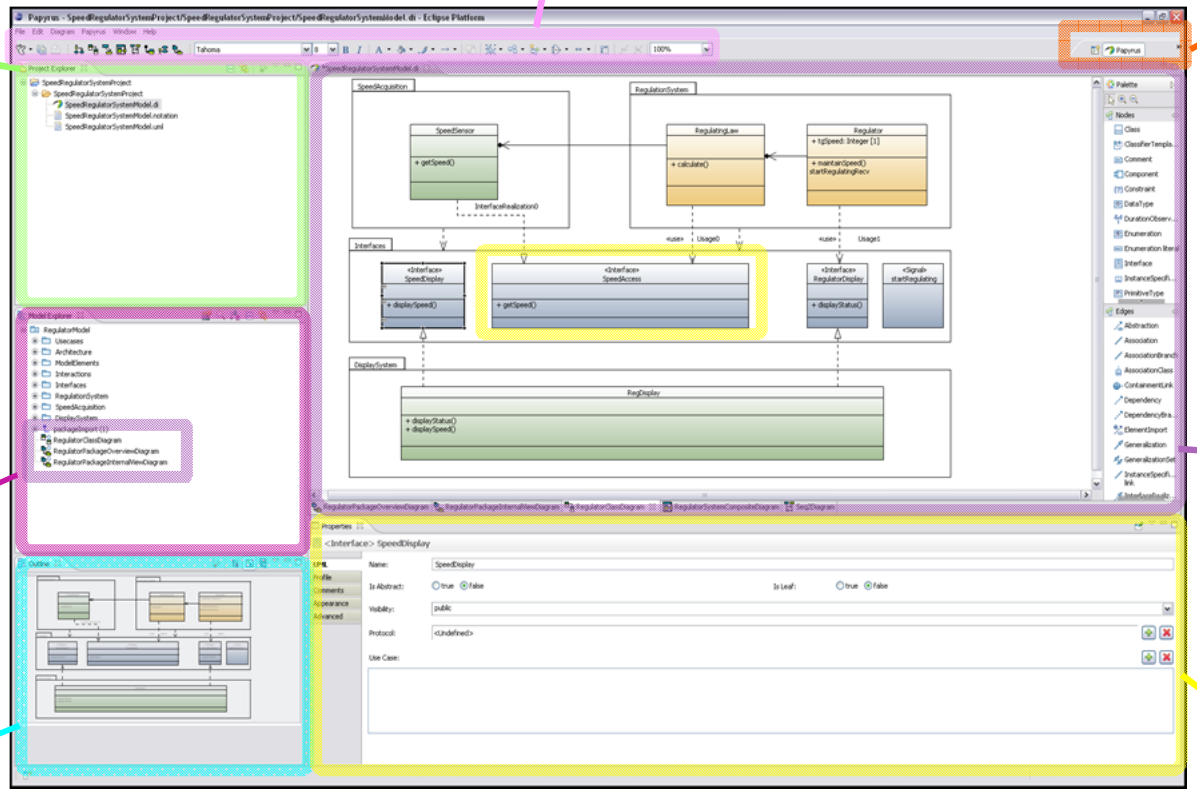
Papyrus facilities in a nutshell

- **Support for Major OMG Standard Modeling Languages**
  - UML2, SysML and MARTE.
- **Support for DSMLs**
  - Based on the UML2 extended by specific profiles
  - Supporting any specific domain notations, either graphical or textual
  - Providing powerful and easy-to-use tool customization facilities
- **Enabler for a full model-based engineering**
  - Model compare and merge
  - Team working
  - Documentation description and generation
  - Model validation

Outlines of the Papyrus perspective

Project explorer: used to manage Papyrus projects at file system level.

Main toolbar: diagram creation, graphical editing (align, distribute…), show /hyde, …



Perspective: switch the modeling context, define windows (eclipse views) arrangement, define the list of available diagrams, define the available menus and toolbars.

Model editors: model editor enabling to edit models through a given modeling language.

Property view: form-based model editor enabling to view & edit model element properties.

Outline view: provide overview of the model (read only).

Model explorer: tree-based model editor covering the whole model.

Authoring models in graphics, but also in texts and tables!

CESAR

- **Graphical-based editors**
  - UML2
    - Support for Class, Composite Structure, Deployment, Component, Use Case, Sequence, Statemachine, Activity and Profile diagrams.
  - SysML
    - Support for requirements, BDD, IBD and Parametric diagrams.
- **Textual-based editors**
  - Extension point to embed textual editors within Papyrus to edit partially a diagram.
    - E.g.: Attributes or Operation of Classes, Port of Composite Structure, State of Statemachine.
  - Framework for using the Xtext technology.
  - Miscellaneous:
    - Textual editor for ALF (“Action Language for fUML”) .
    - Textual editor for VSL in the MARTE plug-in of Papyrus.
- **Table-based Editors**
  - Based on NatTable and customizable via EMF Facet.
  - Available generic table editor allowing to show any kinds of element including its stereotypes and related properties.
  - Two customizations for SysML: Requirement and Allocation table editors.

… but modeling is also:

- **Validation**
  - Based on EMF Validation project of Eclipse
  - Integrated within Papyrus GUI
- **Compare and merge**
  - Based on EMF Compare project of Eclipse
    - Specialized for UML models (e.g., specific case of the Stereotypes applying)
  - Integrated within Papyrus GUI
- **Team working**
  - Control mode: enables to split one model into several files to be able to use versioning systems on sub-parts of one model.
- **Documentation generation**
  - Integration of the Gendoc2 component of TopCASED
  - Enable generation of ODT and DOCX documents
  - Available via the market place of Eclipse: http://marketplace.eclipse.org/
- **Code generation and model transformation**
  - All code generators and model transformation engines can be used and connected to Papyrus

Agenda

2. MDT Papyrus for DSML

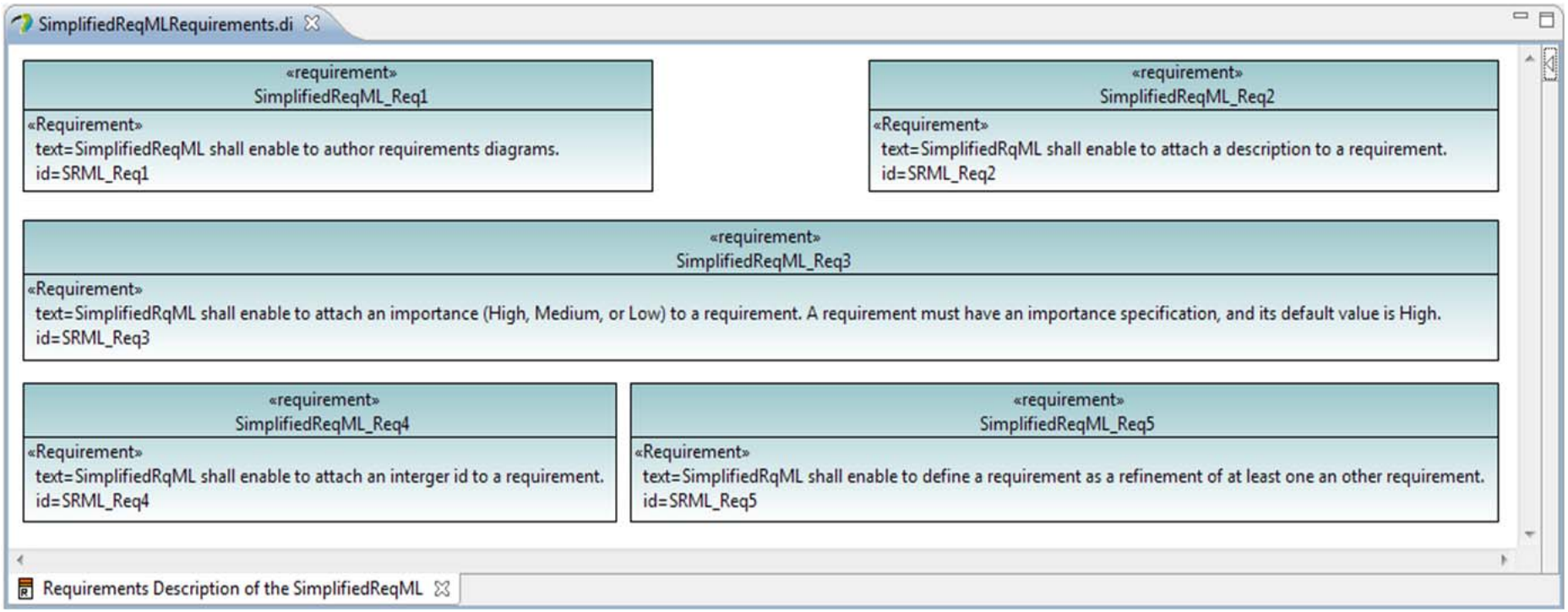
3. MDT Papyrus support of MARTE and scheduling analysis
4. MDT Papyrus integration in SCADE
5. Q&A.

About the development process of a DSML in Papyrus

- **3 iterated activities for designing the language itself**
  - Activity 1: defining the language requirements
    - Used of the SysML requirements diagrams or tables.
  - Activity 2: designing the language concepts
    - Meta-modeling of the language concepts: domain model of the language.
    - Used of a restricted class diagram conforming to construct package of the UML2 infrastructure.
  - Activity 3: profiling UML2 for the language
    - Defining the UML extensions to implement the concepts of the domain model of activity 2.
    - Used of profile diagrams.
- **A language needs a tool to be useful**
  - Option 1: based on standard UML extensions implemented within Papyrus
  - Option 2: based on customizing facilities of Papyrus

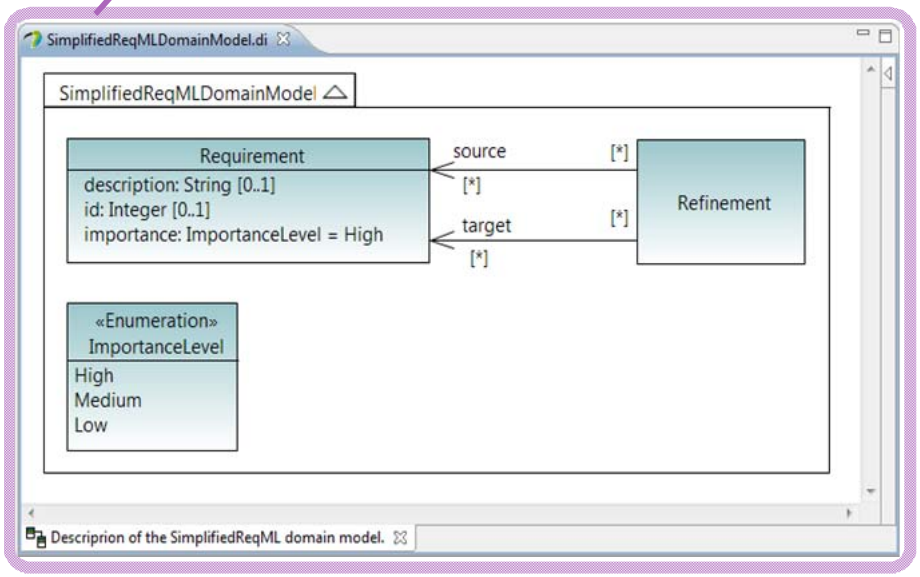
Activity 1: requirements modeling of SRML

- **Purpose: design a DSML enabling basic requirements modeling**
  - SRML ("Simplified Requirement Modeling Language")
- **Requirements modeling of SRML**



Activity 2: specification modeling of SRML

Domain model of SRML.



Requirements traceability table

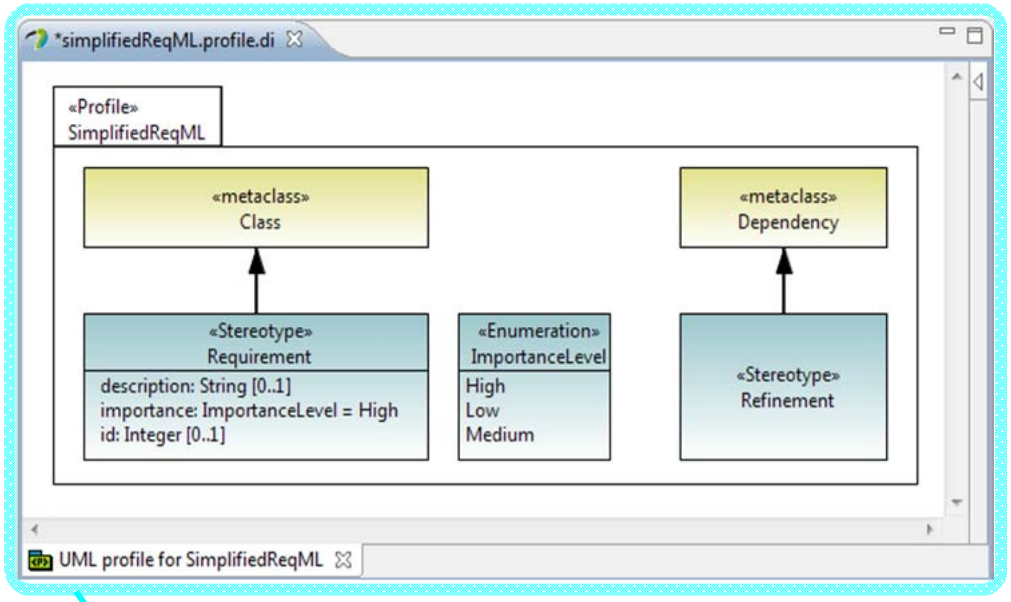
Domain concept	Satisfied Req. Id
Requirement	SRML_req1
Requirement:: description	SRML_req2
Requirement:: id	SRML_req4
Requirement:: importance	SRML_req3
ImportanceLevel	SRML_req3
Refinement	SReqML_req5

All requirements expressed in Activity 1 are satisfied!

Step3: UML profile design of SRML

- **From SimplifiedReqML domain model to UML2**
  - Choice of UML2 extensions
    - SimplifiedReqML::Requirement will extend UML::Class
    - SimplifiedReqML::Refinement will extend UML::Dependency
  - Reused of UML Class diagrams

Profile Element	Icons
Requirement.importance = High	
Requirement.importance = Medium	
Requirement.importance = Low	
Refinement	

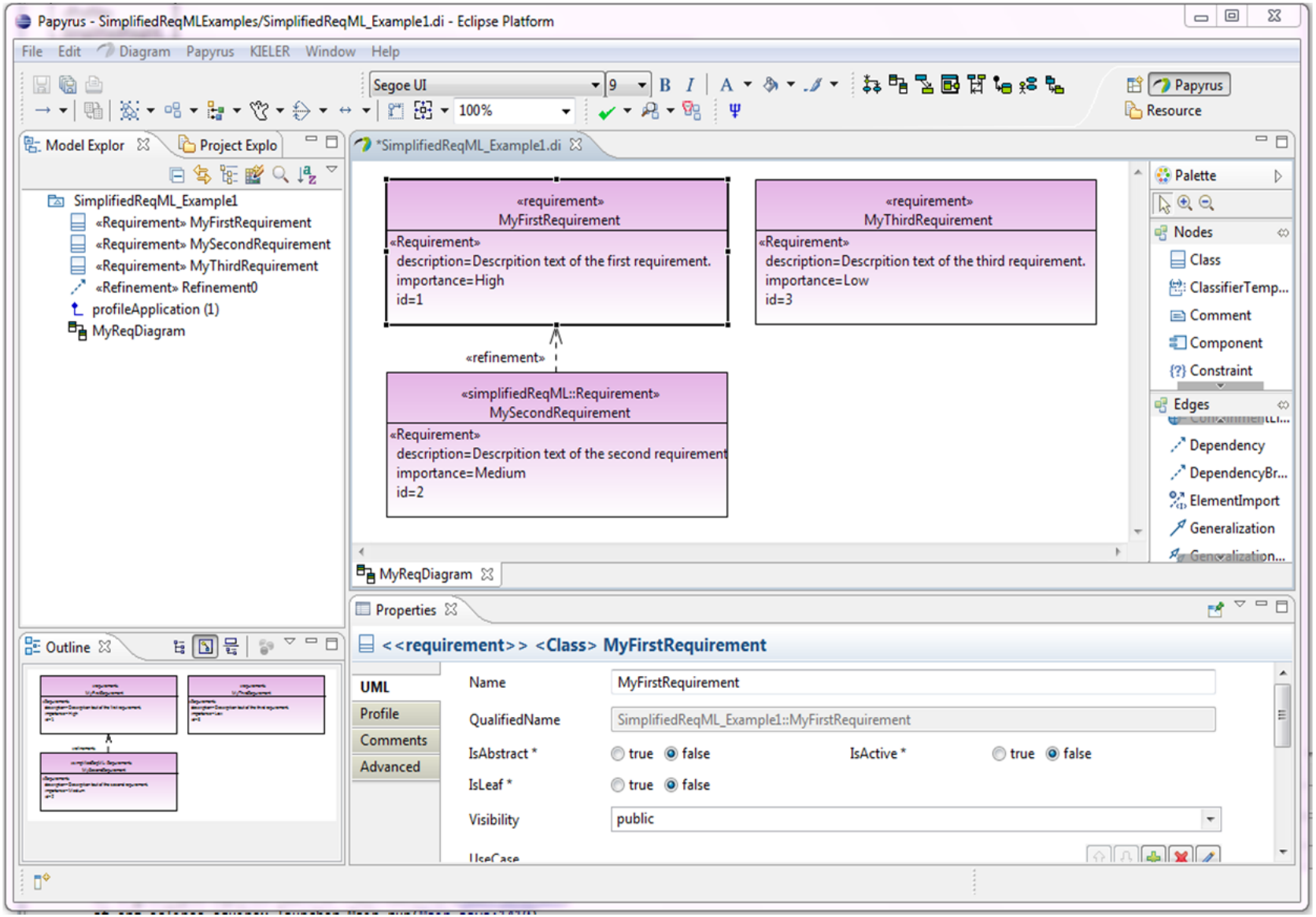


UML Profile for SimplifiedReqML.

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

Domain model to Profile traceability table

View of the SRML tool without Papyrus specific customisation

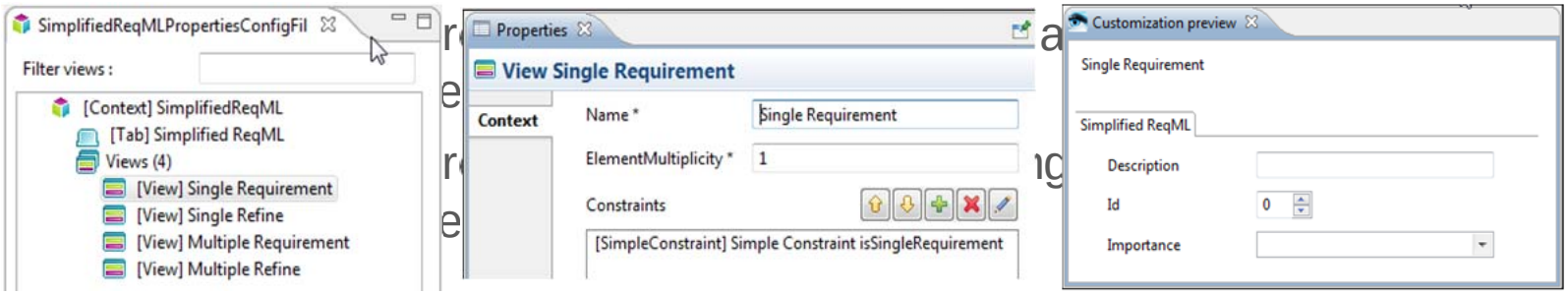


UML profile-based DSML within Papyrus

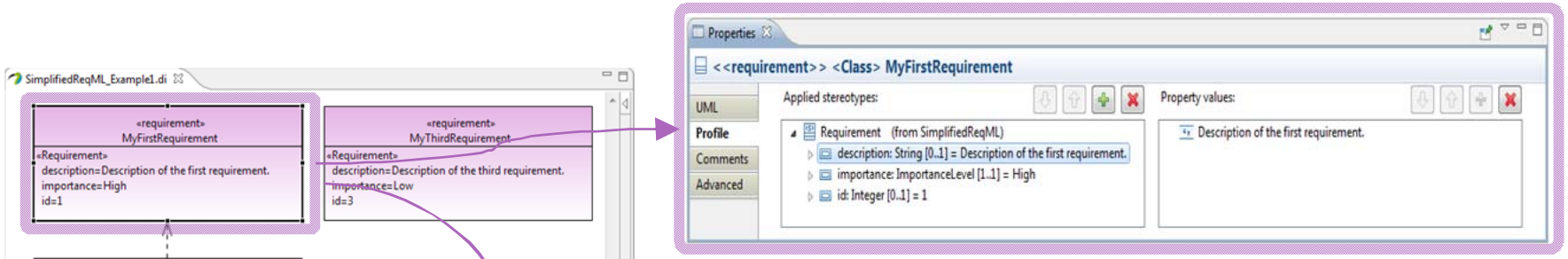
- **By-default definition of UML profile-based DSML**
  - UML profile are used to design the DSML in terms of UML extensions
    - Define the abstract syntax of the Language
    - Propose a default notation used to annotate UML model elements: «StereotypeName »
      - Include possible specific icons and shapes!
- **Advanced design of UML profile-based DSML**
  - Specific palettes
  - Specific properties editor
  - Specific model browser
  - Specific model validation rules
  - Specific editors
    - Custom Tabular Editor
    - Specific textual editor (based on Xtext)
    - Inherited Diagram Editors
    - Brand new editors (based on either EMF or GMF technologies)
  - Specific model wizzards

Specific Properties Editor

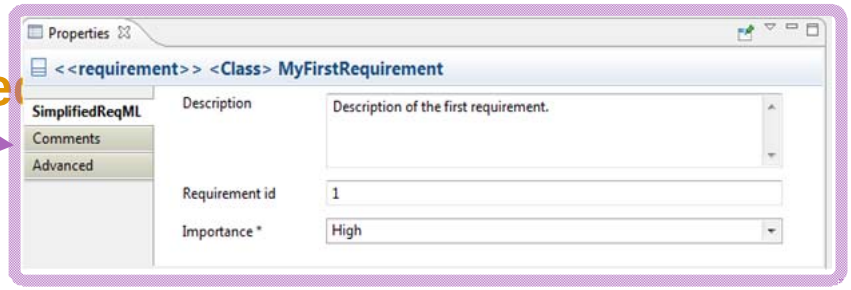
- **Papyrus properties editor is customizable enabling to show only specific properties**



- UML-like properties editors

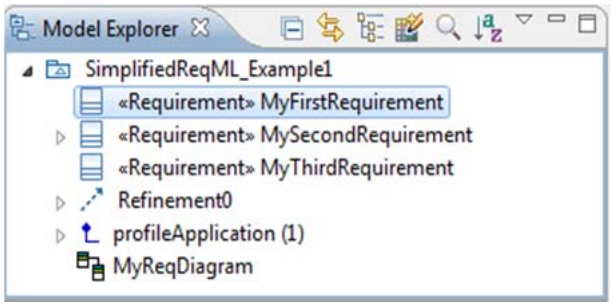


- DSML-like properties editors

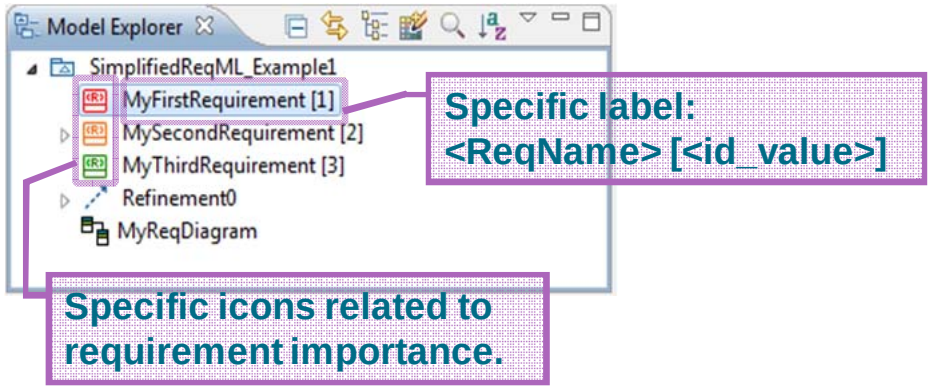


Specific Model Browser

- **Papyrus model browser is customizable enabling to show and browse model elements according to specific rules**
  - Based on the EMF Facet component of Eclipse
    - http://www.eclipse.org/modeling/emft/facet/
- **Custom feature possibilities**
  - Appearance of elements in model browser can be customized
    - Icons, content of labels, appearance of labels…
    - Customizations can be static of dynamic
    - Based on a “ui customization” file
  - Grouping of elements in model browser can be customized
    - Based on EMF facet definitions
- **Illustration on SimplifiedReqML**
  - “By default” Papyrus model browser

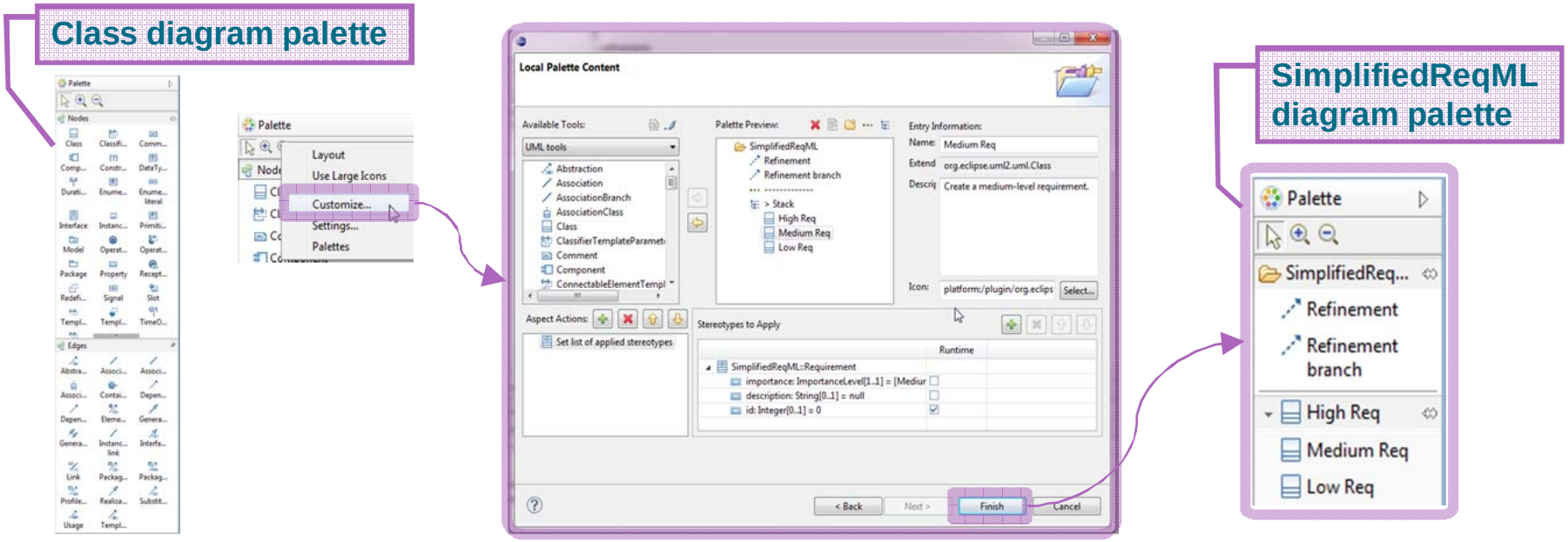


  - Customized model browser for SimplifiedReqML

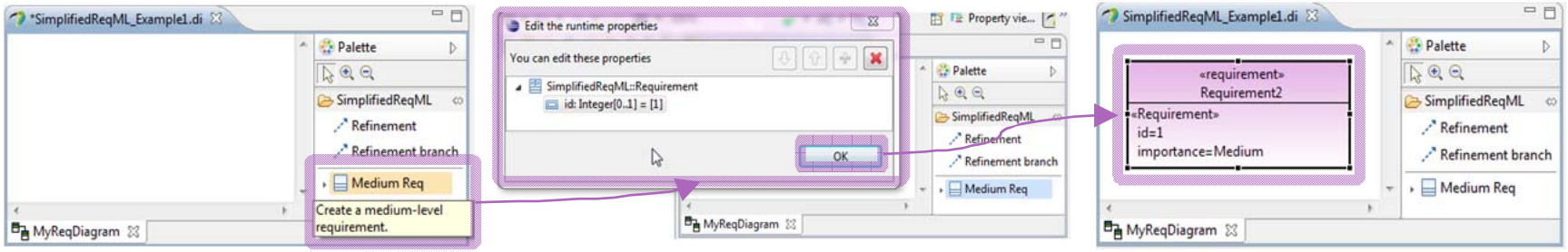


Specific Palette in Graphical Editors

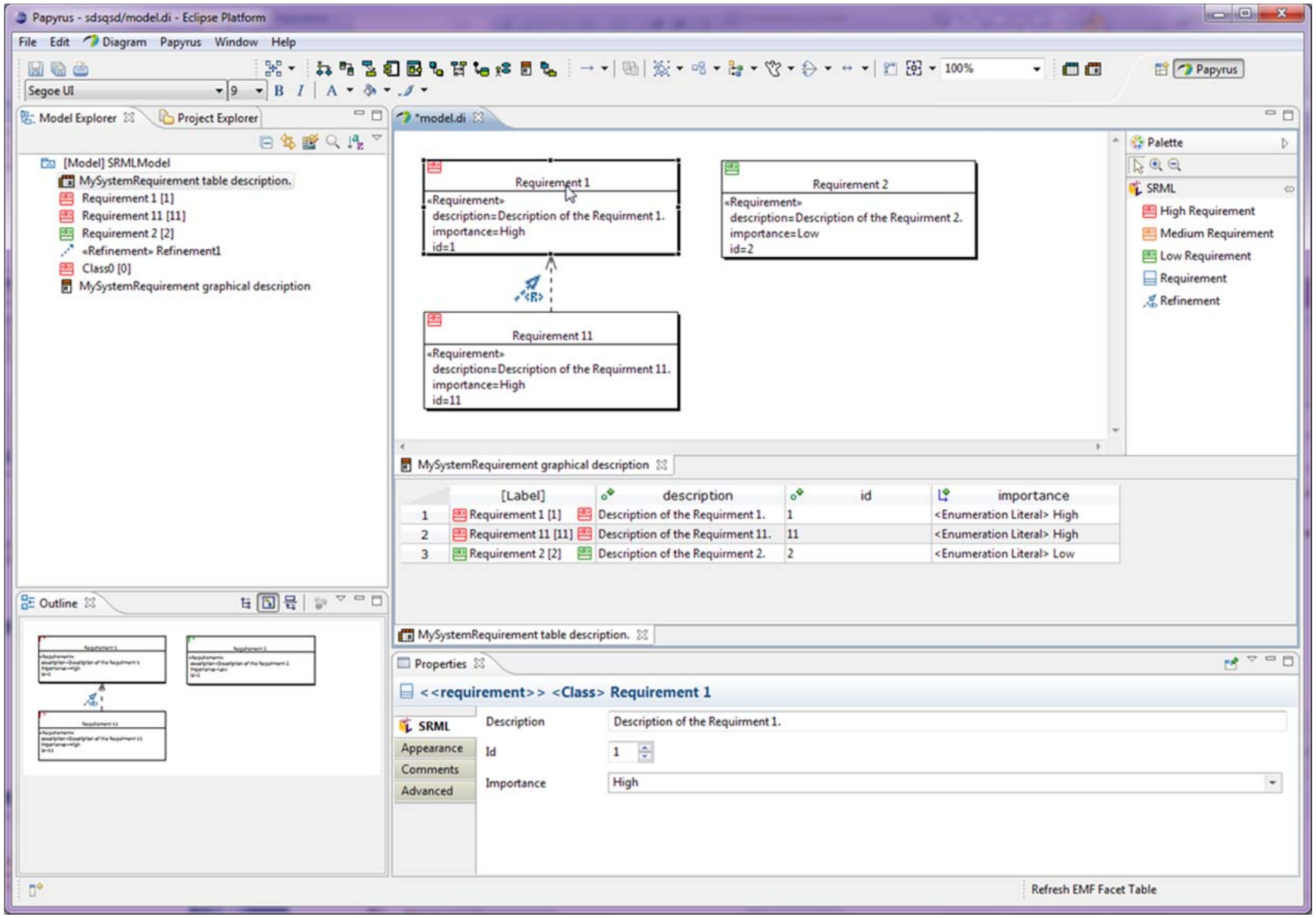
- **Papyrus enables to customize the palette of its graphical editors**
  - Provide a very simple to use editors to adapt the palette content to specific needs



- Usage illustration on SimplifiedReqML

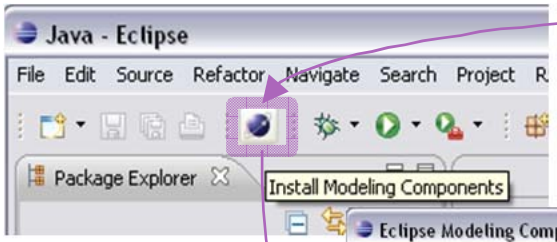


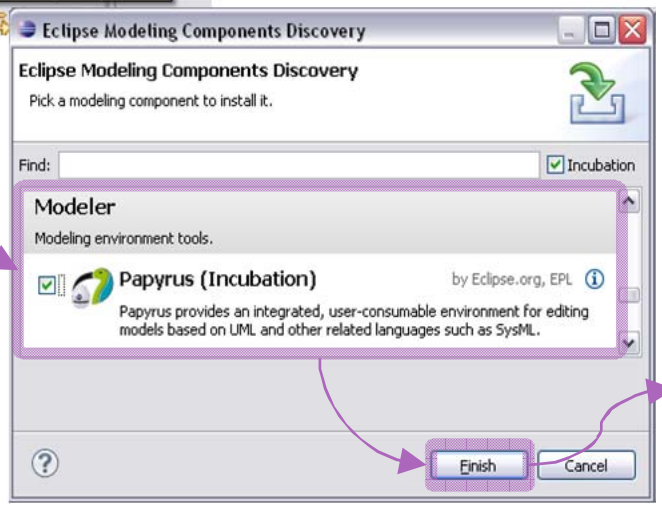
Final view of the SRML tool with Papyrus customization.

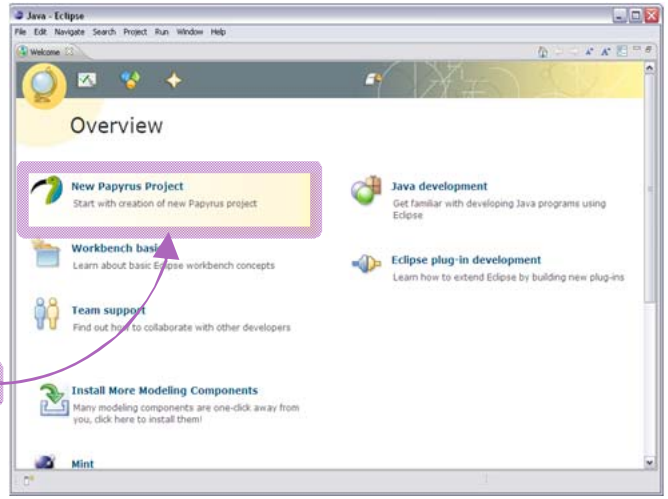


Papyrus standard download

- **Via the standard Eclipse Modeling Platform**
  - Download the Eclipse Modeling Platform
    - Helios: www.eclipse.org/downloads
    - Indigo: Indigo: http://download.eclipse.org/releases/indigo/
  - Unzip the downloaded file and start Eclipse.exe,
  - Launch the Modeling discovery site update,
  - Check Papyrus and start installation.







Papyrus in life...

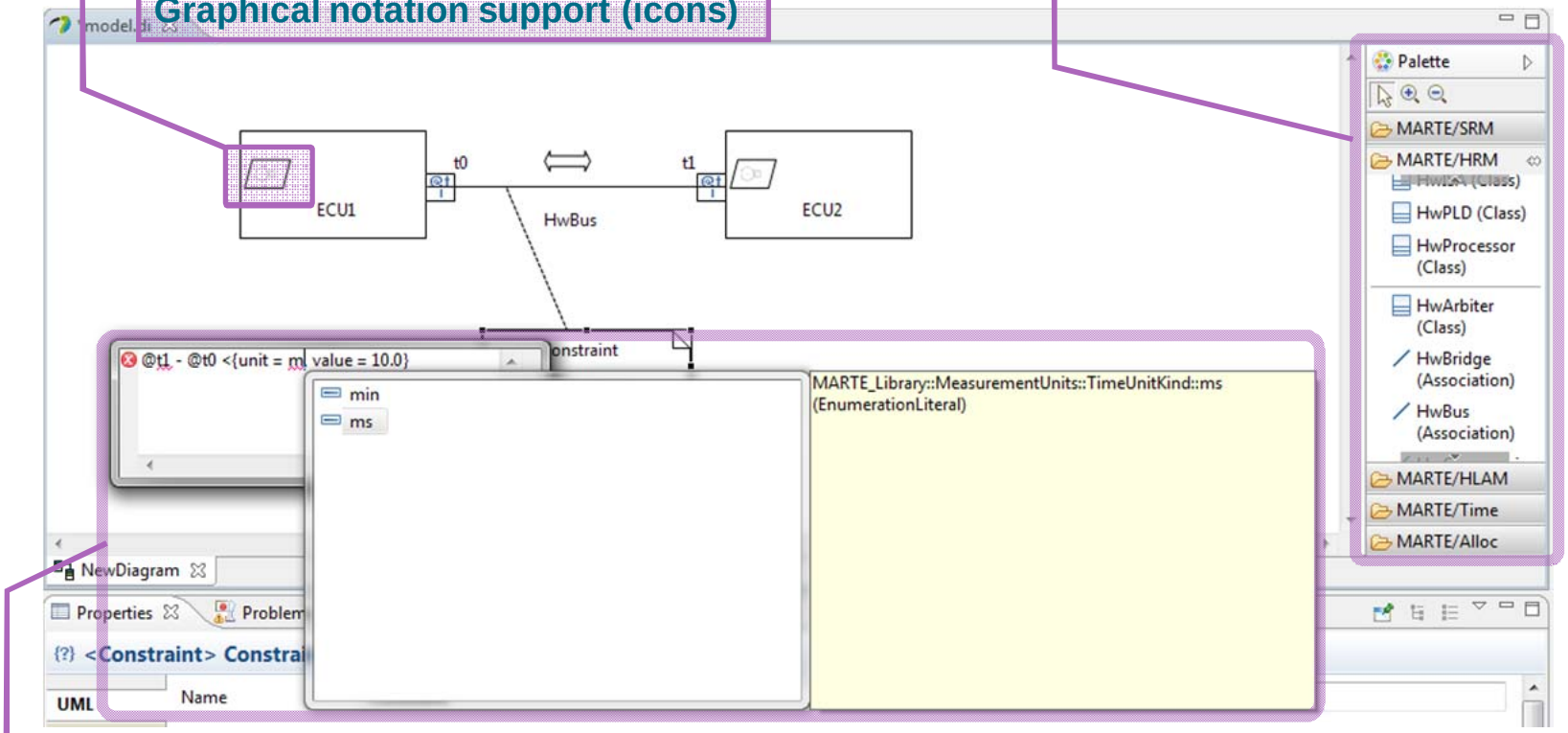
Agenda

3. MDT Papyrus support of MARTE and scheduling analysis
4. MDT Papyrus integration in SCADE
5. Q&A.

MARTE v1.1 Support

Dedicated palette for direct creation of MARTE elements

Graphical notation support (icons)

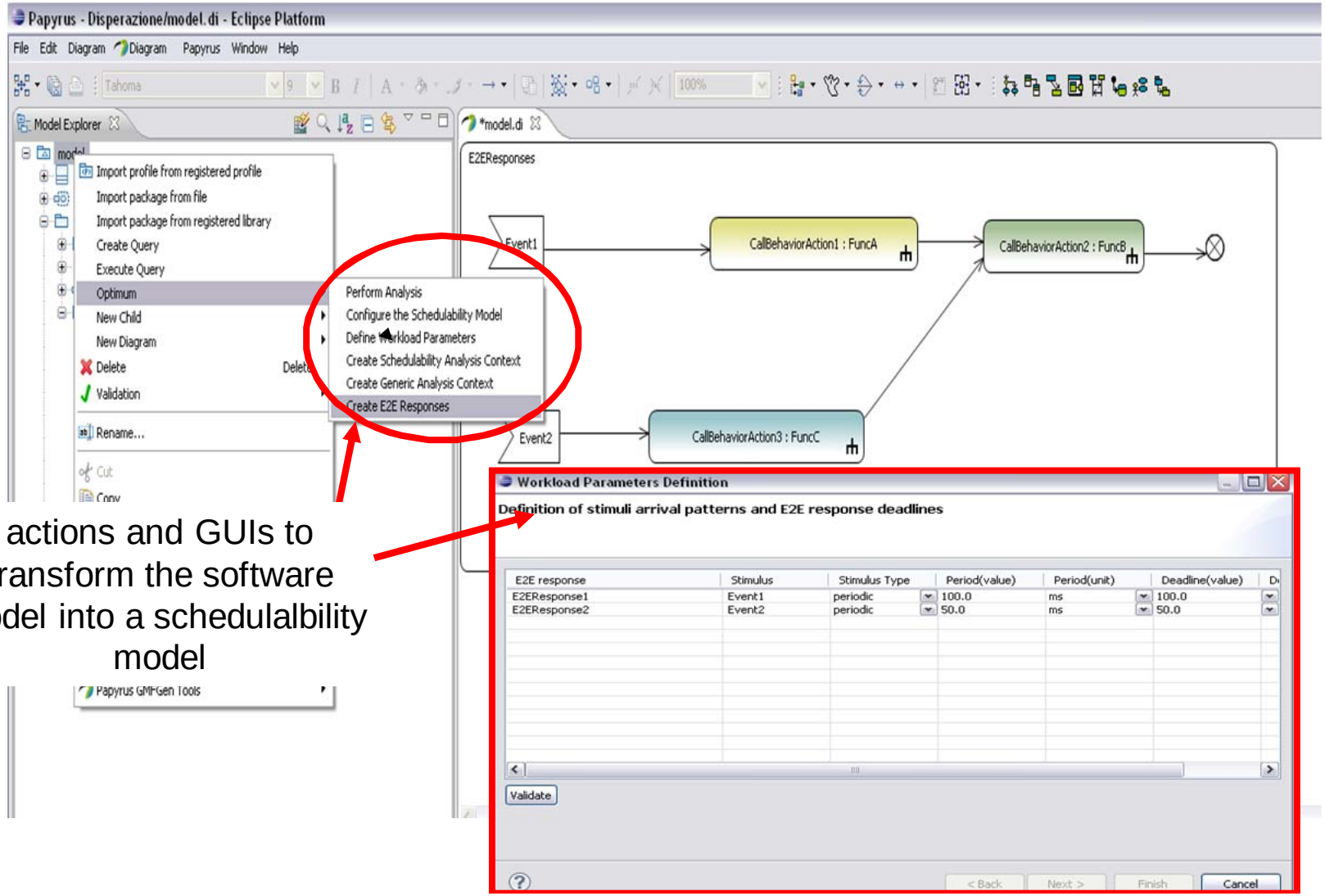


VSL editor with completion, syntactic and semantic verification

Installation update site for MARTE within Papyrus:
http://download.eclipse.org/modeling/mdt/papyrus/extra/updates/nightly/indigo

Model-level schedulability analysis: Optimum

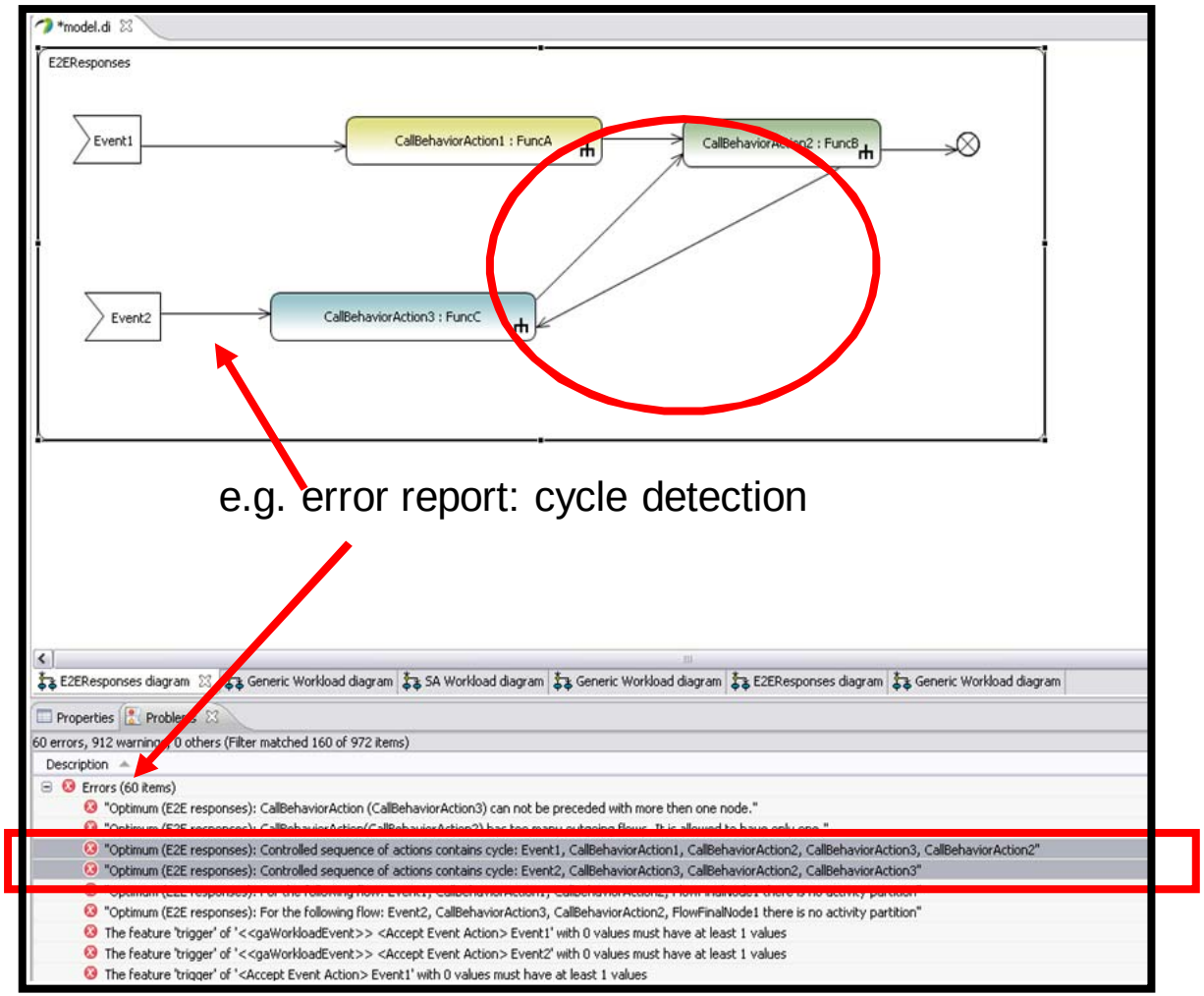
- Methodological support: from software model to schedulability model



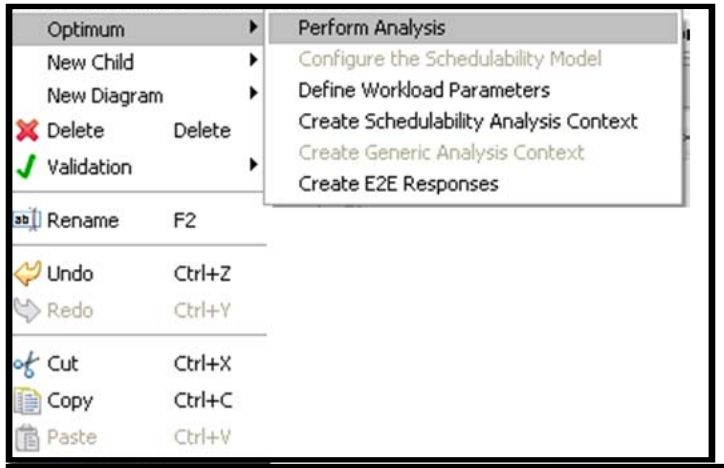
actions and GUIs to transform the software model into a schedulalbility model

Model-level schedulability analysis: Optimum

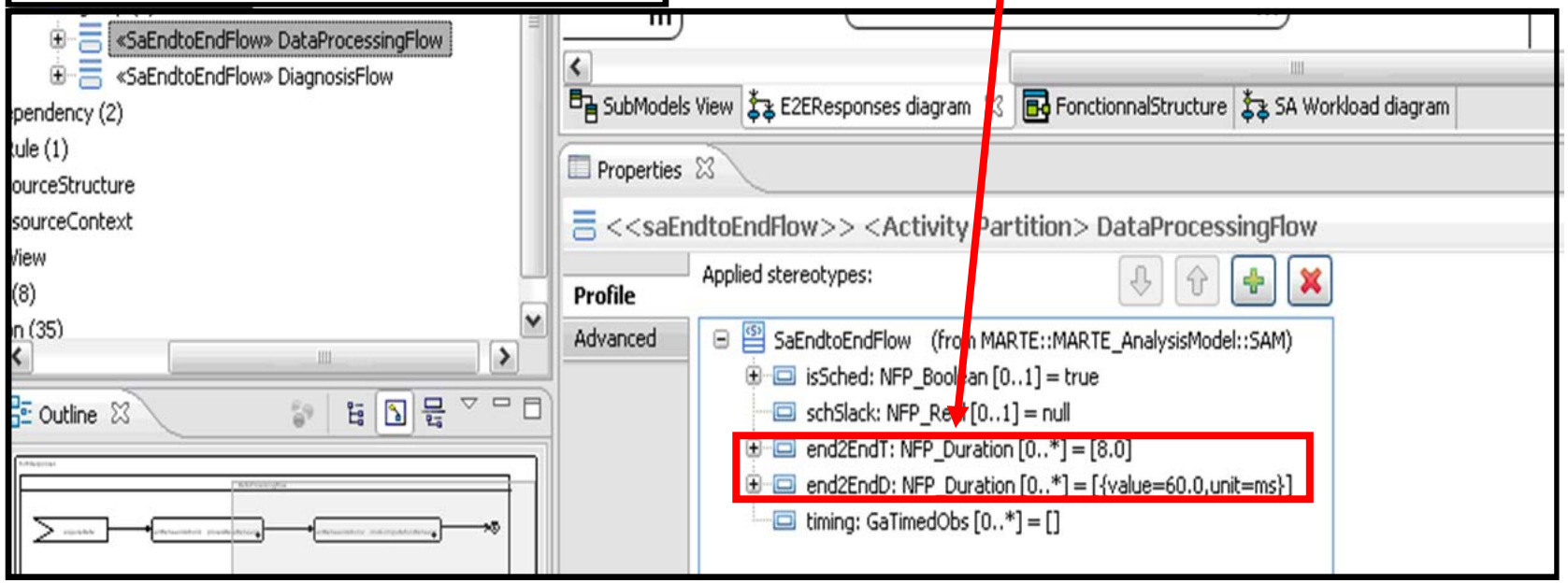
- Methodological support: model validation



- Schedulability analysis (fixed priority)



analysis results in the MARTE model



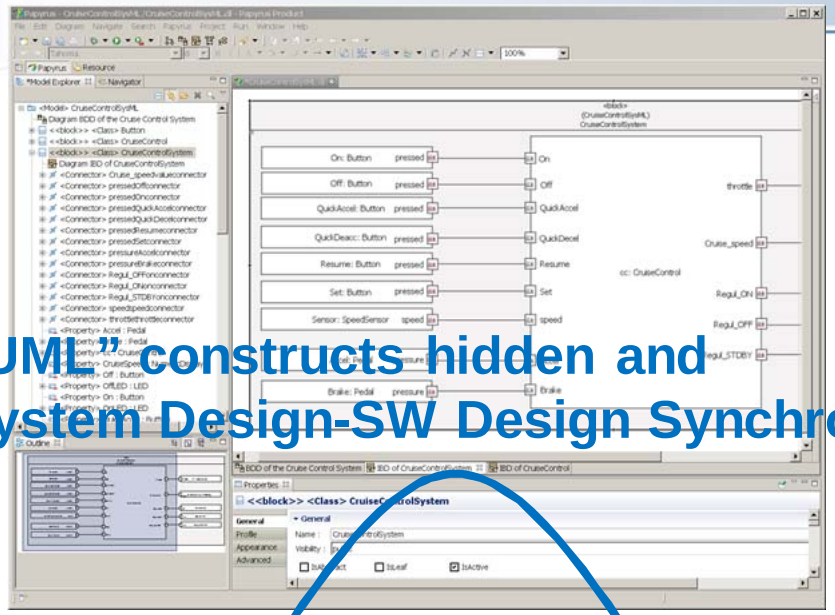
Agenda

4. MDT Papyrus integration in SCADE
5. Q&A.

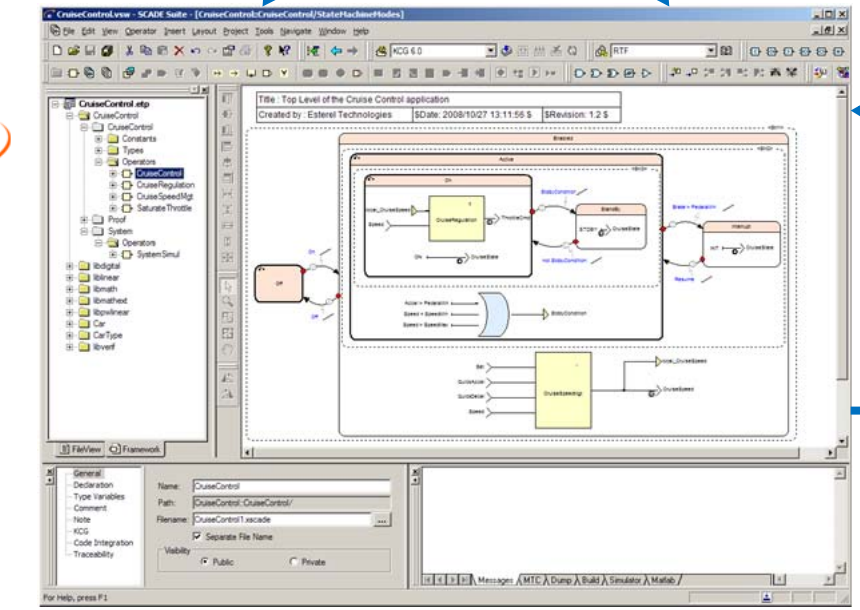
Main Objectives

- Provide a model based description of the Software Architecture
- Add an Architectural Dataflow view of the system in SCADE
- Ease communication between teams and provide a way to remove redundancies between activities (System architecture, Control engineering, Safety analysis, Software design)
- Dataflow and Interfaces management (Consistency, checks, synchronization with Software tools and central repository)
- Provide assistance for System Integration (Code generation)

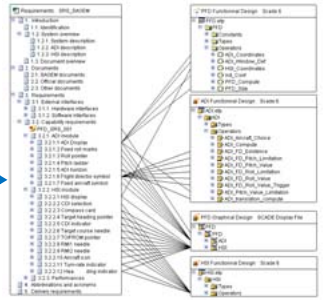
SCADE System Concept



"UML" constructs hidden and
System Design-SW Design Synchronization

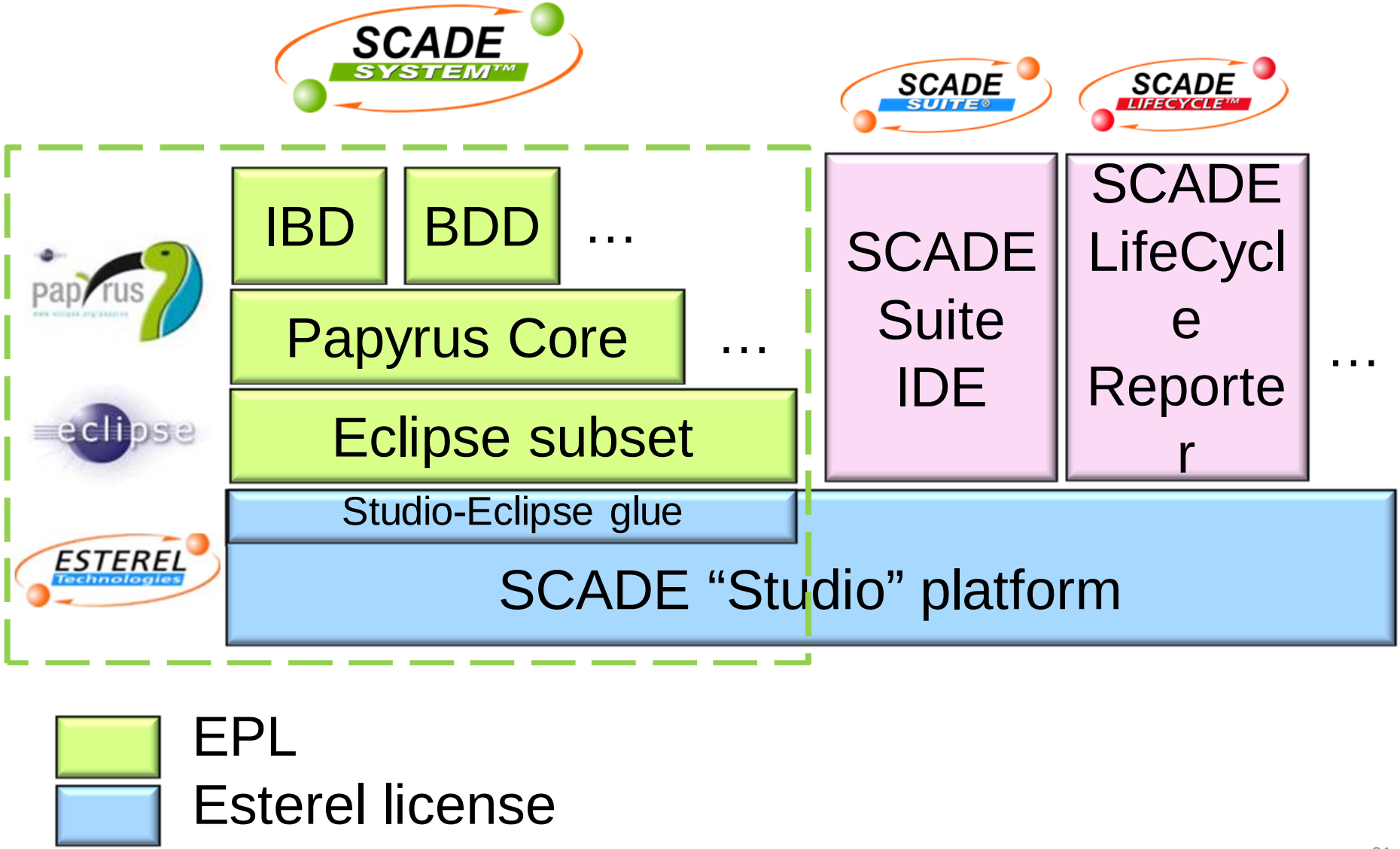


RMgateway



Reporter

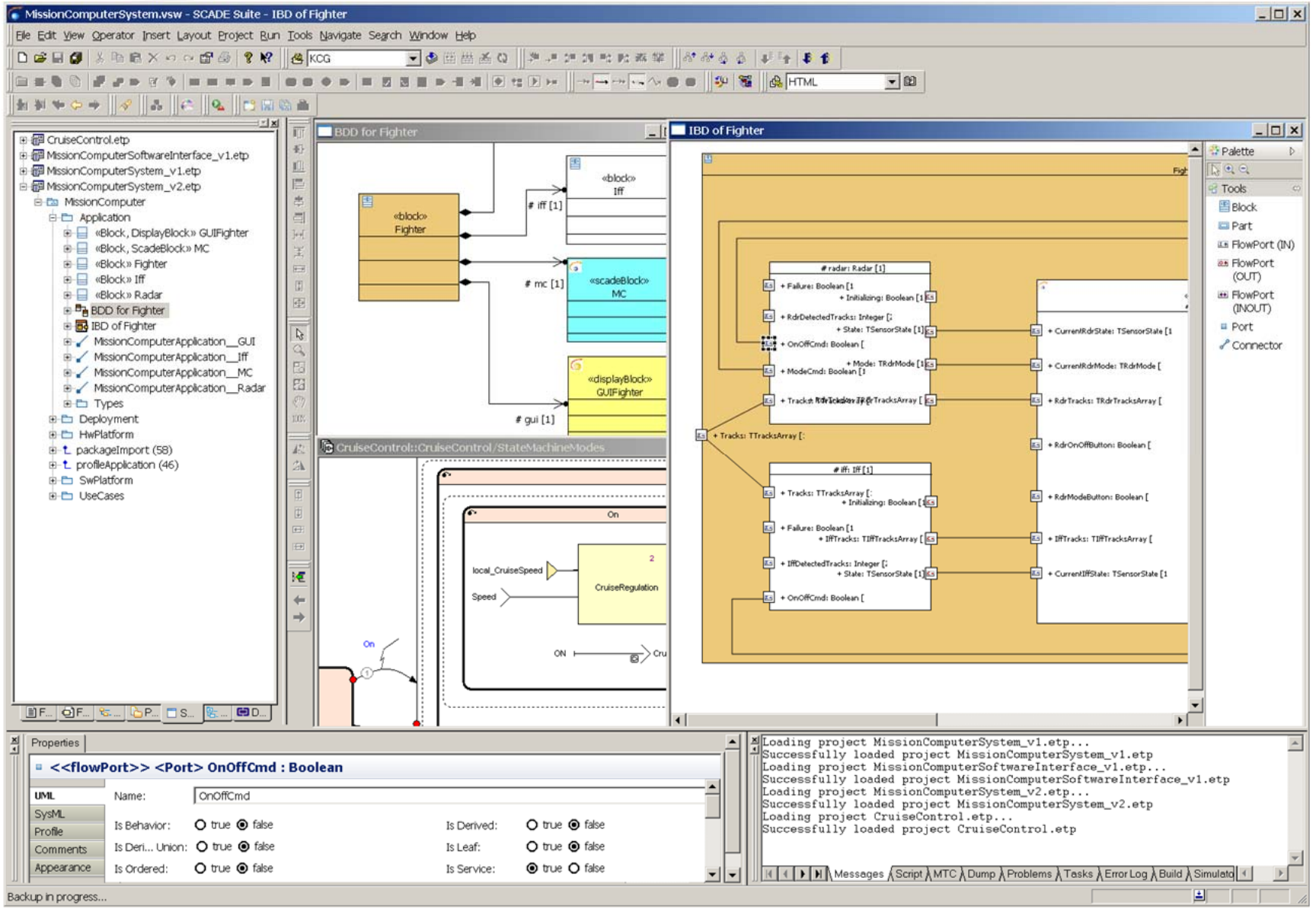
Open Tool Architecture



Open Tool Architecture - Benefits

- **Open Source**
  - Developed within the framework of the joint Esterel Technologies / French Atomic Energy Commission (CEA LIST) Laboratory
  - MDT Papyrus components shared with SCADE System Designer
  - EPL License
- **Open Eclipse Architecture**
  - Interoperable, and extensible with other MDT Papyrus components
  - Framework opened to plug-in other Eclipse based tools
- **Open Tool w.r.t User Designs**
  - SysML model API for data extract, import or modifications
- **Professional Support**
  - Unified look and feel with SCADE

Overview


MissionComputerSystem.vsw - SCADE Suite - IBD of Fighter
File Edit View Operator Insert Layout Project Run Tools Navigate Search Window Help
CruiseControl.etp
MissionComputerSoftwareInterface_v1.etp
MissionComputerSystem_v1.etp
MissionComputerSystem_v2.etp
MissionComputer
Application
«Block, DisplayBlock» GUIFighter
«Block, ScadeBlock» MC
«Block» Fighter
«Block» Iff
«Block» Radar
BDD for Fighter
IBD of Fighter
MissionComputerApplication__GUI
MissionComputerApplication__Iff
MissionComputerApplication__MC
MissionComputerApplication__Radar
Types
Deployment
HwPlatform
packageImport (58)
profileApplication (46)
SwPlatform
UseCases
BDD for Fighter
«block» Fighter
«block» Iff
«scadeBlock» MC
«displayBlock» GUIFighter
iff [1]
mc [1]
gui [1]
CruiseControl::CruiseControl/StateMachineModes
On
local_CruiseSpeed
Speed
CruiseRegulation
ON
IBD of Fighter
Palette
Tools
Block
Part
FlowPort (IN)
FlowPort (OUT)
FlowPort (INOUT)
Port
Connector
radar: Radar [1]
iff: Iff [1]
Properties
<<flowPort>> <Port> OnOffCmd : Boolean
UML
SysML
Profile
Comments
Appearance
Name: OnOffCmd
Is Behavior: true false
Is Derived: true false
Is Deri... Union: true false
Is Leaf: true false
Is Ordered: true false
Is Service: true false
Loading project MissionComputerSystem_v1.etp...
Successfully loaded project MissionComputerSystem_v1.etp
Loading project MissionComputerSoftwareInterface_v1.etp...
Successfully loaded project MissionComputerSoftwareInterface_v1.etp
Loading project MissionComputerSystem_v2.etp...
Successfully loaded project MissionComputerSystem_v2.etp
Loading project CruiseControl.etp...
Successfully loaded project CruiseControl.etp
Messages Script MTC Dump Problems Tasks Error Log Build Simulato
Backup in progress...

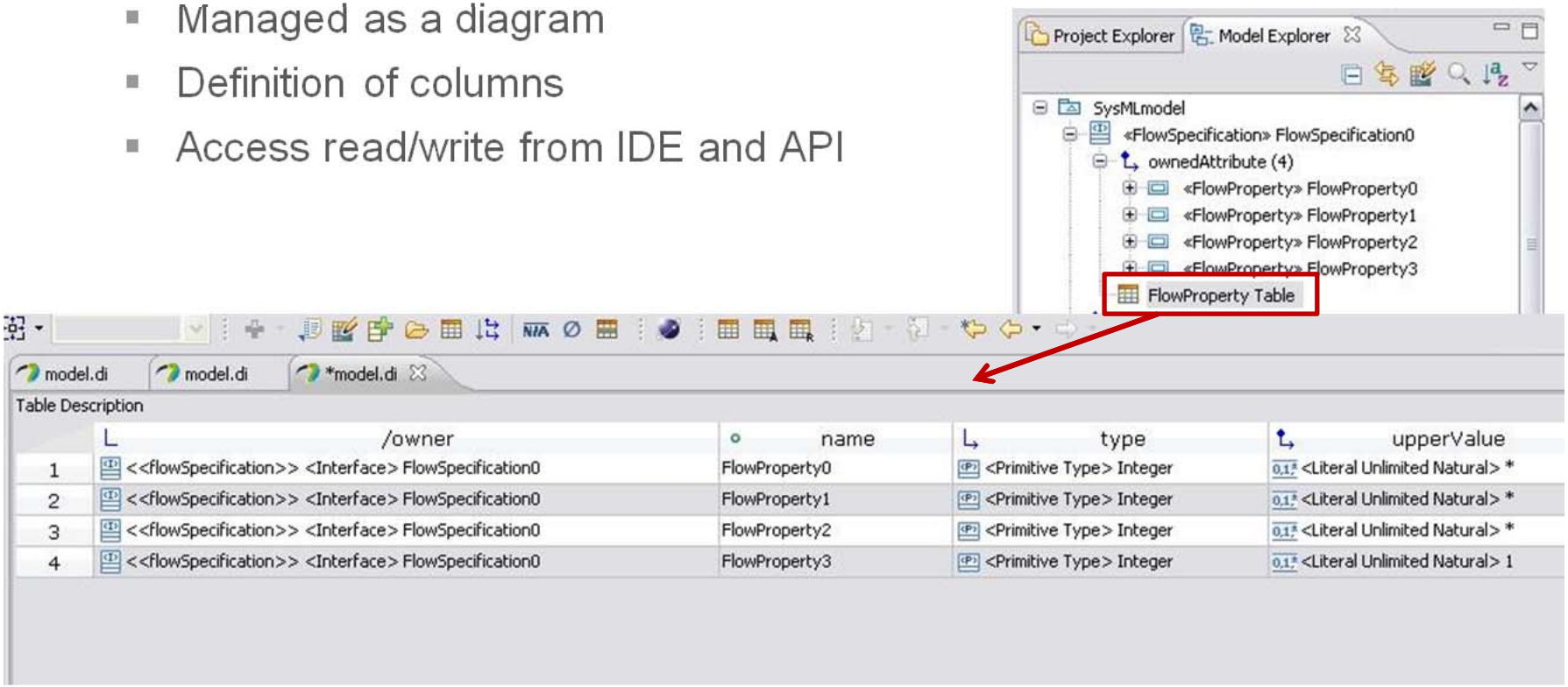
Features

- **System Architecture Modeling**
  - Focus on **ease of use**
    - Hide the “based on a UML profile” feeling
    - Copy/Paste, DD, popup menus, line-routing, etc.
  - Allows **graphical multi-views** on a system model composing packages, diagrams, blocks, ports, connectors
  - Subset of SysML standard
    - IBD and BDD first

- **Open Architecture**
  - **Model API** (Java and Tcl API) provided for data extract, import or modifications

Features

- **Tabular views**
  - Managed as a diagram
  - Definition of columns
  - Access read/write from IDE and API



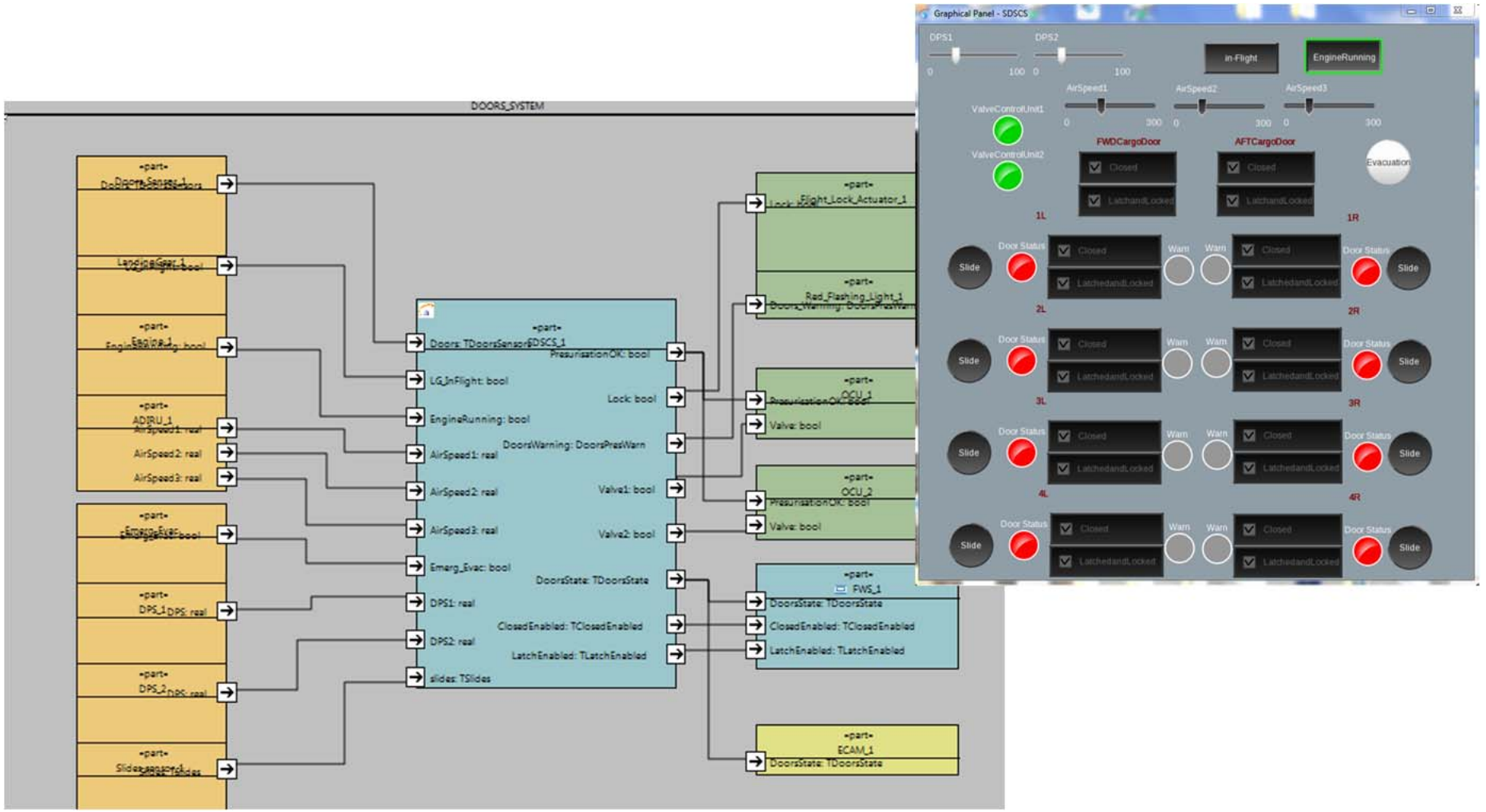
System Engineering Process Support

- **Model Diff**
  - To understand system design evolutions
- **Design Checker**
  - To ensure early model consistency
  - Based on customizable rules
- **Model Component Export**
  - To preserve IP of other system design parts, e.g. to subcontract the SW development of one system block
- **System/SW Model Synchronization**
  - Avoid duplication of efforts and inconsistencies between system structural description and software behavioral description
  - System design and components evolve independently
  - On-demand re-synchronization/reconciliation of interfaces

SCADE lifecycle Support

- **A common traceability tool** (RM Gateway)
  - Fine grained traceability, customizable
  - Can be used for SCADE System, Suite, Display
- **A common Document generation module** (Reporter)
  - Cross references between system and SW designs
  - Share RTF and HTML renderers: same, customizable look
  - Can be used for SCADE System, Suite, Display models

SCADE System Demo: Doors Management Simplified System



Agenda

1. MDT Papyrus versus Papyrus 1.x

5. Q&A.

More Information

- **For developers...**
  - http://wiki.eclipse.org/Papyrus_Developer_Guide
  - http://dev.eclipse.org/mailman/listinfo/mdt-papyrus.dev
- **For users...**
  - http://www.eclipse.org/papyrus
  - news://news.eclipse.org/eclipse.papyrus
- **Papyrus project lead contact:**
  - sebastien.gerard@cea.fr