

ModelicaML – Tutorial

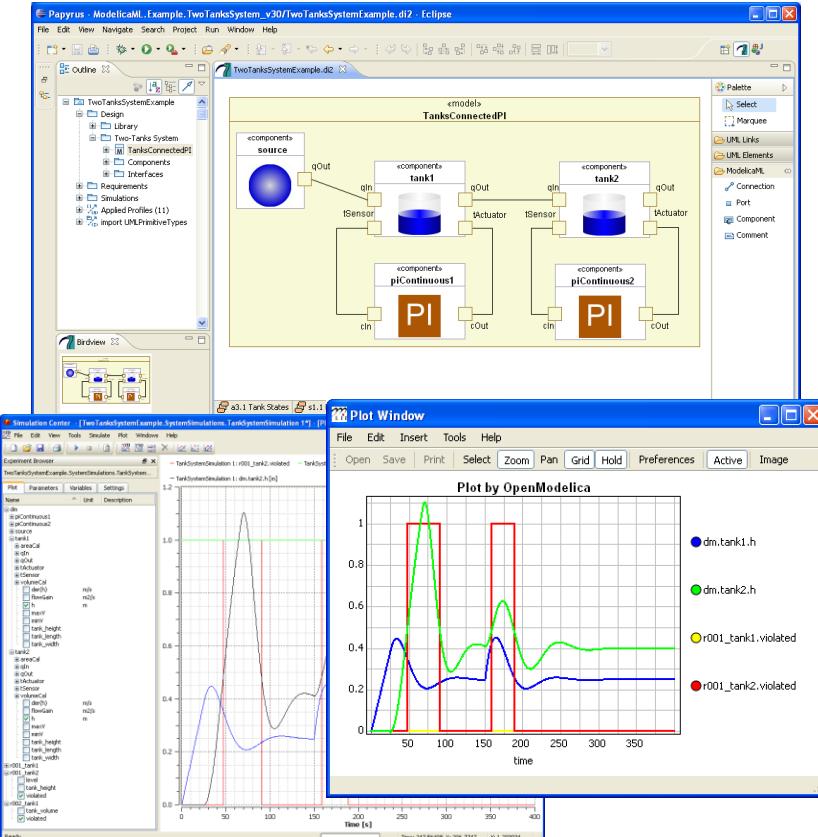
Getting Started

Wladimir Schamai
EADS Innovation Works
Systems Engineering

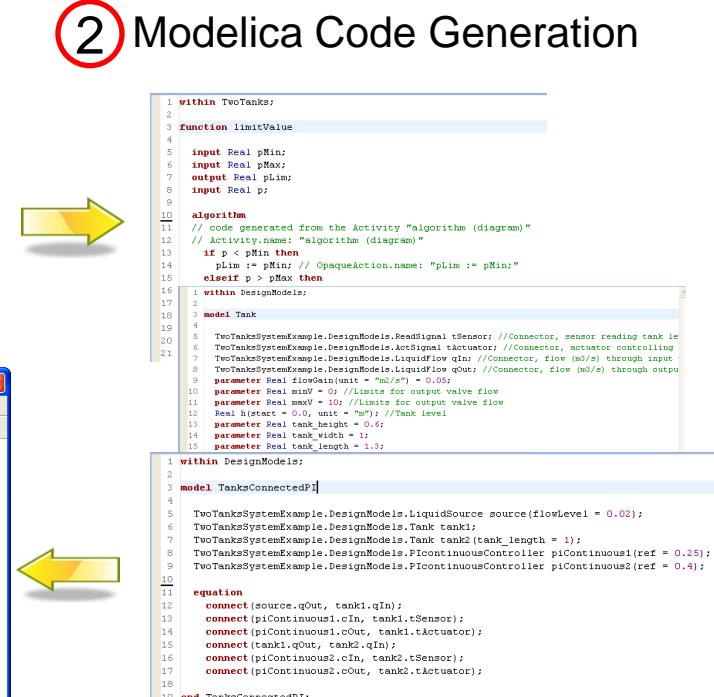


ModelicaML: Technology

1 System Modeling with ModelicaML



2 Modelica Code Generation



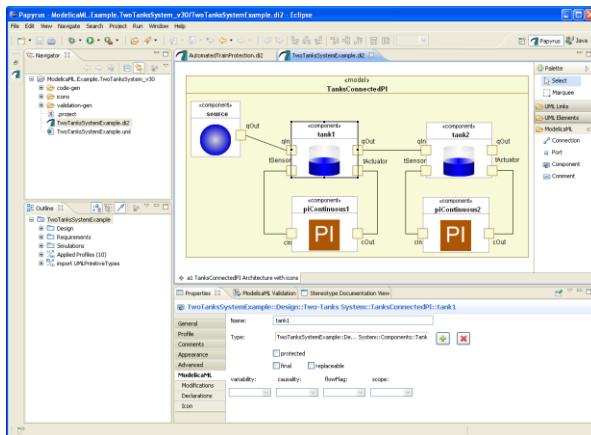
3 System Simulation with Modelica Tools

ModelicaML: Technology



Papyrus UML

ModelicaML Profile (Eclipse Plug-In)



Any Modelica Simulation Tool



Acceleo



Acceleo

ModelicaML Code Generator
(Eclipse Plug-In)



Model to Text Transformation



Generated Modelica Code
.mo files)

EADS

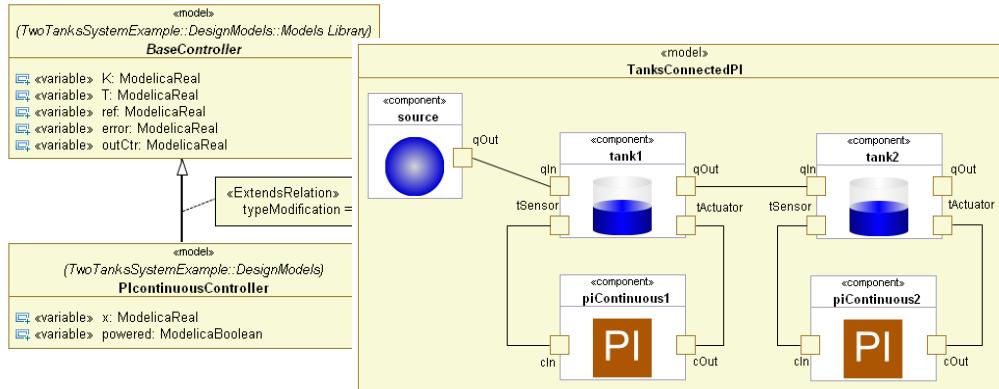
ModelicaML

General Description of the
UML-Based Graphical Notation



ModelicaML: Graphical Notation

Structure



Requirements

«Requirement»
id = 001
text = The level of liquid in a tank shall never exceed 80% of the tank-height.
specifiesType = [Tank]

«Requirement»
id = 002
text = The volume of the tank1 shall be 0.8m3.
specifiesObject = [TanksConnectedPI.tank1]

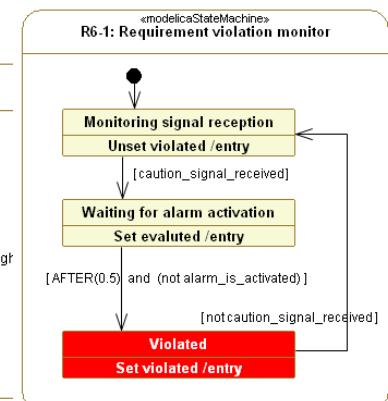
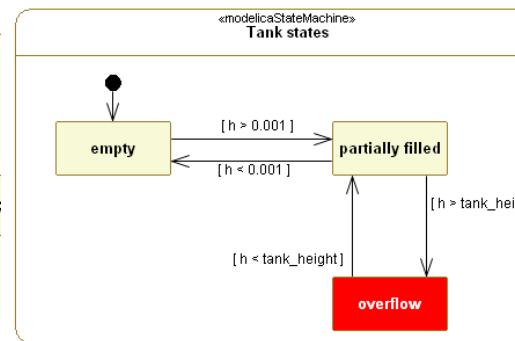
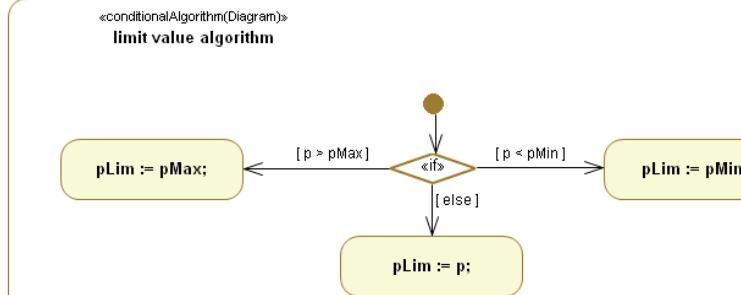
«requirement»
Max level of liquid in a tank

- «variable» maxLevel: ModelicaReal
- «variable» tank_height: ModelicaReal
- «variable» level: ModelicaReal

«requirement»
Volume of the tank1

- «variable» tank_volume: ModelicaReal
- «variable» design_value: ModelicaReal

Behavior

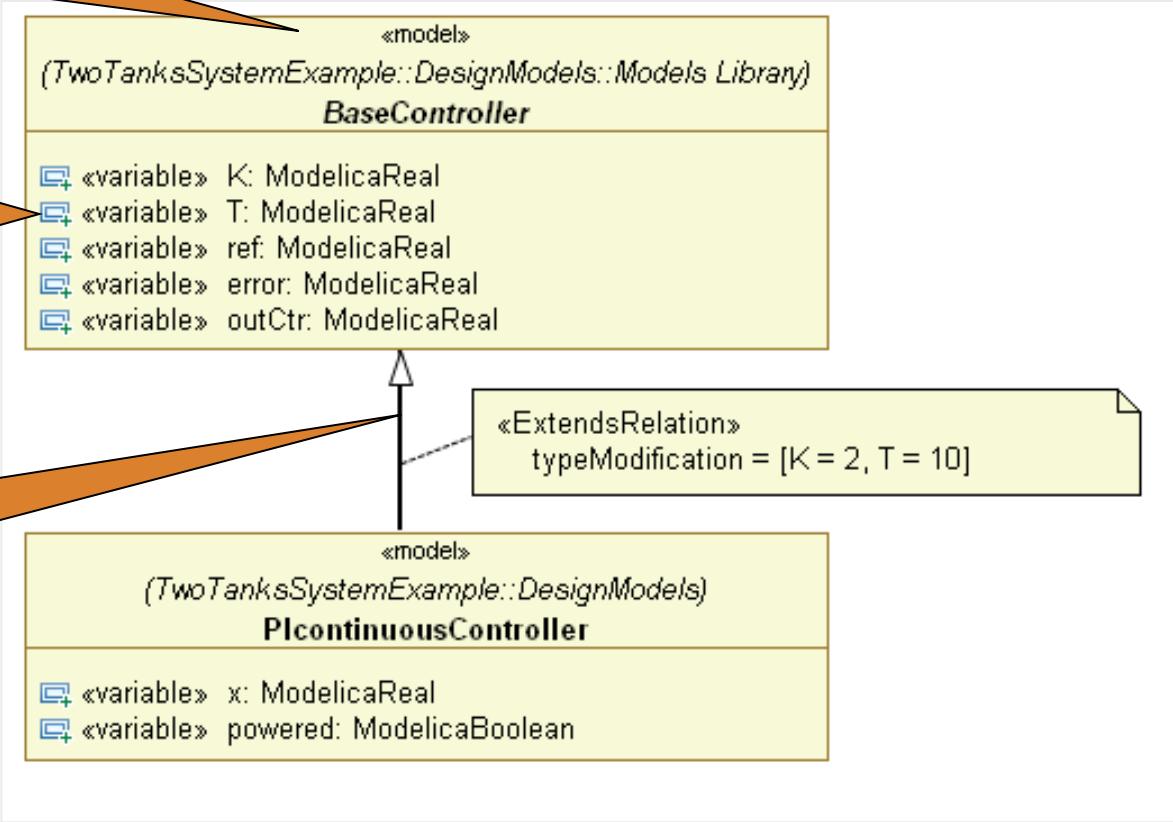


ModelicaML: Class Diagram

Class (model, block, record, connector)

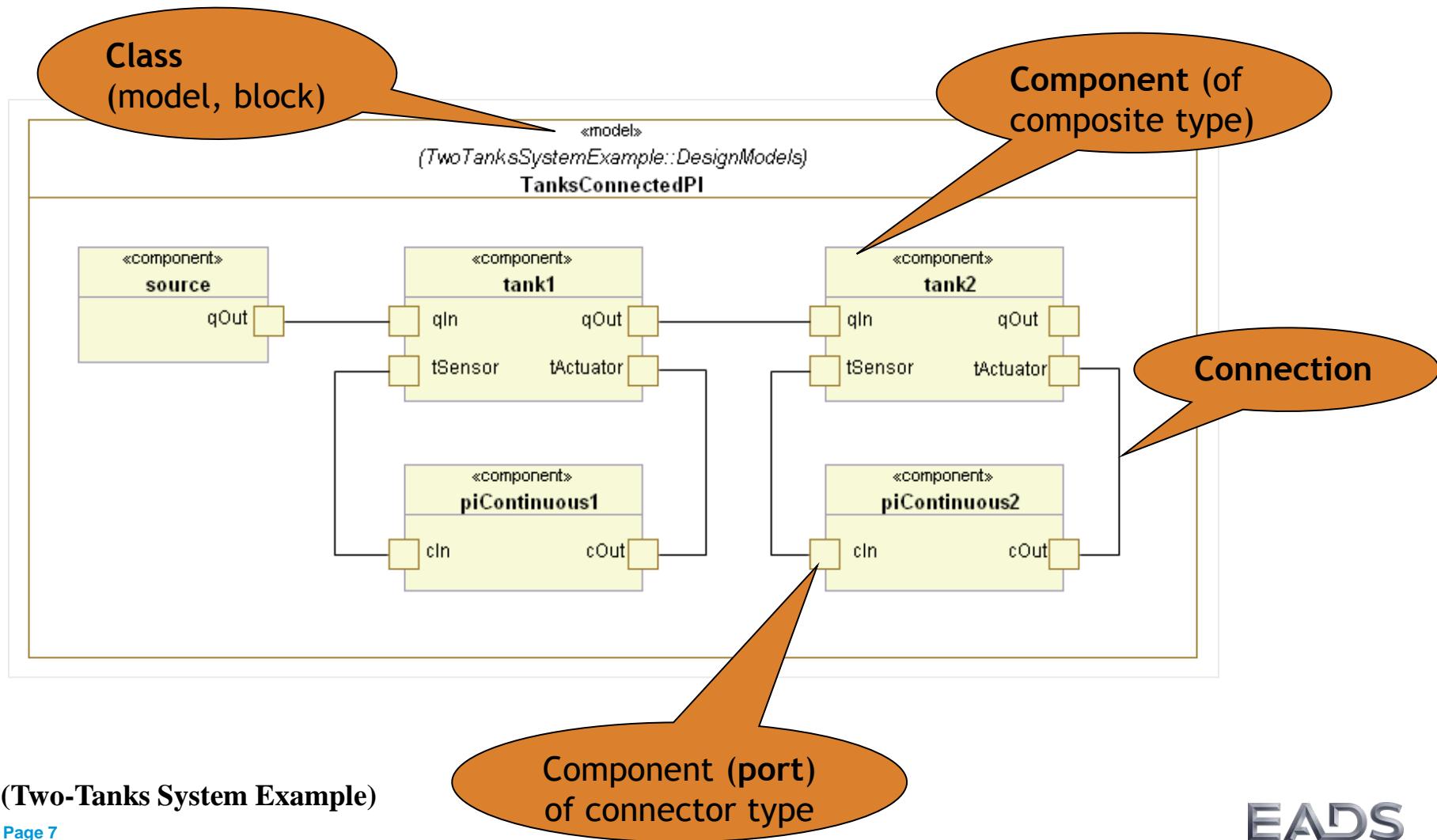
variables (of primitive type) or components (of composite type)

extends relation (with type modifications)

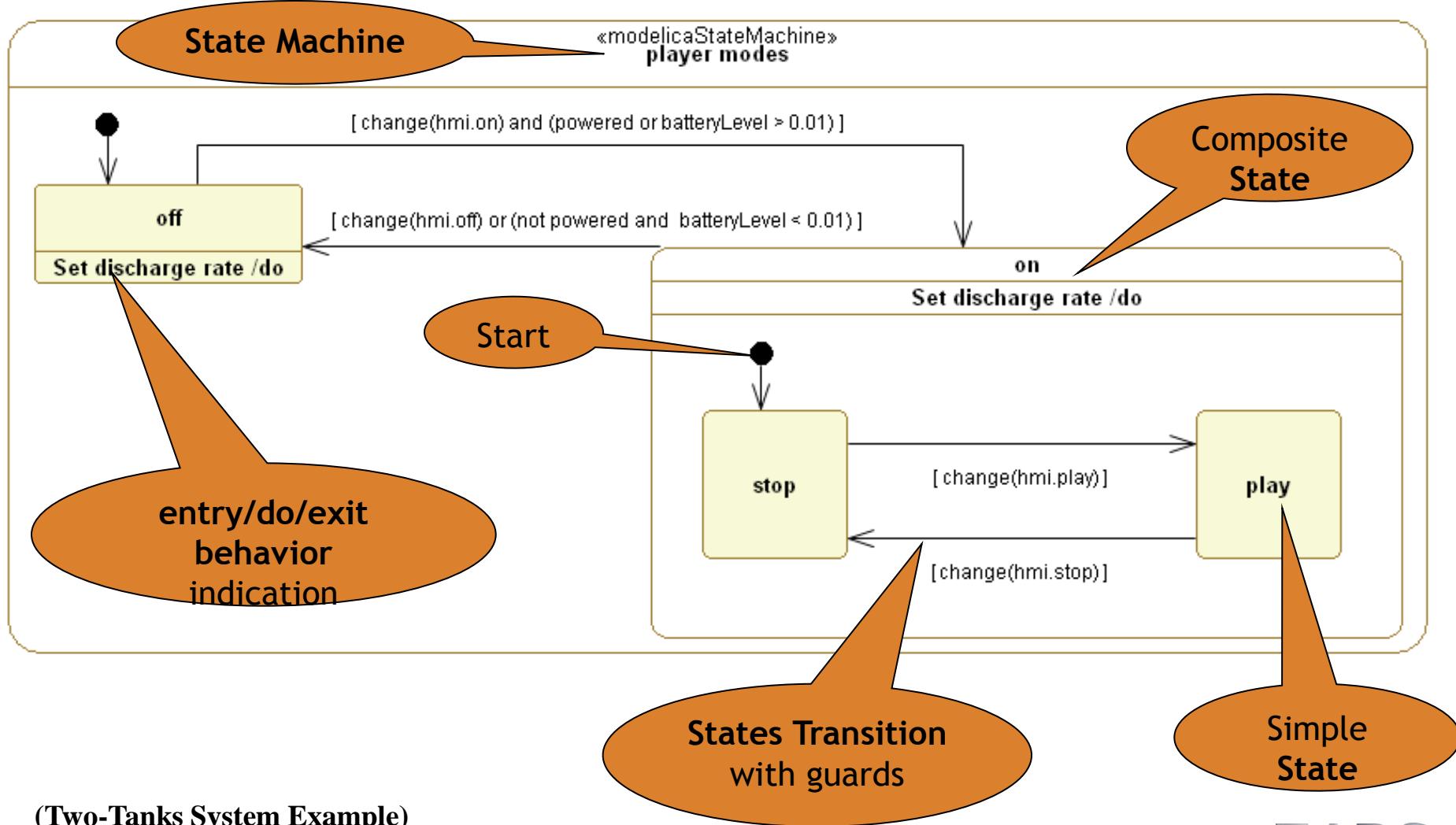


(Two-Tanks System Example)

ModelicaML: Connection Diagram

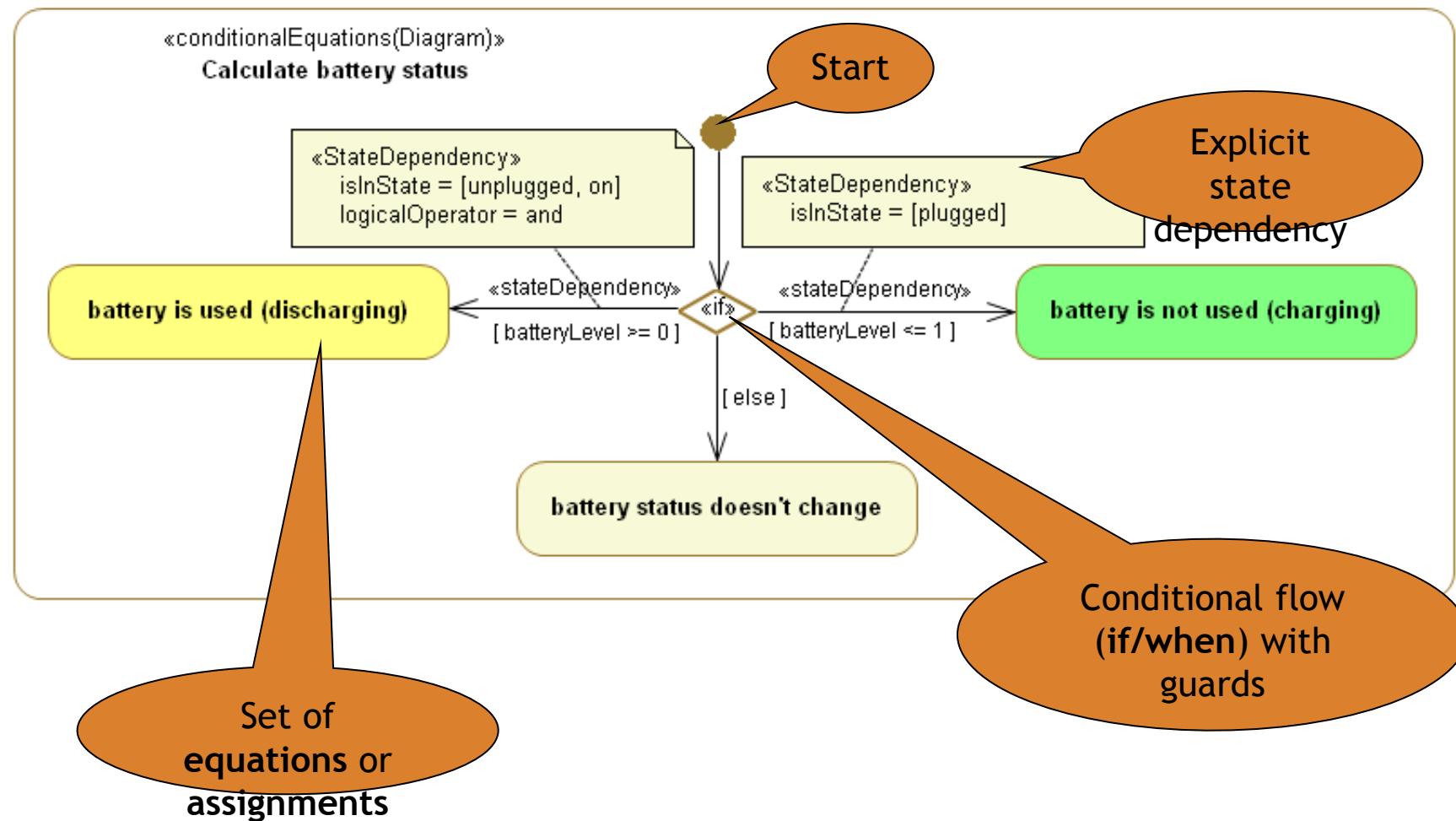


ModelicaML: State Machine Diagram



(Two-Tanks System Example)

ModelicaML: Conditional Eq./Alg. Diagram



(Player Example)

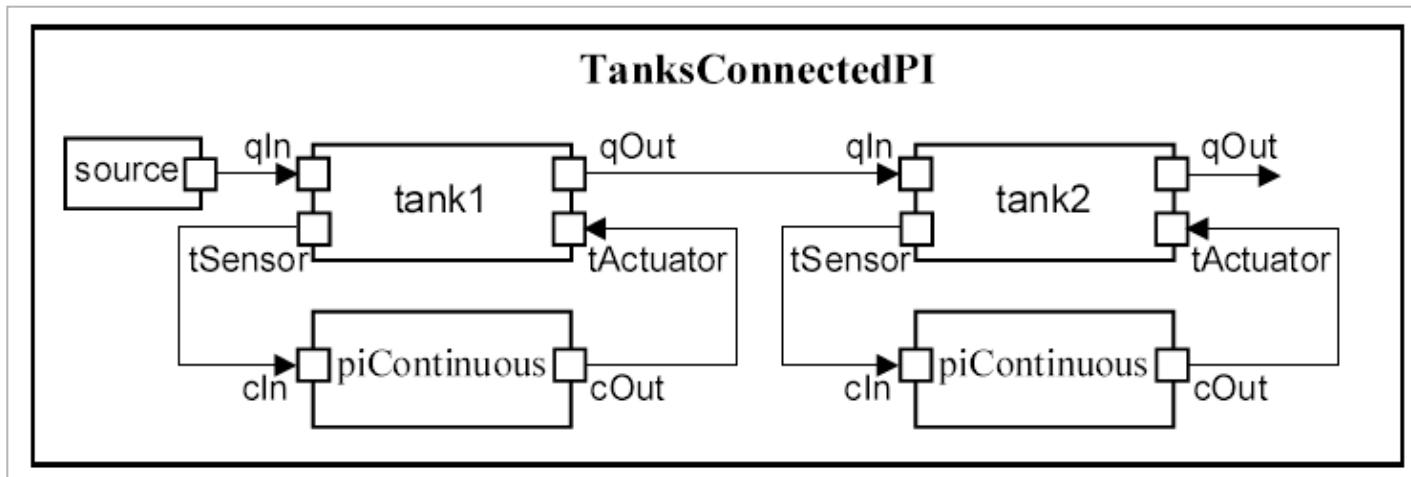
ModelicaML

Hands-on Modeling Tutorial



Example: Two Tanks System

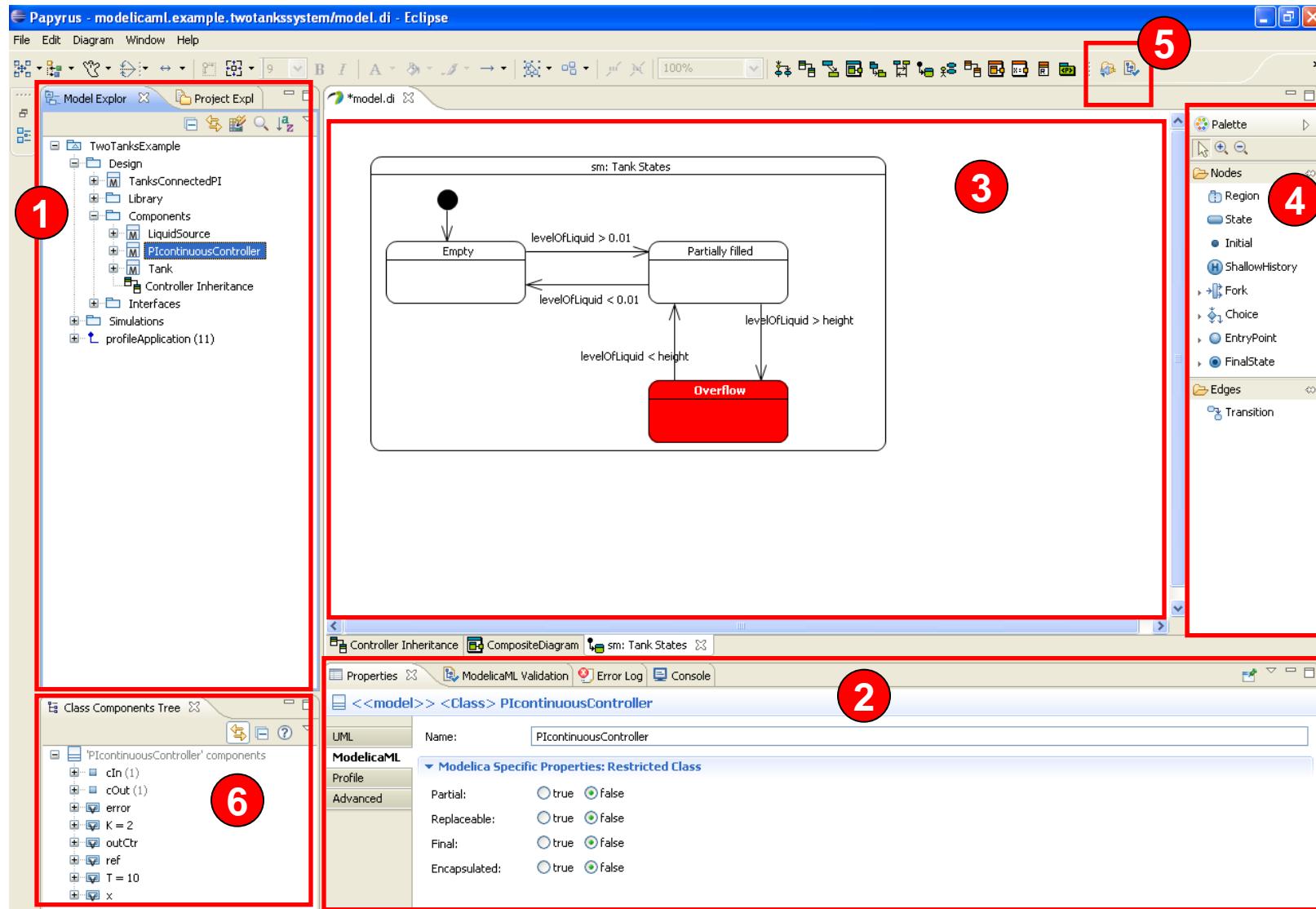
- From “source” liquid flows into the “tank1”
- Controller “piContinuous1” controls the level of liquid in “tank1”, based on a predefined reference value, by opening and closing the tank outflow valve.
- Liquid flows from “tank1” into “tank2”
- Controller “piContinuous2” controls the level of liquid in “tank2”



Source: Peter Fritzson. Principles of Object-Oriented Modeling and Simulation with Modelica 2.1. Wiley-IEEE Press, 2004. page 391.



ModelicaML Papyrus MDT GUI Overview





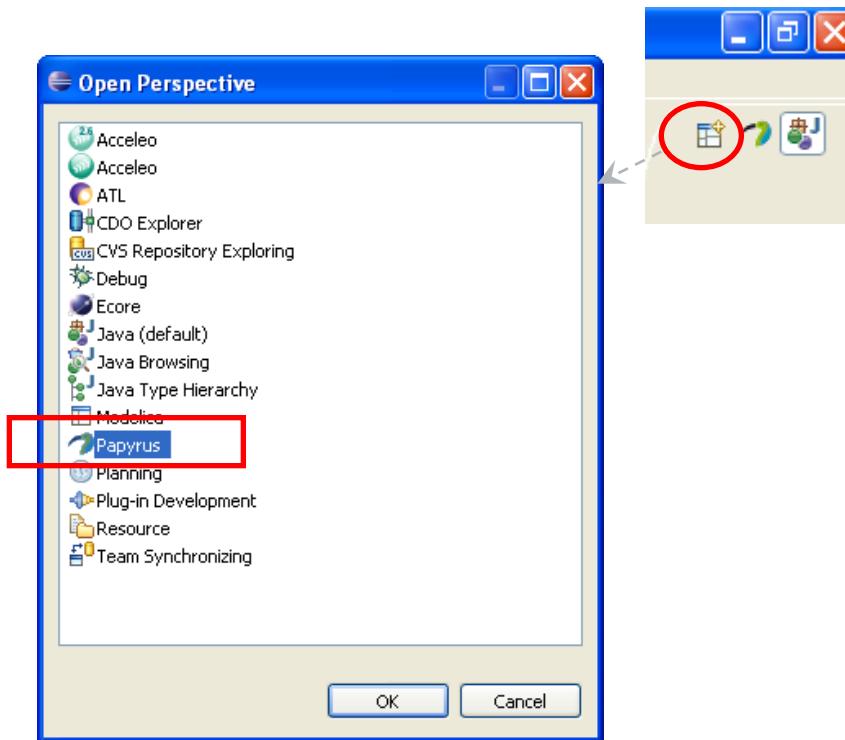
ModelicaML Papyrus MDT GUI Overview

- ① Model Browser: Shows model elements
- ② Properties View: Shows the properties of selected element
- ③ Diagram Editors (different UML-based diagrams)
- ④ Palette (different for each diagram)
- ⑤ ModelicaML code generation and validation buttons
- ⑥ Component tree: Shows the components hierarchy of the selected class

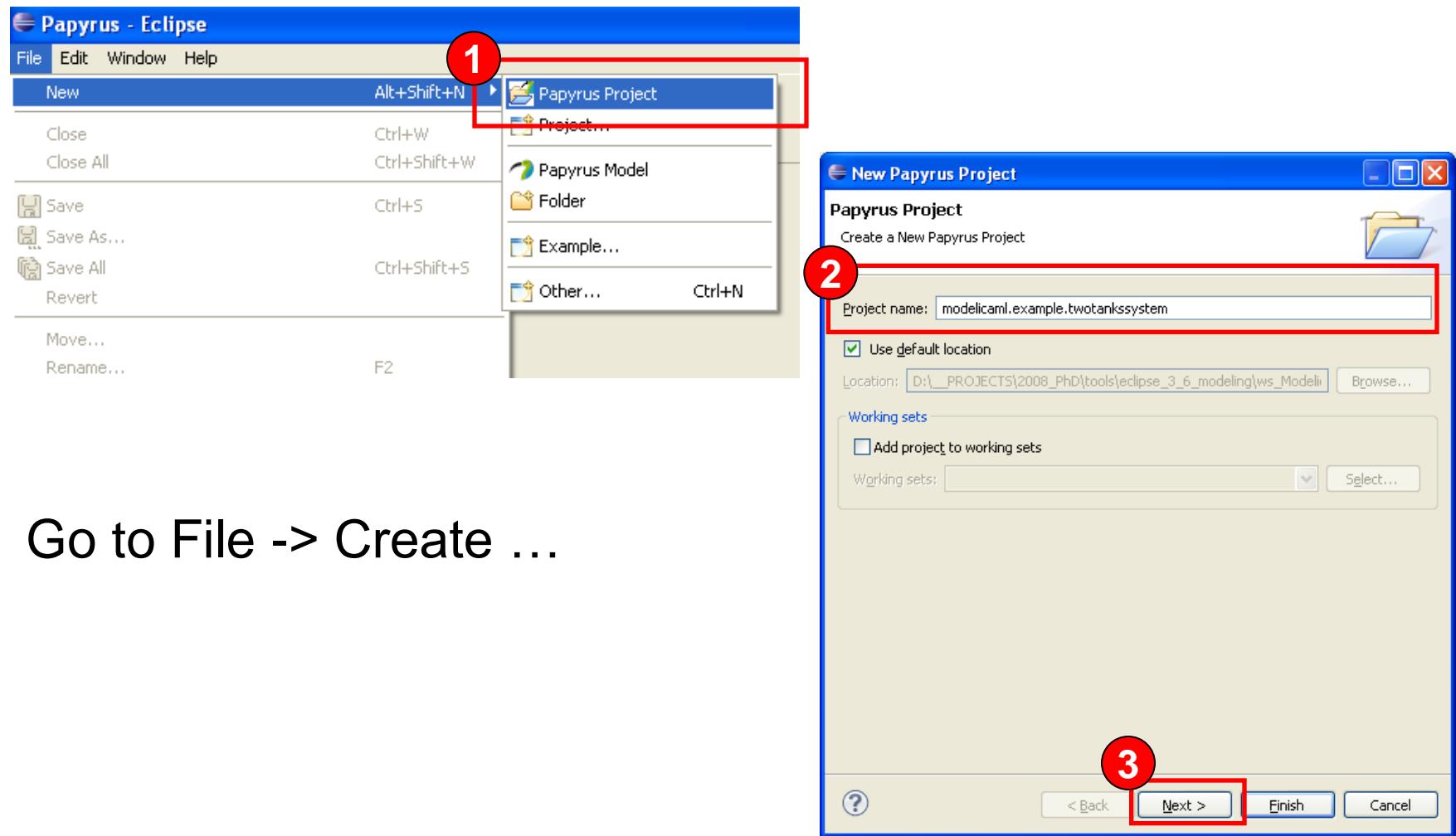
ModelicaML Project Setup

Create ModelicaML Project

- Open Eclipse
- Change the Perspective to Papyrus Perspective

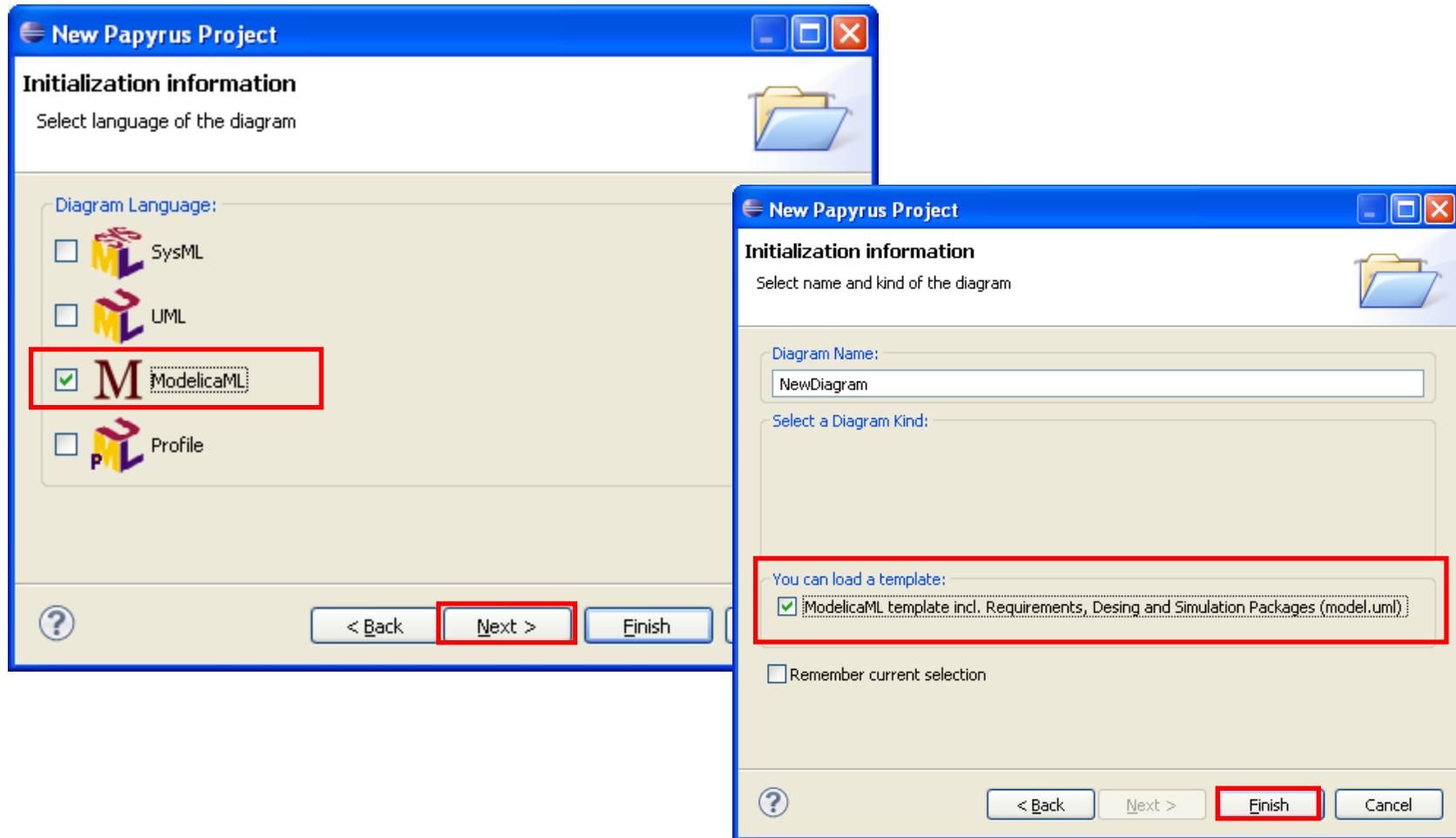


Create ModelicaML Project



Go to File -> Create ...

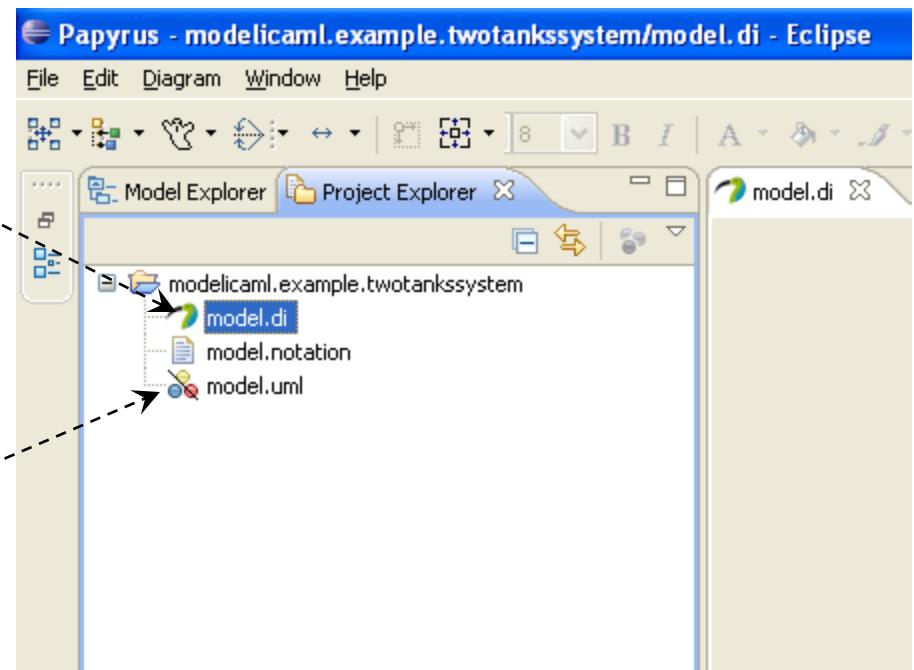
Create ModelicaML Project



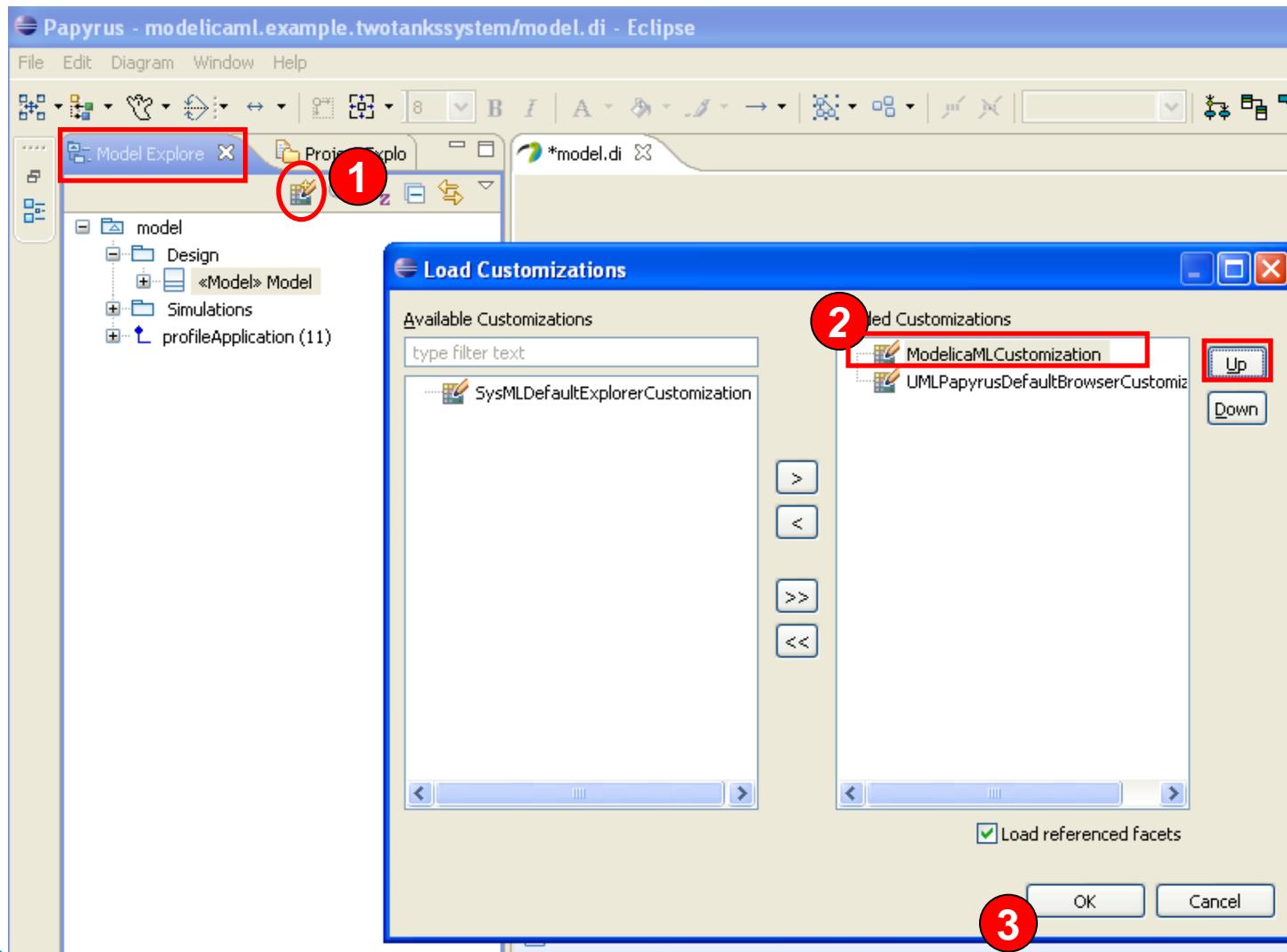
Papyrus Project Files

Diagram file
(can only be edited using Papyrus MDT)

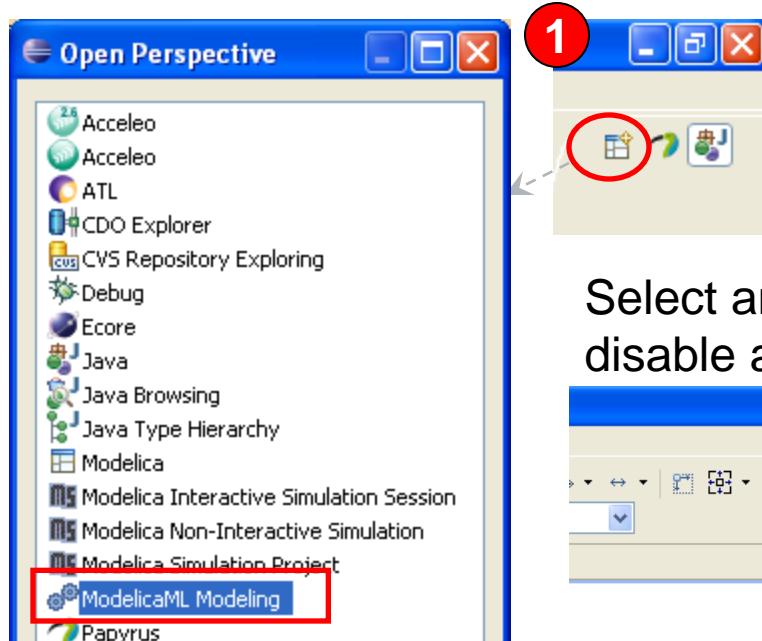
Model file
(can be edited using any UML2 tool)



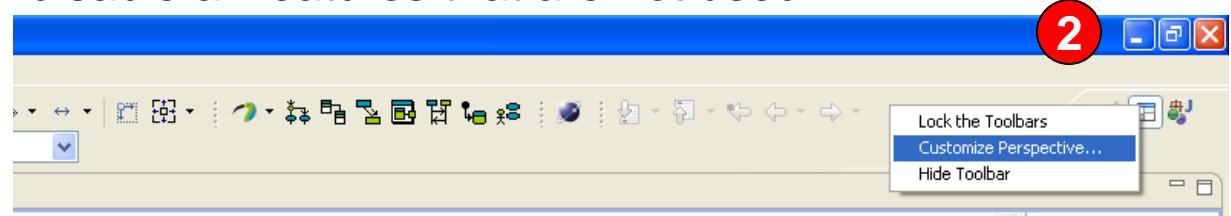
Configure Model Explorer



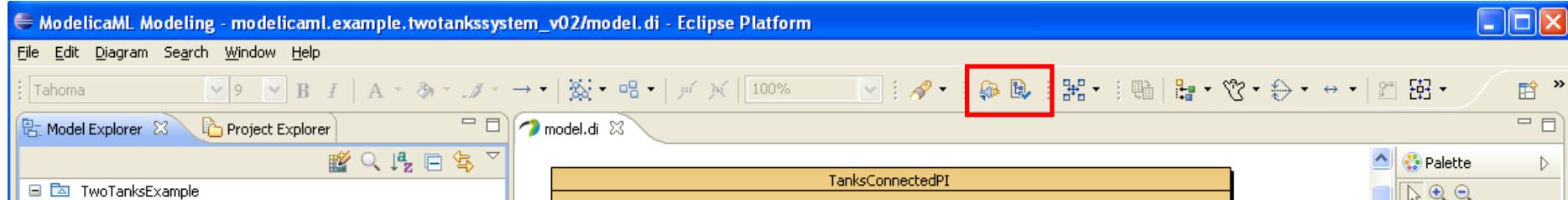
Change and Customize the Perspective



Select and customize the ModelicaML perspective – disable all features that are not used



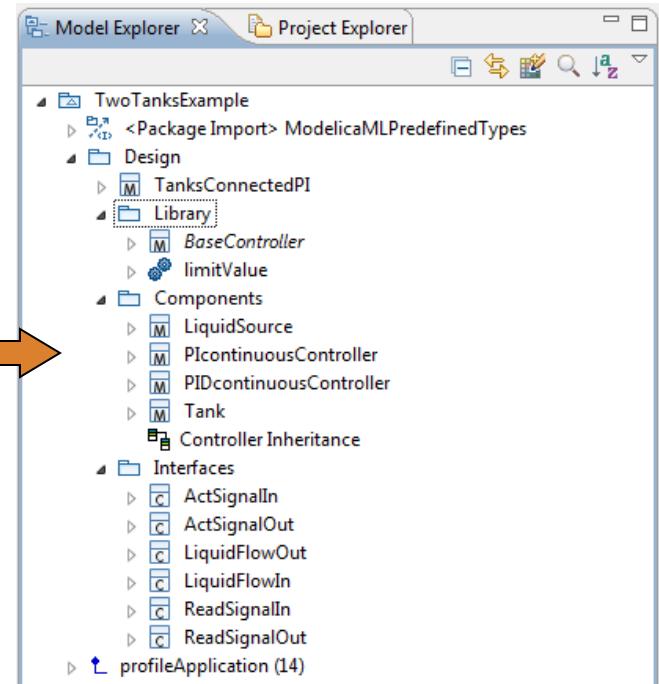
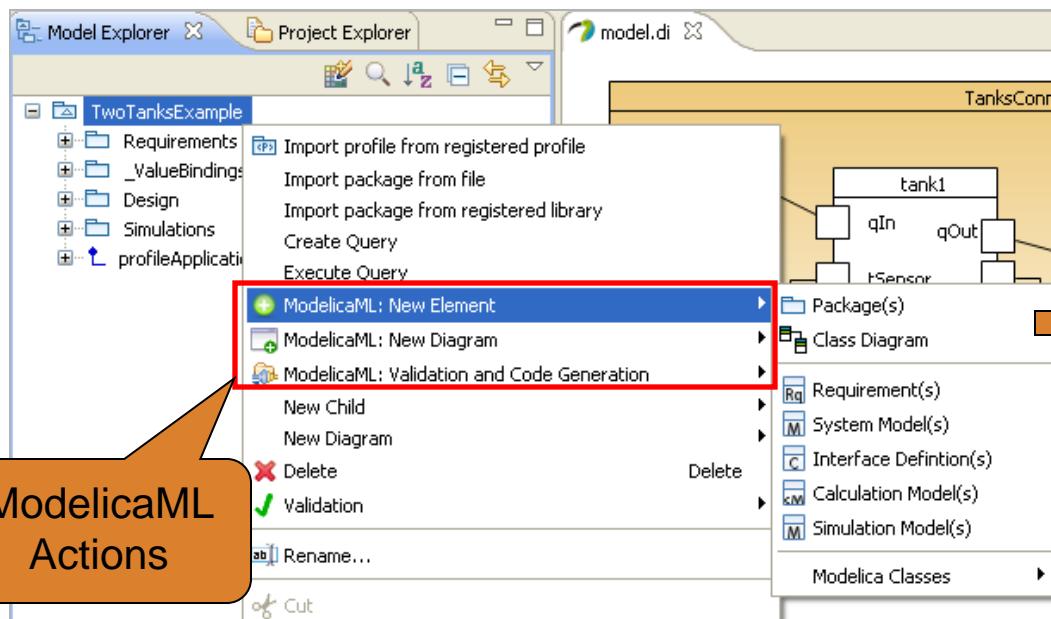
Typical ModelicaML customized perspective:



Model Setup

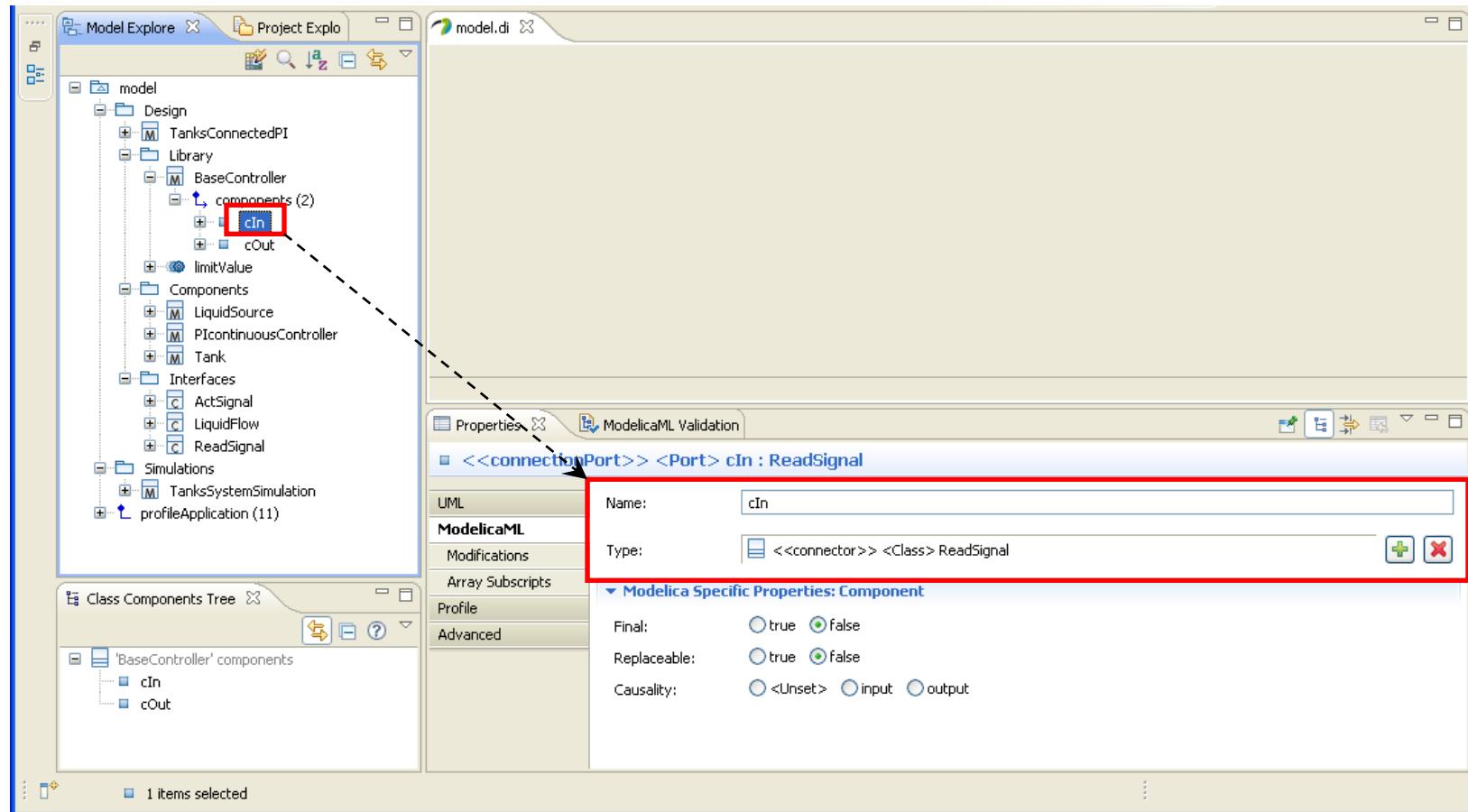
Create Model Structure

Create Packages and Classes using ModelicaML menus





Hint: Setting type of components





Hint: Setting type of components

ModelicaML Primitive Types are:

- ModelicaReal
- ModelicaInteger
- ModelicaString
- ModelicaBoolean

Select Type:
Select the reference you want for Type:
Tooltip: This information is derived from the return result for this Operation. The type of t
modelicareal
Matching items:
<Primitive Type> ModelicaReal

Properties X ModelicaML Validation Error Log Console

<<variable>> <Property> flowLevel : ModelicaReal

UML	Name:	flowLevel
ModelicaML	Type:	<Primitive Type> ModelicaReal
Modifications		
Declaration		
Cond. Expression		
Array Subscripts		
Profile		

▼ Modelica Specific Properties: Component

Variability: continuous

Causality: <Unset> input output

Cancel



Hint: Setting of the component properties (Declaration, Causality, Variability, etc.)

The screenshot shows the Model Explorer and the Properties view in a Modelica editor.

Model Explorer: Shows a project structure for "TwoTanksExample". Under "Components", there is a "Tank" component which contains 10 components: tSensor, tActuator, qIn, qOut, flowGain, minV, maxV, area, height, and levelOfLiquid. The "flowGain" component is selected and highlighted with a red box.

Properties View: The title bar shows "Properties" and other tabs: ModelicaML Validation, Error Log, and Console. The main content displays the properties for the selected "flowGain" variable.

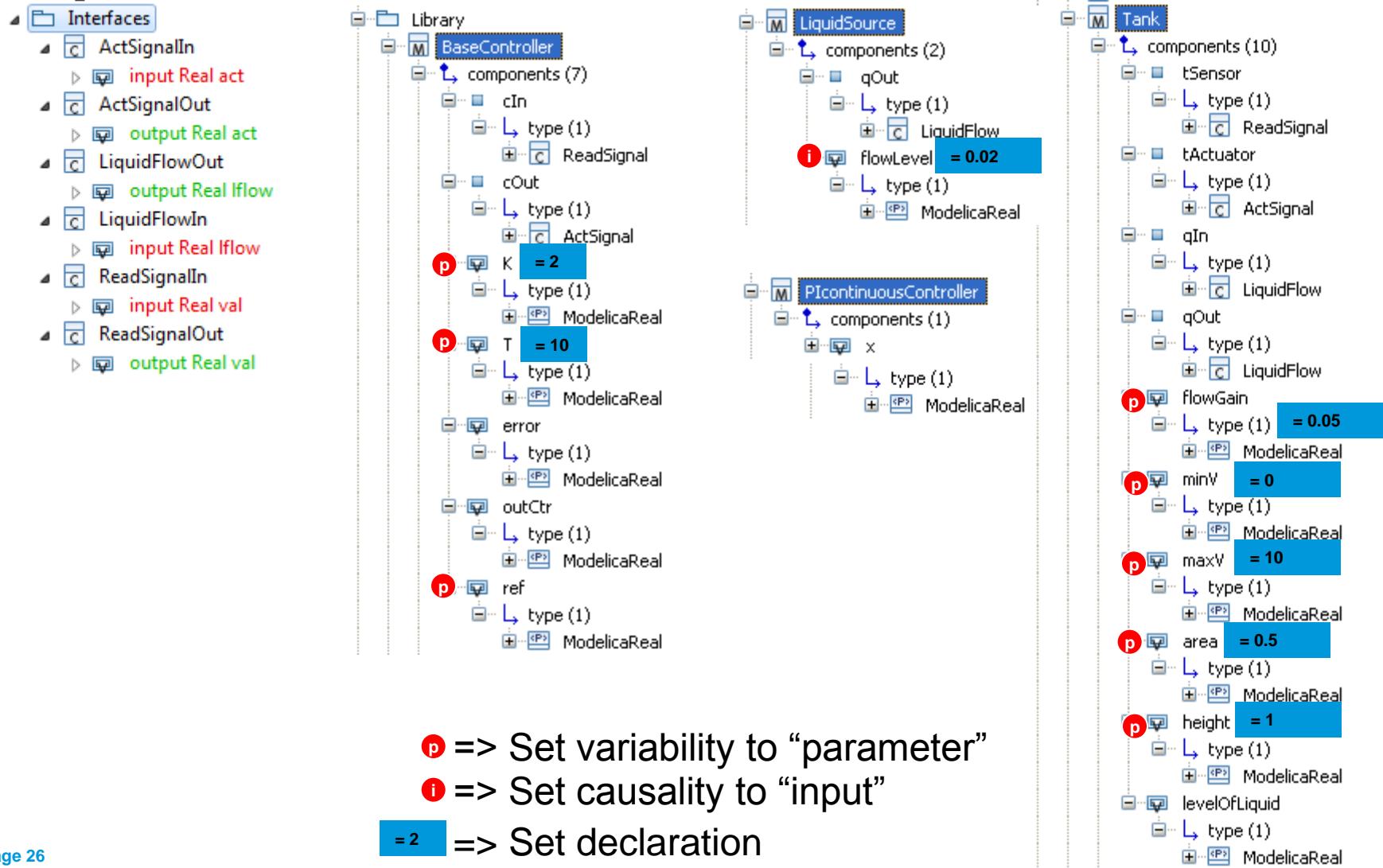
Properties for flowGain:

- Name:** flowGain
- Type:** <Primitive Type> ModelicaReal
- Variability:** parameter
- Causality:** <Unset> input output
- Flow Flag:** <Unset> flow stream
- Scope:** <Unset> inner outer
- Final:** true false
- Replaceable:** true false

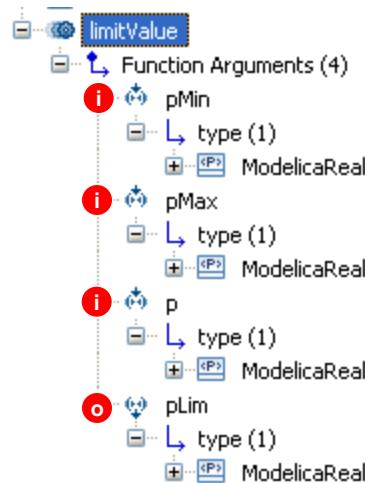
Declaration Tab: This tab is shown at the bottom left of the Properties view. It includes sections for UML, ModelicaML, Modifications, Cond. Expression, Array Subscripts, Profile, and Declaration. The "Declaration" section is highlighted with a red box and has an arrow pointing to it from the "flowGain" entry in the Model Explorer.

Value: = 0.05

Create Class Components



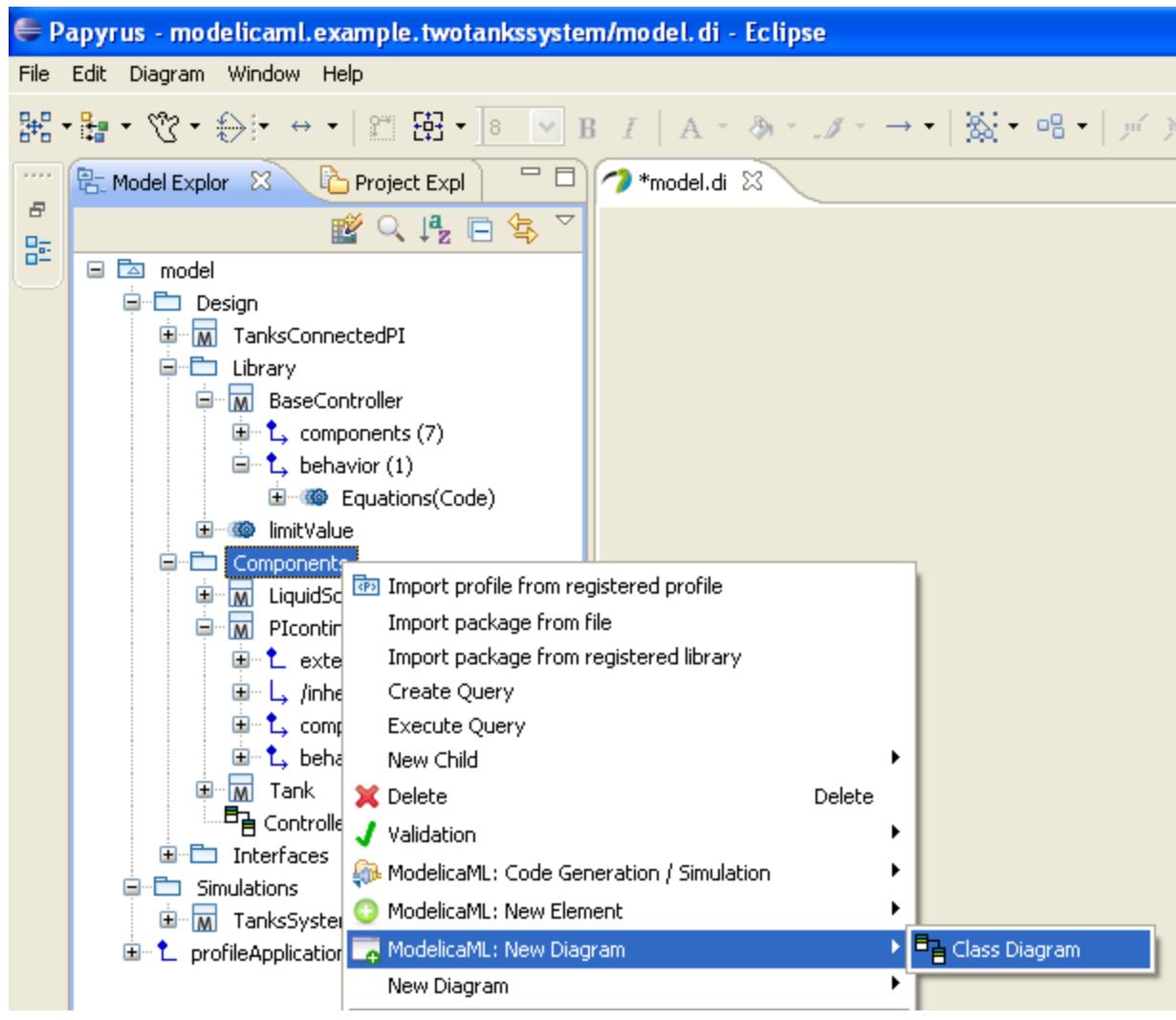
Create Function Arguments



- => Set causality to “input”
- => Set causality to “output”

Inheritance/Extension Modeling

Create Class Diagram

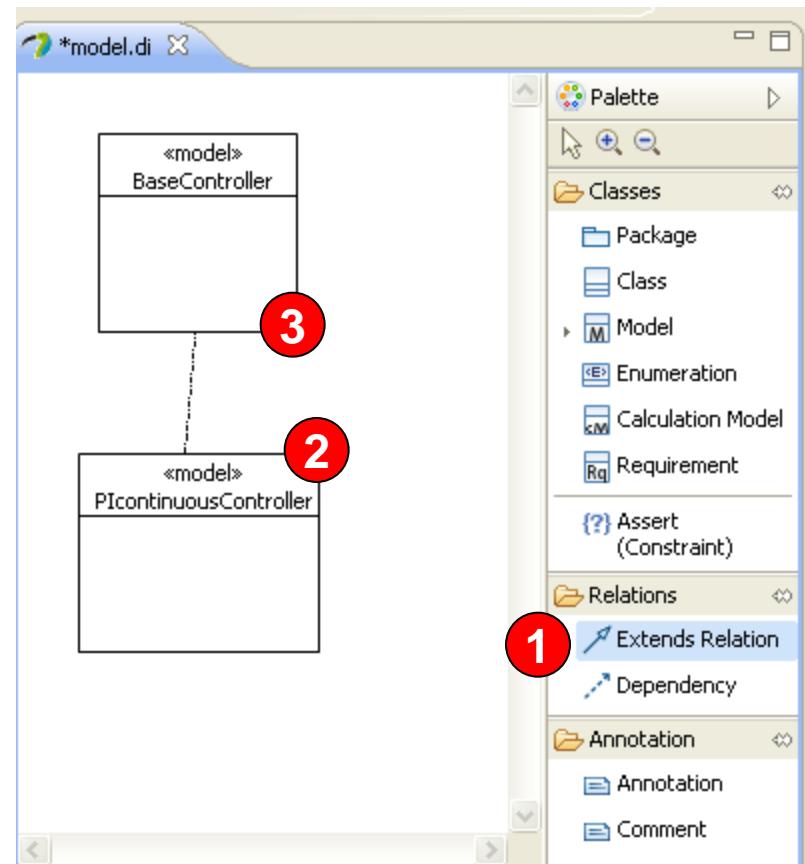




General: Working with diagrams

Creating edges:

- 1 Select the palette tool
- 2 Click on the edge source element and hold the mouse button
- 3 Move the mouse to the target element and release the mouse button

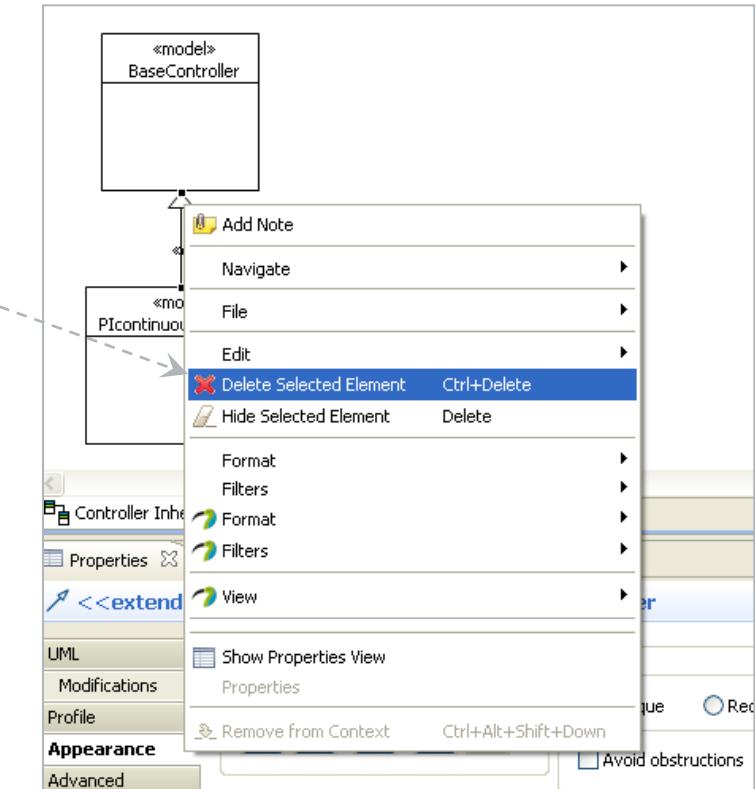




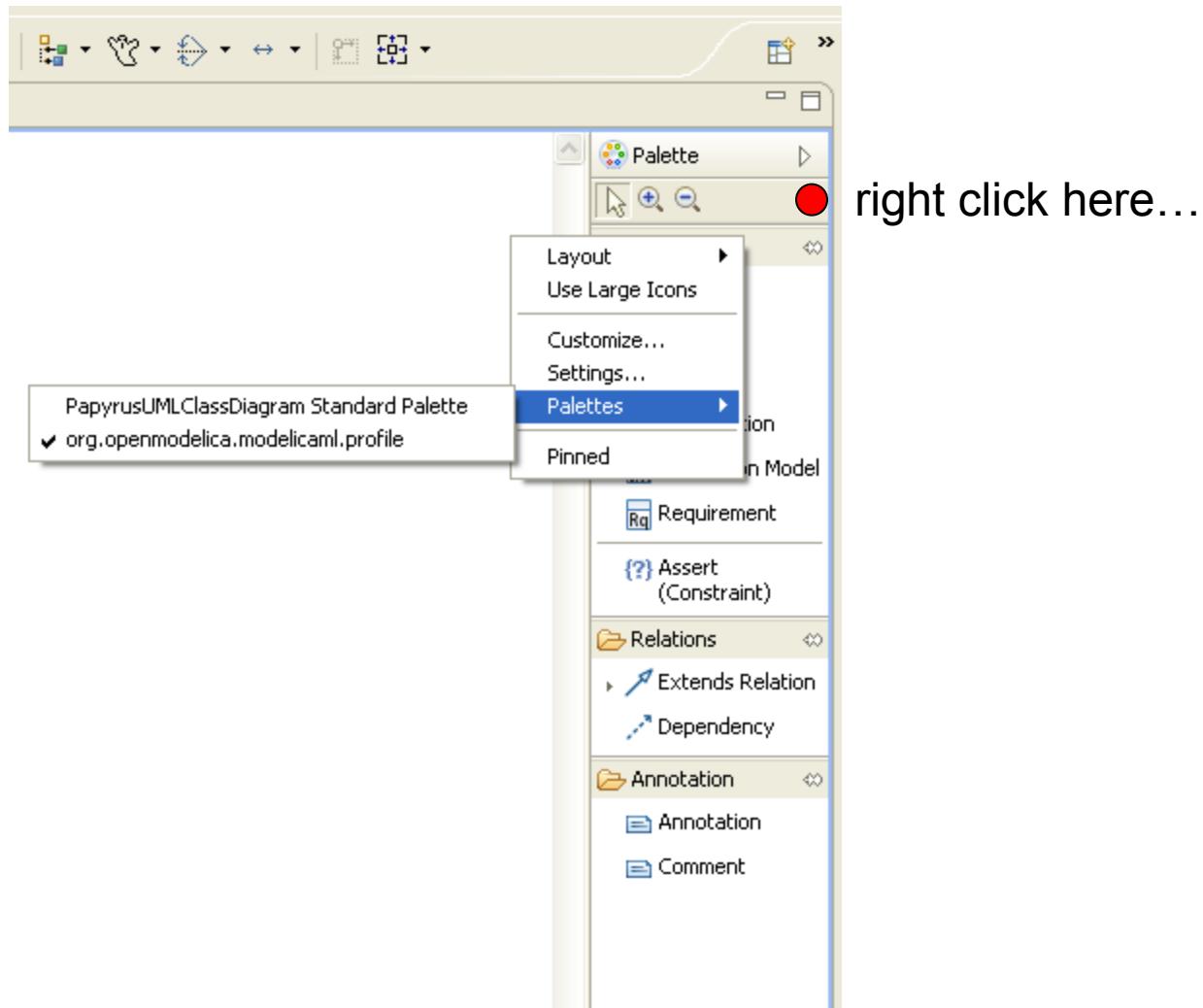
General: Working with diagrams

Deletion of elements:

- Right click on a diagram element
- (preferred) Select the option “Delete Selected Element” in order to delete it from the model. This is recommended in order to keep the model and the diagram consistent
- Select the option “Hide Selected Element” in order to remove the element from the diagram. The element will still exist in the model and can be shown on the diagram by drag&drop.

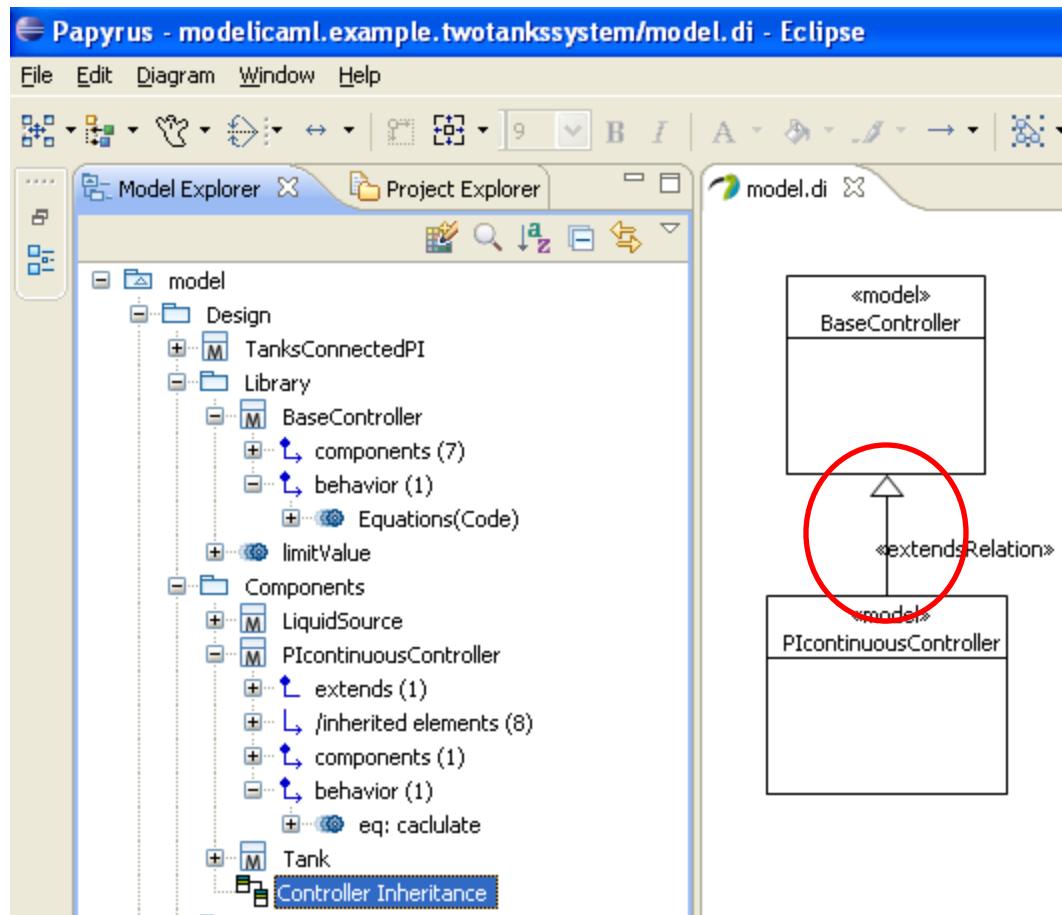


Configure Diagram Palette



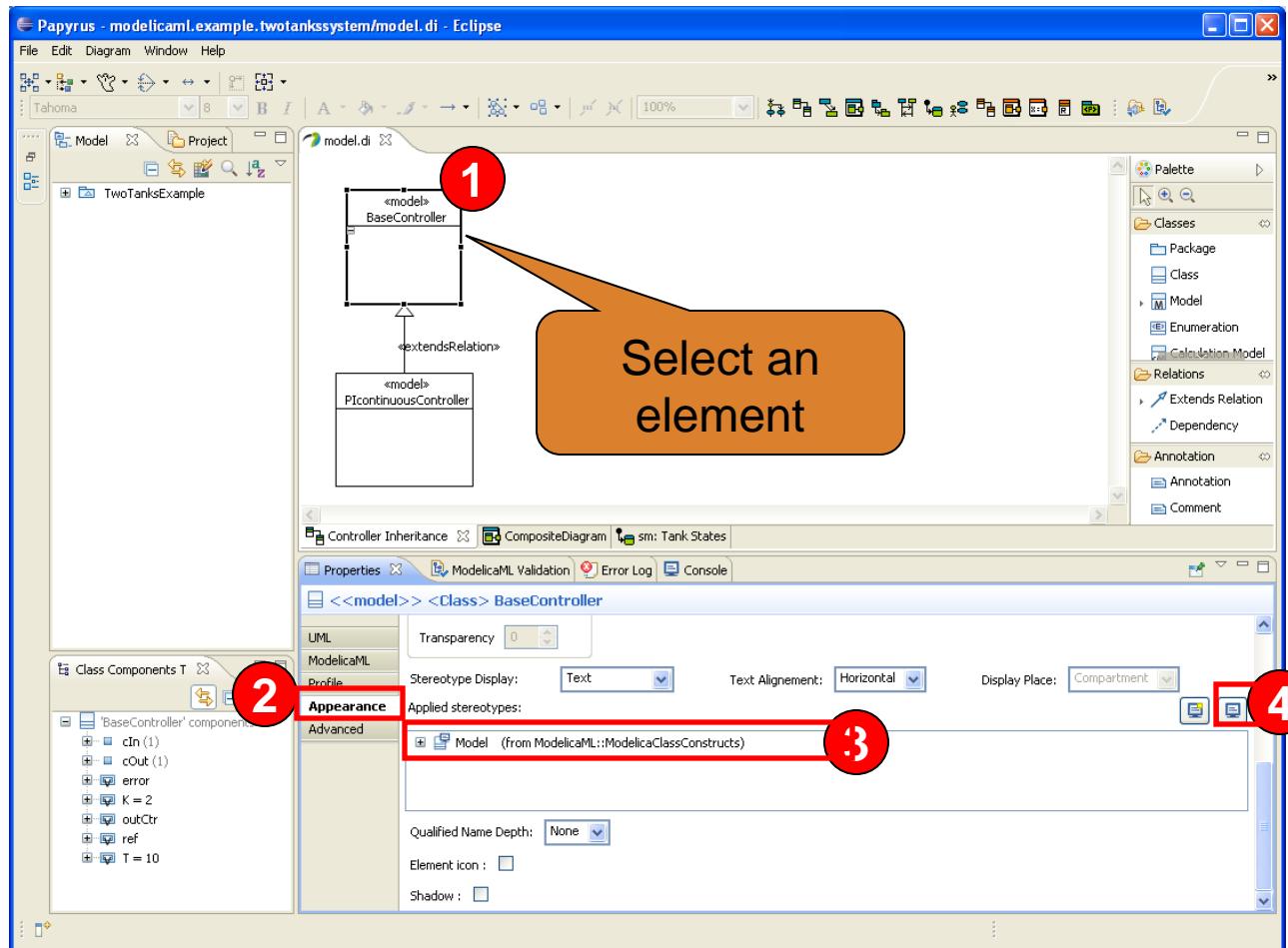
Create Extends Relation

- Drag & drop BaseController and PIcontinuousController onto diagram
- Use the palette tool “Extends Relation”



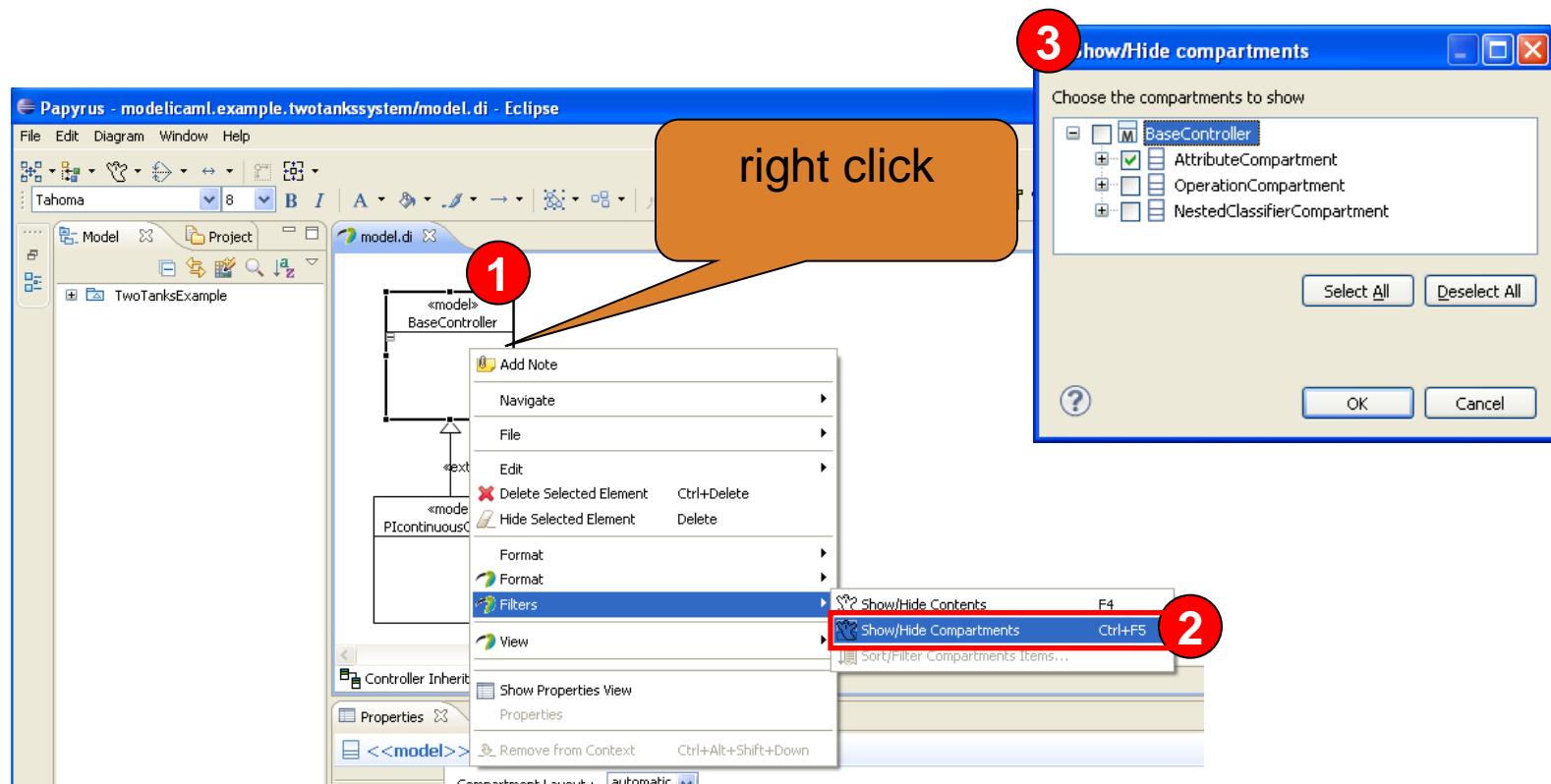


Hint: Element Appearance: Show stereotype name





Hint: Element Appearance: Compartments

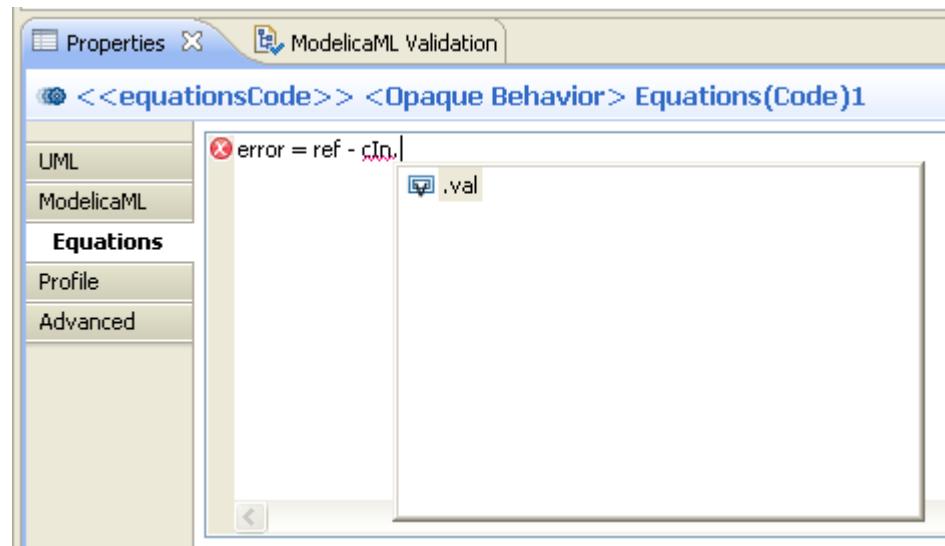


Behavior Modeling

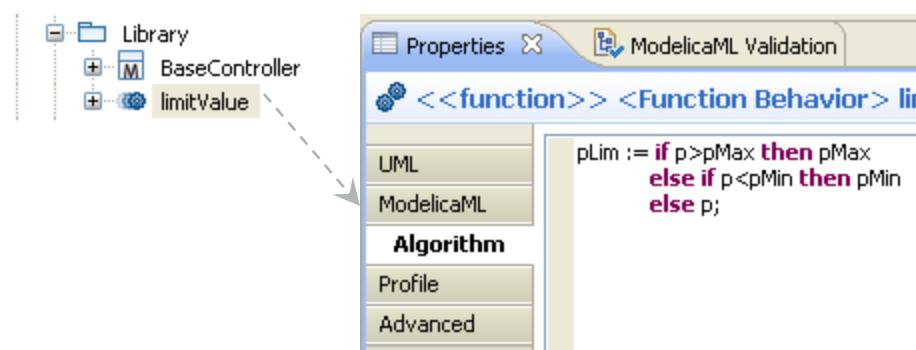
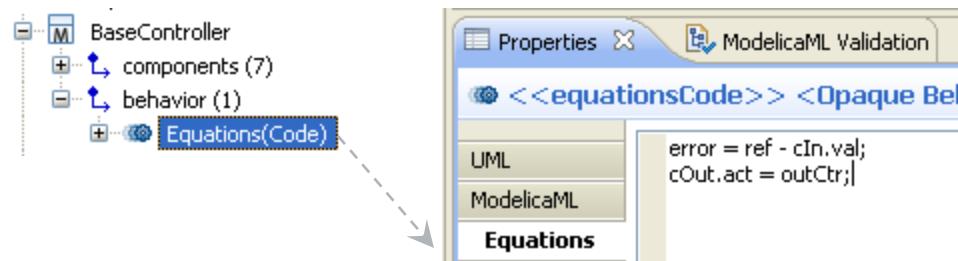


Hint: Editing Modelica Code

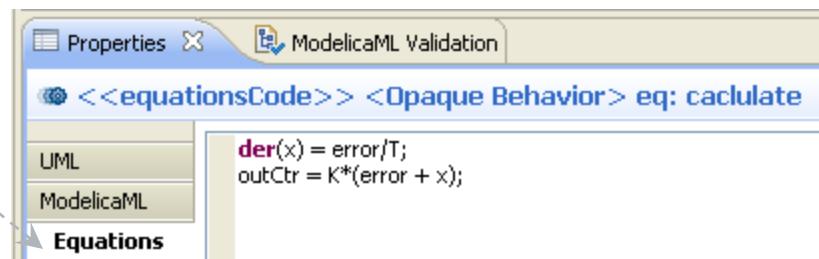
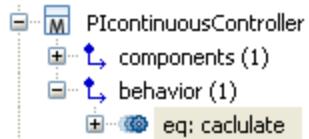
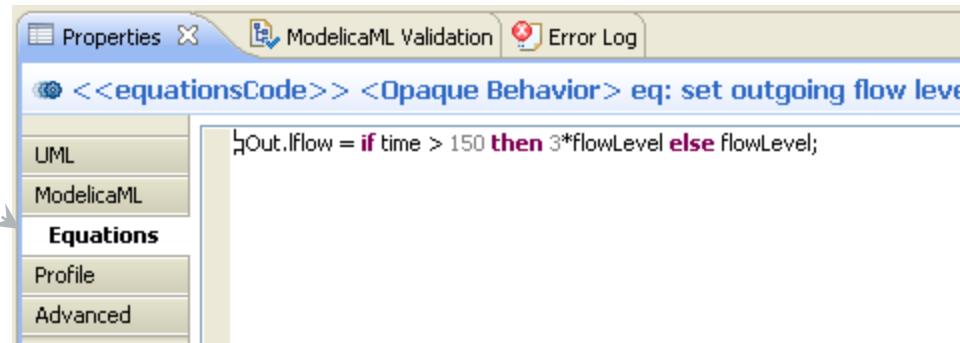
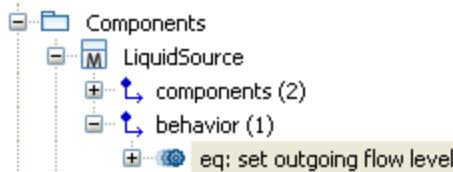
- Syntax highlighting and code completion is supported in code editors
- Hit **Ctrl + Space** for code completion when editing Modelica code



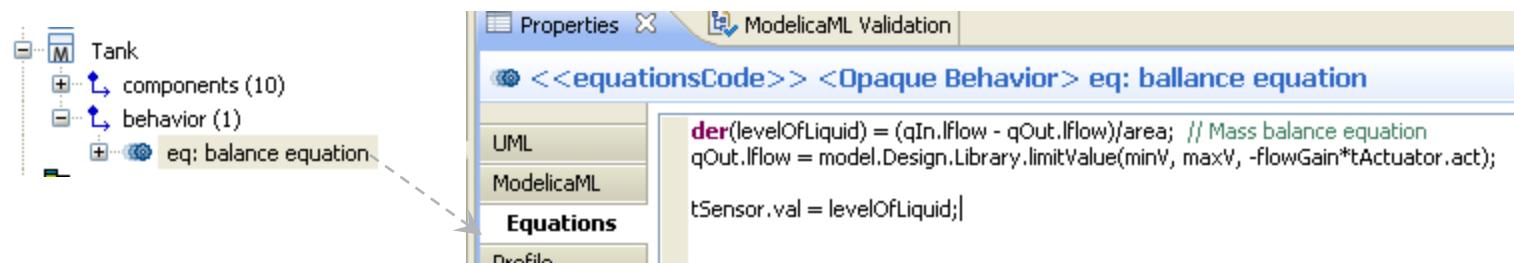
Create Behavior



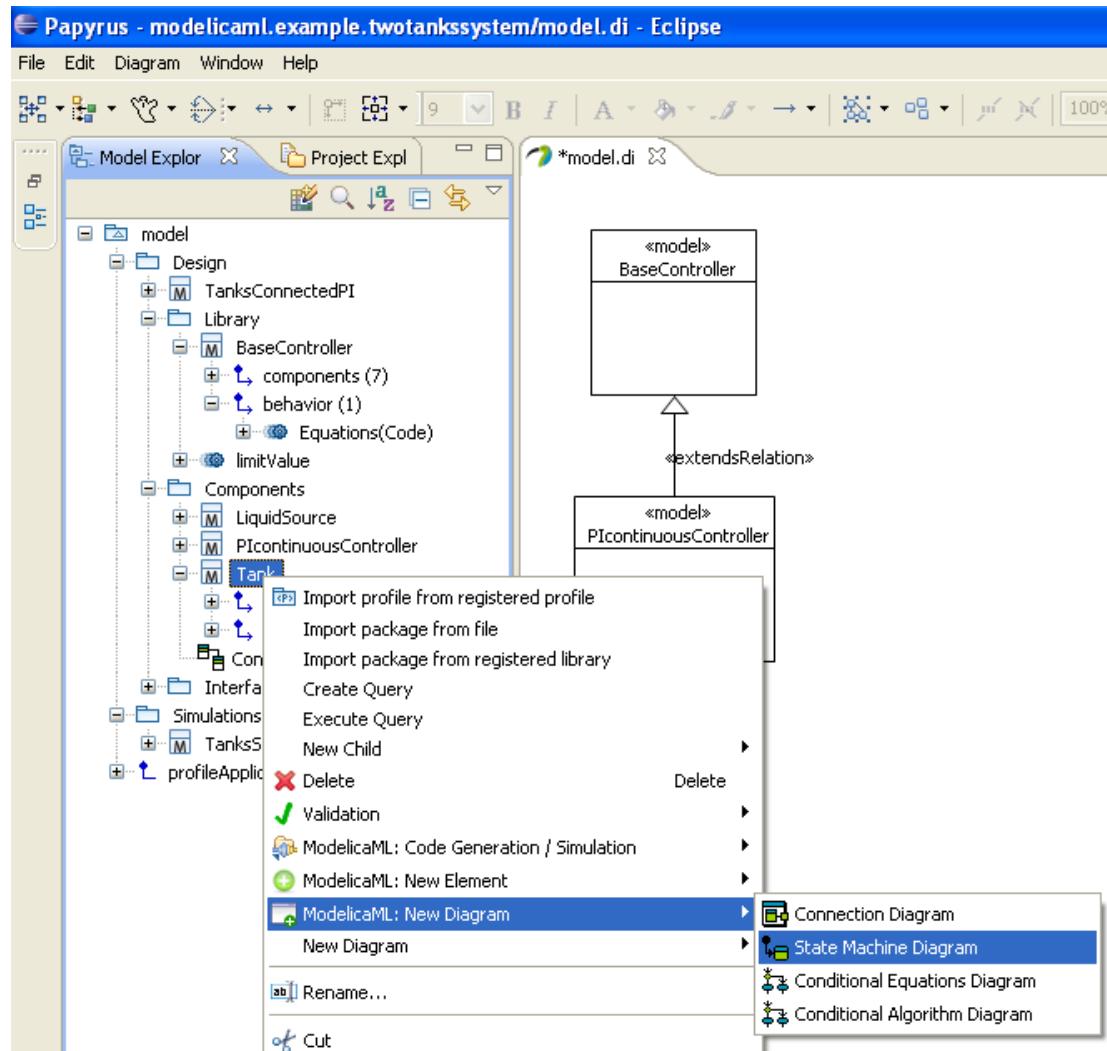
Create Behavior



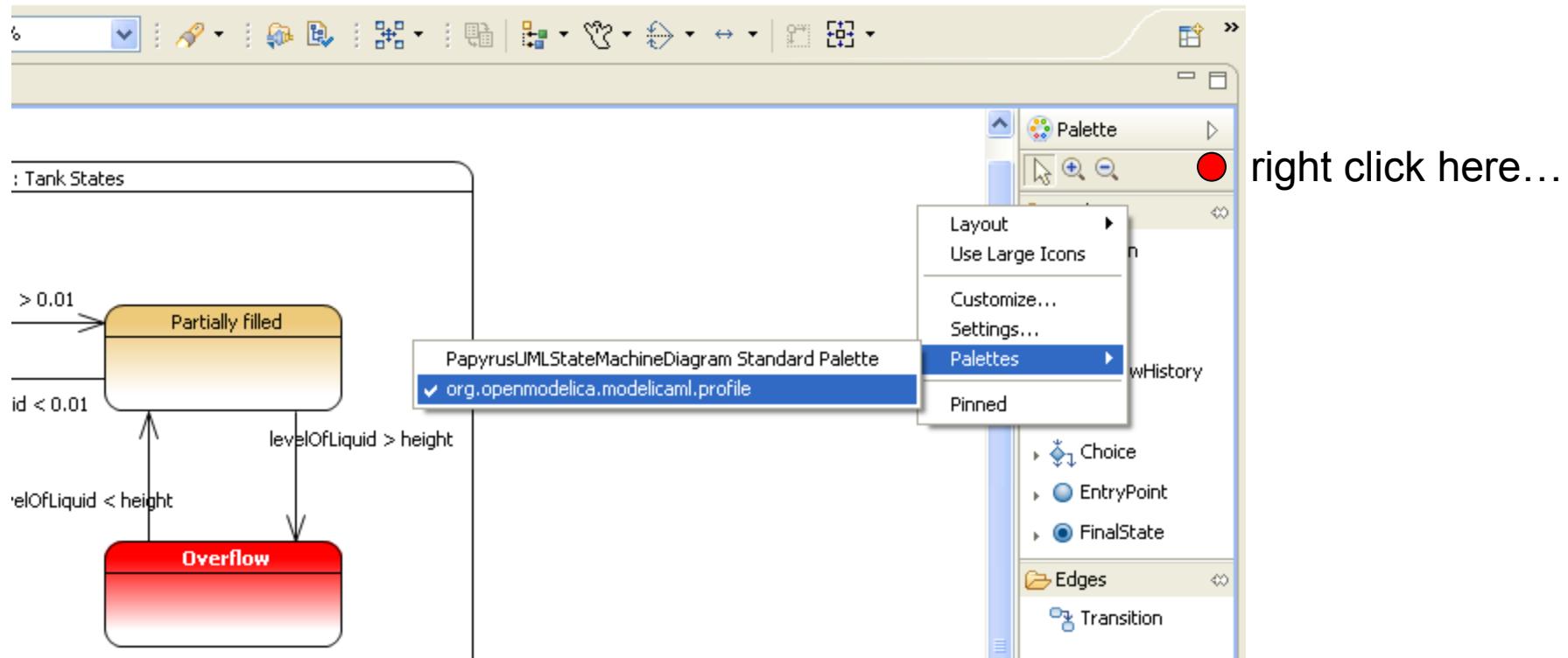
Create Behavior



Create State Machine



Configure Diagram Palette

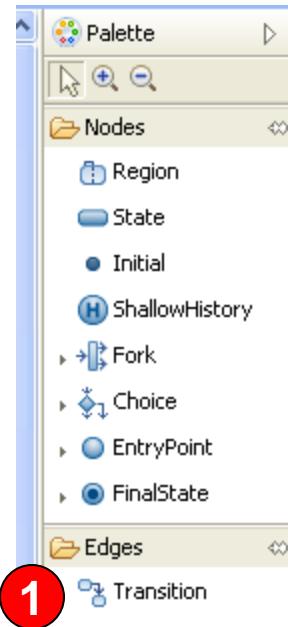
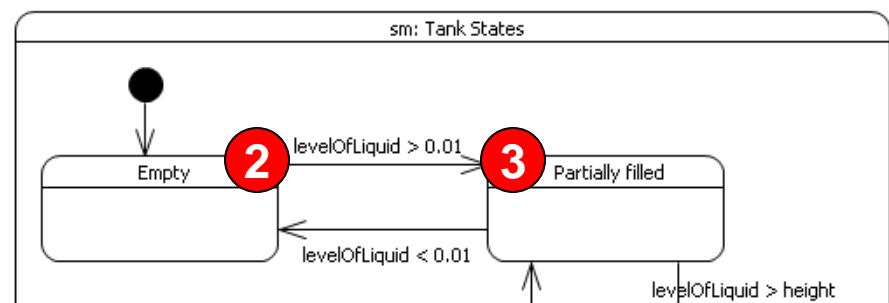




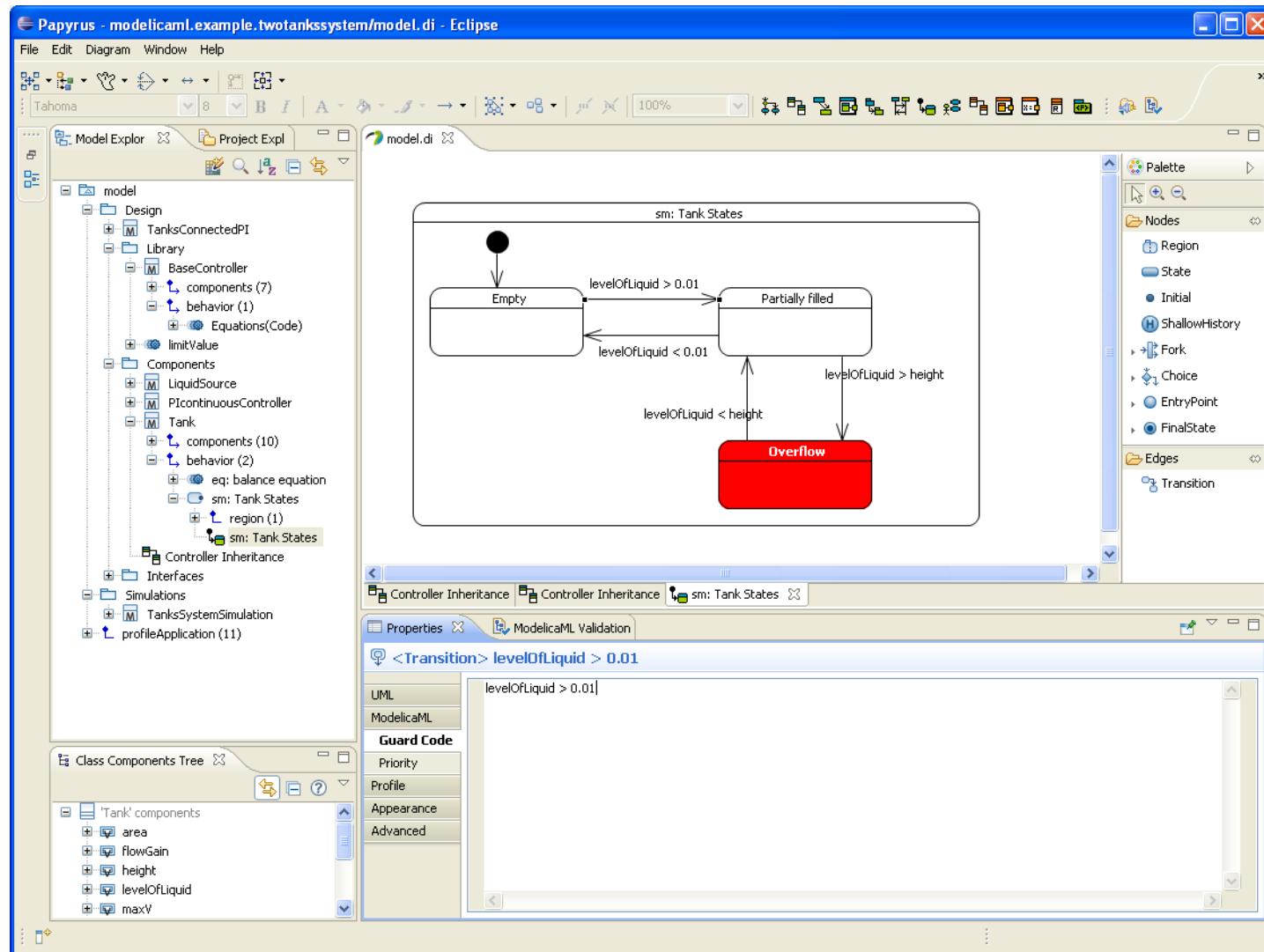
General: Working with diagrams

State Transitions:

- ① Select the palette tool
- ② Click on the transition source state (click on label) and hold the mouse button
- ③ Move the mouse to the target state (to its label) and release the mouse button



Create State Machine

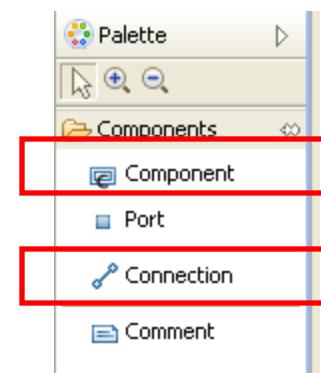


Architecture Modeling

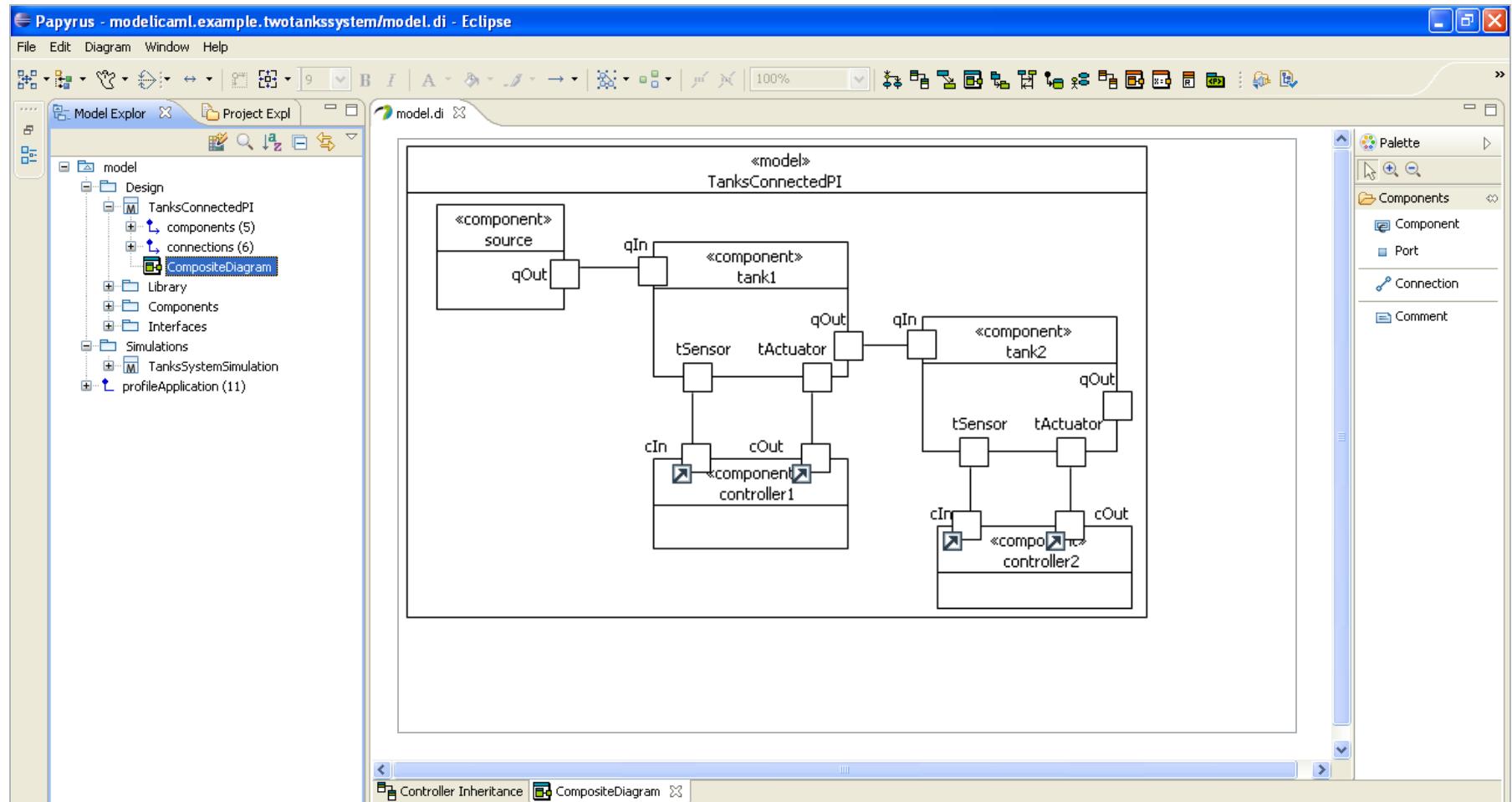


Create Connection Diagram

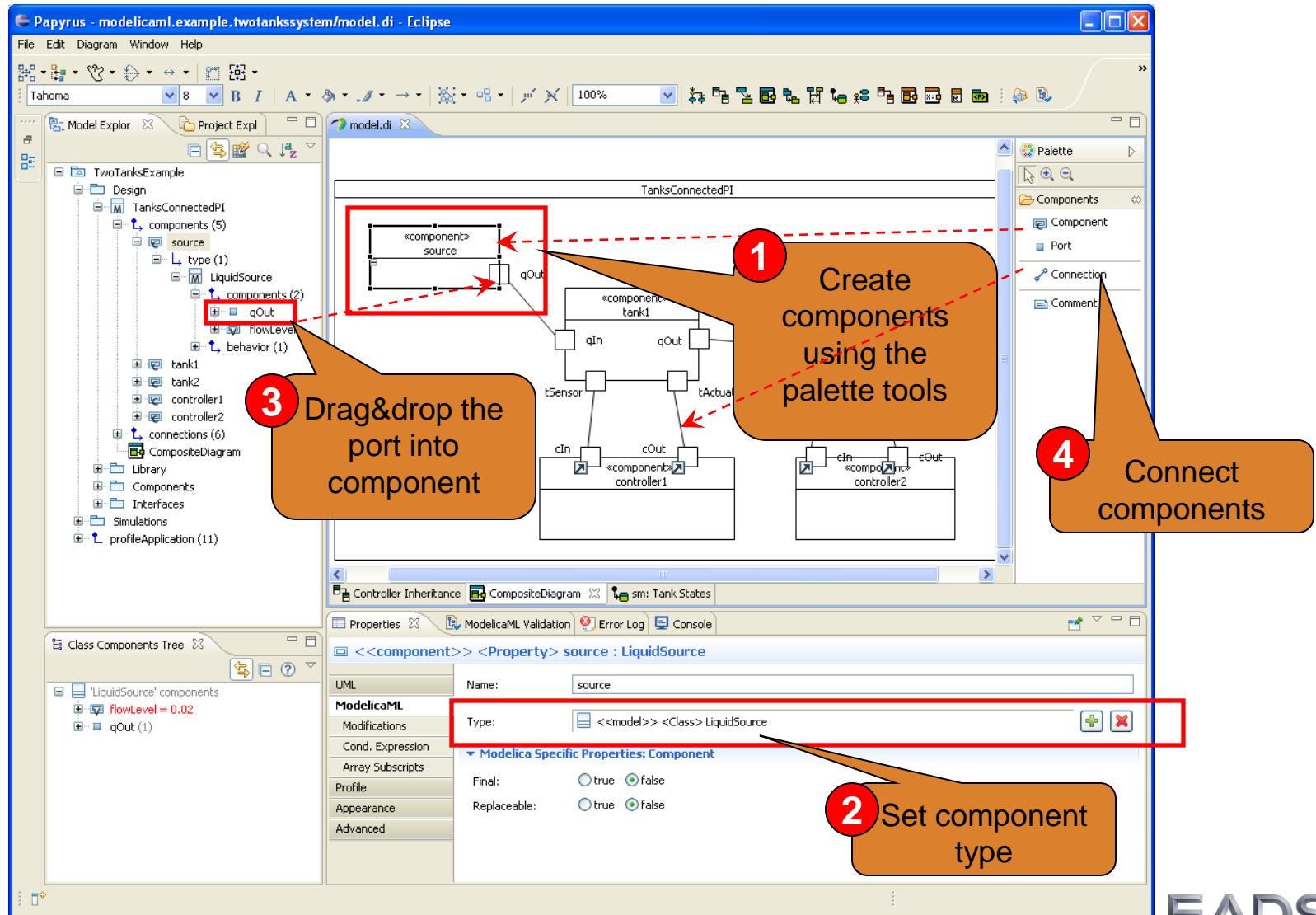
- Create a ModelicaML Connection Diagram under the TanksConnectedPI class
- Use components tool from the palette to create components inside the class on the diagram
- Define the types of components
- Use Model Explorer to find the ports
- Drag&Prop ports into respective components
- Arrange the components
- Use the “Connection” tool from the palette for connecting ports



Model System Architecture



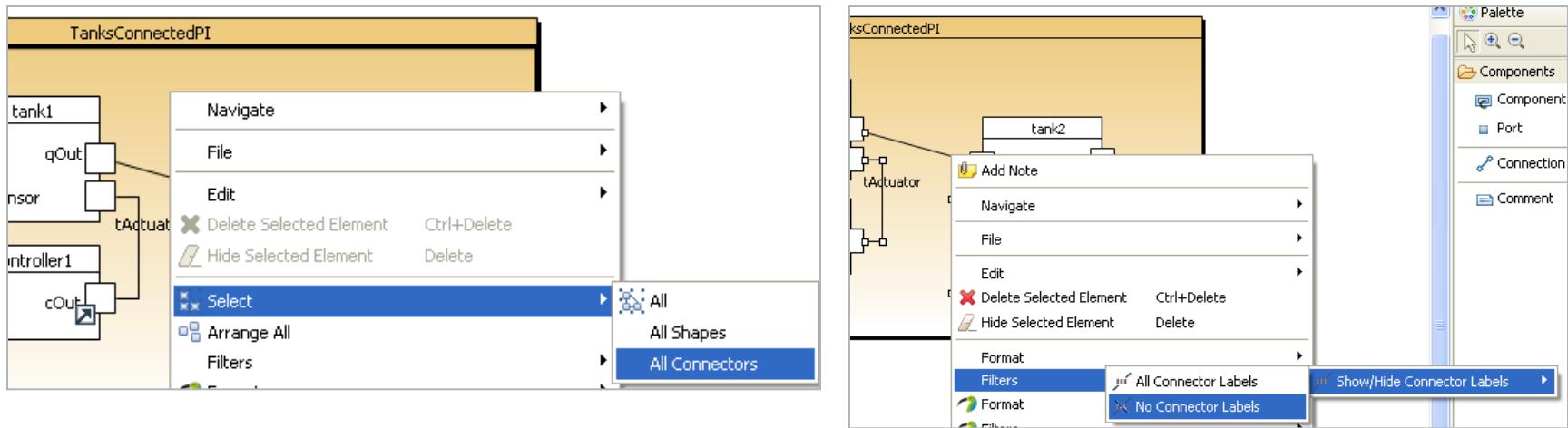
Create Connection Diagram





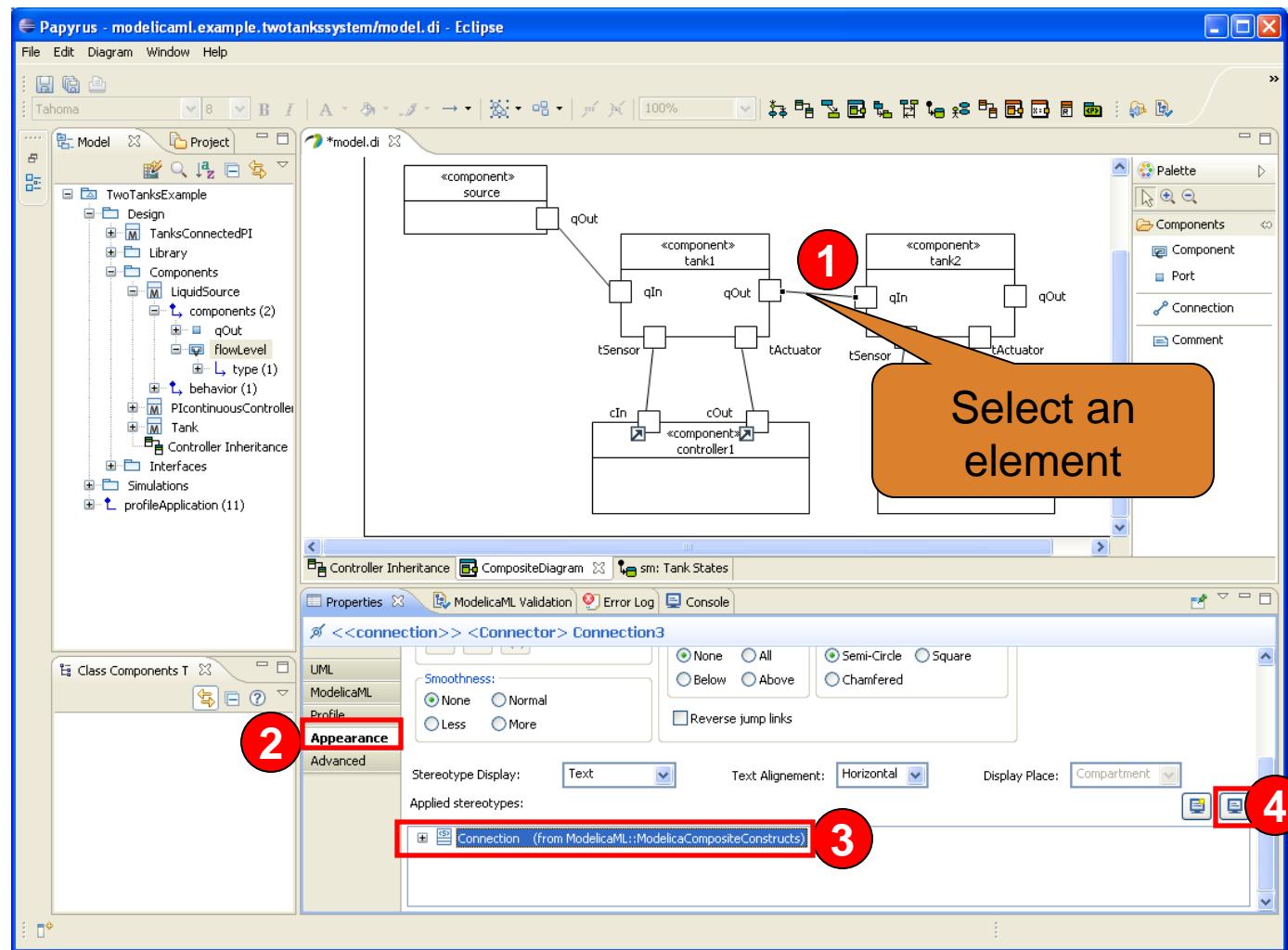
Hint: Hide the name of all connectors

- Click on the compartment of the class
- Right-click -> “Select” -> “All Connectors”
- Right-click on one of the selected connectors -> “Filters” -> “Show/Hide connector Labels” -> “No connector Labels”



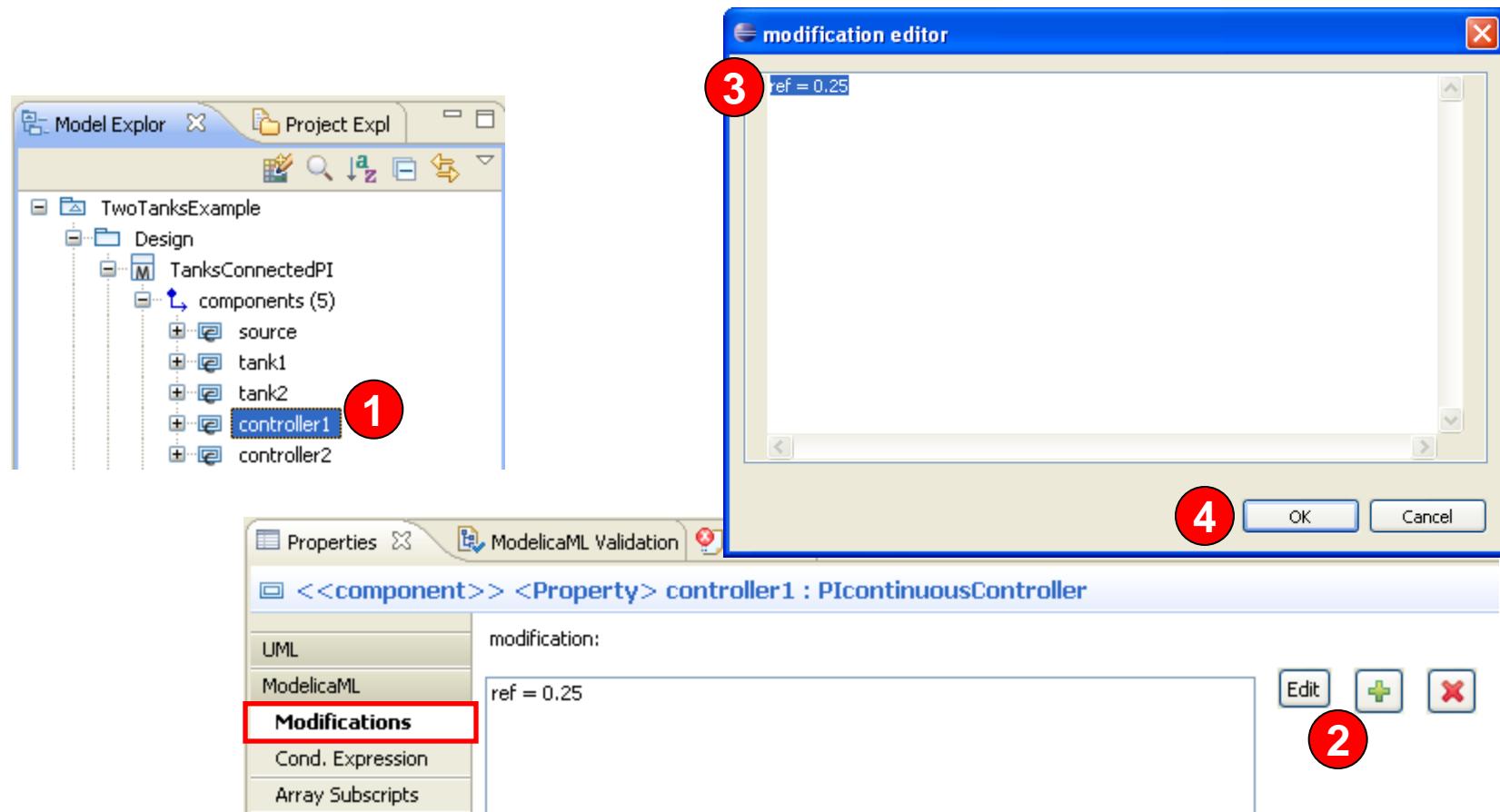


Hint: Element Appearance: Hide the name of the connection stereotype

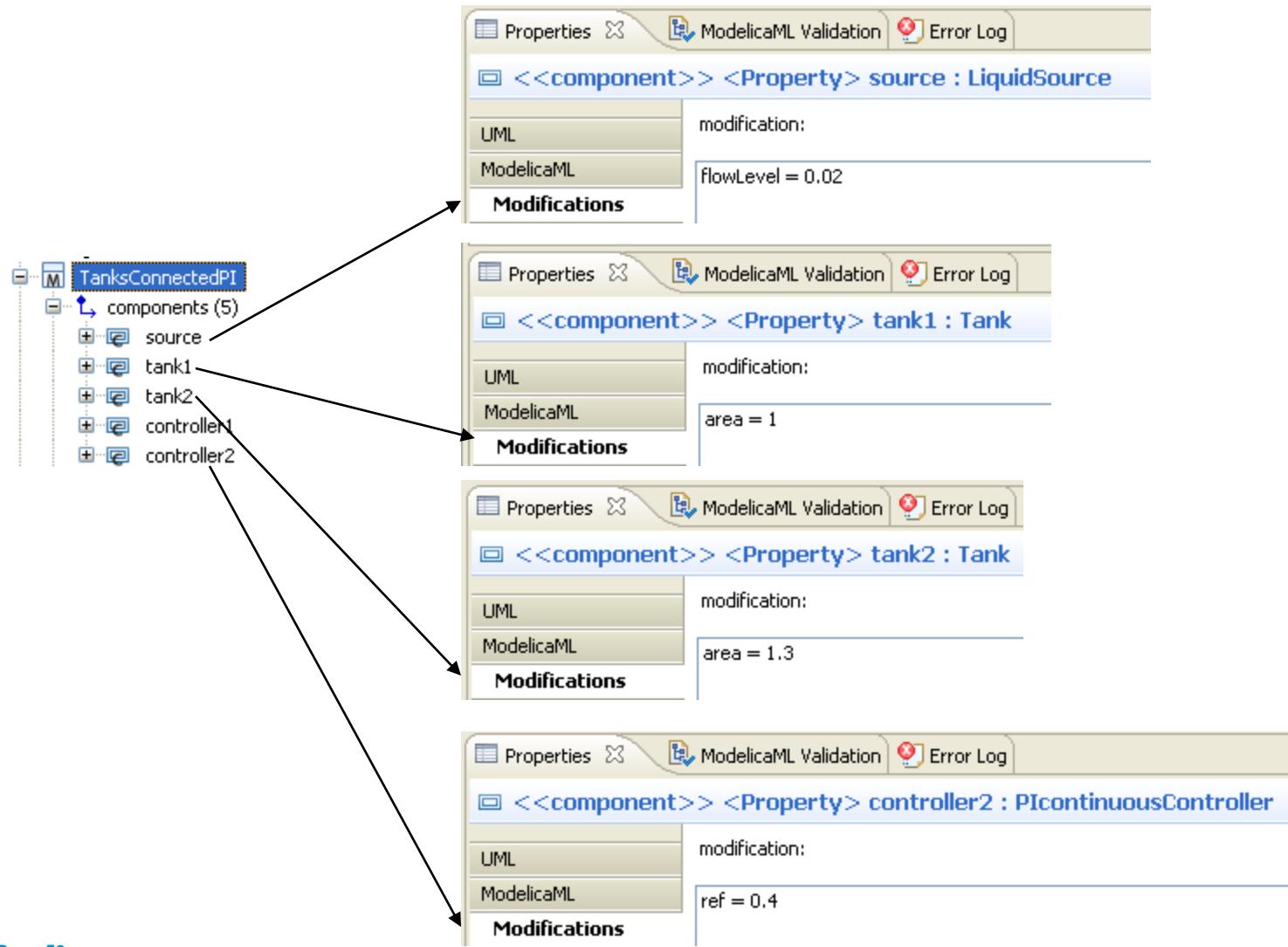


Component Modifications

Define Component Modifications

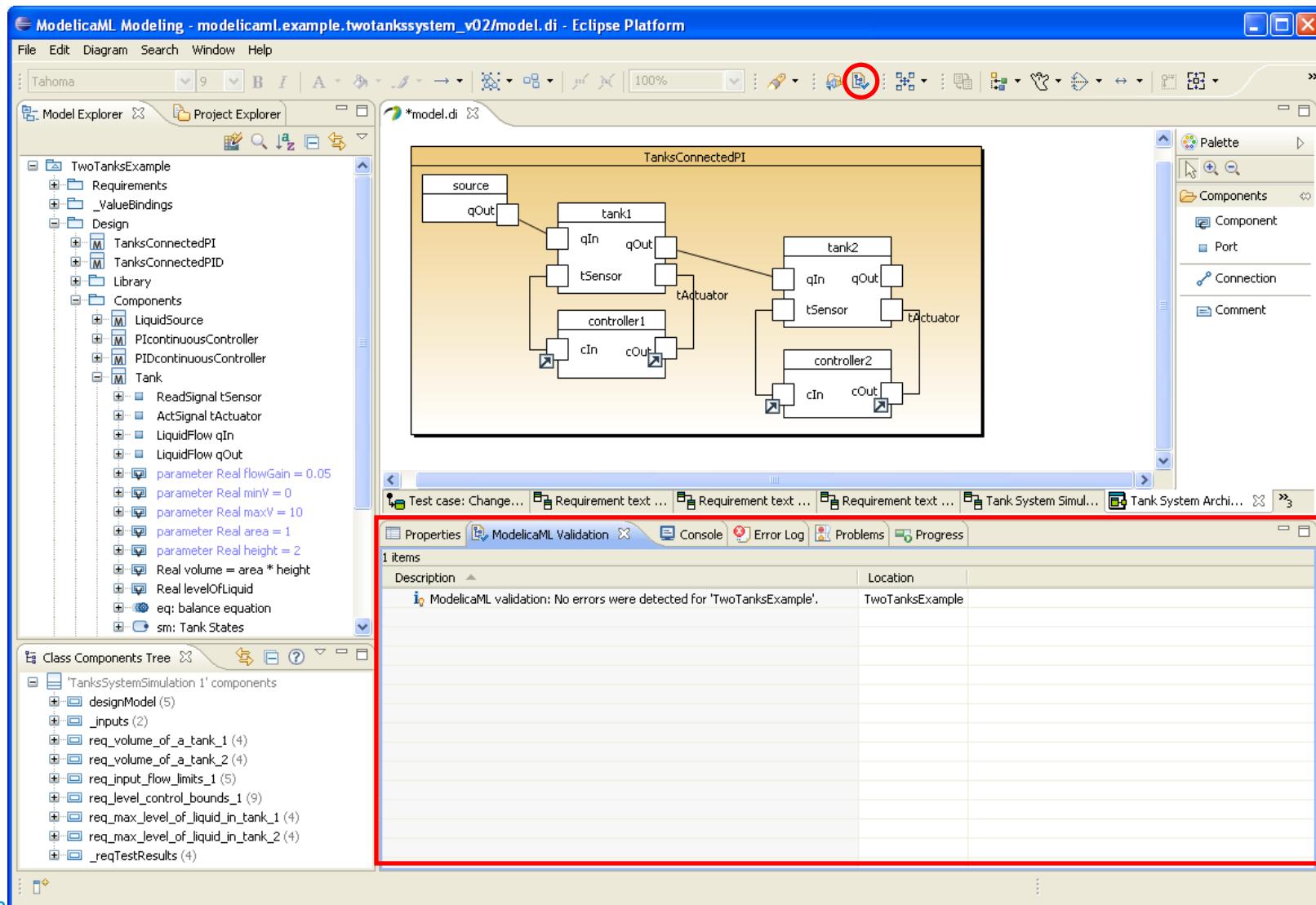


Define Component Modifications



Model Validation

Validate Model



Modelica Code Generation



Launch Modelica Code Generation

ModelicaML Modeling - modelicaml.example.twotankssystem_v02/model.di - Eclipse Platform

File Edit Diagram Search Window Help

Tahoma 9 B I A 100% 100%

Model Explorer Project Explorer

ModelicaML Validation Console Error Log Problems Progress

Pa 1 items selected

TwoTanksExample Requirements

- Rq Volume of a tank
 - input Real inputFlowLevelTank1
 - constant Real min = 0
 - constant Real max = 0.08
 - output Boolean violated = inputFlowLevelTank1 < min or in
 - output Boolean evaluated = true
- Comments (1)
- Requirement text and measurable properties
- Rq Max level of liquid in tank
- Rq Settling time and bounds after a change of input flow
- _ValueBindings
- Design

 - M TanksConnectedPI
 - M TanksConnectedPID
 - Library
 - Components
 - M LiquidSource
 - M PIcontinuousController
 - M PIDcontinuousController
 - M Tank
 - Controller Inheritance
 - Interfaces

sm: Tank States

```
graph TD; Empty -- "levelOfLiquid > 0.01" --> PartiallyFilled; PartiallyFilled -- "levelOfLiquid < 0.01" --> Overflow; Overflow -- "levelOfLiquid < height" --> PartiallyFilled; Overflow -- "levelOfLiquid > height" --> Empty;
```

Nodes

- Region
- State
- Initial
- ShallowHistory
- Fork
- Choice
- EntryPoint
- FinalState

Edges

- Transition

Properties

<<requirement>> <Class> Input flow limits

UML	Name:	Input flow limits
ModelicaML	Is Abstract:	<input type="radio"/> true <input checked="" type="radio"/> false
Profile	Is Leaf:	<input type="radio"/> true <input checked="" type="radio"/> false
Advanced	Visibility:	public
	Use Case:	

DS



Generated Modelica Code

ModelicaML Modeling - modelicaml.example.twotankssystem_v02/model.di - Eclipse Platform

File Edit Diagram Search Window Help

Tahoma 100%

Model Explorer Project Explorer

*model.di

sm: Tank States

```
graph TD; Empty -- "levelOfLiquid > 0.01" --> PartiallyFilled; PartiallyFilled -- "levelOfLiquid < 0.01" --> Overflow; Overflow -- "levelOfLiquid < height" --> PartiallyFilled; Overflow -- "levelOfLiquid > height" --> Empty;
```

Palette

Nodes

- Region
- State
- Initial
- ShallowHistory
- Fork
- Choice
- EntryPoint
- FinalState

Edges

- Transition

Requirement text ... Requirement text ... Tank System Simul... Controller Inheri... sm: Tank States

Properties ModelicaML Validation Console Error Log Problems Progress

model.di - modelicaml.example.twotankssystem_v02

Resource

Property	Value
Info	derived editable last modified linked location name path size
Info	false true 16. März 2011 15:07:01 false D:_PROJECTS\2008_PhD\tools\eclipse_3_6_modeling\runtime-New_config.. model.di /modelicaml.example.twotankssystem_v02/model.di 2.971 bytes

Class Components Tree

'TanksSystemSimulation 1' components

- designModel (5)
- _inputs (2)
- req_volume_of_a_tank_1 (4)
- req_volume_of_a_tank_2 (4)
- req_input_flow_limits_1 (5)
- req_level_control_bounds_1 (9)

model.di - modelicaml.example.twotankssystem_v02

EADS