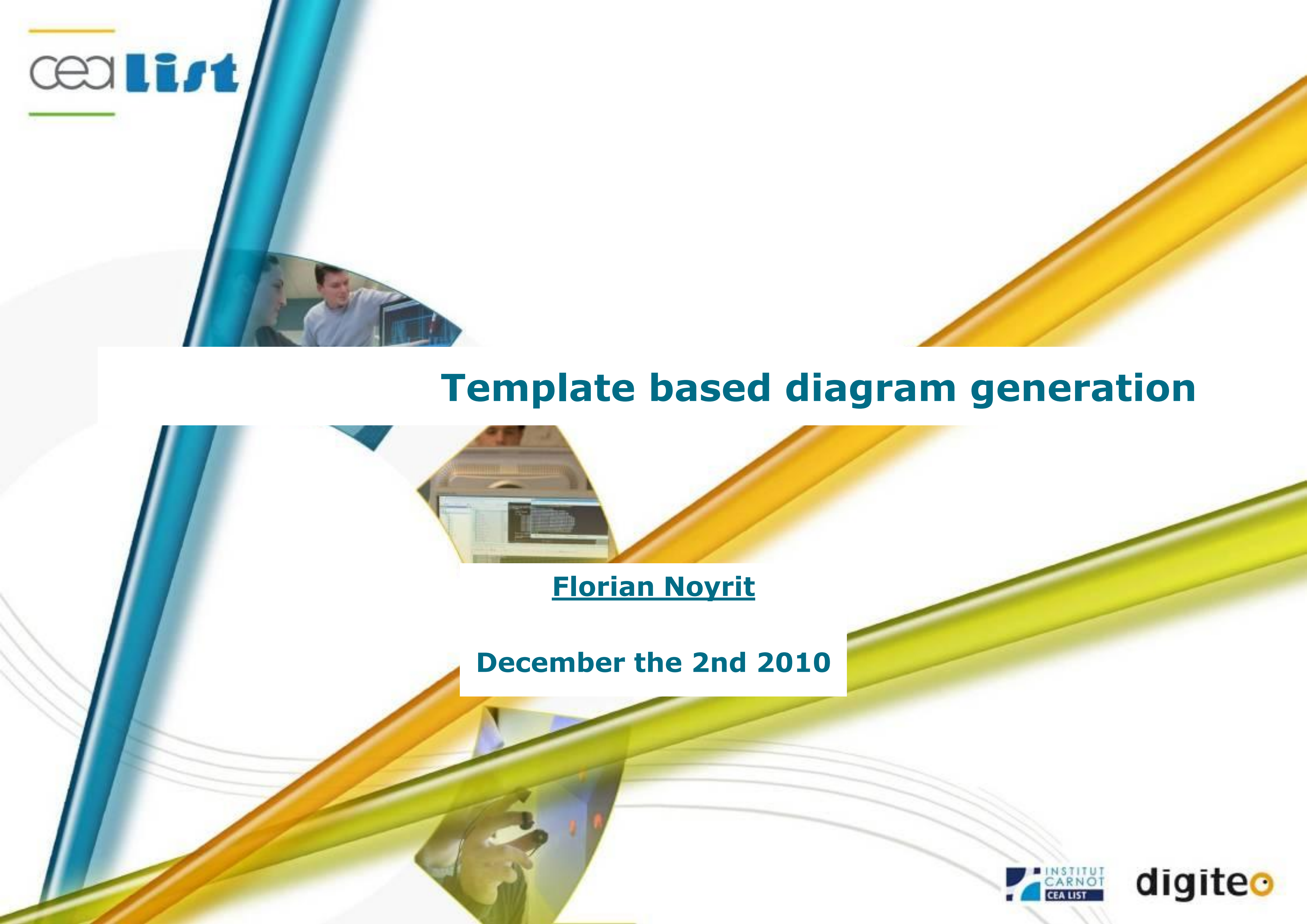
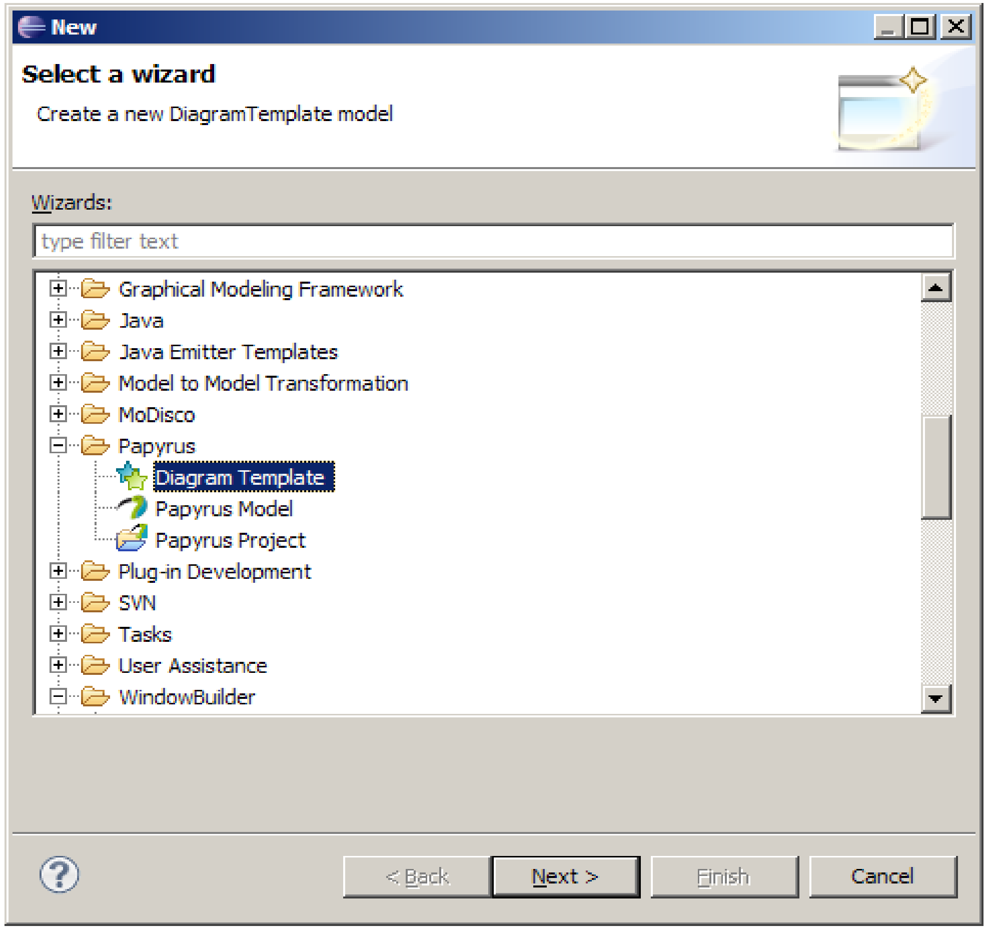


Template based diagram generation

Florian Noyrit

December the 2nd 2010

Create a new diagram template



Select a wizard

Create a new DiagramTemplate model

Wizards:

type filter text

- Graphical Modeling Framework
- Java
- Java Emitter Templates
- Model to Model Transformation
- MoDisco
- Papyrus
  - Diagram Template
  - Papyrus Model
  - Papyrus Project
- Plug-in Development
- SVN
- Tasks
- User Assistance
- WindowBuilder

Editor overview

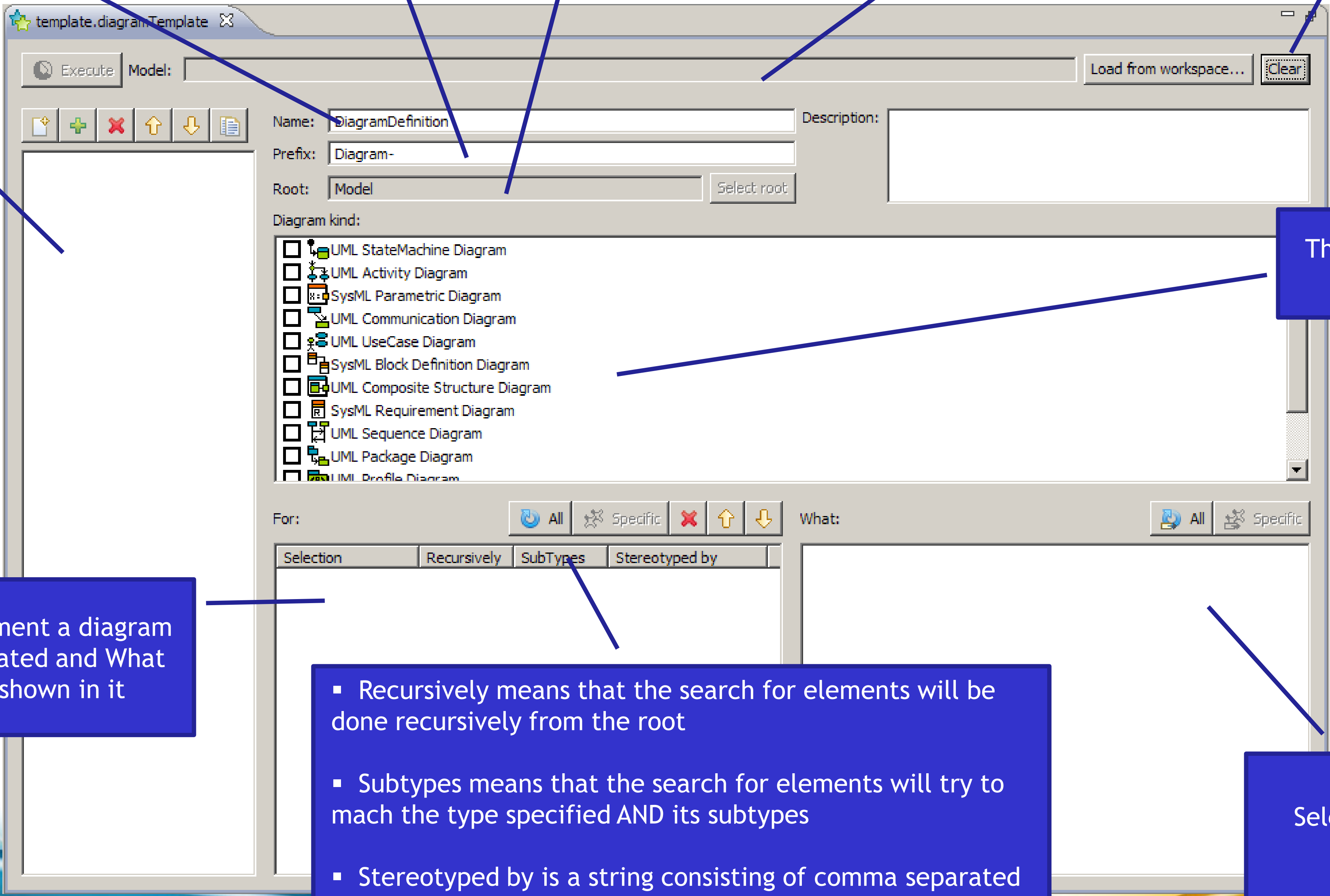
Name of the diagram definition

Prefix used to name the diagram: prefix+name of the owner

Root to start search

The model to apply the template on

Clear specific information: remove application on a specific model



Model:

Name: DiagramDefinition

Prefix: Diagram-

Root: Model

Description:

Diagram kind:

- ☐ UML StateMachine Diagram
- ☐ UML Activity Diagram
- ☐ SysML Parametric Diagram
- ☐ UML Communication Diagram
- ☐ UML UseCase Diagram
- ☐ SysML Block Definition Diagram
- ☐ UML Composite Structure Diagram
- ☐ SysML Requirement Diagram
- ☐ UML Sequence Diagram
- ☐ UML Package Diagram
- ☐ UML Profile Diagram

For:

Selection	Recursively	SubTypes	Stereotyped by

What:

Diagram definitions

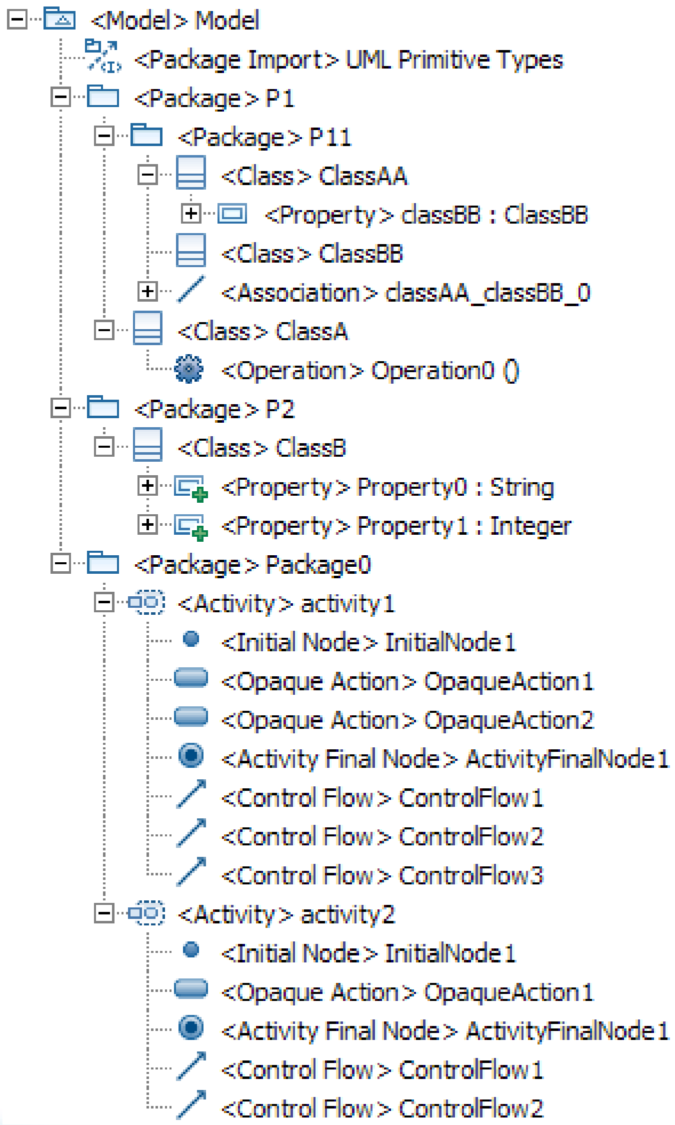
The kind of diagram to create

For which element a diagram should be created and What should be shown in it

- Recursively means that the search for elements will be done recursively from the root
- Subtypes means that the search for elements will try to mach the type specified AND its subtypes
- Stereotyped by is a string consisting of comma separated qualified names of stereotypes (e.g. SysML::Blocks::Block). The search will try to match this stereotype applications

Select what should be shown

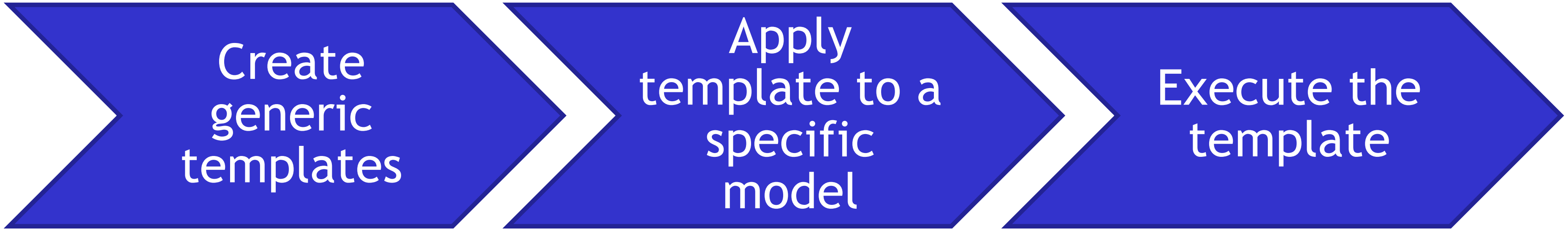
Case study



```
<Model> Model
    <Package Import> UML Primitive Types
    <Package> P1
        <Package> P11
            <Class> ClassAA
                <Property> classBB : ClassBB
            <Class> ClassBB
            <Association> classAA_classBB_0
        <Class> ClassA
            <Operation> Operation0 ()
    <Package> P2
        <Class> ClassB
            <Property> Property0 : String
            <Property> Property1 : Integer
    <Package> Package0
        <Activity> activity1
            <Initial Node> InitialNode1
            <Opaque Action> OpaqueAction1
            <Opaque Action> OpaqueAction2
            <Activity Final Node> ActivityFinalNode1
            <Control Flow> ControlFlow1
            <Control Flow> ControlFlow2
            <Control Flow> ControlFlow3
        <Activity> activity2
            <Initial Node> InitialNode1
            <Opaque Action> OpaqueAction1
            <Activity Final Node> ActivityFinalNode1
            <Control Flow> ControlFlow1
            <Control Flow> ControlFlow2
```

- Create class diagrams for all packages
  - Show classes
    - Show attributes and operations
  - Show associations
- Create an activity diagram for activity1
  - Show ControlFlows

Template editor



Create generic templates

Apply template to a specific model

Execute the template

Create generic templates
Create class diagrams for all packages

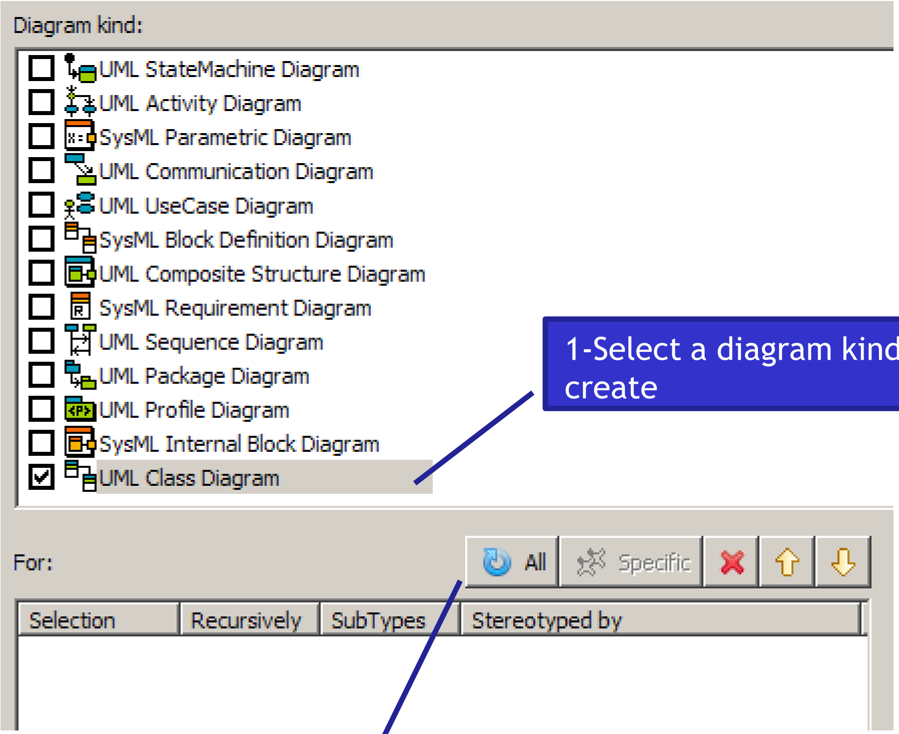


Diagram kind:

- ☐ UML StateMachine Diagram
- ☐ UML Activity Diagram
- ☐ SysML Parametric Diagram
- ☐ UML Communication Diagram
- ☐ UML UseCase Diagram
- ☐ SysML Block Definition Diagram
- ☐ UML Composite Structure Diagram
- ☐ SysML Requirement Diagram
- ☐ UML Sequence Diagram
- ☐ UML Package Diagram
- ☐ UML Profile Diagram
- ☐ SysML Internal Block Diagram
- ☑ UML Class Diagram

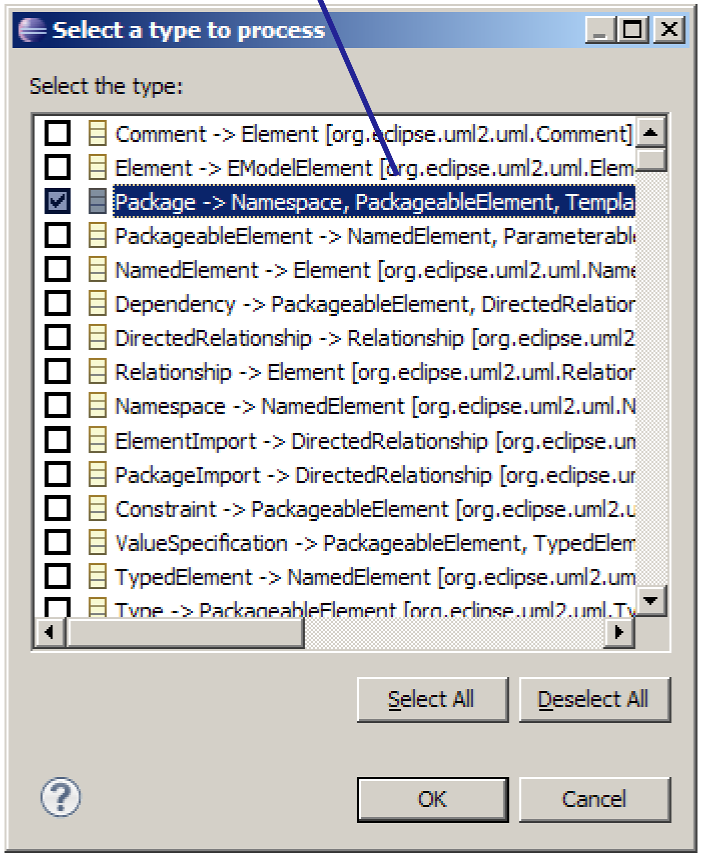
For:

Selection	Recursively	SubTypes	Stereotyped by

1-Select a diagram kind to create

2-Select “All”

3-Select the type of the element you want to create diagrams for. Here, select Package



Select a type to process

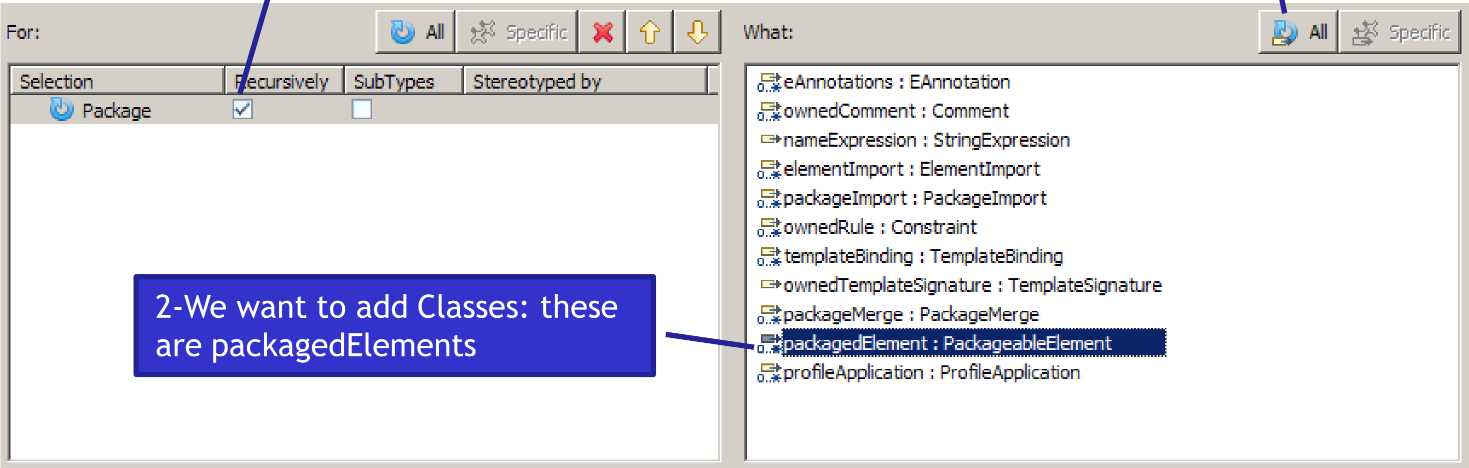
Select the type:

- ☐ Comment -> Element [org.eclipse.uml2.uml.Comment]
- ☐ Element -> EModelElement [org.eclipse.uml2.uml.Elem
- ☑ Package -> Namespace, PackageableElement, Templa
- ☐ PackageableElement -> NamedElement, Parameterabl
- ☐ NamedElement -> Element [org.eclipse.uml2.uml.Name
- ☐ Dependency -> PackageableElement, DirectedRelatior
- ☐ DirectedRelationship -> Relationship [org.eclipse.uml2
- ☐ Relationship -> Element [org.eclipse.uml2.uml.Relatior
- ☐ Namespace -> NamedElement [org.eclipse.uml2.uml.N
- ☐ ElementImport -> DirectedRelationship [org.eclipse.un
- ☐ PackageImport -> DirectedRelationship [org.eclipse.ur
- ☐ Constraint -> PackageableElement [org.eclipse.uml2.u
- ☐ ValueSpecification -> PackageableElement, TypedElem
- ☐ TypedElement -> NamedElement [org.eclipse.uml2.um
- ☐ Type -> PackageableElement [org.eclipse.uml2.uml.Ty

Create generic templates
Show classes

1-We want to create diagrams for all packages recursively

3-We want to process all the classes



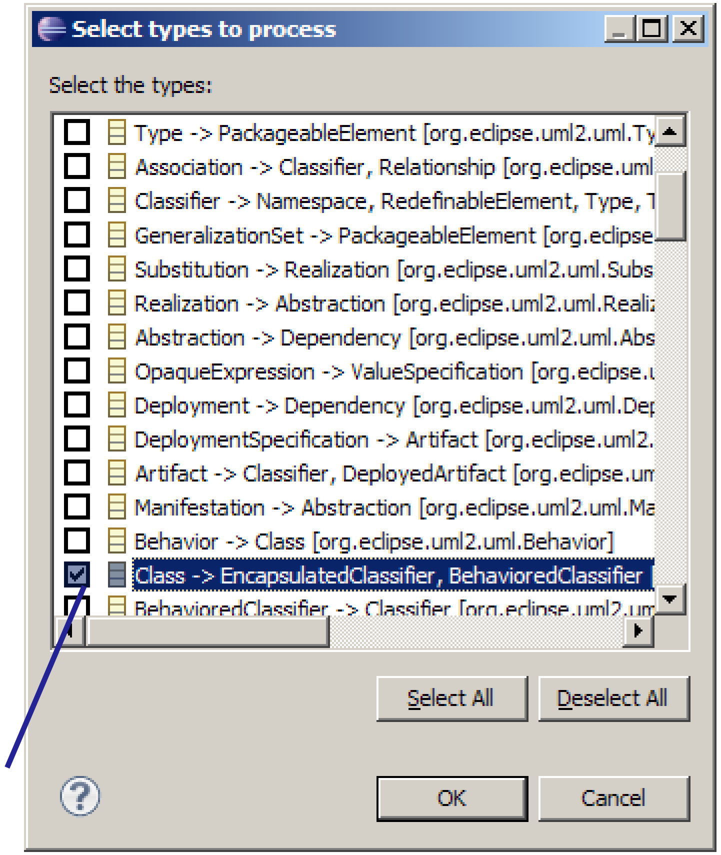
For:

Selection	Recursively	SubTypes	Stereotyped by
Package	☑	☐	

What:

- eAnnotations : EAnnotation
- ownedComment : Comment
- nameExpression : StringExpression
- elementImport : ElementImport
- packageImport : PackageImport
- ownedRule : Constraint
- templateBinding : TemplateBinding
- ownedTemplateSignature : TemplateSignature
- packageMerge : PackageMerge
- packagedElement : PackageableElement
- profileApplication : ProfileApplication

2-We want to add Classes: these are packagedElements



Select types to process

Select the types:

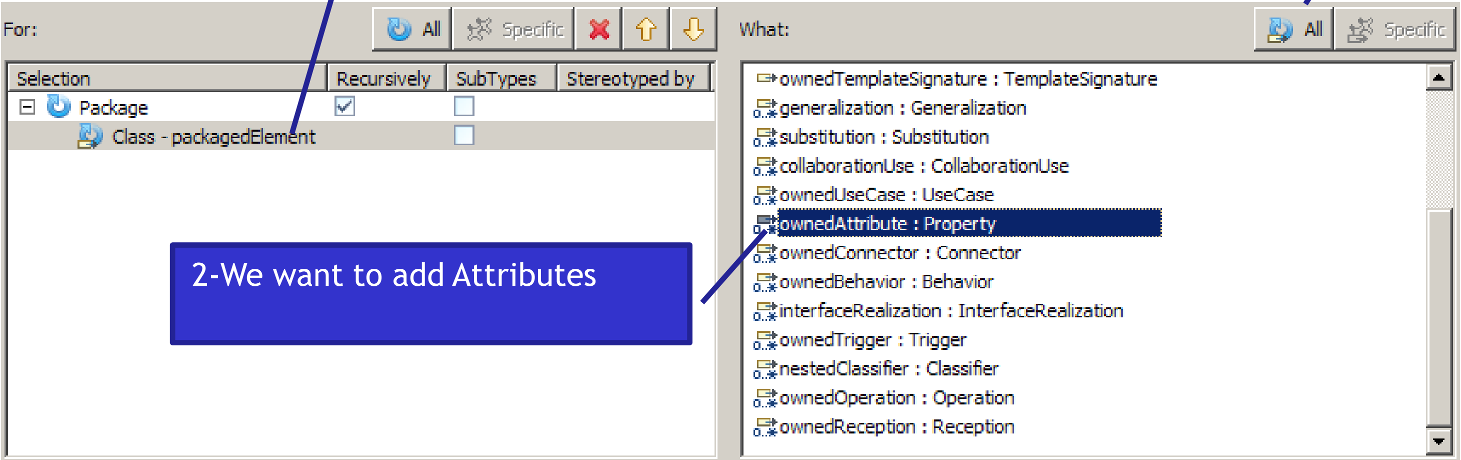
- ☐ Type -> PackageableElement [org.eclipse.uml2.uml.Ty
- ☐ Association -> Classifier, Relationship [org.eclipse.uml
- ☐ Classifier -> Namespace, RedefinableElement, Type, T
- ☐ GeneralizationSet -> PackageableElement [org.eclipse
- ☐ Substitution -> Realization [org.eclipse.uml2.uml.Subs
- ☐ Realization -> Abstraction [org.eclipse.uml2.uml.Reali
- ☐ Abstraction -> Dependency [org.eclipse.uml2.uml.Abs
- ☐ OpaqueExpression -> ValueSpecification [org.eclipse.u
- ☐ Deployment -> Dependency [org.eclipse.uml2.uml.Dep
- ☐ DeploymentSpecification -> Artifact [org.eclipse.uml2.
- ☐ Artifact -> Classifier, DeployedArtifact [org.eclipse.um
- ☐ Manifestation -> Abstraction [org.eclipse.uml2.uml.Ma
- ☐ Behavior -> Class [org.eclipse.uml2.uml.Behavior]
- ☑ Class -> EncapsulatedClassifier, BehavioredClassifier
- ☐ BehavioredClassifier -> Classifier [org.eclipse.uml2.um

4-Select Class

Create generic templates
Show attributes and operations

1-Select the “All classes”

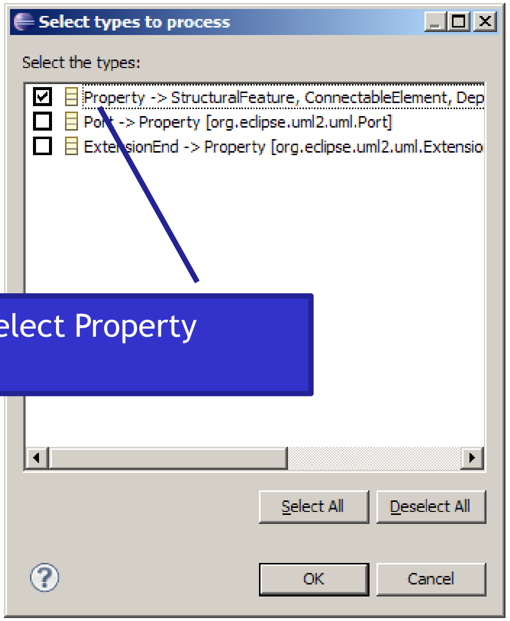
3-We want to process them all



Selection	Recursively	SubTypes	Stereotyped by
⊟ Package	☑	□	
Class - packagedElement		□	

- ownedTemplateSignature : TemplateSignature
- generalization : Generalization
- substitution : Substitution
- collaborationUse : CollaborationUse
- ownedUseCase : UseCase
- ownedAttribute : Property
- ownedConnector : Connector
- ownedBehavior : Behavior
- interfaceRealization : InterfaceRealization
- ownedTrigger : Trigger
- nestedClassifier : Classifier
- ownedOperation : Operation
- ownedReception : Reception

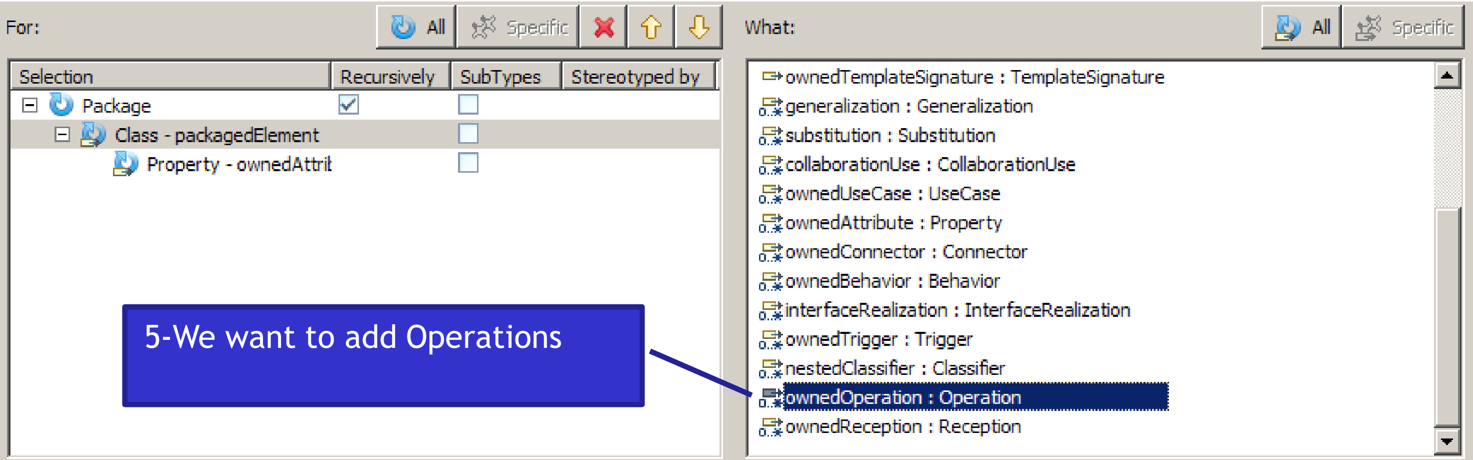
2-We want to add Attributes



Select the types:

☑ Property -> StructuralFeature, ConnectableElement, Dep
□ Port -> Property [org.eclipse.uml2.uml.Port]
□ ExtensionEnd -> Property [org.eclipse.uml2.uml.Extensio

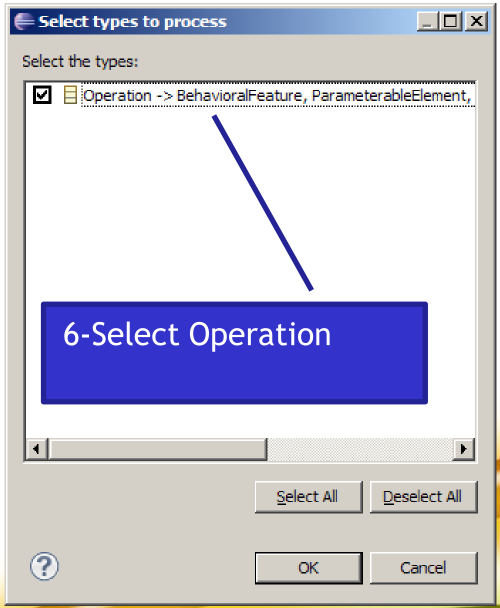
4-Select Property



Selection	Recursively	SubTypes	Stereotyped by
⊟ Package	☑	□	
⊟ Class - packagedElement		□	
Property - ownedAttrit		□	

- ownedTemplateSignature : TemplateSignature
- generalization : Generalization
- substitution : Substitution
- collaborationUse : CollaborationUse
- ownedUseCase : UseCase
- ownedAttribute : Property
- ownedConnector : Connector
- ownedBehavior : Behavior
- interfaceRealization : InterfaceRealization
- ownedTrigger : Trigger
- nestedClassifier : Classifier
- ownedOperation : Operation
- ownedReception : Reception

5-We want to add Operations



Select the types:

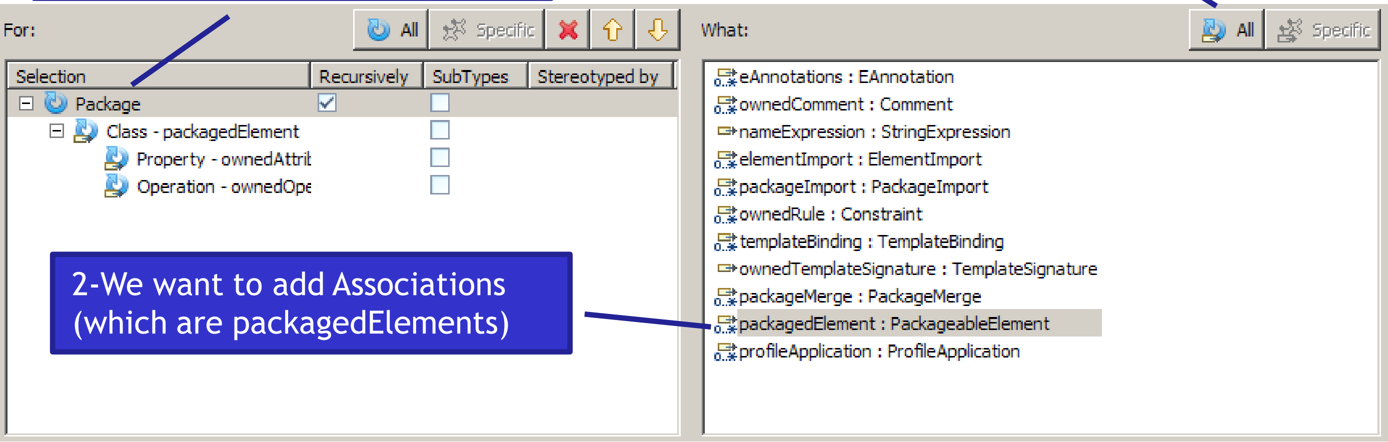
☑ Operation -> BehavioralFeature, ParameterableElement,

6-Select Operation

Create generic templates
Show associations

1-Select the “All packages”

3-We want to add them all



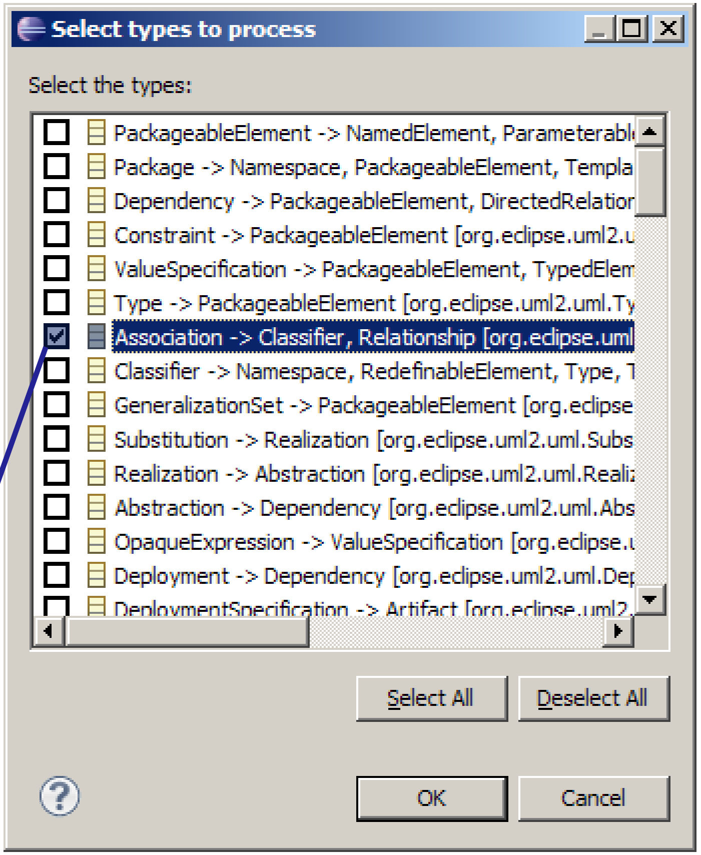
For: All Specific

Selection	Recursively	SubTypes	Stereotyped by
⊟ Package	☑	□	
⊟ Class - packagedElement		□	
Property - ownedAttrit		□	
Operation - ownedOpe		□	

What: All Specific

- eAnnotations : EAnnotation
- ownedComment : Comment
- nameExpression : StringExpression
- elementImport : ElementImport
- packageImport : PackageImport
- ownedRule : Constraint
- templateBinding : TemplateBinding
- ownedTemplateSignature : TemplateSignature
- packageMerge : PackageMerge
- packagedElement : PackageableElement
- profileApplication : ProfileApplication

2-We want to add Associations (which are packagedElements)



Select types to process

Select the types:

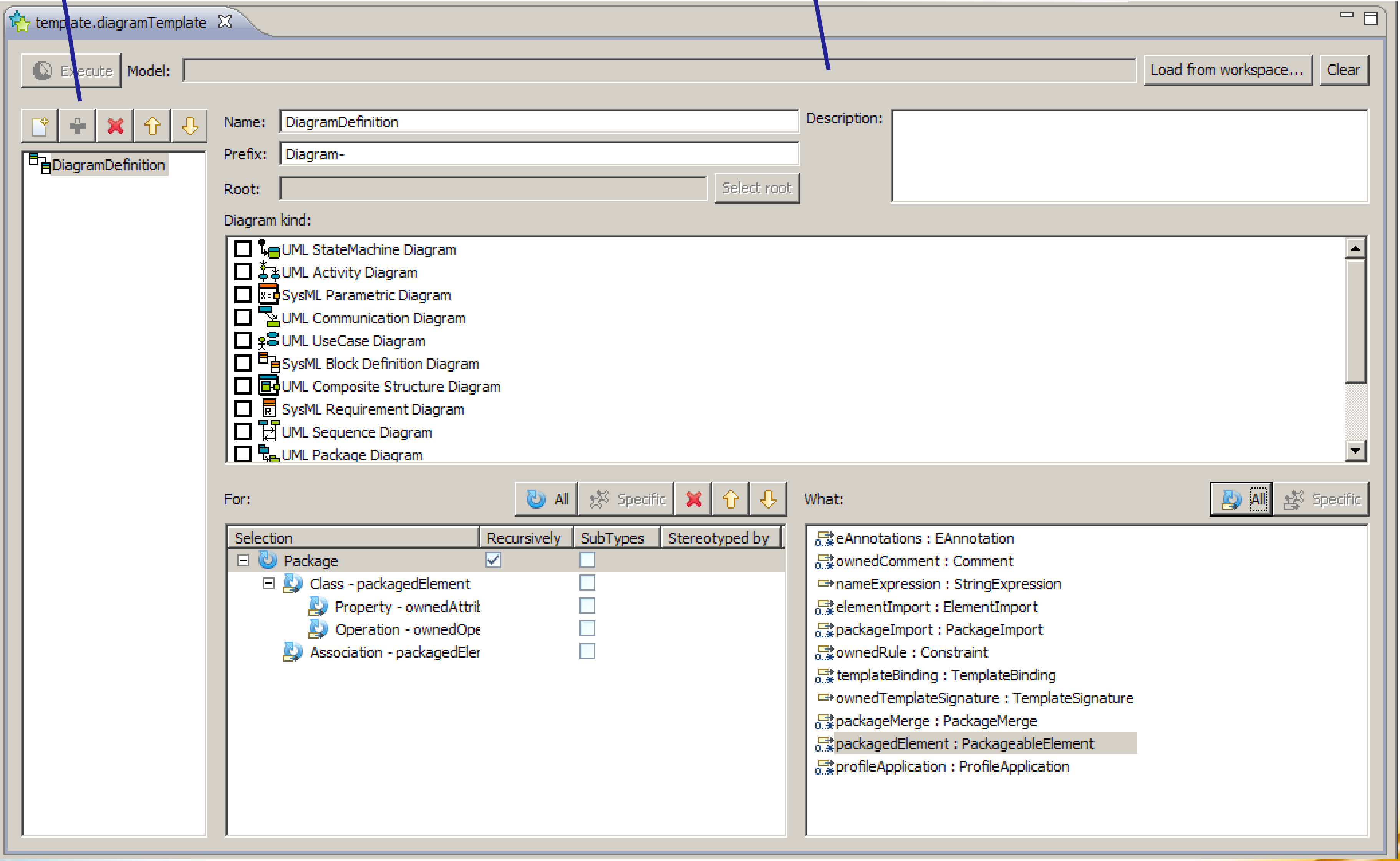
- □ PackageableElement -> NamedElement, Parameterabl
- □ Package -> Namespace, PackageableElement, Templa
- □ Dependency -> PackageableElement, DirectedRelatior
- □ Constraint -> PackageableElement [org.eclipse.uml2.u
- □ ValueSpecification -> PackageableElement, TypedElem
- □ Type -> PackageableElement [org.eclipse.uml2.uml.Ty
- ☑ Association -> Classifier, Relationship [org.eclipse.uml
- □ Classifier -> Namespace, RedefinableElement, Type, T
- □ GeneralizationSet -> PackageableElement [org.eclipse
- □ Substitution -> Realization [org.eclipse.uml2.uml.Subs
- □ Realization -> Abstraction [org.eclipse.uml2.uml.Realiz
- □ Abstraction -> Dependency [org.eclipse.uml2.uml.Abs
- □ OpaqueExpression -> ValueSpecification [org.eclipse.u
- □ Deployment -> Dependency [org.eclipse.uml2.uml.Dep
- □ DeploymentSpecification -> Artifact [org.eclipse.uml2.

4-Select Association

Create generic templates

Add this diagram definition

At this stage, the template is generic



Model:

DiagramDefinition

Name: DiagramDefinition

Prefix: Diagram-

Root: Select root

Description:

Diagram kind:

- □ UML StateMachine Diagram
- □ UML Activity Diagram
- □ SysML Parametric Diagram
- □ UML Communication Diagram
- □ UML UseCase Diagram
- □ SysML Block Definition Diagram
- □ UML Composite Structure Diagram
- □ SysML Requirement Diagram
- □ UML Sequence Diagram
- □ UML Package Diagram

For:

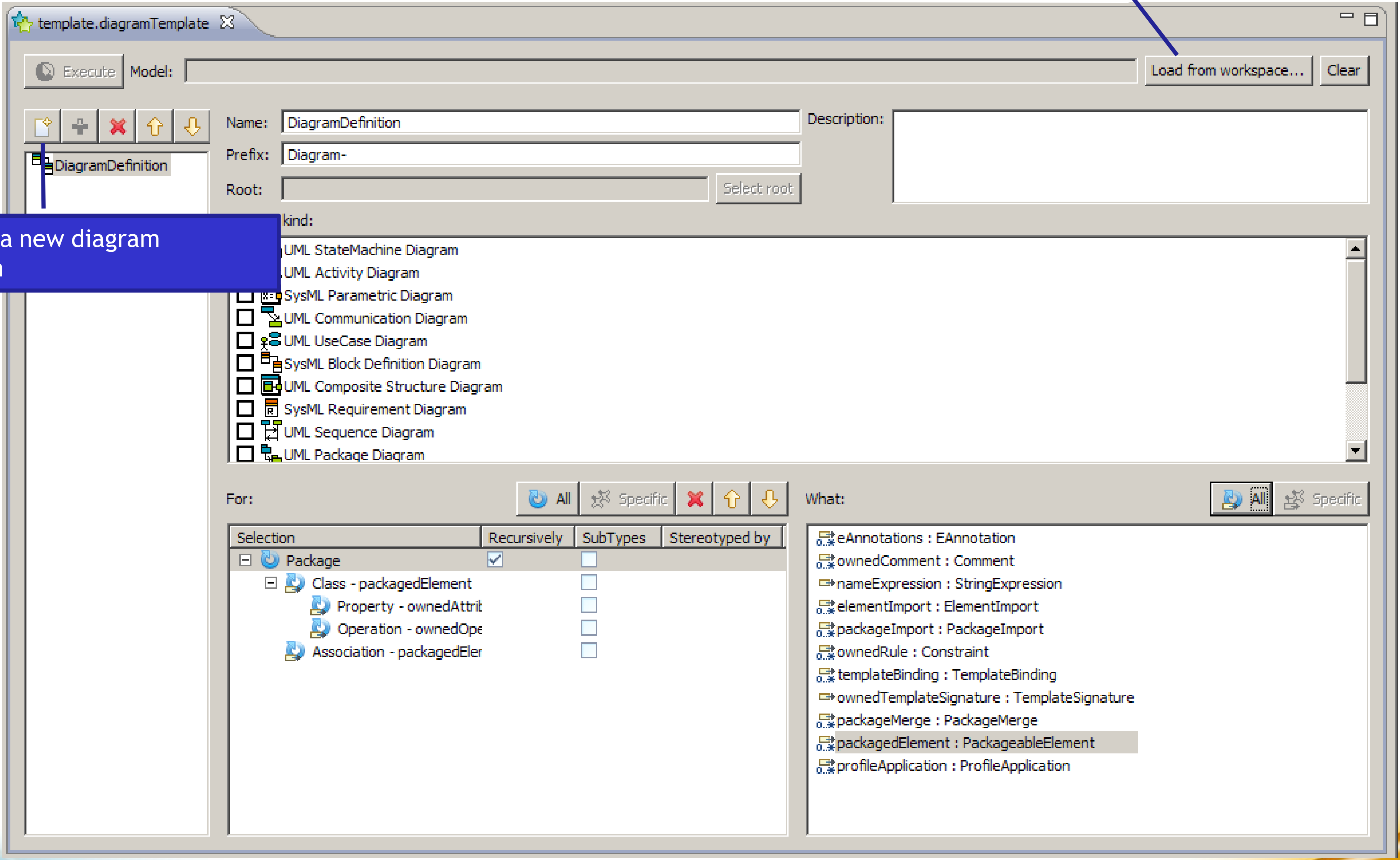
Selection	Recursively	SubTypes	Stereotyped by
Package	☑	□	
Class - packagedElement		□	
Property - ownedAttrit		□	
Operation - ownedOpe		□	
Association - packagedEler		□	

What:

- eAnnotations : EAnnotation
- ownedComment : Comment
- nameExpression : StringExpression
- elementImport : ElementImport
- packageImport : PackageImport
- ownedRule : Constraint
- templateBinding : TemplateBinding
- ownedTemplateSignature : TemplateSignature
- packageMerge : PackageMerge
- packagedElement : PackageableElement
- profileApplication : ProfileApplication

Apply template to a specific model

1-Select a UML model to apply this template on



template.diagramTemplate

Execute Model: Load from workspace... Clear

DiagramDefinition

Name: DiagramDefinition

Prefix: Diagram-

Root: Select root

Description:

kind:

- UML StateMachine Diagram
- UML Activity Diagram
- ☐ SysML Parametric Diagram
- ☐ UML Communication Diagram
- ☐ UML UseCase Diagram
- ☐ SysML Block Definition Diagram
- ☐ UML Composite Structure Diagram
- ☐ SysML Requirement Diagram
- ☐ UML Sequence Diagram
- ☐ UML Package Diagram

For: All Specific

Selection	Recursively	SubTypes	Stereotyped by
Package	☑	☐	
Class - packagedElement		☐	
Property - ownedAttrit		☐	
Operation - ownedOpe		☐	
Association - packagedEler		☐	

What: All Specific

- eAnnotations : EAnnotation
- ownedComment : Comment
- nameExpression : StringExpression
- elementImport : ElementImport
- packageImport : PackageImport
- ownedRule : Constraint
- templateBinding : TemplateBinding
- ownedTemplateSignature : TemplateSignature
- packageMerge : PackageMerge
- packagedElement : PackageableElement
- profileApplication : ProfileApplication

2-Create a new diagram definition

Apply template to a specific model
Create an activity diagram for activity1

1-Select a diagram kind to create

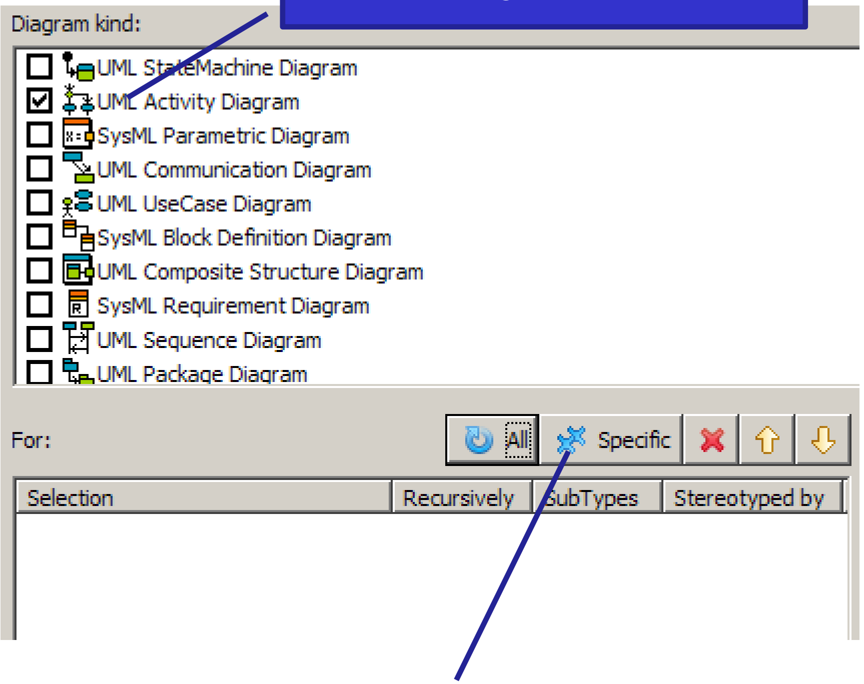


Diagram kind:

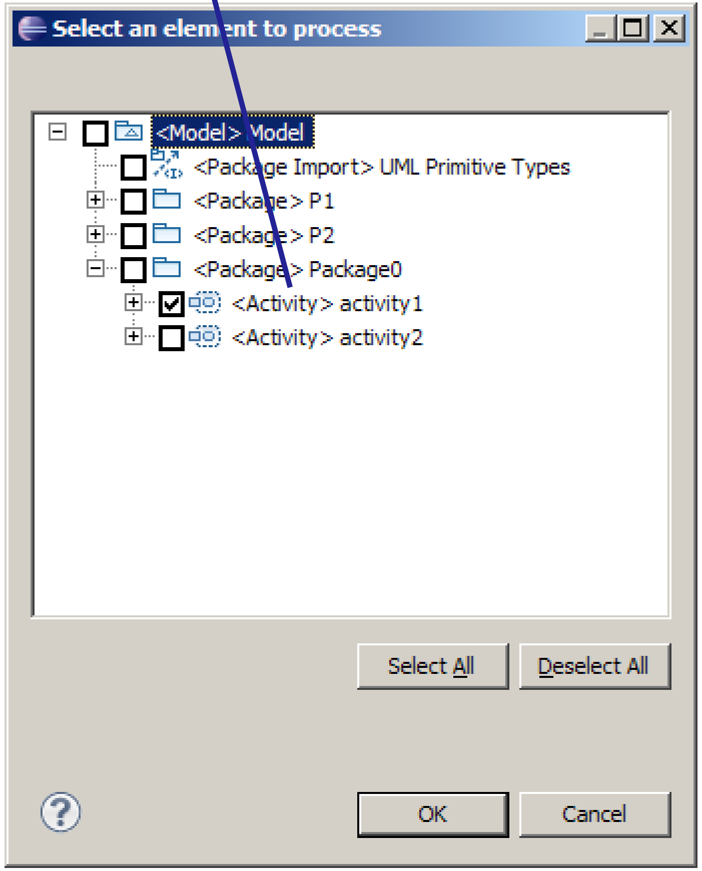
- □ UML StateMachine Diagram
- ☑ UML Activity Diagram
- □ SysML Parametric Diagram
- □ UML Communication Diagram
- □ UML UseCase Diagram
- □ SysML Block Definition Diagram
- □ UML Composite Structure Diagram
- □ SysML Requirement Diagram
- □ UML Sequence Diagram
- □ UML Package Diagram

For:

Selection	Recursively	SubTypes	Stereotyped by

2-Select “Specific”

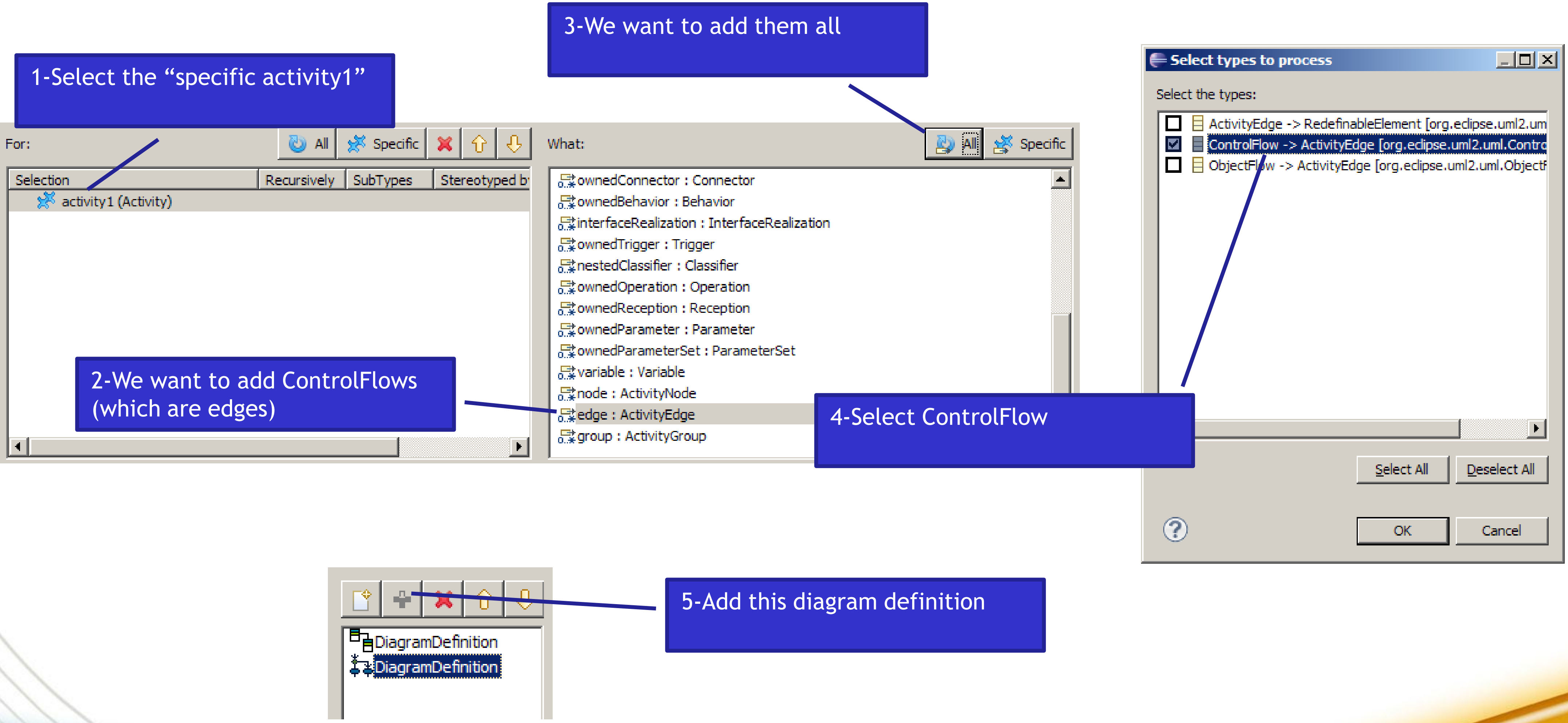
3-Select the element you want to create the diagram for. Here, select activity1



Select an element to process

- □ <Model> Model
  - □ <Package Import> UML Primitive Types
  - □ <Package> P1
  - □ <Package> P2
  - □ <Package> Package0
    - ☑ <Activity> activity1
    - □ <Activity> activity2

Apply template to a specific model
Show ControlFlows



1-Select the “specific activity1”

2-We want to add ControlFlows (which are edges)

3-We want to add them all

4-Select ControlFlow

5-Add this diagram definition

Selection	Recursively	SubTypes	Stereotyped b
activity1 (Activity)			

- ownedConnector : Connector
- ownedBehavior : Behavior
- interfaceRealization : InterfaceRealization
- ownedTrigger : Trigger
- nestedClassifier : Classifier
- ownedOperation : Operation
- ownedReception : Reception
- ownedParameter : Parameter
- ownedParameterSet : ParameterSet
- variable : Variable
- node : ActivityNode
- edge : ActivityEdge
- group : ActivityGroup

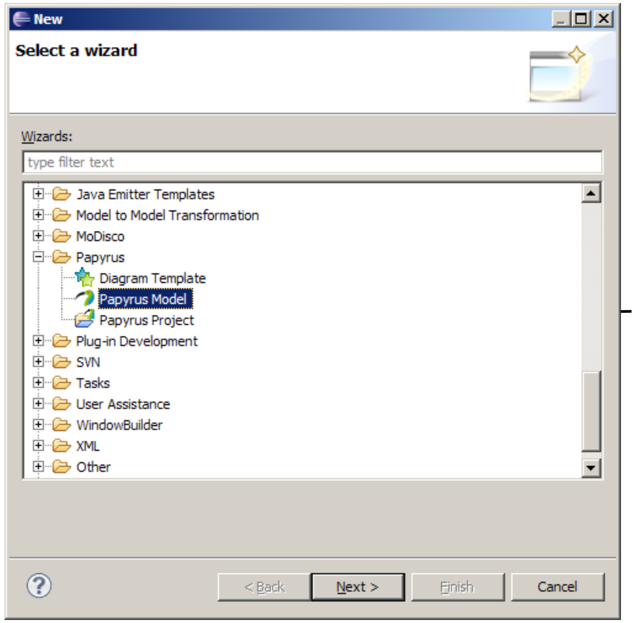
Select the types:

- □ ActivityEdge -> RedefinableElement [org.eclipse.uml2.um
- ☑ ControlFlow -> ActivityEdge [org.eclipse.uml2.uml.Contro
- □ ObjectFlow -> ActivityEdge [org.eclipse.uml2.uml.ObjectF

- DiagramDefinition
- DiagramDefinition

Execute the template

- An empty Papyrus diagram must be created for the UML model: use the Papyrus wizard



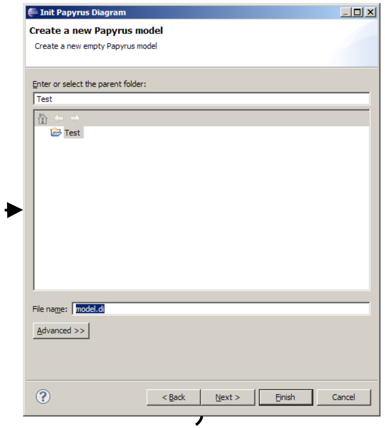
New

Select a wizard

Wizards:

type filter text

- Java Emitter Templates
- Model to Model Transformation
- MoDisco
- Papyrus
  - Diagram Template
  - Papyrus Model
  - Papyrus Project
- Plug-in Development
- SVN
- Tasks
- User Assistance
- WindowBuilder
- XML
- Other



Init Papyrus Diagram

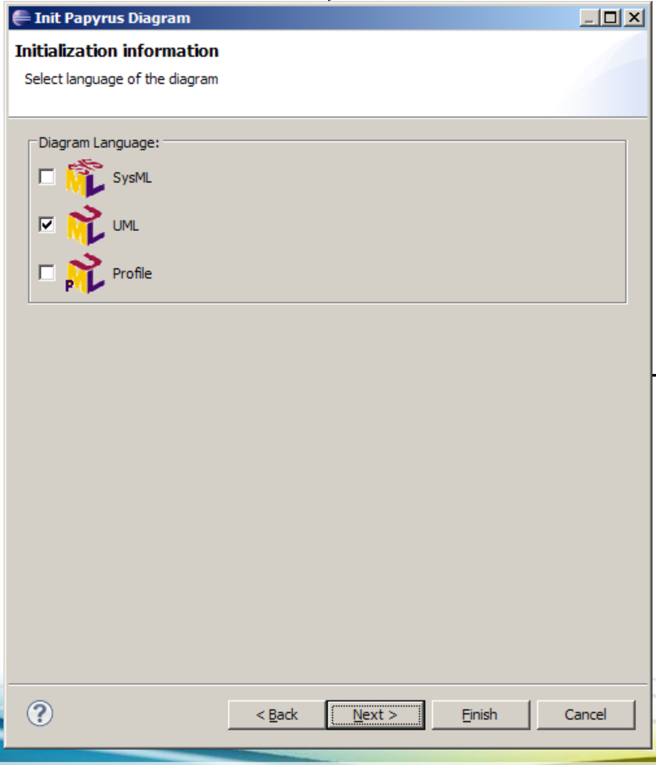
Create a new Papyrus model

Create a new empty Papyrus model

Enter or select the parent folder: Test

- Test

File name: model.di



Init Papyrus Diagram

Initialization information

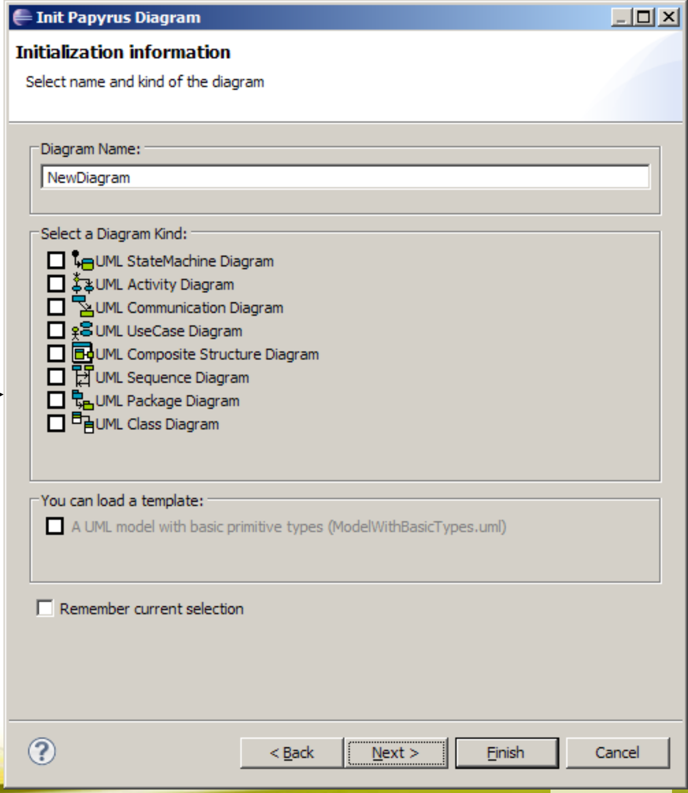
Select language of the diagram

Diagram Language:

☐ SysML

☑ UML

☐ Profile



Init Papyrus Diagram

Initialization information

Select name and kind of the diagram

Diagram Name: NewDiagram

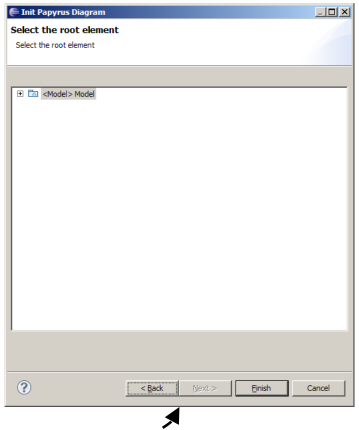
Select a Diagram Kind:

☐ UML StateMachine Diagram
☐ UML Activity Diagram
☐ UML Communication Diagram
☐ UML UseCase Diagram
☐ UML Composite Structure Diagram
☐ UML Sequence Diagram
☐ UML Package Diagram
☐ UML Class Diagram

You can load a template:

☐ A UML model with basic primitive types (ModelWithBasicTypes.uml)

☐ Remember current selection



Init Papyrus Diagram

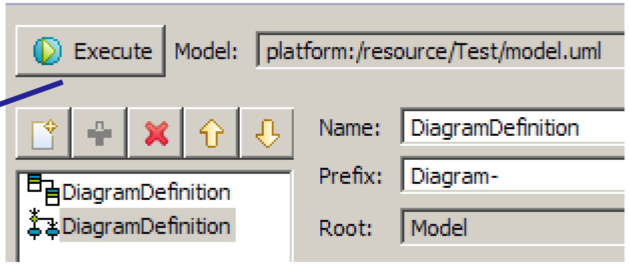
Select the root element

Select the root element

- <Model> Model

Execute the template

1-When your template is defined : Execute



Model: platform:/resource/Test/model.uml

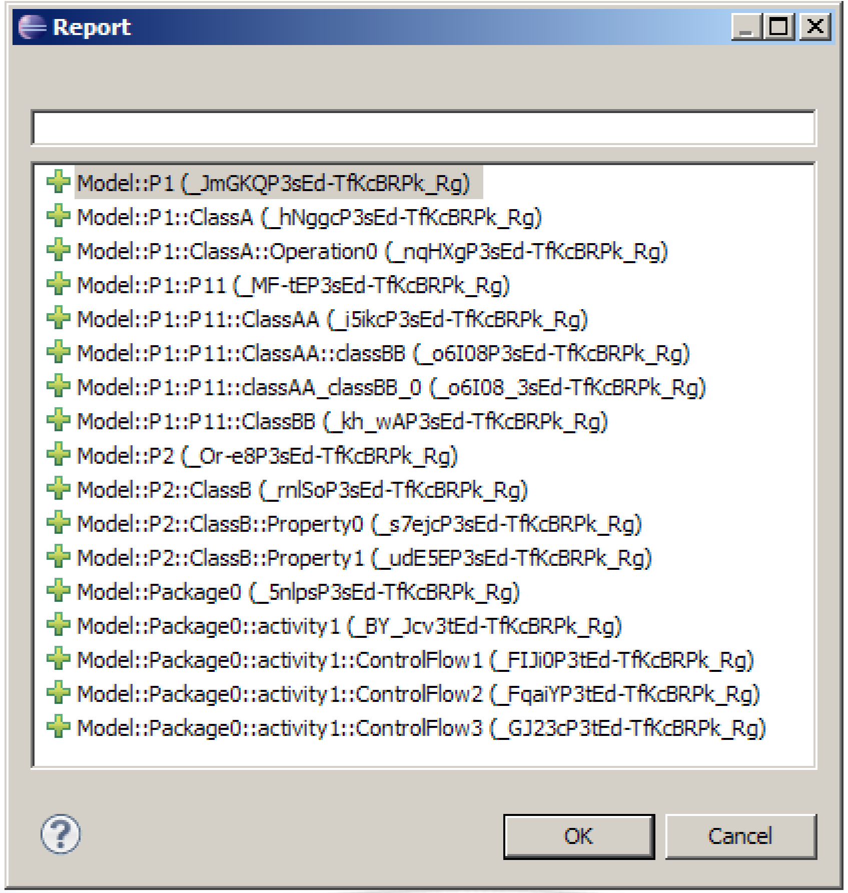
DiagramDefinition
DiagramDefinition

Name: DiagramDefinition

Prefix: Diagram-

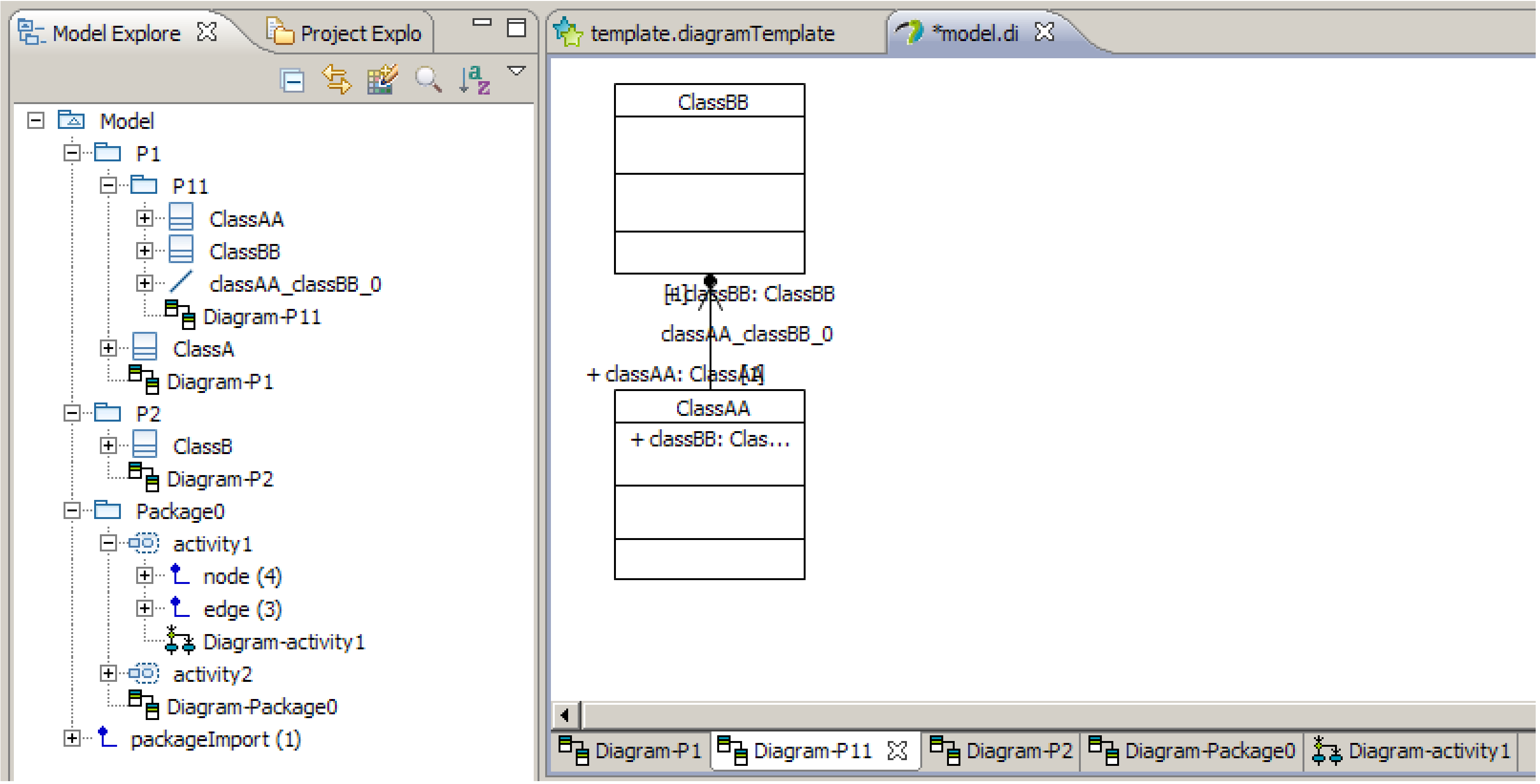
Root: Model

2-You get a report of what has been added (green +). If something matches the template definition but failed to be shown on a diagram or a diagram cannot be created, you will get a red X for it



```
+ Model::P1 (_JmGKQP3sEd-TfKcBRPk_Rg)
+ Model::P1::ClassA (_hNggcP3sEd-TfKcBRPk_Rg)
+ Model::P1::ClassA::Operation0 (_nqHXgP3sEd-TfKcBRPk_Rg)
+ Model::P1::P11 (_MF-tEP3sEd-TfKcBRPk_Rg)
+ Model::P1::P11::ClassAA (_i5ikcP3sEd-TfKcBRPk_Rg)
+ Model::P1::P11::ClassAA::classBB (_o6I08P3sEd-TfKcBRPk_Rg)
+ Model::P1::P11::classAA_classBB_0 (_o6I08_3sEd-TfKcBRPk_Rg)
+ Model::P1::P11::ClassBB (_kh_wAP3sEd-TfKcBRPk_Rg)
+ Model::P2 (_Or-e8P3sEd-TfKcBRPk_Rg)
+ Model::P2::ClassB (_rnlSoP3sEd-TfKcBRPk_Rg)
+ Model::P2::ClassB::Property0 (_s7ejcP3sEd-TfKcBRPk_Rg)
+ Model::P2::ClassB::Property1 (_udE5EP3sEd-TfKcBRPk_Rg)
+ Model::Package0 (_5nlpsP3sEd-TfKcBRPk_Rg)
+ Model::Package0::activity1 (_BY_Jcv3tEd-TfKcBRPk_Rg)
+ Model::Package0::activity1::ControlFlow1 (_FIJi0P3tEd-TfKcBRPk_Rg)
+ Model::Package0::activity1::ControlFlow2 (_FqaiYP3tEd-TfKcBRPk_Rg)
+ Model::Package0::activity1::ControlFlow3 (_GJ23cP3tEd-TfKcBRPk_Rg)
```

Result



Model Explore
Project Explo
template.diagramTemplate
*model.di
Model
P1
P11
ClassAA
ClassBB
classAA_classBB_0
Diagram-P11
ClassA
Diagram-P1
P2
ClassB
Diagram-P2
Package0
activity1
node (4)
edge (3)
Diagram-activity1
activity2
Diagram-Package0
packageImport (1)
ClassBB
[1] classBB: ClassBB
classAA_classBB_0
+ classAA: ClassAA[1]
ClassAA
+ classBB: Clas…
Diagram-P1
Diagram-P11
Diagram-P2
Diagram-Package0
Diagram-activity1

Limitations and Future work

- **If the drag and drop feature is not well implemented then the diagram generation will fail**
  - Post a bug to the developer responsible for the diagram that failed.
- **Expressivity is clearly not enough**
  - Use queries (Modisco) to define :
    - for which elements diagrams are to be created
    - what to show on each diagram
- **Layout of generated diagrams is not always adequate**
  - Post process generation with advanced auto layout algos
  - Let the template designer choose which layout to apply on a diagram definition
- **Report is too laconic**
  - A verbose trace should be generated