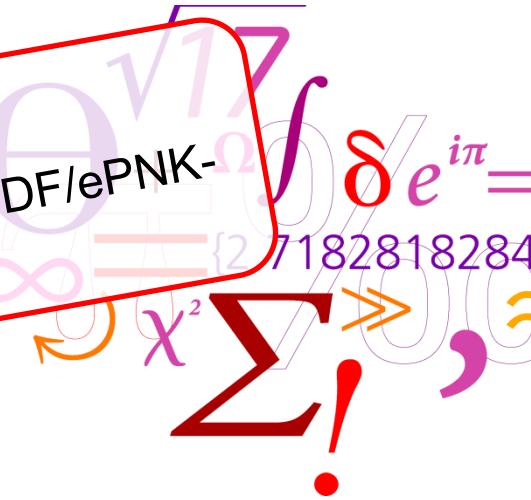
The ePNK:
An extensible Petri net tool for PNML

Ekkart Kindler

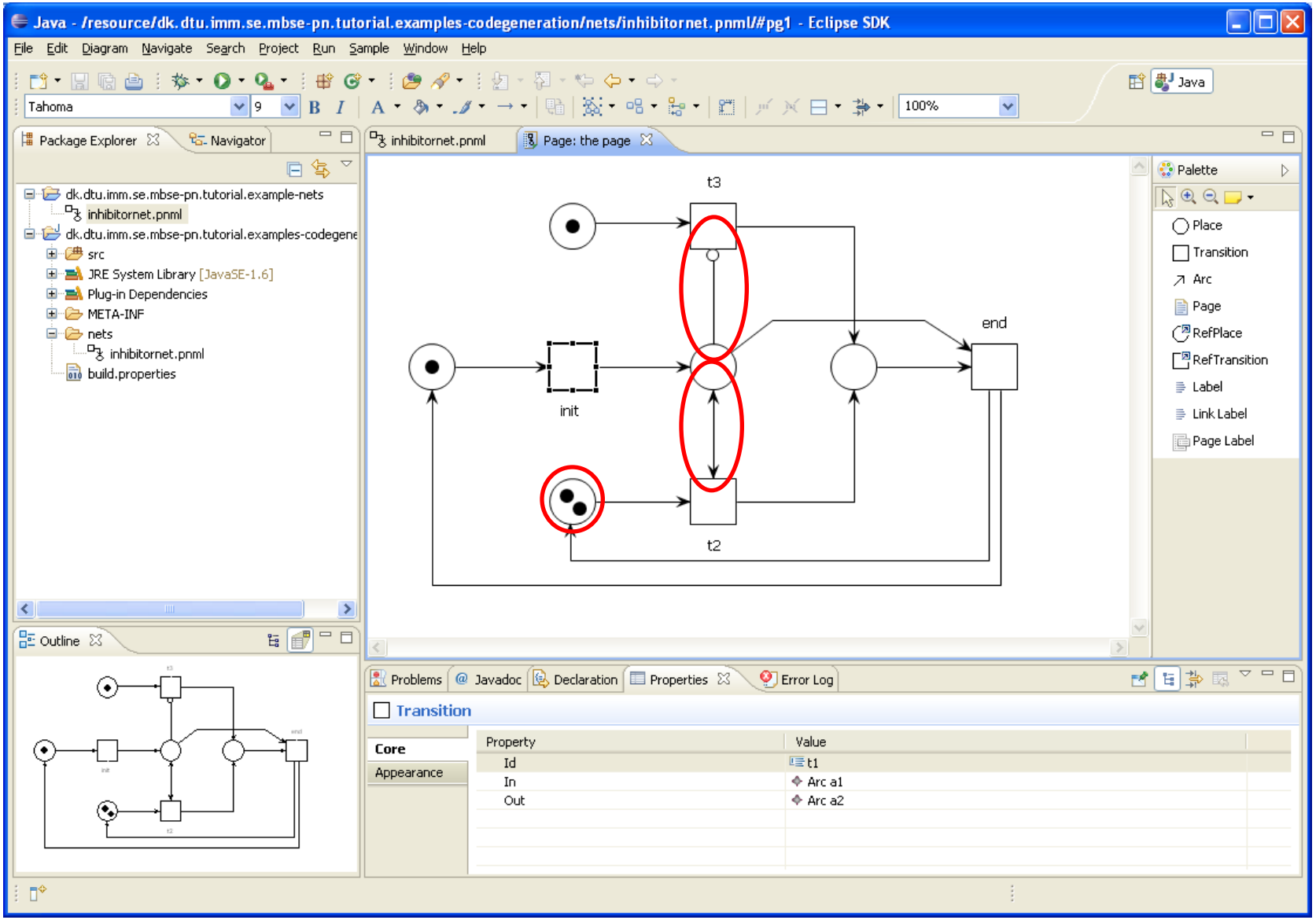
DTU Informatics
Department of Informatics and Mathematical Modeling

For more details see ePNK documentation:
http://www2.compute.dtu.dk/~ekki/projects/ePNK/PDF/ePNK-manual-1.0.0.pdf



New Petri net type

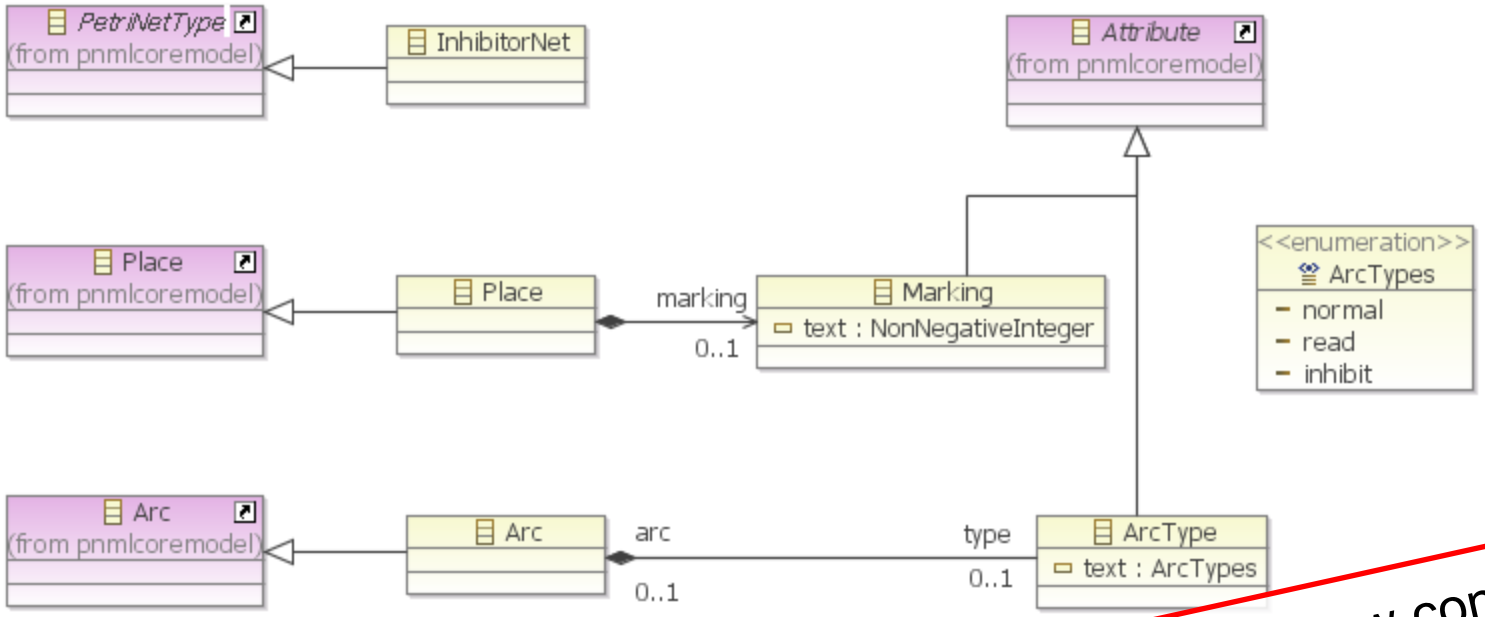
DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler



Define a Petri Net Type

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

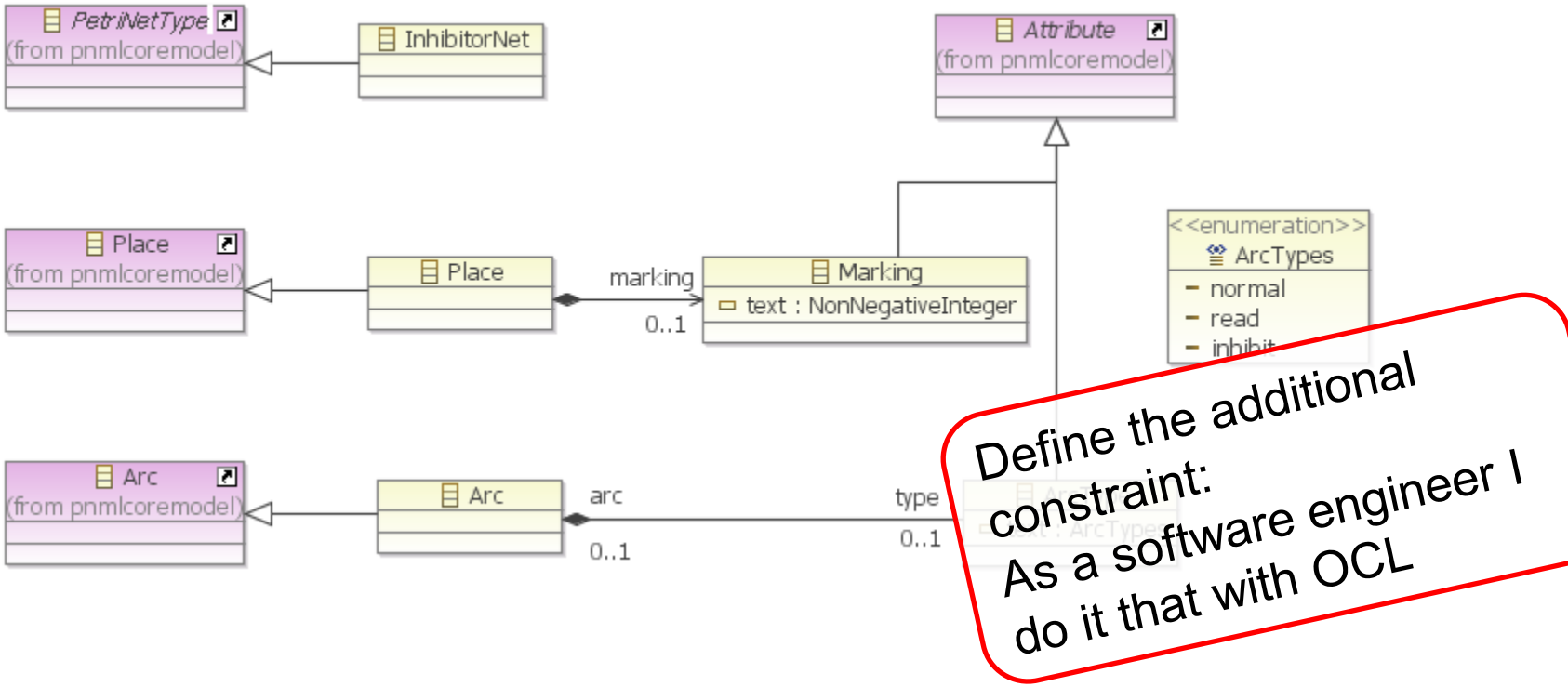
What should it take to define this new Petri Net type conceptually?



Define the new concepts:
As a software engineer, I do it that with a class diagram

Define a Petri Net Type

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

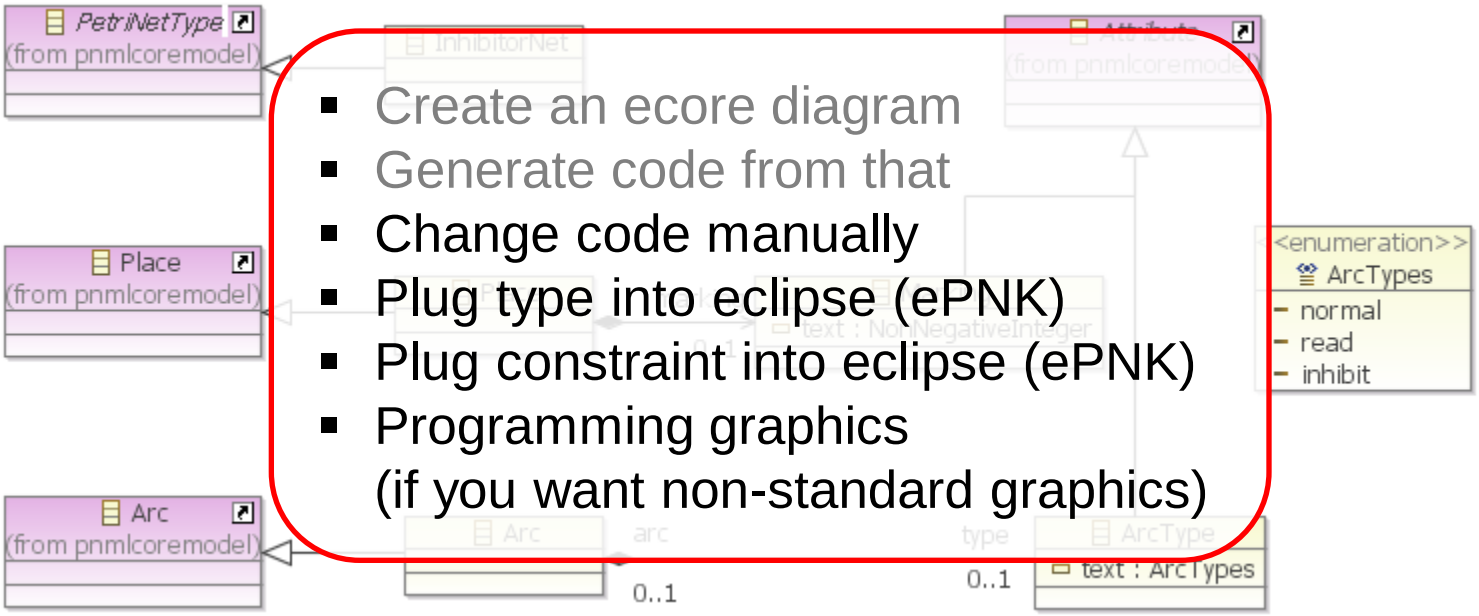



```
( self.source.oclIsKindOf(pnmlcoremodel::PlaceNode) and
  self.target.oclIsKindOf(pnmlcoremodel::TransitionNode) )
or
( self.source.oclIsKindOf(pnmlcoremodel::TransitionNode) and
  self.target.oclIsKindOf(pnmlcoremodel::PlaceNode) and
  not ( self.type.text = ArcTypes::inhibit ) )
```

Define a Petri Net Type

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

What does it take to implement it now?



- Create an ecore diagram
- Generate code from that
- Change code manually
- Plug type into eclipse (ePNK)
- Plug constraint into eclipse (ePNK)
- Programming graphics
  (if you want non-standard graphics)

```
( self.source.oclIsKindOf(pnmlcoremodel::PlaceNode) and
  self.target.oclIsKindOf(pnmlcoremodel::TransitionNode) )
or
( self.source.oclIsKindOf(pnmlcoremodel::TransitionNode) and
  self.target.oclIsKindOf(pnmlcoremodel::PlaceNode) and
  not ( self.type.text = ArcTypes::inhibit ) )
```

Change code manually

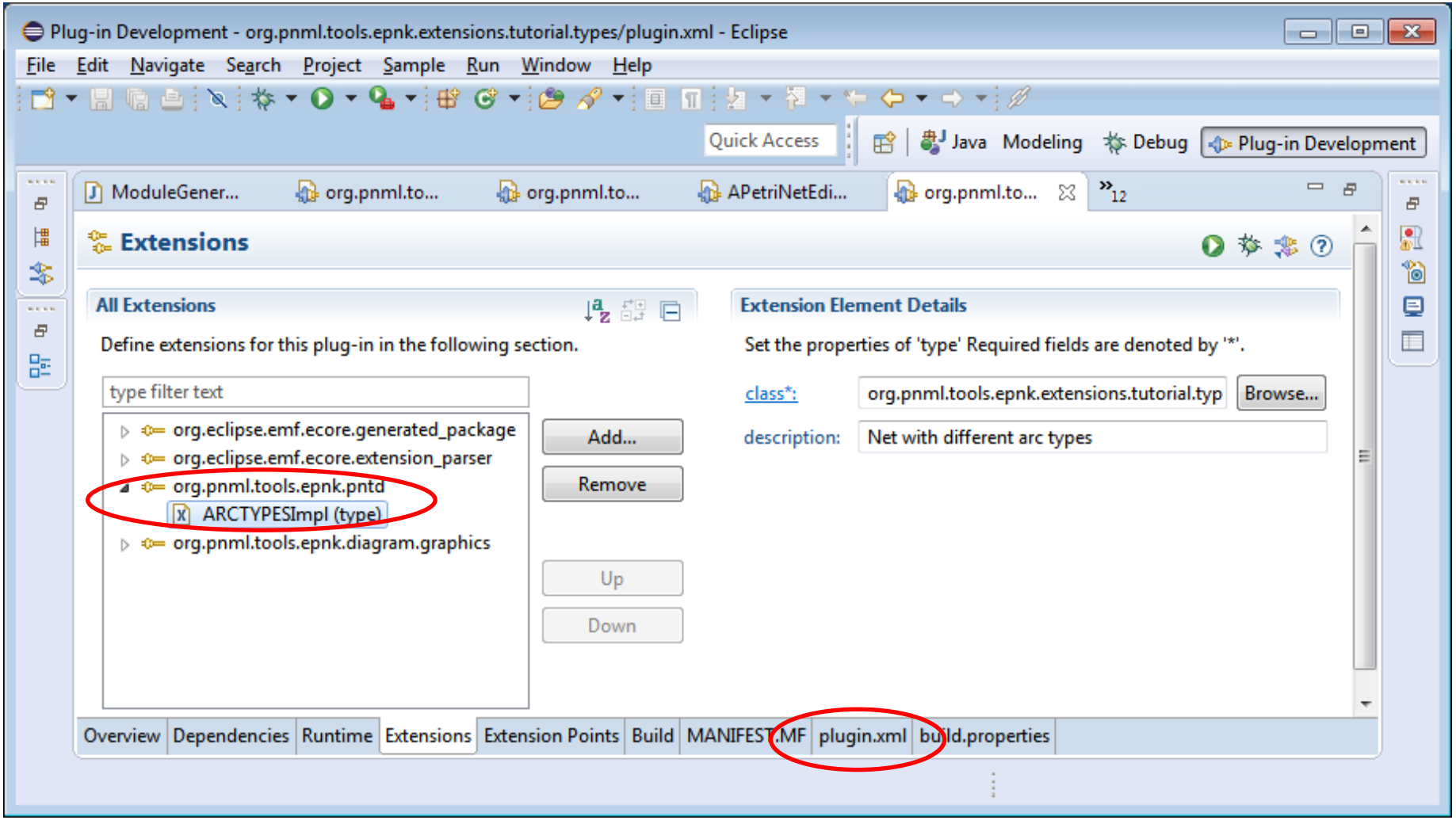
DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

```
package inhibitornets.impl;

[…]

public class InhibitorNetImpl extends PetriNetTypeImpl implements
   InhibitorNet {

   public InhibitorNetImpl() {
        super();
   }

   protected EClass eStaticClass() {
        return InhibitornetsPackage.Literals.INHIBITOR_NET;
   }

   public String toString() {
        return "http://dk.dtu.imm.se.mbse-pn.tutorial.inhibitornet";

   }

}
```

Making constructor of generated code public is not so nice! Extend the class and make the constructor of the extended class public.

Plug in net type

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler



plugin.xml

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

```
<extension
      id="org.pnml.tools.epnk.extensions.tutorial.types.arctypes"
      name="ArcTypes"
      point="org.pnml.tools.epnk.pntd">
   <type
       class="org.pnml.[…].tutorial.types.arctypes.arctypes.impl.ARCTYPESImpl"
       description="Net with different arc types">
   </type>
</extension>
<extension
      id="org.pnml.tools.epnk.extensions.tutorial.types.arctypes.graphics"
      name="ArcTypes Net graphical extensions"
      point="org.pnml.tools.epnk.diagram.graphics">
   <graphicsextension
         class= "org.pnml.tools.epnk.[…].ArcTypesGraphicalExtension"
         description="Special graphics for arcs in this kind of net">
   </graphicsextension>
</extension>
```

Plug in constraint

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

```
<extension
    point="org.eclipse.emf.validation.constraintProviders">

      [about 40 other boring but technically important lines]


  <![CDATA[
  ( self.source.oclIsKindOf(pnmlcoremodel::PlaceNode) and
    self.target.oclIsKindOf(pnmlcoremodel::TransitionNode) )
  or
  ( self.source.oclIsKindOf(pnmlcoremodel::TransitionNode) and
    self.target.oclIsKindOf(pnmlcoremodel::PlaceNode) and
    not ( self.type.text = ArcTypes::inhibit ) )
  ]]>
     </constraint>
    </constraints>
  </constraintProvider>
</extension>
```

Unfortunately, this is a bit technical

Constraints can also be programmed

Programm Graphics

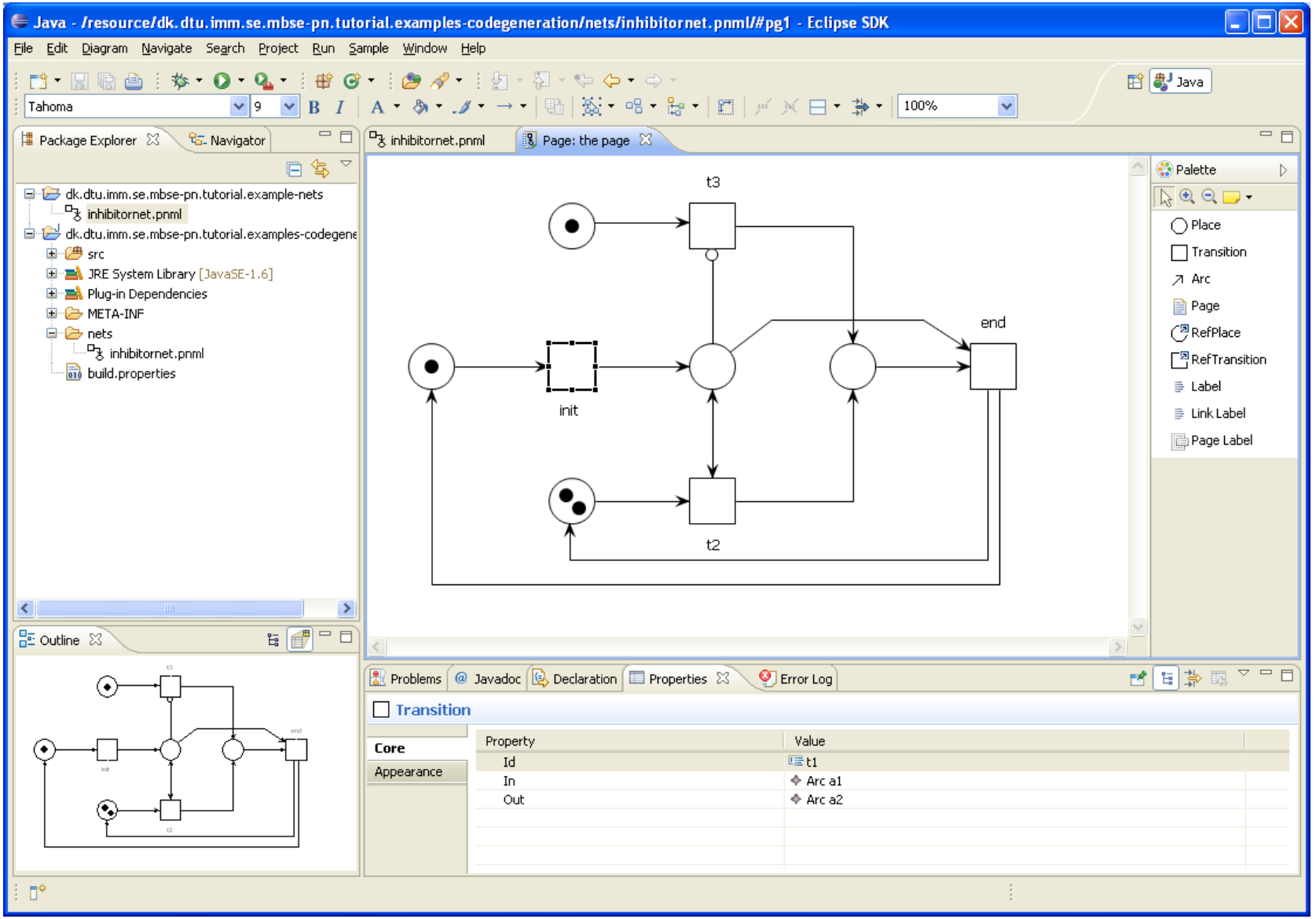
DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

```
public class InhibitornetsArcFigure extends ArcFigure {
[...]
  private void setGraphics() {
    RotatableDecoration targetDecorator = null;
    RotatableDecoration sourceDecorator = null;

    if (type.equals(ArcTypes.NORMAL)) {
      targetDecorator = new ReisigsArrowHeadDecoration();

    } else if (type.equals(ArcTypes.INHIBIT)) {
      CircleDecoration circleDecoration =
        new CircleDecoration();
      circleDecoration.setLineWidth(this.getLineWidth());
      targetDecorator = circleDecoration;

    } else if (type.equals(ArcTypes.READ)) {
      targetDecorator = new ReisigsArrowHeadDecoration();
      sourceDecorator = new ReisigsArrowHeadDecoration();
[...]
```

Just a glimpse!

Result

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler



Technical details

DTU Informatics
Department of Informatics and Mathematical Modelling
Ekkart Kindler

You can look up some details in the version of the ePNK that was deployed in the SE2 course:

`org.pnml.tools.epnk.extensions.tutorial.types`

`org.pnml.tools.epnk.extensions.tutorial.types.edit`

In order to see these projects to your workspace:

- Open “Plug-ins” view
- Select the resp. plug-in project(s)
- Right-click and choose
  Import As → Source Project