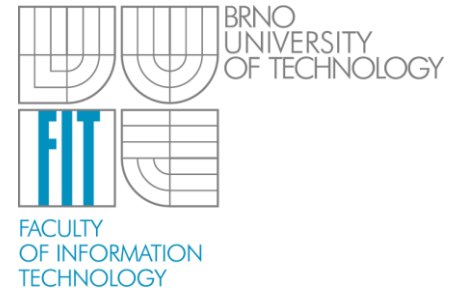


Tvorba editoru pomocí GMF pro Eclipse

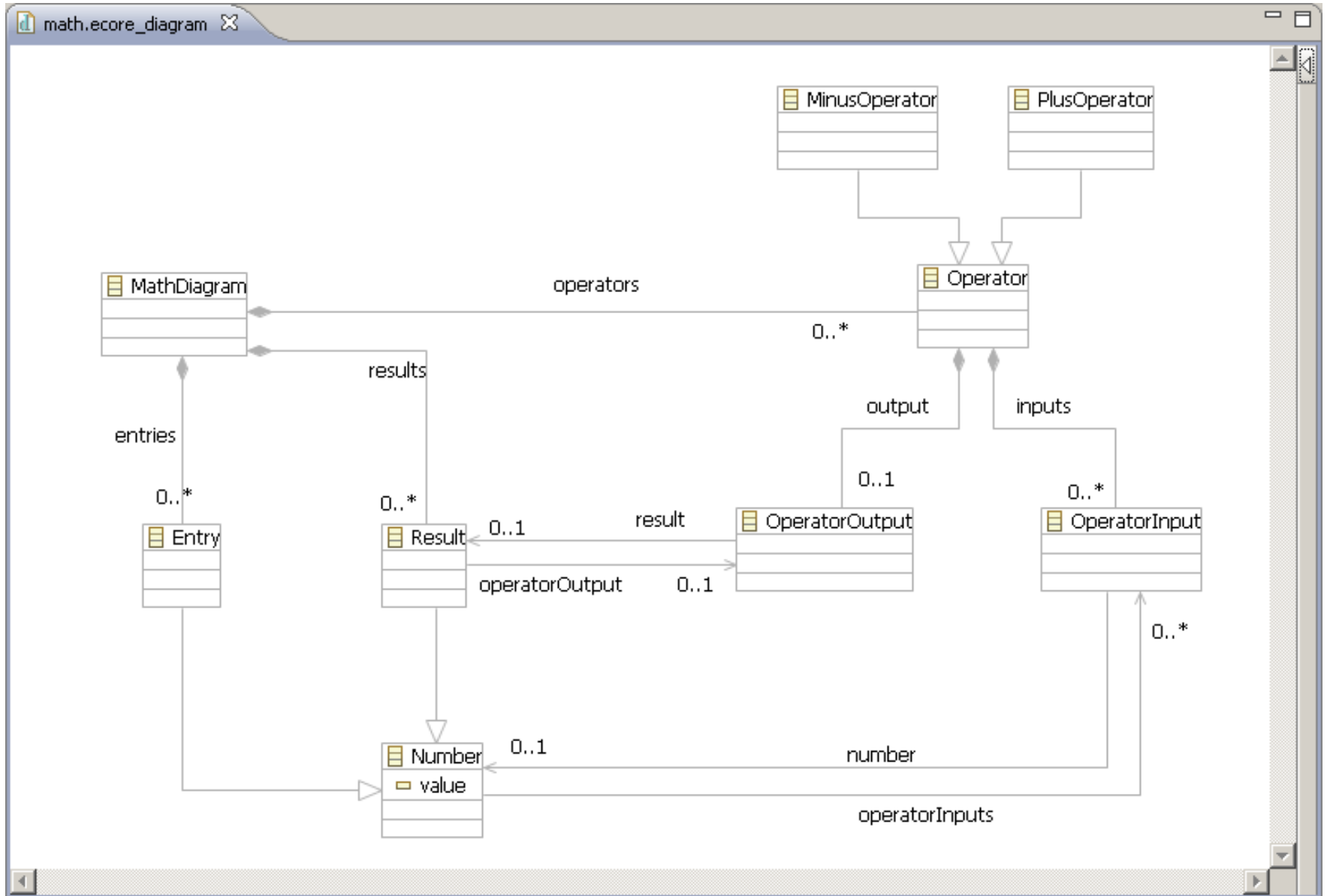
Ota Jirák

Vysoké učení technické v Brně, Fakulta informačních technologií
Božetěchova 2, 612 66 Brno
ijirak@fit.vutbr.cz



20.10.2010

- Definice Modelu
- Tvorba Editoru – dashboard
- Úpravy modelu
- Programový přístup



Definice modelu

Anotovaná Java

```

package jfb.examples.gmf.math;

import java.util.List;

/**
 * @model abstract="true"
 */
public interface Operator {

    /**
     * @model containment="true"
     */
    public List<OperatorInput> getInputs();

    /**
     * @model containment="true"
     */
    public OperatorOutput getOutput();
}

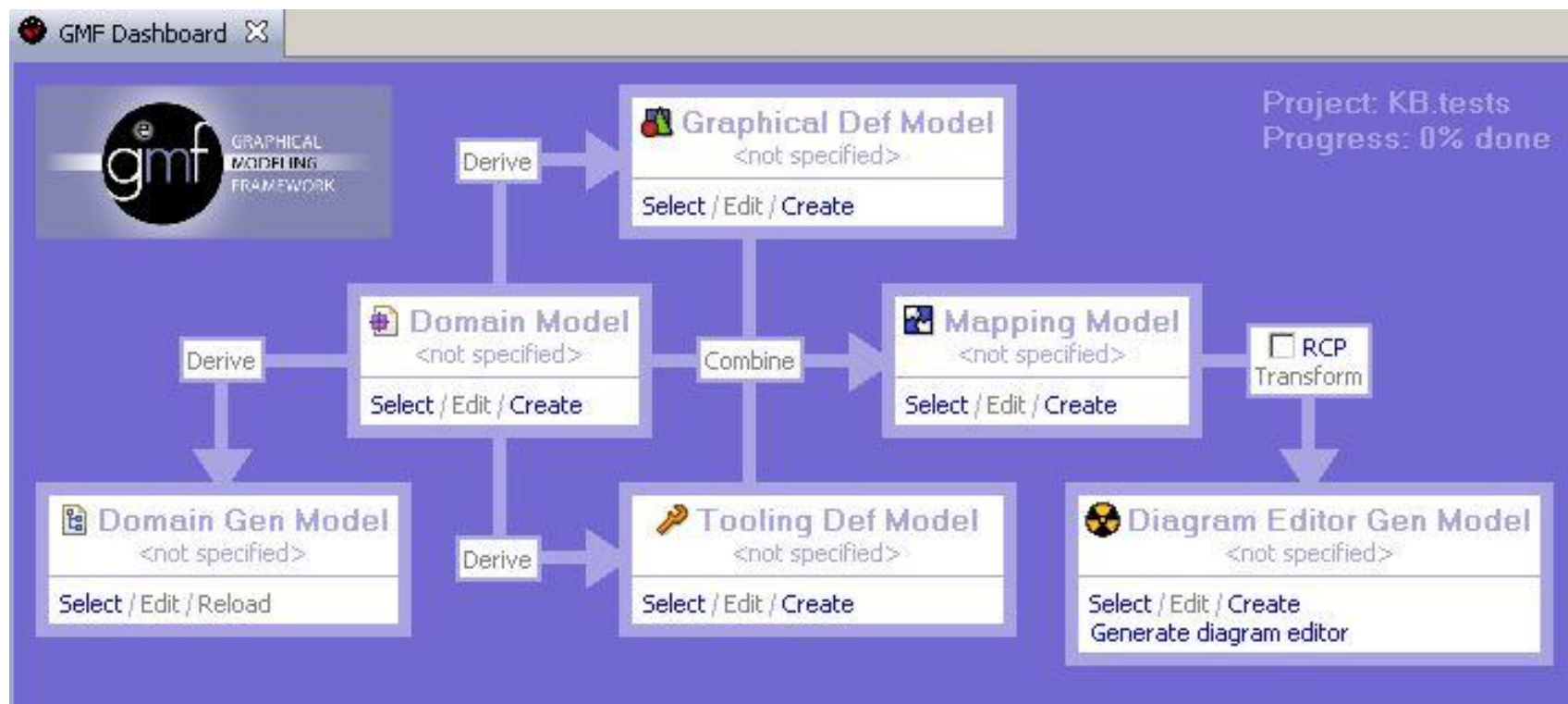
```

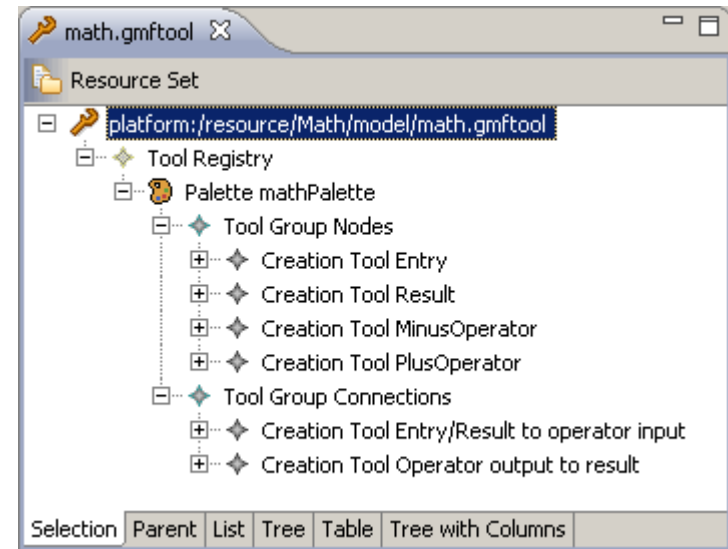
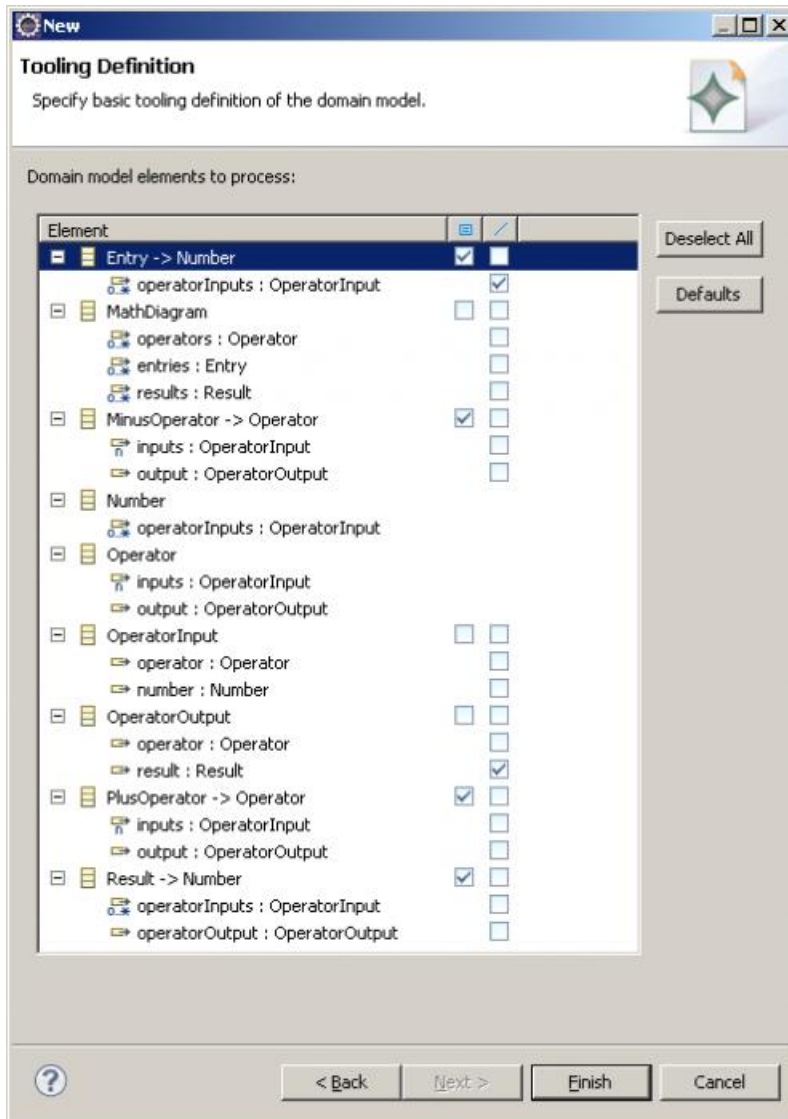
Emfatic

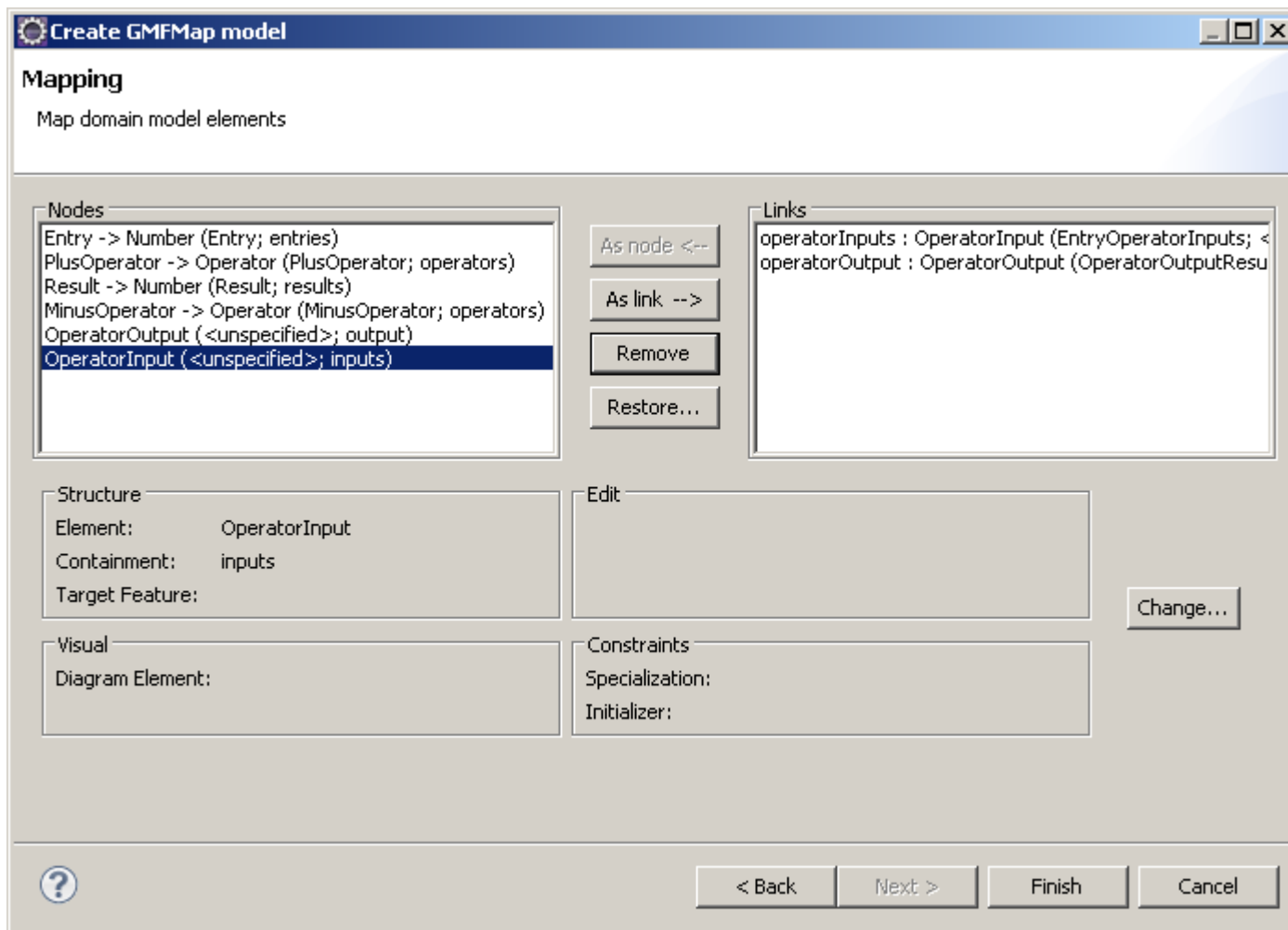
```

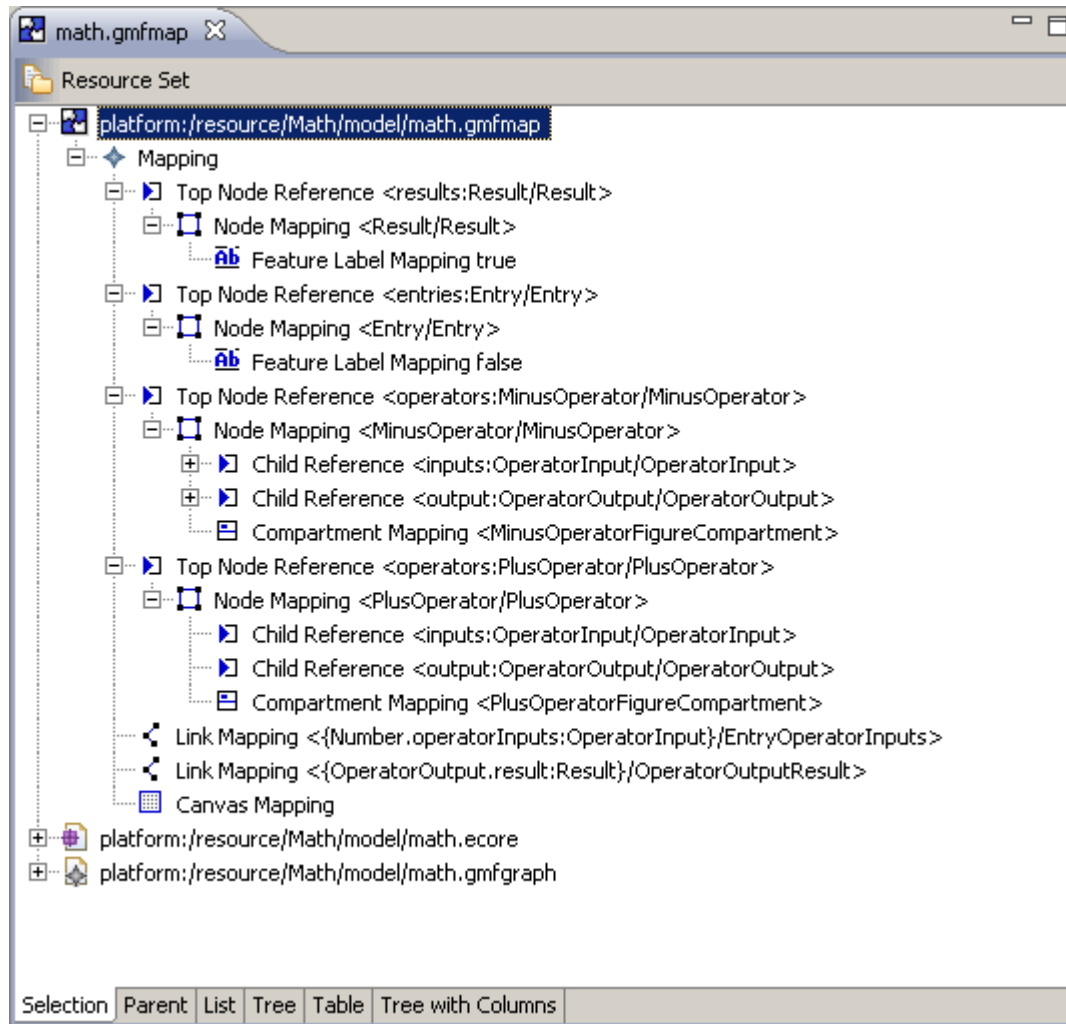
62
63 @gmf.node(label="name")
64 class Port extends Interface {
65     op boolean          isVirtual();
66     attr int            busLimit;
67     attr Sensitivity    sensitivity;
68     attr int            width;
69     attr ValidityType   vt_width;
70     attr int            occupied;
71
72     @gmf.compartment(foo="bar")
73     !resolve val Port[*] subPort;
74     attr int cHeight;
75     attr ValidityType vt_cHeight;
76 }

```





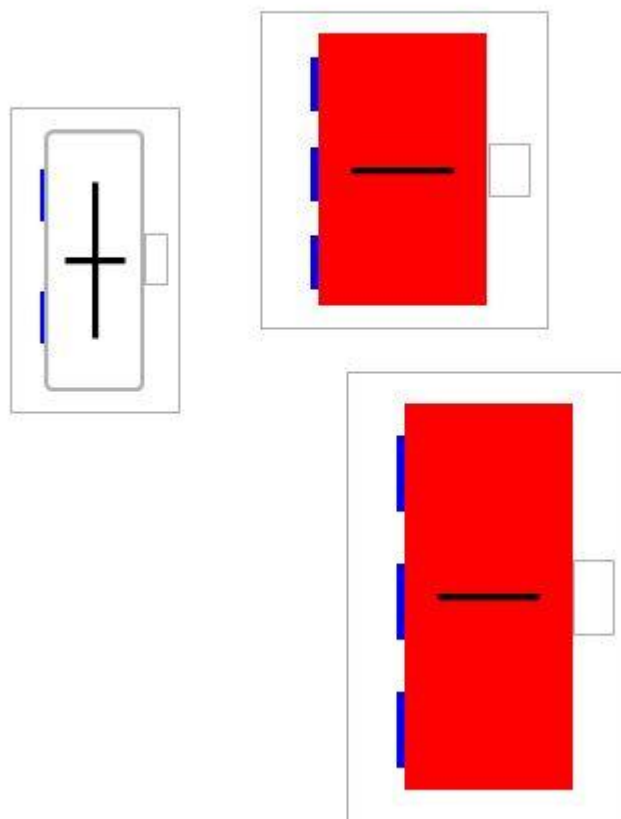




The screenshot shows a window titled 'math.gmfmap' with a 'Resource Set' containing the following structure:

- platform:/resource/Math/model/math.gmfmap
 - Mapping
 - Top Node Reference <results:Result/Result>
 - Node Mapping <Result/Result>
 - Feature Label Mapping true
 - Top Node Reference <entries:Entry/Entry>
 - Node Mapping <Entry/Entry>
 - Feature Label Mapping false
 - Top Node Reference <operators:MinusOperator/MinusOperator>
 - Node Mapping <MinusOperator/MinusOperator>
 - Child Reference <inputs:OperatorInput/OperatorInput>
 - Child Reference <output:OperatorOutput/OperatorOutput>
 - Compartment Mapping <MinusOperatorFigureCompartment>
 - Top Node Reference <operators:PlusOperator/PlusOperator>
 - Node Mapping <PlusOperator/PlusOperator>
 - Child Reference <inputs:OperatorInput/OperatorInput>
 - Child Reference <output:OperatorOutput/OperatorOutput>
 - Compartment Mapping <PlusOperatorFigureCompartment>
 - Link Mapping <{Number.operatorInputs:OperatorInput}/EntryOperatorInputs>
 - Link Mapping <{OperatorOutput.result:Result}/OperatorOutputResult>
 - Canvas Mapping
 - platform:/resource/Math/model/math.ecore
 - platform:/resource/Math/model/math.gmfgraph

At the bottom, there is a navigation bar with the following options: Selection, Parent, List, Tree, Table, Tree with Columns.



```
59 org.eclipse.emf.common.util.URI modelURI = URI.createFileURI("D:/model.kb");
60 KbPackageImpl.init();
61 Resource.Factory.Registry.INSTANCE.getExtensionToFactoryMap().put("kb", new XMIResourceFactoryImpl()
62 {
63
64     public Resource createResource(URI uri)
65     {
66         XMIResource xmiResource = new XMIResourceImpl(uri);
67         return xmiResource;
68     }
69 });
70
71 ResourceSet resSet = new ResourceSetImpl();
72 org.eclipse.emf.ecore.resource.Resource myModel = resSet.createResource(modelURI);
73 KbDiagram diagram = KbFactory.eINSTANCE.createKbDiagram();
74 myModel.getContents().add(diagram);
75
76 Architecture architecture = KbFactory.eINSTANCE.createArchitecture();
77 diagram.getArchitectures().add(architecture);
78
79 try {
80     myModel.save(null);
81 } catch (IOException e) {
82     // TODO Auto-generated catch block
83     e.printStackTrace();
84 }
```

```
23 org.eclipse.emf.common.util.URI modelURI = URI.createFileURI(kbPath);
24 KbPackageImpl.init();
25 Resource.Factory.Registry.INSTANCE.getExtensionToFactoryMap().put("kb", new XMIResourceFactoryImpl()
26 {
27
28     public Resource createResource(URI uri)
29     {
30         XMIResource xmiResource = new XMIResourceImpl(uri);
31         return xmiResource;
32     }
33 });
34
35 ResourceSet resSet = new ResourceSetImpl();
36 org.eclipse.emf.ecore.resource.Resource myModel = resSet.getResource(modelURI, true);
37
38 kb = (KbDiagram)myModel.getContents().get(0);
```

```
150 public void testGetSubInterfaces() {
151     System.out.println("testGetSubInterfaces()");
152     Port p1 = KbFactory.eINSTANCE.createPort();
153     p1.setName("p1");
154     Port p2 = KbFactory.eINSTANCE.createPort();
155     p2.setName("p2");
156     Port p3 = KbFactory.eINSTANCE.createPort();
157     p3.setName("p3");
158
159     EList<Port> list = new BasicEList<Port>();
160     list.add(p2);
161     list.add(p3);
162     p1.getSubPort().addAll(list);
163
164     EList<Interface> ret = p1.getSubInterfaces();
165     System.out.println(p1.toString()+"");
166     for (Interface inf: ret)
167     {
168         System.out.println("- "+inf.toString());
169     }
170     System.out.println("-----");
171 }
```

```
59 org.eclipse.emf.common.util.URI modelURI = URI.createFileURI("D:/model.kb");
60 KbPackageImpl.init();
61 Resource.Factory.Registry.INSTANCE.getExtensionToFactoryMap().put("kb", new XMIResourceFactoryImpl()
62 {
63
64     public Resource createResource(URI uri)
65     {
66         XMIResource xmiResource = new XMIResourceImpl(uri);
67         return xmiResource;
68     }
69 });
70
71 ResourceSet resSet = new ResourceSetImpl();
72 org.eclipse.emf.ecore.resource.Resource myModel = resSet.createResource(modelURI);
73 KbDiagram diagram = KbFactory.eINSTANCE.createKbDiagram();
74 myModel.getContents().add(diagram);
75
76 Architecture architecture = KbFactory.eINSTANCE.createArchitecture();
77 diagram.getArchitectures().add(architecture);
78
79 try {
80     myModel.save(null);
81 } catch (IOException e) {
82     // TODO Auto-generated catch block
83     e.printStackTrace();
84 }
```

- <http://www.ibm.com/developerworks/open-source/library/os-ecl-gmf/>
- http://gmfsamples.tuxfamily.org/wiki/doku.php?id=gmf_tutorial5

Děkuji Vám za pozornost!