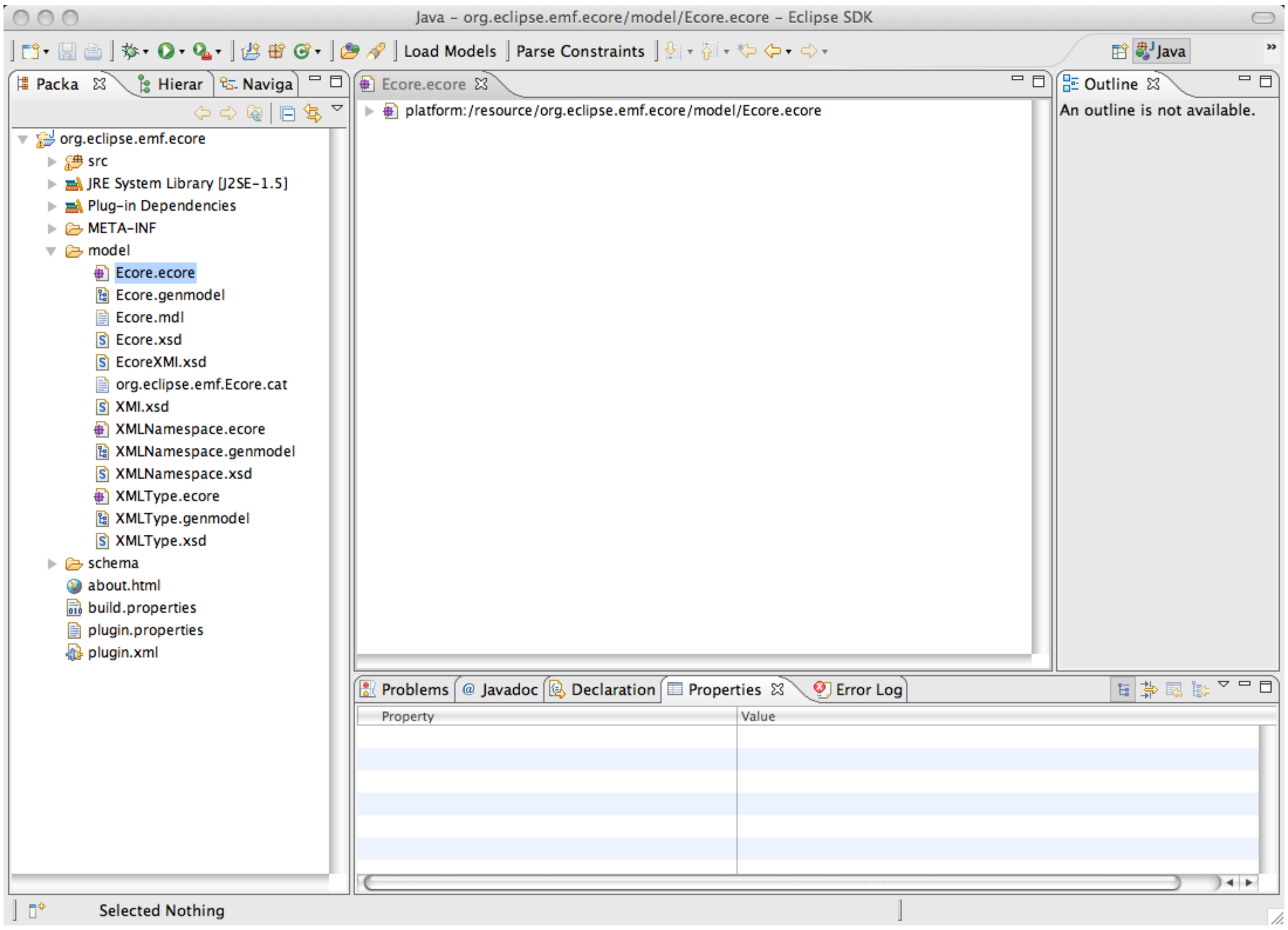
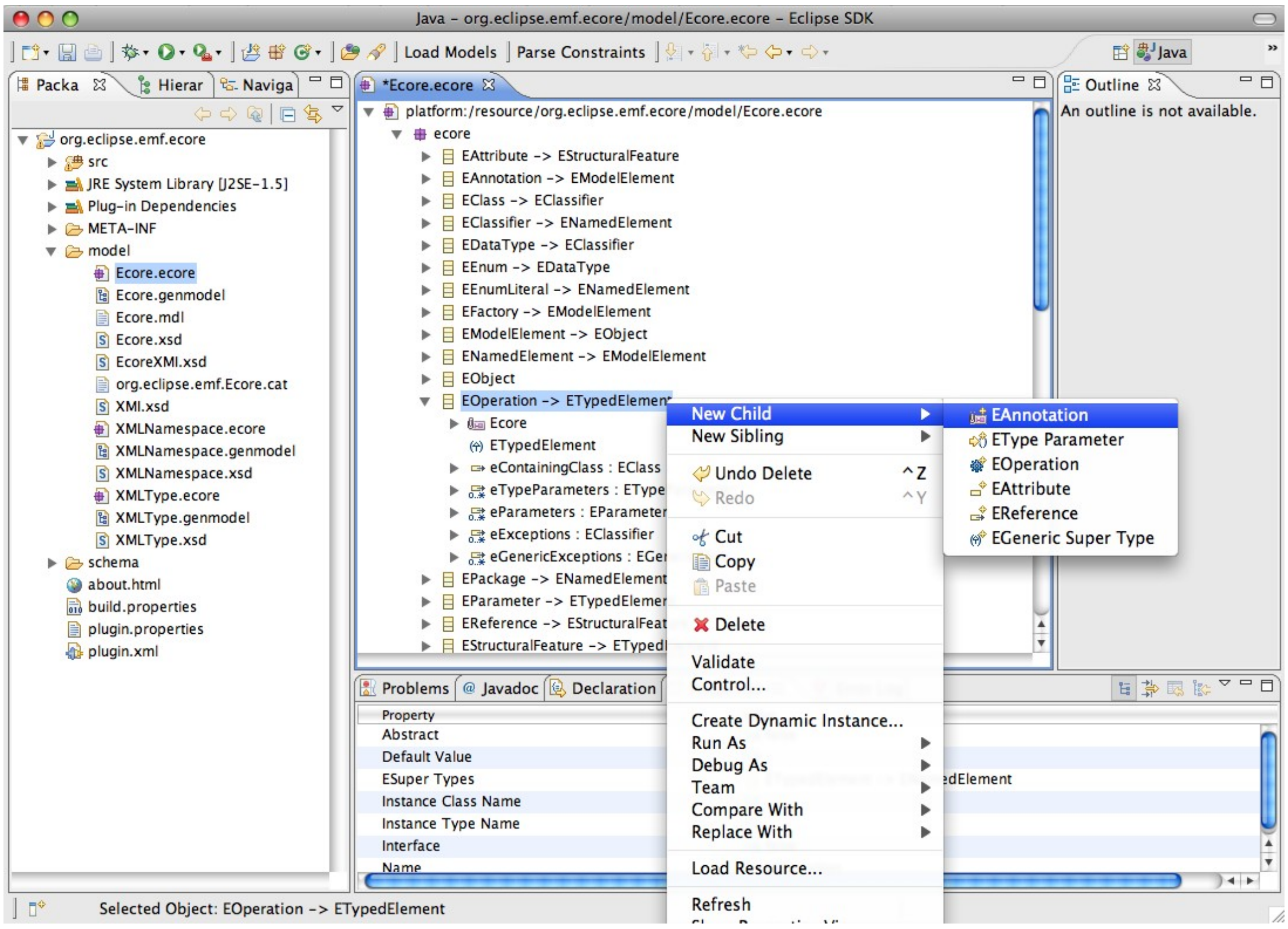
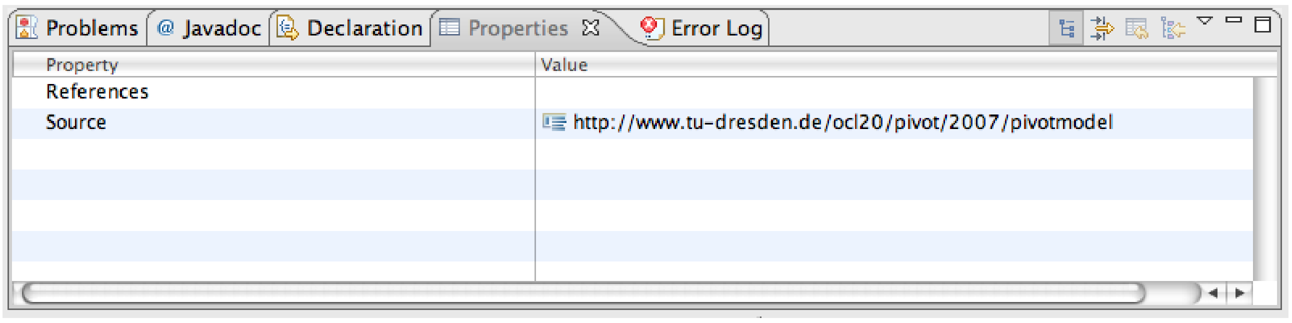
- Start Eclipse with all neccessary OCL2.0-Plugins
  - org.apache.commons.lang
  - tudresden.ocl20.logging
  - tudresden.ocl20.pivot.codegen.adapter
  - tudresden.ocl20.pivot.essentialocl
  - tudresden.ocl20.pivot.essentialocl.edit
  - tudresden.ocl20.pivot.essentialocl.editor
  - tudresden.ocl20.pivot.essentialocl.standardlibrary
  - tudresden.ocl20.pivot.essentialocl.tests
  - tudresden.ocl20.pivot.modelbus
  - tudresden.ocl20.pivot.modelbus.ui
  - tudresden.ocl20.pivot.parser
  - tudresden.ocl20.pivot.parser.ui
  - tudresden.ocl20.pivot.pivotmodel
  - tudresden.ocl20.pivot.pivotmodel.edit
  - tudresden.ocl20.pivot.pivotmodel.tests
  - tudresden.ocl20.pivot.standardlibrary
  - tudresden.ocl20.pivot.xocl
  - tudresden.ocl20.pivot.xocl.edit
  - tudresden.ocl20.pivot.xocl.editor
- To integrate your DSL with the Dresden OCL2.0 Toolkit you need an EMF representation of the DSL's metamodel.

- If your metamodel is completed, you have to annotate all DSL model elements that are to be mapped to Pivot Model types. For example, if your DSL is Ecore itself, you want to map the EOperation to the Operation Pivot Model type.



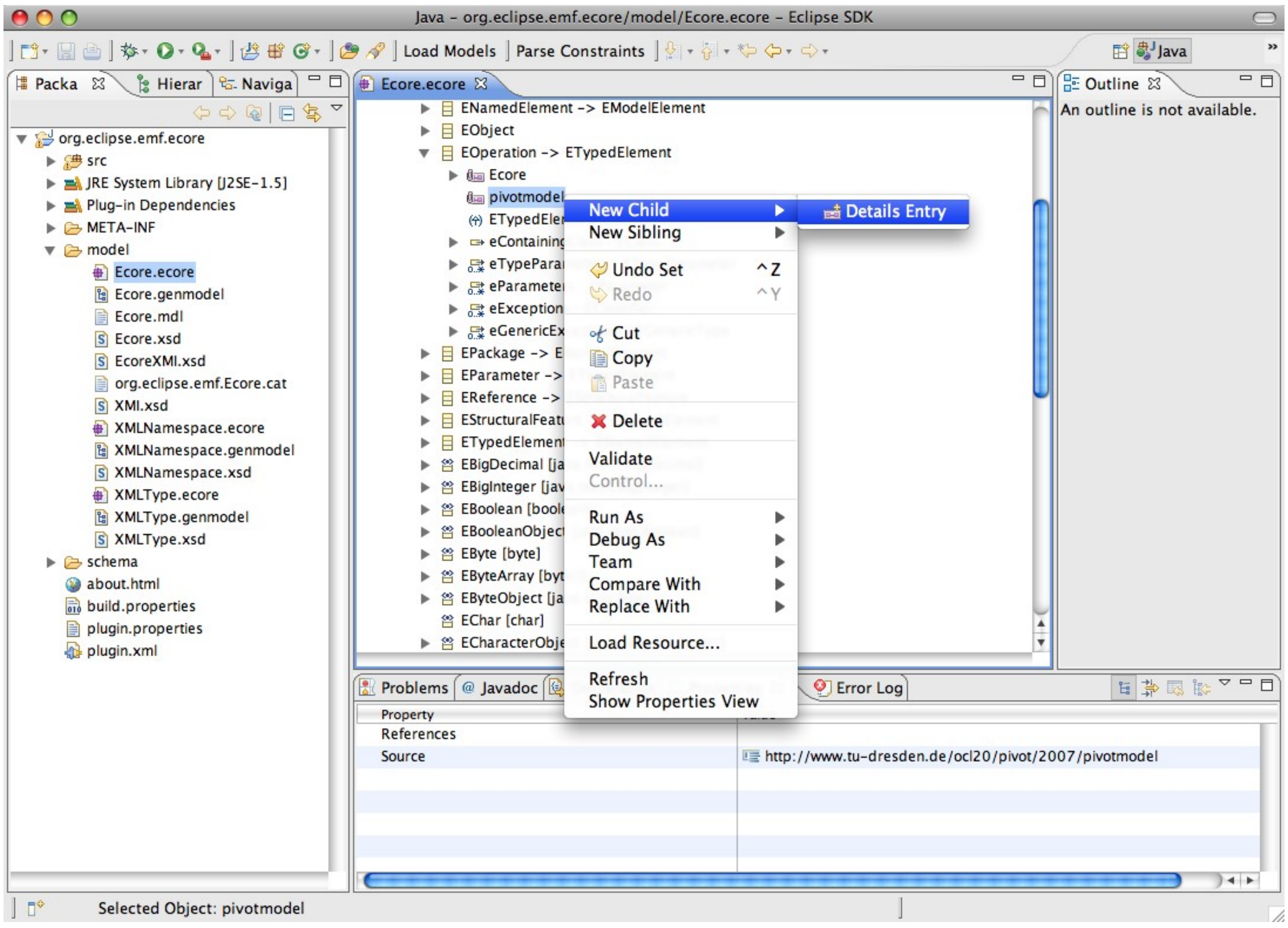
- To do this, select the EOperation in the EMF editor. Right-click and select „New Child -> EAnnotation“. In the properties view, name the Source „http://www.tu-dresden.de/ocl20/pivot/2007/pivotmodel“.



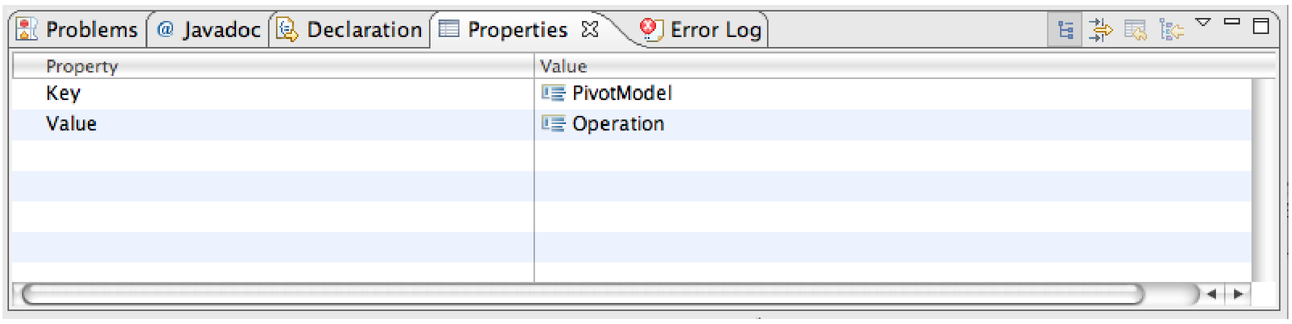


Property	Value
References	
Source	http://www.tu-dresden.de/ocl20/pivot/2007/pivotmodel

- Select the annotation, right-click and select „New Child -> Details Entry“. In the properties view add „PivotModel“ as the Key. The Value in this example would be „Operation“. This is the name of the Pivot Model type this DSL element is mapped to.

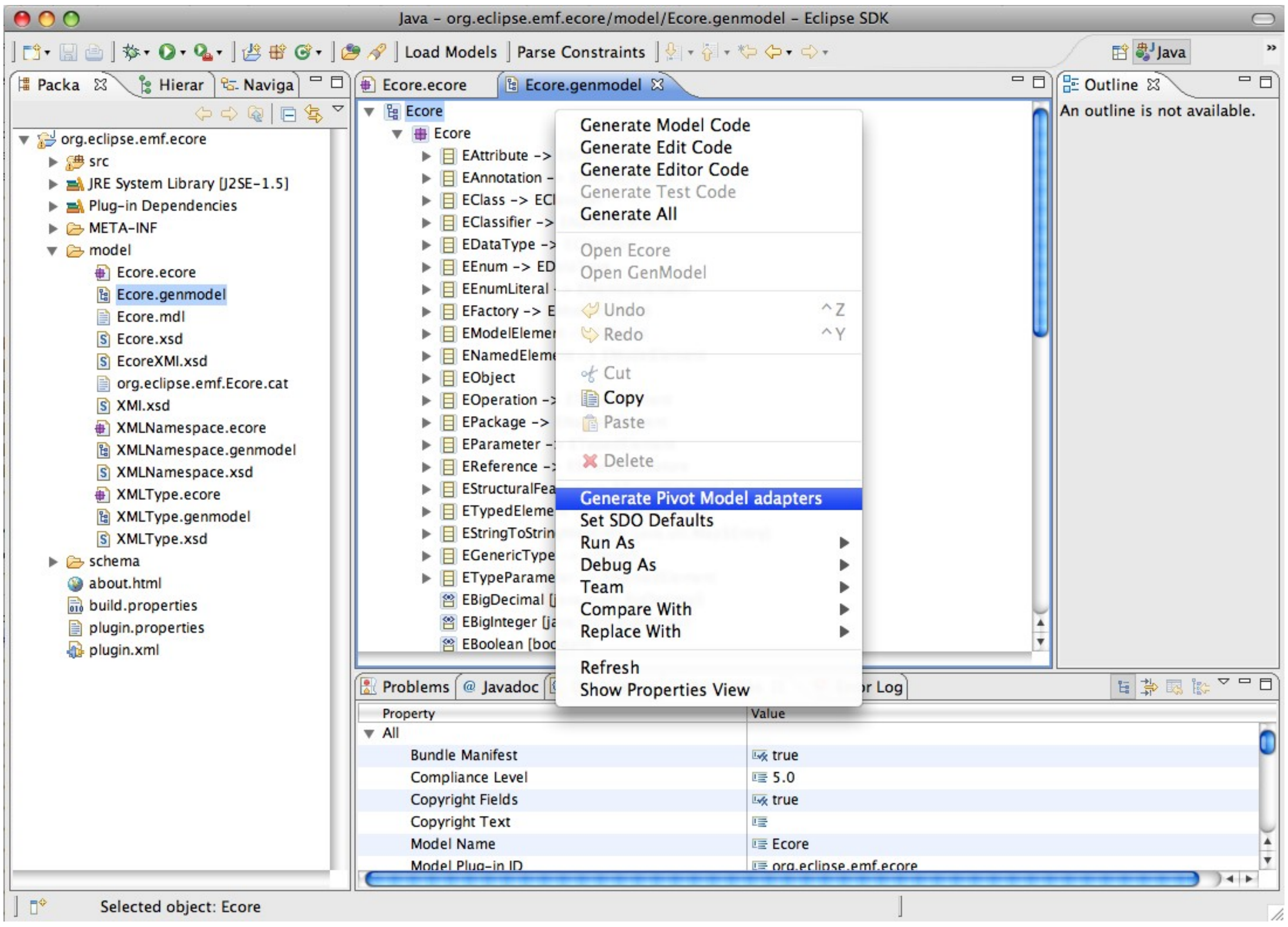


Property	Value
References	
Source	http://www.tu-dresden.de/ocl20/pivot/2007/pivotmodel



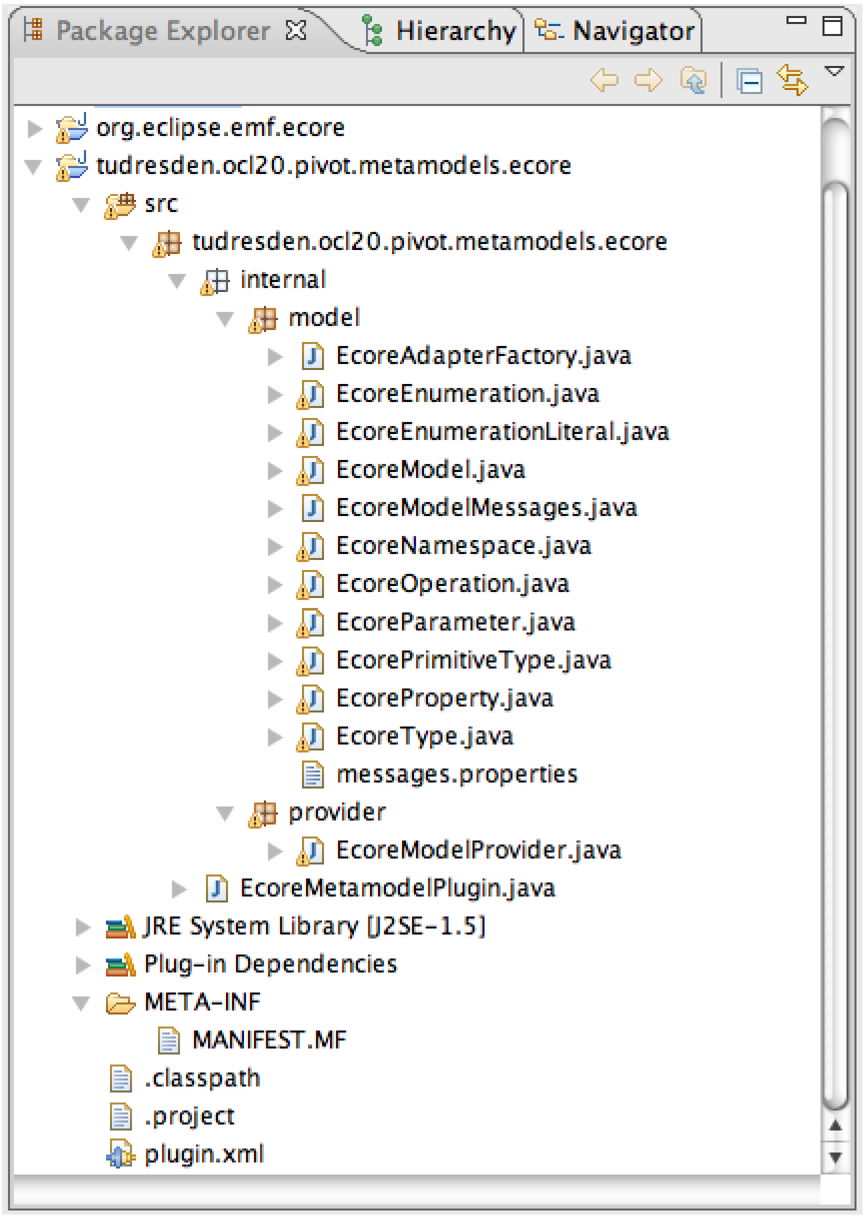
Property	Value
Key	PivotModel
Value	Operation

- Repeat the last step for all Pivot Model elements.
- If you are ready, open the DSL's GenModel. Right-click on the top-level element and select „Generate Pivot Model adapters“.

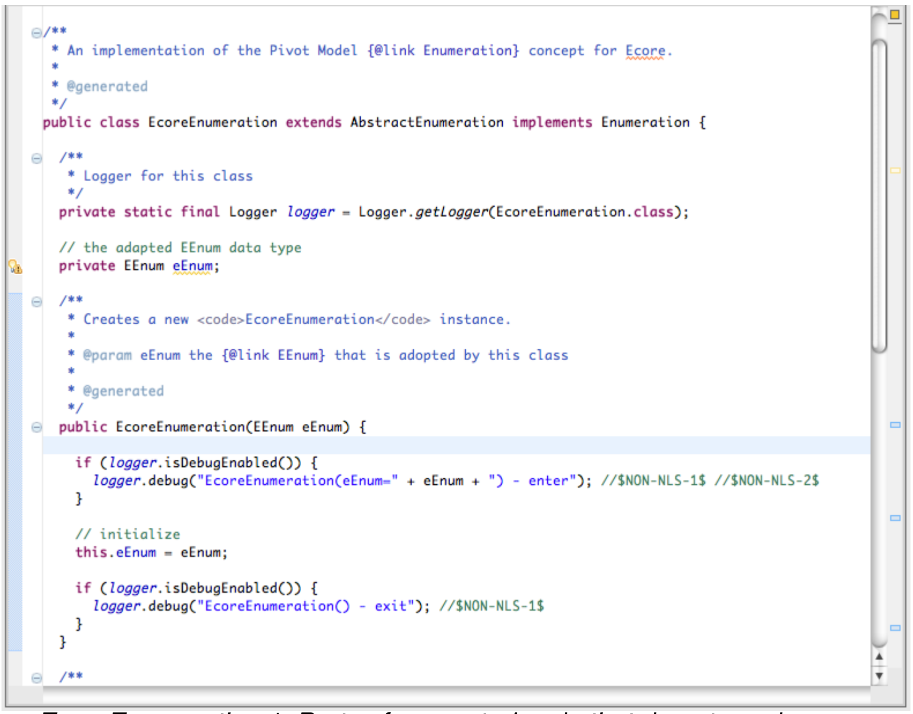


Property	Value
All	
Bundle Manifest	true
Compliance Level	5.0
Copyright Fields	true
Copyright Text	
Model Name	Ecore
Model Plug-in ID	org.eclipse.emf.ecore

- A new Eclipse Plug-in is generated. It is named „tudresden.ocl20.pivot.metamodels.ecore“.

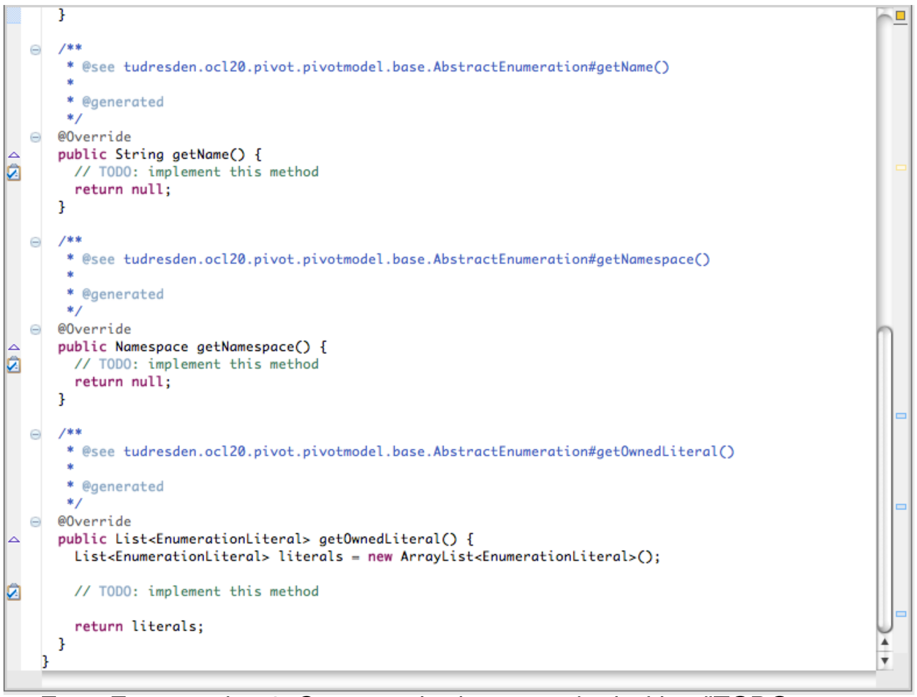


```
org.eclipse.emf.ecore
tudresden.ocl20.pivot.metamodels.ecore
  src
    tudresden.ocl20.pivot.metamodels.ecore
      internal
        model
          EcoreAdapterFactory.java
          EcoreEnumeration.java
          EcoreEnumerationLiteral.java
          EcoreModel.java
          EcoreModelMessages.java
          EcoreNamespace.java
          EcoreOperation.java
          EcoreParameter.java
          EcorePrimitiveType.java
          EcoreProperty.java
          EcoreType.java
          messages.properties
        provider
          EcoreModelProvider.java
      EcoreMetamodelPlugin.java
  JRE System Library [J2SE-1.5]
  Plug-in Dependencies
  META-INF
    MANIFEST.MF
  .classpath
  .project
  plugin.xml
```



```
/**
 * An implementation of the Pivot Model {@link Enumeration} concept for Ecore.
 *
 * @generated
 */
public class EcoreEnumeration extends AbstractEnumeration implements Enumeration {

  /**
   * Logger for this class
   */
  private static final Logger logger = Logger.getLogger(EcoreEnumeration.class);

  // the adapted EEnum data type
  private EEnum eEnum;

  /**
   * Creates a new <code>EcoreEnumeration</code> instance.
   *
   * @param eEnum the {@link EEnum} that is adopted by this class
   *
   * @generated
   */
  public EcoreEnumeration(EEnum eEnum) {

    if (logger.isDebugEnabled()) {
      logger.debug("EcoreEnumeration(eEnum=" + eEnum + ") - enter"); //$NON-NLS-1$ //$NON-NLS-2$
    }

    // initialize
    this.eEnum = eEnum;

    if (logger.isDebugEnabled()) {
      logger.debug("EcoreEnumeration() - exit"); //$NON-NLS-1$
    }
  }

  /**
```

EcoreEnumeration 1: Parts of generated code that do not need any refinement



```
  }

  /**
   * @see tudresden.ocl20.pivot.pivotmodel.base.AbstractEnumeration#getName()
   *
   * @generated
   */
  @Override
  public String getName() {
    // TODO: implement this method
    return null;
  }

  /**
   * @see tudresden.ocl20.pivot.pivotmodel.base.AbstractEnumeration#getNamespace()
   *
   * @generated
   */
  @Override
  public Namespace getNamespace() {
    // TODO: implement this method
    return null;
  }

  /**
   * @see tudresden.ocl20.pivot.pivotmodel.base.AbstractEnumeration#getOwnedLiteral()
   *
   * @generated
   */
  @Override
  public List<EnumerationLiteral> getOwnedLiteral() {
    List<EnumerationLiteral> literals = new ArrayList<EnumerationLiteral>();

    // TODO: implement this method

    return literals;
  }
}
```

EcoreEnumeration 2: Some methods are marked with a "TODO: implement this method" or other TODOs.