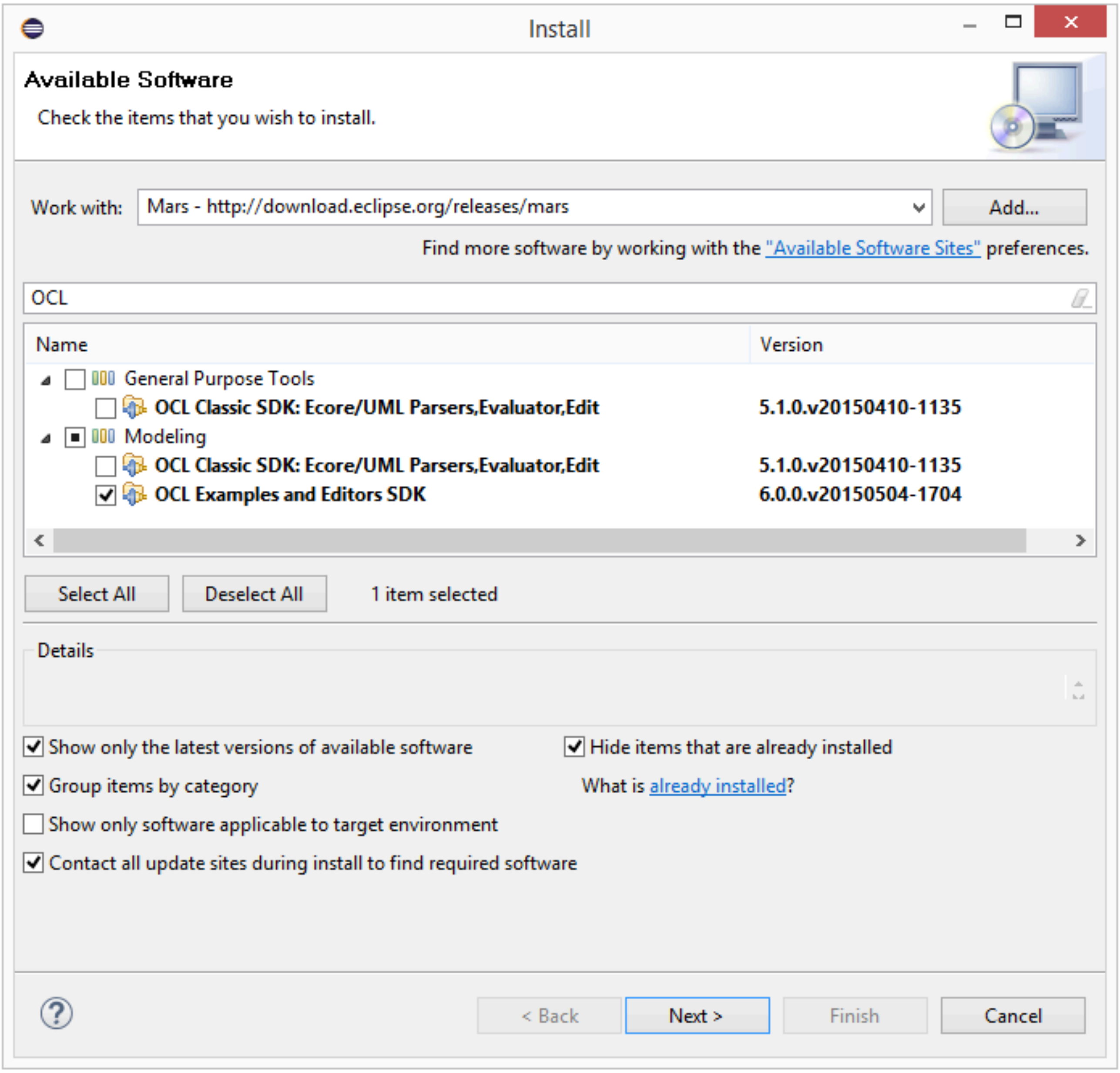
## 1. Install OCL and the additional Editors

From Eclipse menu, **Help -> Install New Software**…

From default update site: expand **Modelling** category and select **OCL Examples and Editors SDK**



Install

**Available Software**

Check the items that you wish to install.

Work with: Mars - http://download.eclipse.org/releases/mars

Find more software by working with the "Available Software Sites" preferences.

OCL

| Name | Version |
|---|---|
| □ General Purpose Tools | |
| □ **OCL Classic SDK: Ecore/UML Parsers,Evaluator,Edit** | **5.1.0.v20150410-1135** |
| ■ Modeling | |
| □ **OCL Classic SDK: Ecore/UML Parsers,Evaluator,Edit** | **5.1.0.v20150410-1135** |
| ☑ **OCL Examples and Editors SDK** | **6.0.0.v20150504-1704** |

1 item selected

Details

☑ Show only the latest versions of available software

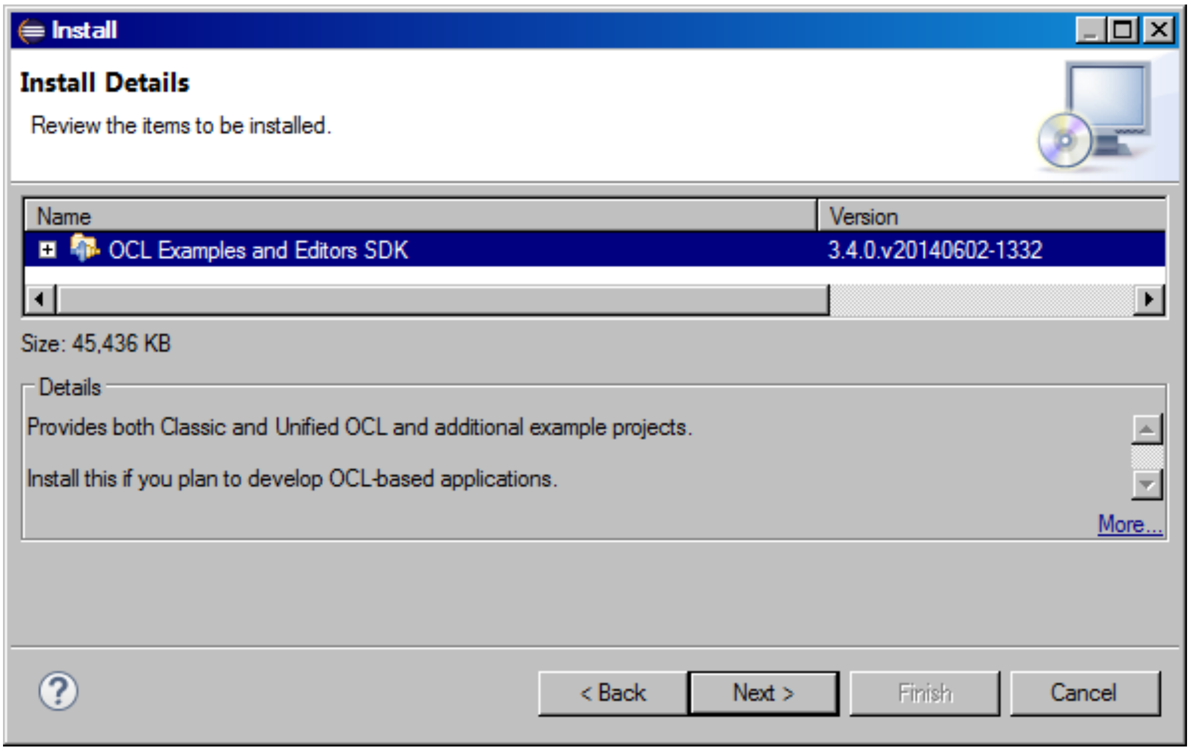
☑ Hide items that are already installed

☑ Group items by category

What is already installed?

□ Show only software applicable to target environment

☑ Contact all update sites during install to find required software



**Install Details**

Review the items to be installed.

| Name | Version |
|---|---|
| OCL Examples and Editors SDK | 3.4.0.v20140602-1332 |

Size: 45,436 KB

Details

Provides both Classic and Unified OCL and additional example projects.

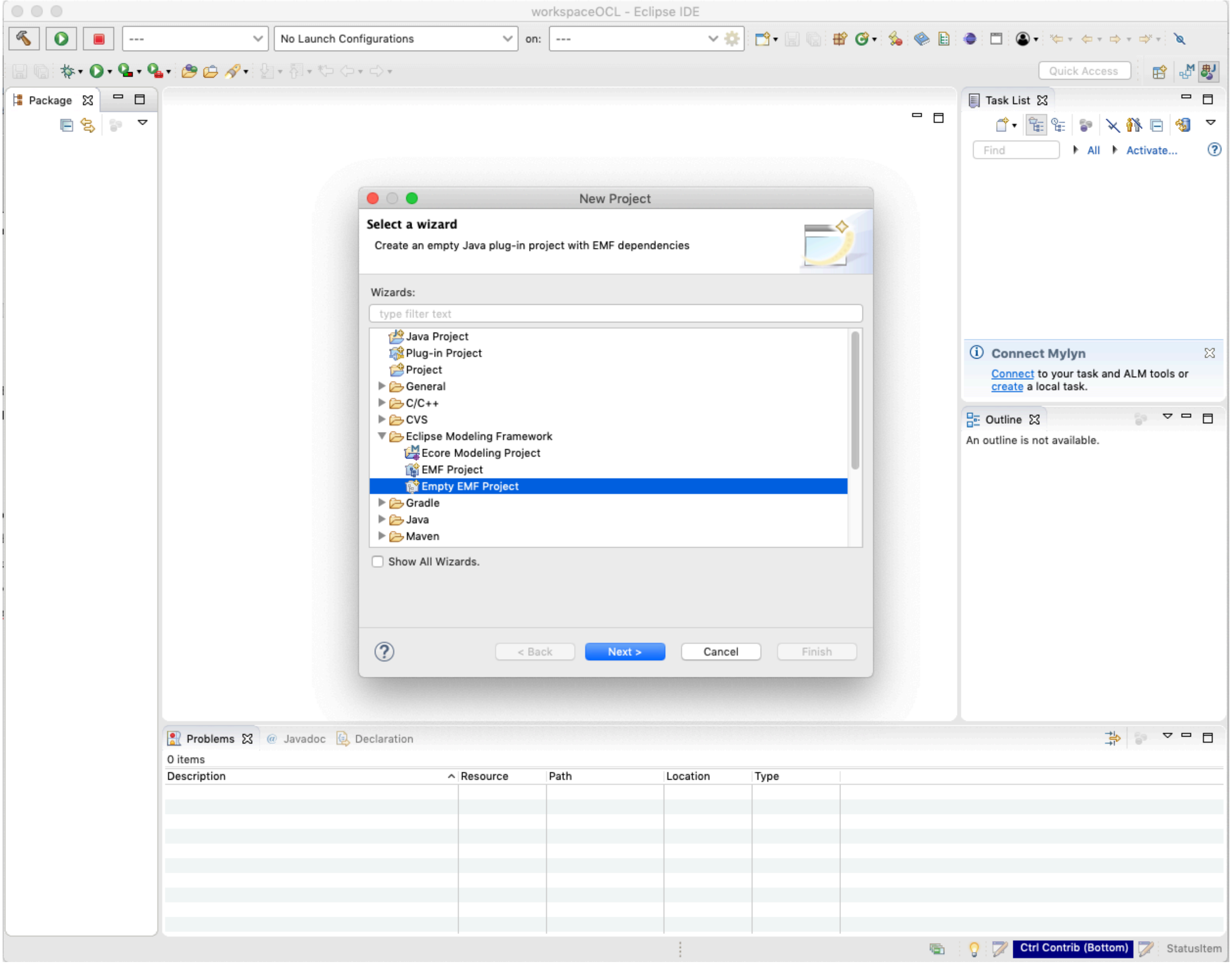
Install this if you plan to develop OCL-based applications.

**Restart** Eclipse when finished.

## 2. Use the OCLinEcore editor

### 2.1 Create an Ecore model using the OCLinEcore text editor

From Eclipse menu, **File -> New -> Project** then **Eclipse Modeling Framework -> Empty EMF Project**



New Project

Select a wizard

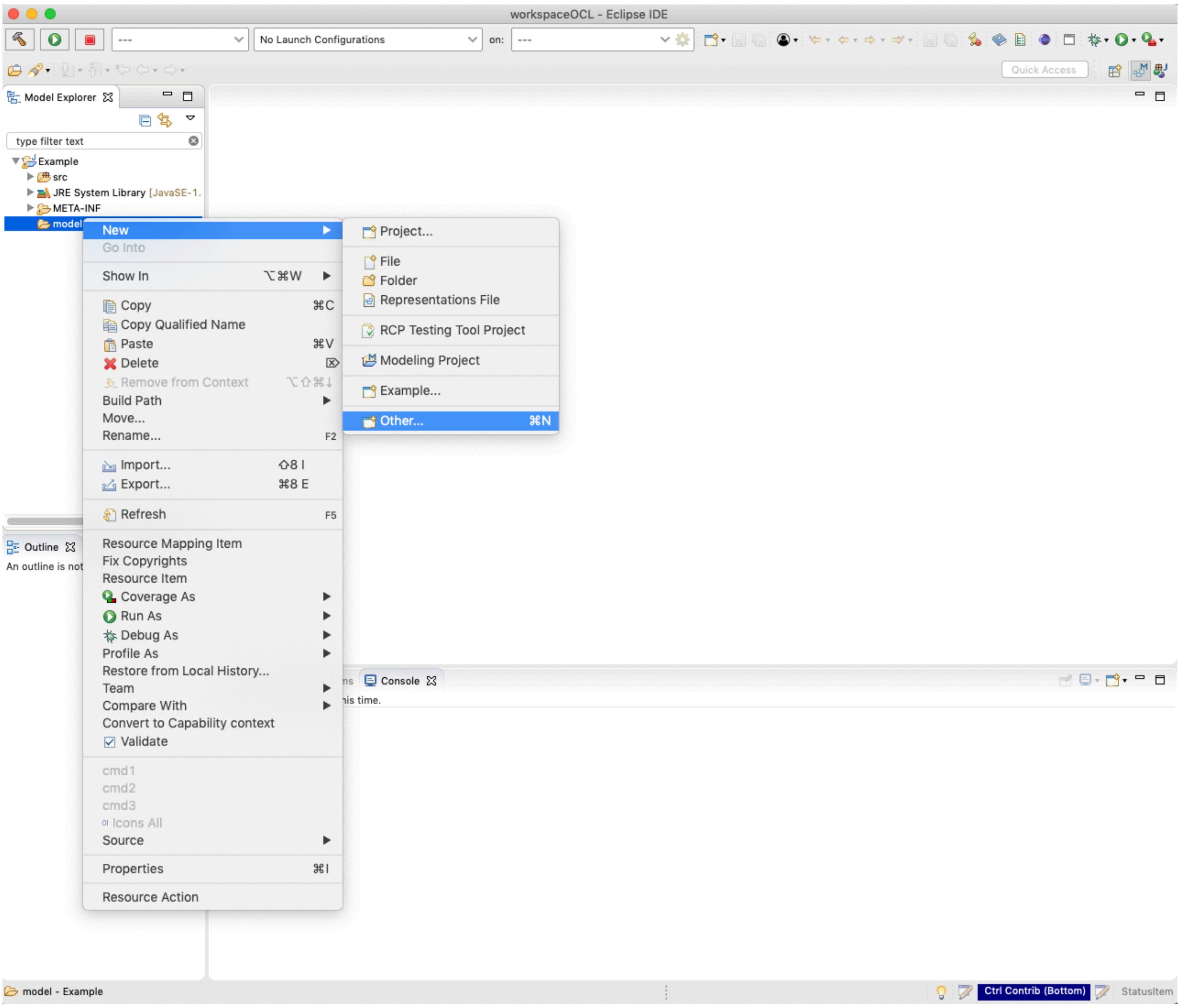
Create an empty Java plug-in project with EMF dependencies

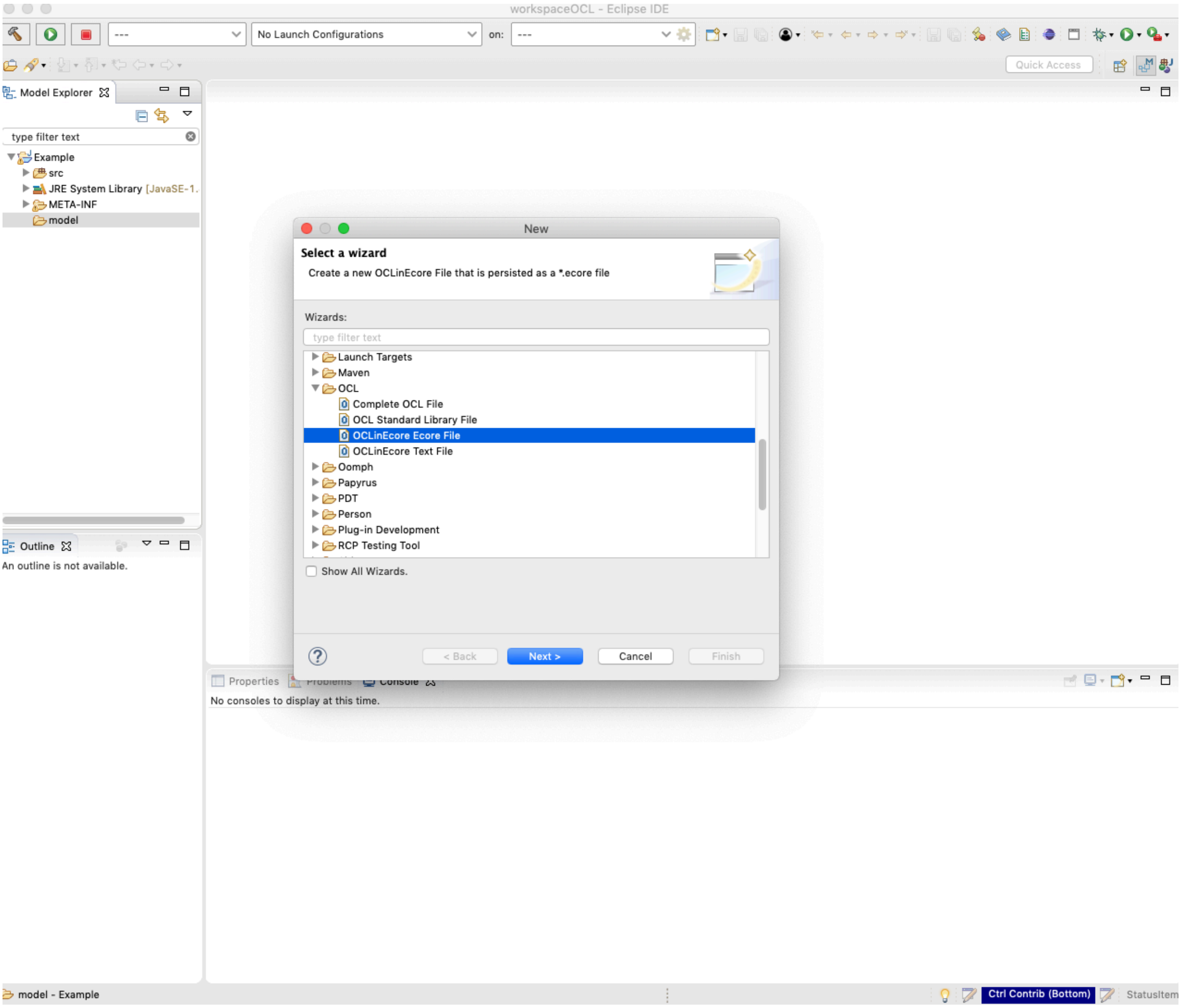
Wizards:

Give a project name -> **Finish**

Then, create a new Ecore model:

Right-click on the **model** folder in the **Library** project to define the target folder and pop-up the context-sensitive menu. Select **New -> Other...** then select the **OCLinEcore Ecore File** from the **OCL** category.


workspaceOCL - Eclipse IDE
No Launch Configurations
on:
Quick Access
Model Explorer
type filter text
Example
src
JRE System Library [JavaSE-1.
META-INF
model
New
Go Into
Show In ⌥⌘W
Copy ⌘C
Copy Qualified Name
Paste ⌘V
Delete
Remove from Context ⌥⇧⌘↓
Build Path
Move...
Rename... F2
Import... ⇧8 I
Export... ⌘8 E
Refresh F5
Resource Mapping Item
Fix Copyrights
Resource Item
Coverage As
Run As
Debug As
Profile As
Restore from Local History...
Team
Compare With
Convert to Capability context
Validate
cmd1
cmd2
cmd3
Icons All
Source
Properties ⌘I
Resource Action
Project...
File
Folder
Representations File
RCP Testing Tool Project
Modeling Project
Example...
Other... ⌘N
Outline
An outline is not
Console
his time.
model - Example
Ctrl Contrib (Bottom)
StatusItem

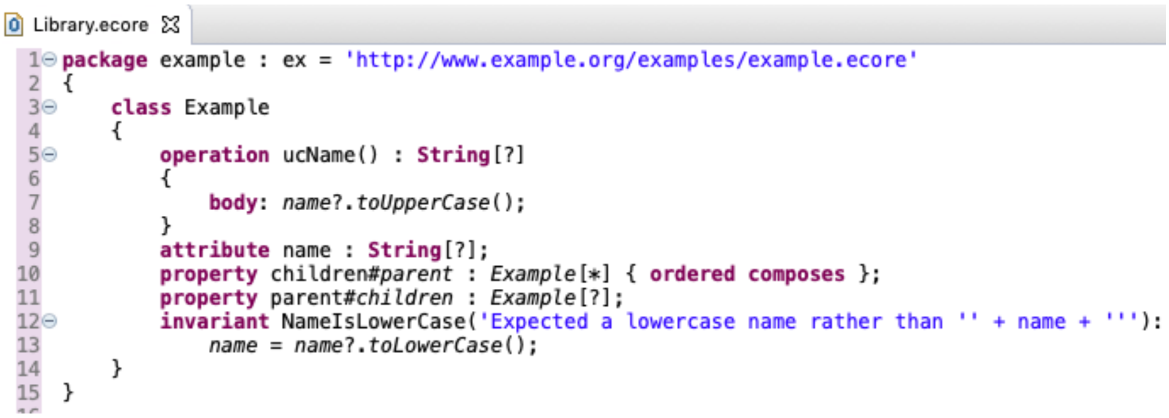


Select a wizard

Create a new OCLinEcore File that is persisted as a *.ecore file

No consoles to display at this time.

Give a name of the model. In this example, let's name it **Library.ecore**
Then, you will have like a file created like below:



```
package example : ex = 'http://www.example.org/examples/example.ecore'
{
    class Example
    {
        operation ucName() : String[?]
        {
            body: name?.toUpperCase();
        }
        attribute name : String[?];
        property children#parent : Example[*] { ordered composes };
        property parent#children : Example[?];
        invariant NameIsLowerCase('Expected a lowercase name rather than '' + name + '''):
            name = name?.toLowerCase();
    }
}
```

Close the editor for now.

To see the Ecore view of the model, Right-click on **Library.ecore** then **Open With -> Sample Ecore Model Editor**.
Close the editor for now.

Until now, an example model is just created. But, we need to create our own model with our classes, attributes, and relationships. To do that we need to edit the Ecore model we just created.

## Edit Ecore Model as OCLinEcore

Now we will open the Ecore model using the OCLinEcore text editor and provide some initial content.

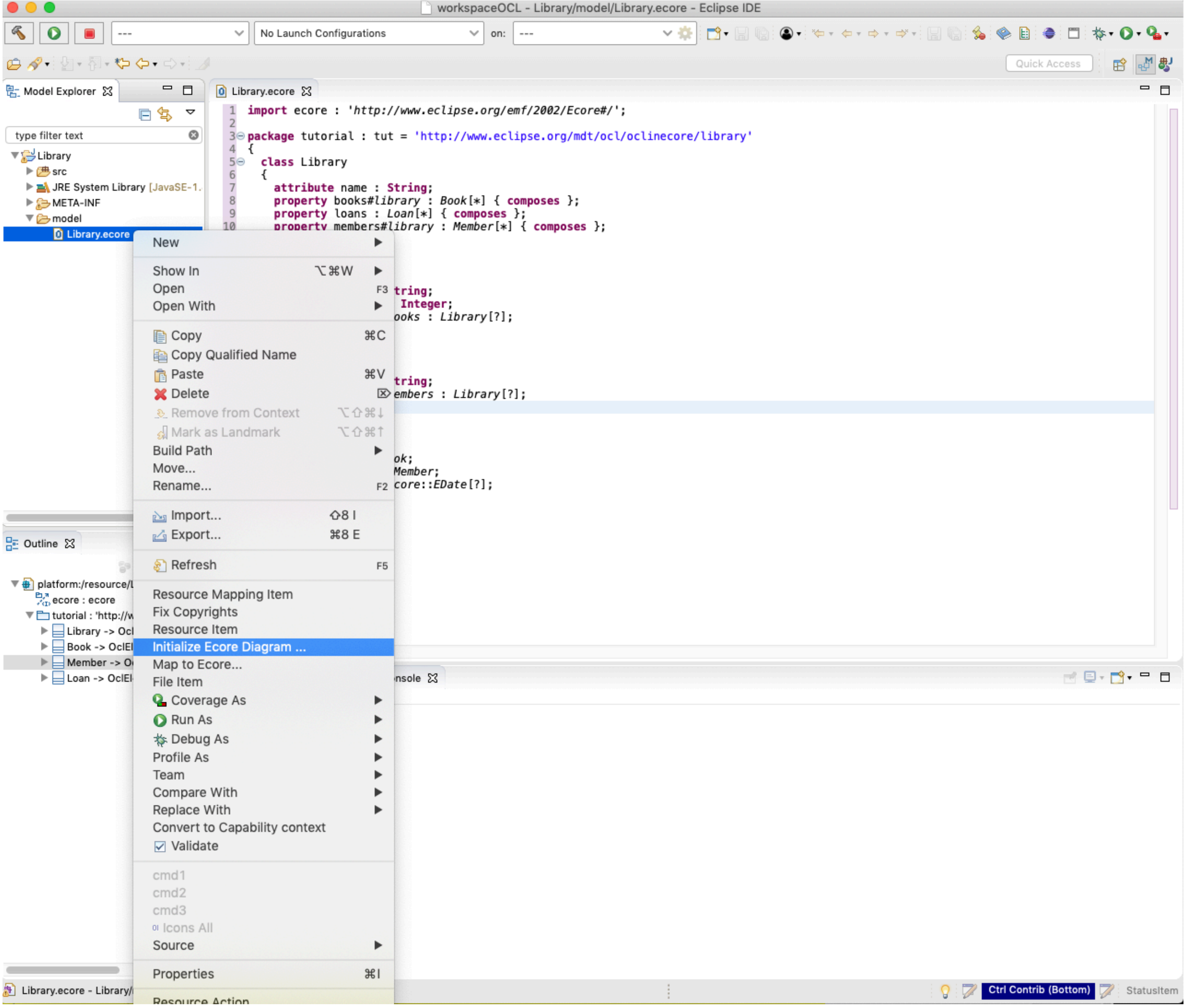
Right-click on the **Library.ecore** file then select **Open With -> OCLinEcore Editor** and replace the content with the following OCL code:

```
import ecore : 'http://www.eclipse.org/emf/2002/Ecore#/';

package Library : lib = 'http://www.eclipse.org/mdt/ocl/oclinecore/library'
{
  class Library
  {
    attribute name : String;
    property books#library : Book[*] { composes };
    property loans : Loan[*] { composes };
    property members#library : Member[*] { composes };
  }

  class Book
  {
    attribute name : String;
    attribute copies : Integer;
    property library#books : Library[?];
  }

  class Member
  {
    attribute name : String;
    property library#members : Library[?];
  }

  class Loan
  {
    property book : Book;
    property member : Member;
    attribute date : ecore::EDate[?];
  }
}
```

The syntax follows the OCLinEcore specification that emulates OMG specifications with **‘name : type[multiplicity] { properties }’.**

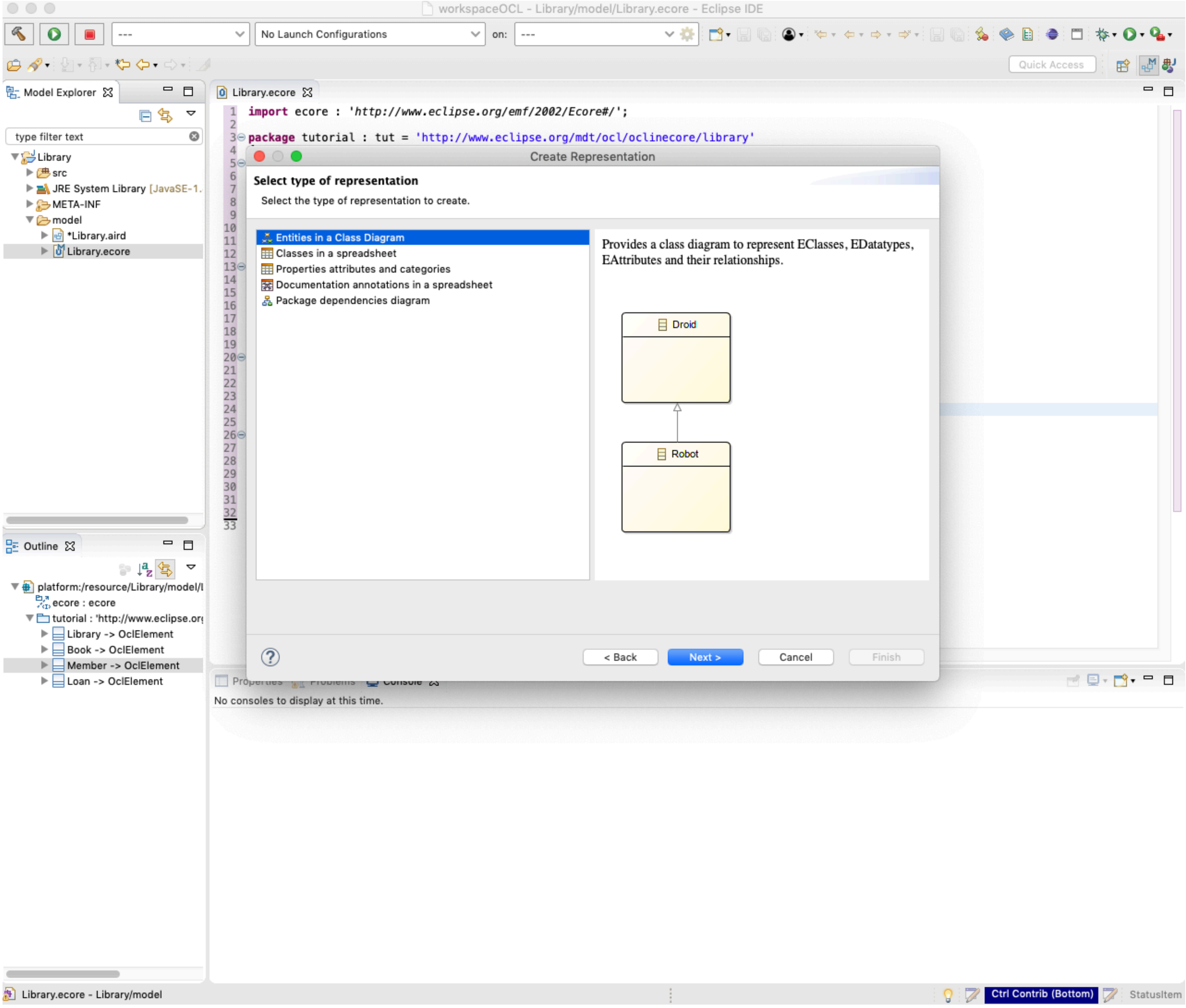
- `import` associates an alias with an external EPackage.
- `package` introduces an EPackage with name, nsPrefix and nsURI.
- `class` introduces an EClass with name and optional superclasses.
- `attribute` introduces a property with a datatype type (an EAttribute).
- `property` introduces a property with a class type (an EReference).
- `#` introduces an opposite role name.

We can generate the visual representation of the model as **Library.aird**:



```
import ecore : 'http://www.eclipse.org/emf/2002/Ecore#/';

package tutorial : tut = 'http://www.eclipse.org/mdt/ocl/oclinecore/library'
{
  class Library
  {
    attribute name : String;
    property books#library : Book[*] { composes };
    property loans : Loan[*] { composes };
    property members#library : Member[*] { composes };
```

```
tring;
 Integer;
ooks : Library[?];
tring;
embers : Library[?];
ok;
Member;
core::EDate[?];
```



workspaceOCL - Library/model/Library.ecore - Eclipse IDE

```
import ecore : 'http://www.eclipse.org/emf/2002/Ecore#';

package tutorial : tut = 'http://www.eclipse.org/mdt/ocl/oclinecore/library'
```

Create Representation

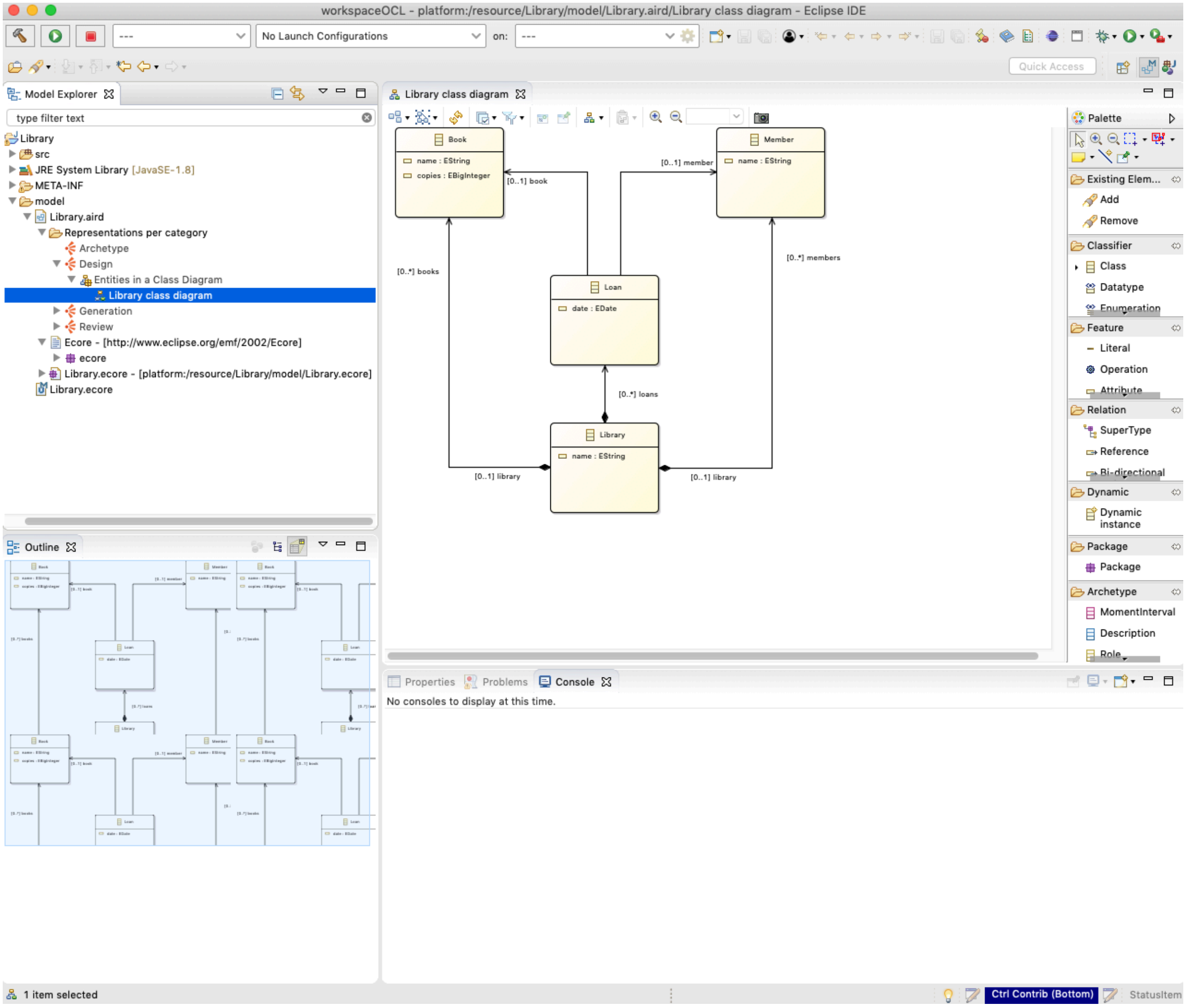
**Select type of representation**

Select the type of representation to create.

- Entities in a Class Diagram
- Classes in a spreadsheet
- Properties attributes and categories
- Documentation annotations in a spreadsheet
- Package dependencies diagram

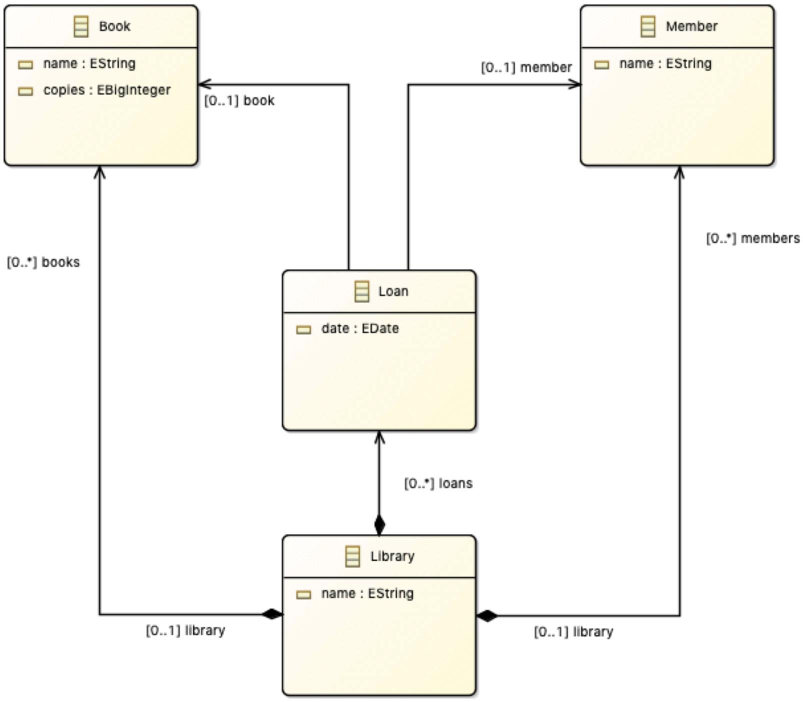
Provides a class diagram to represent EClasses, EDatatypes, EAttributes and their relationships.

No consoles to display at this time.



No consoles to display at this time.

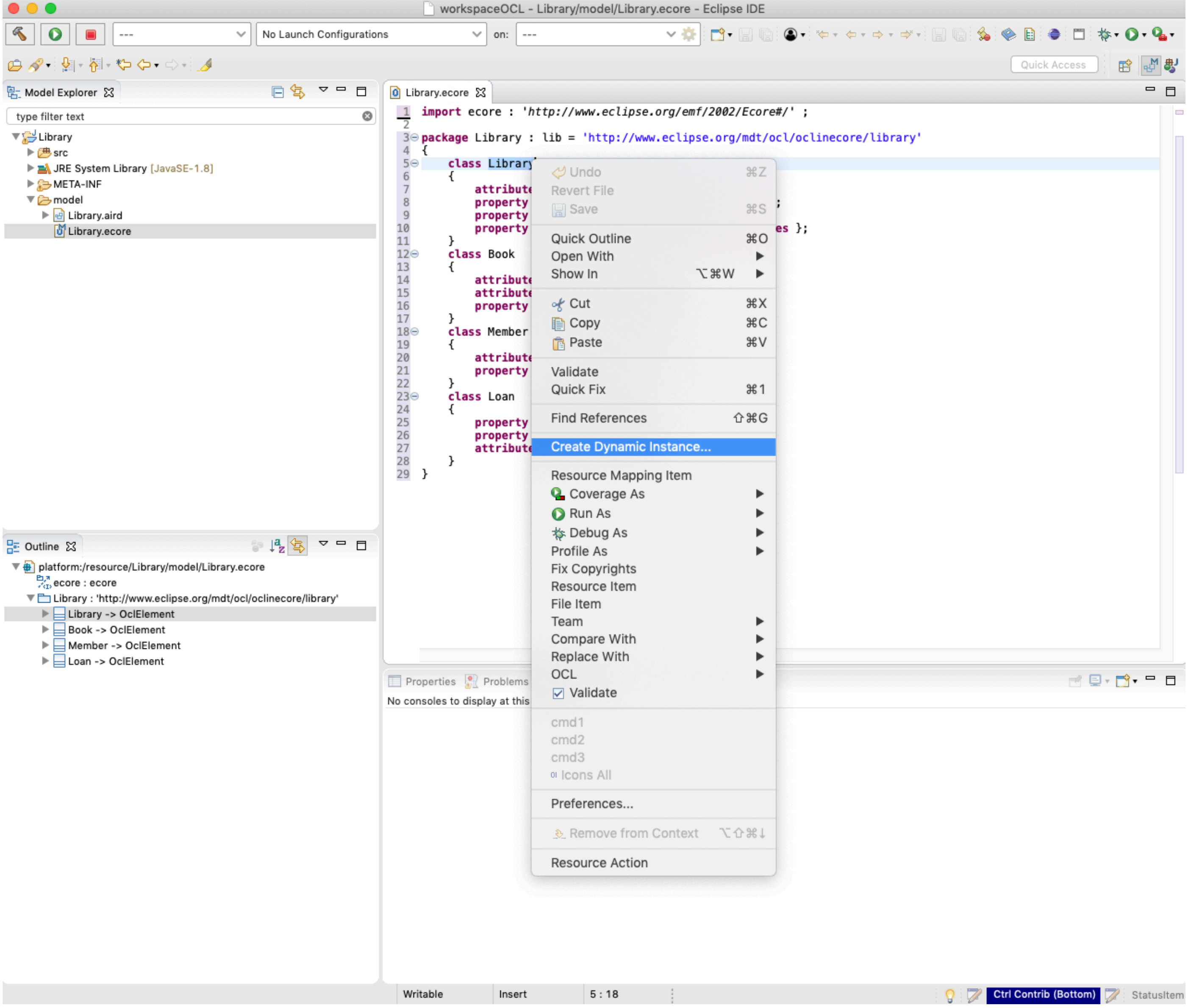
The generated class diagram looks like below:



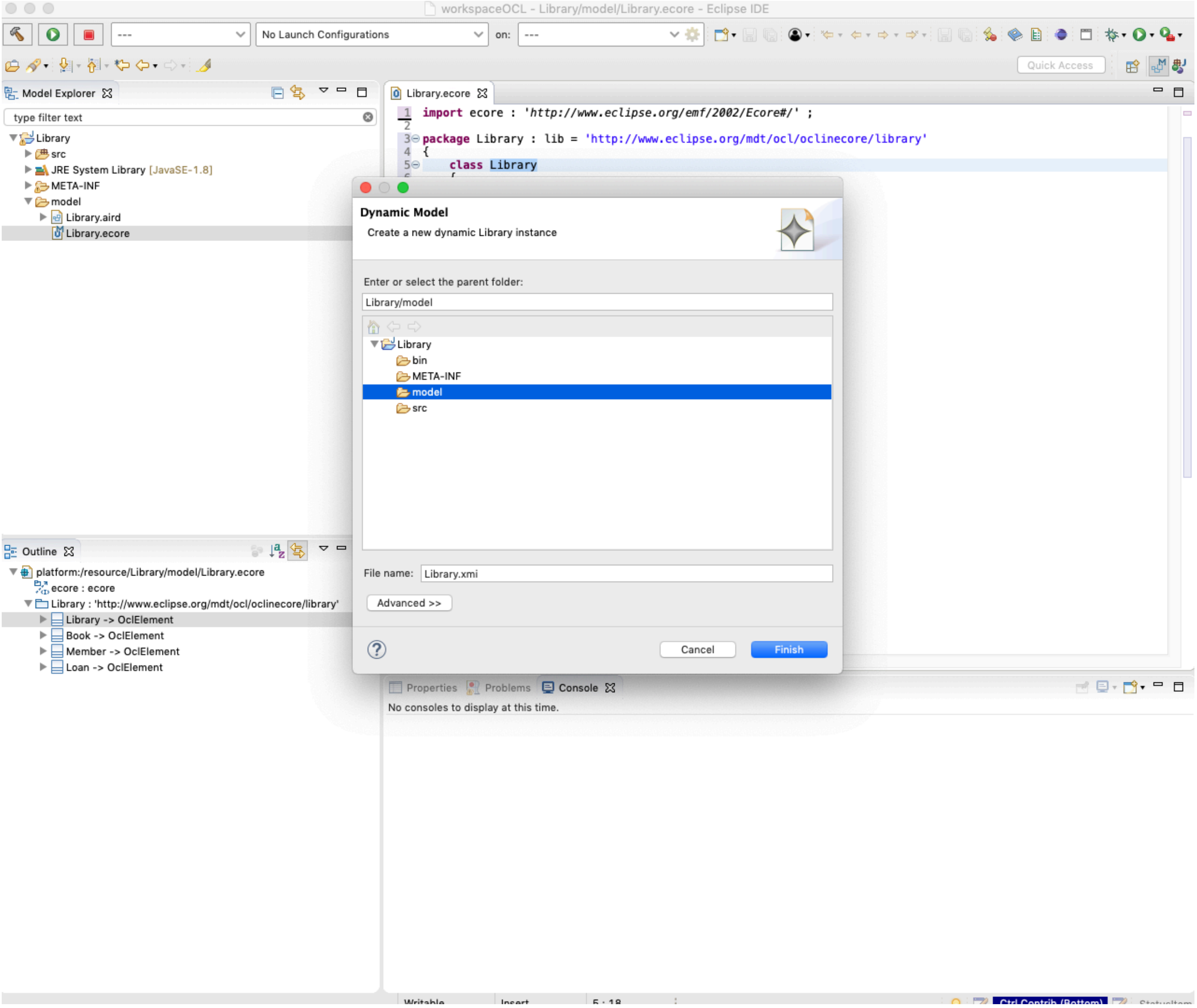
Now that we have the model, we need to create an instance of the metamodel for OCL to be functional.

## 2.2 Create a dynamic instance of that Ecore model

In the **OCLinEcore** Editor view, select the class **Library** then Right-click on it and select **Create Dynamic Instance…**



```
import ecore : 'http://www.eclipse.org/emf/2002/Ecore#/' ;

package Library : lib = 'http://www.eclipse.org/mdt/ocl/oclinecore/library'
{
    class Library
    {
        attribute
        property                                                ;
        property
        property                                                es };
    }
    class Book
    {
        attribute
        attribute
        property
    }
    class Member
    {
        attribute
        property
    }
    class Loan
    {
        property
        property
        attribute
    }
}
```



```
import ecore : 'http://www.eclipse.org/emf/2002/Ecore#' ;

package Library : lib = 'http://www.eclipse.org/mdt/ocl/oclinecore/library'
{
    class Library
```

**Dynamic Model**

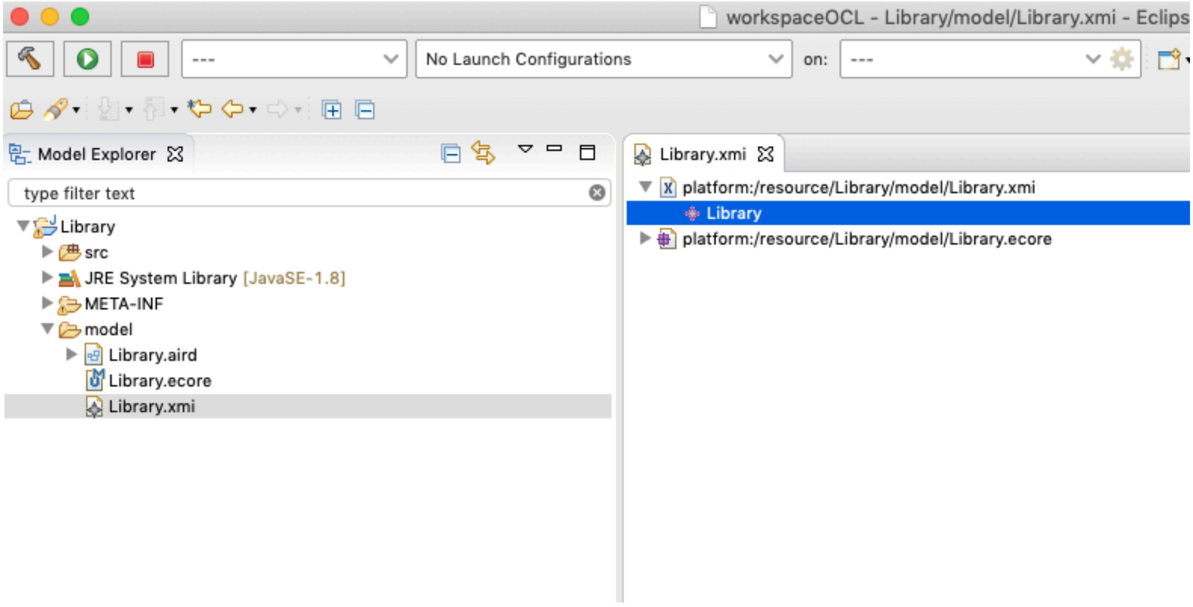
Create a new dynamic Library instance

Enter or select the parent folder: Library/model

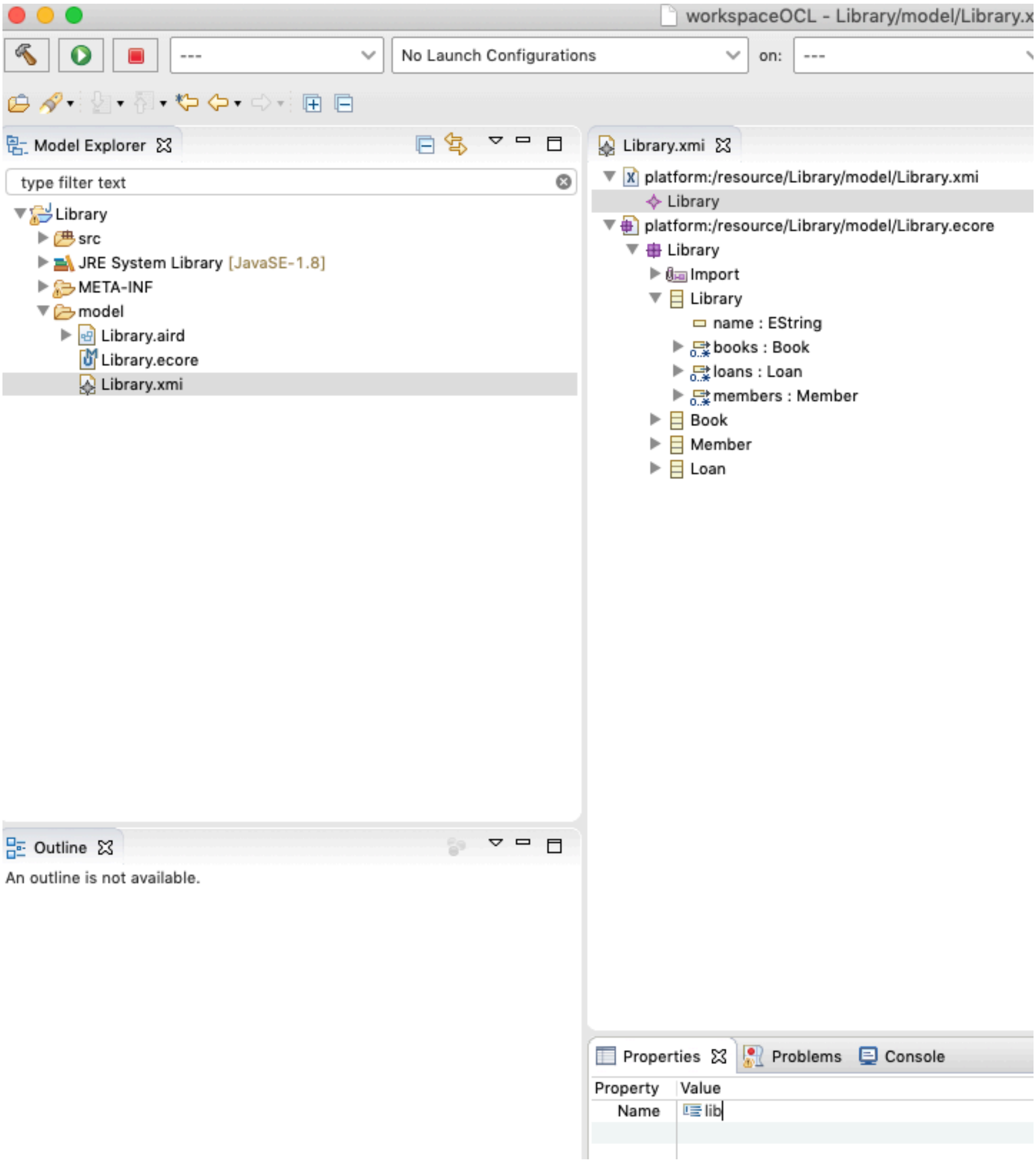
File name: Library.xmi

This will create an instance of the Library.ecore metamodel called **Library.xmi** in **model** folder.

Open the model instance **Library.xmi** using Right-click then **Open With -> Sample Reflective Ecore Model Editor**


workspaceOCL - Library/model/Library.xmi - Eclips
No Launch Configurations
on:
Model Explorer
type filter text
Library
src
JRE System Library [JavaSE-1.8]
META-INF
model
Library.aird
Library.ecore
Library.xmi
Library.xmi
platform:/resource/Library/model/Library.xmi
Library
platform:/resource/Library/model/Library.ecore

From here, select the model instance **Library** , then from **Properties** view give a name **lib.**



| Property | Value |
|---|---|
| Name | lib |

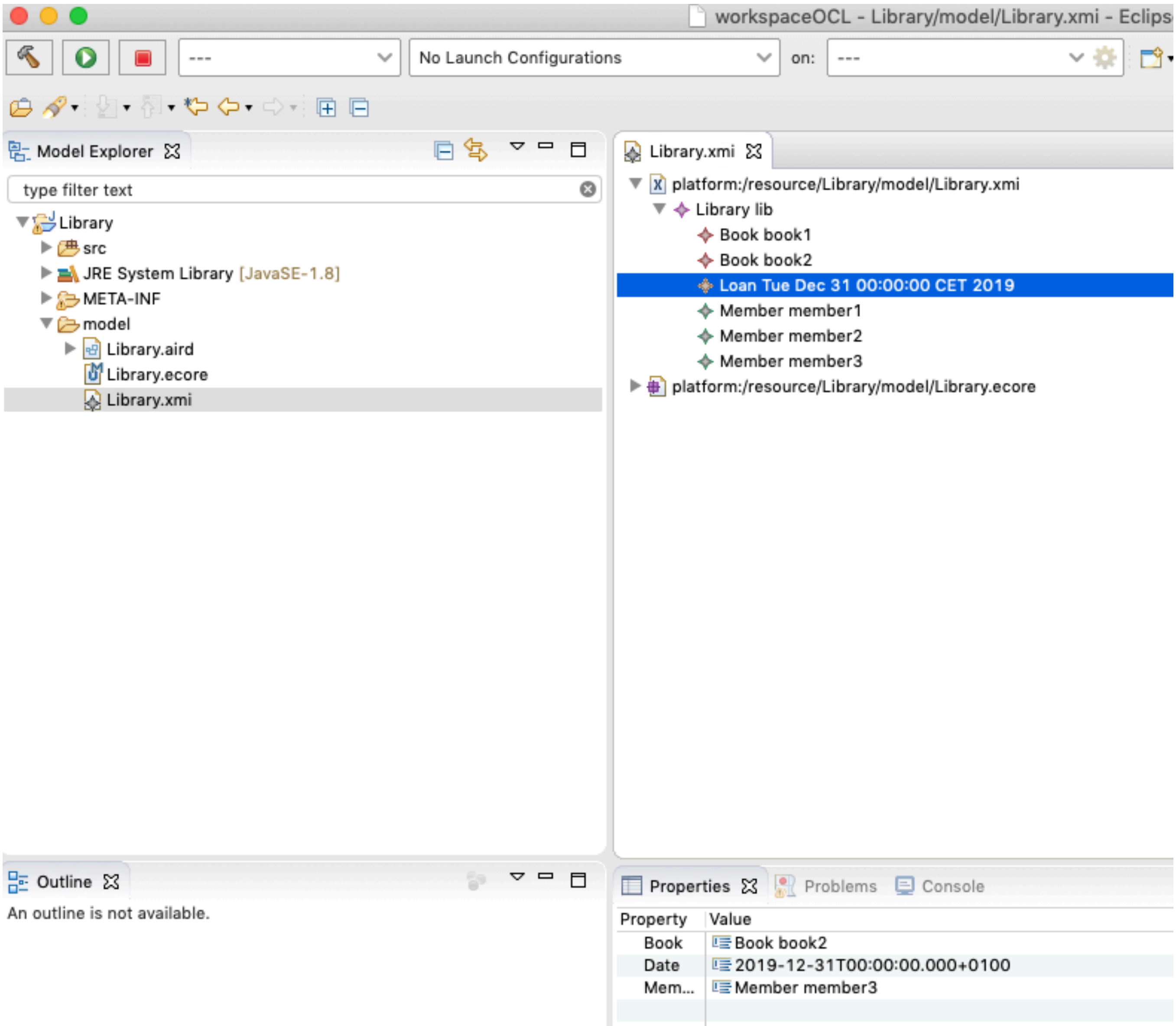
Now that we have the instance of the metamodel, we need to populate the model elements.

From the right-button menu for the instance **Library** select **New Child -> Books Book** twice, use **New Child -> Loans Loan** once and **New Child -> Members Member** three times to populate the model with two books, one loan and three members.

Left-click to select each of the **Books** and **Members** in turn and enter a name such as **book1, book2** and **member1**, **member2**, **member3** using the Properties View. Specify that **book1** has 1 copy and **book2** has 2 copies.

The books and members now have distinct titles in the outline. When you left-click to select the **Loan** and edit its **Book** and **Member** attributes, the associated pull-down has meaningful entries. Specify that the **Loan** is for **book2** by **member3**.

The outcome of above tasks would be like below:



workspaceOCL - Library/model/Library.xmi - Eclips

- Library
  - src
  - JRE System Library [JavaSE-1.8]
  - META-INF
  - model
    - Library.aird
    - Library.ecore
    - Library.xmi

An outline is not available.

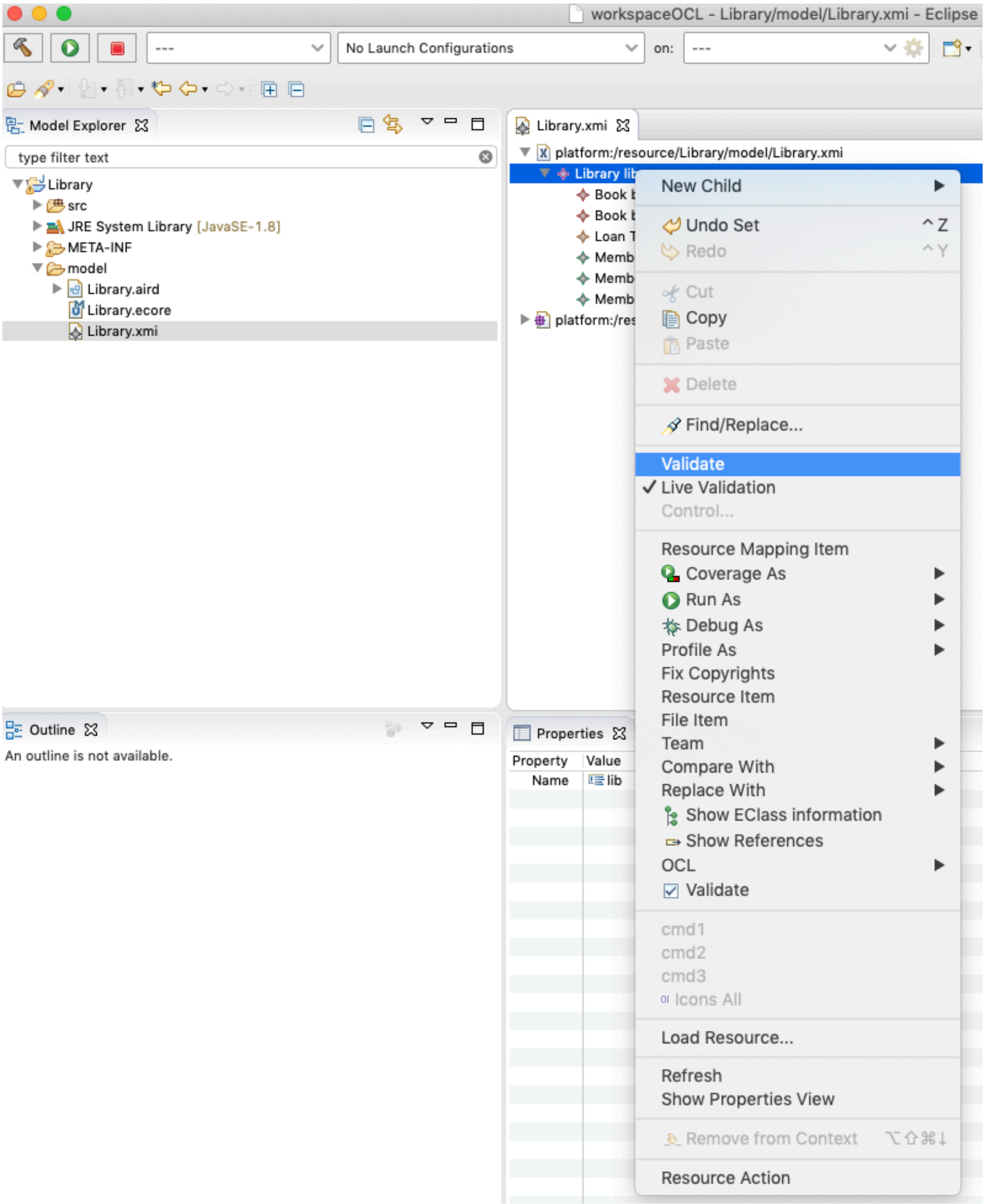
Library.xmi

- platform:/resource/Library/model/Library.xmi
  - Library lib
    - Book book1
    - Book book2
    - Loan Tue Dec 31 00:00:00 CET 2019
    - Member member1
    - Member member2
    - Member member3
- platform:/resource/Library/model/Library.ecore

| Property | Value |
|---|---|
| Book | Book book2 |
| Date | 2019-12-31T00:00:00.000+0100 |
| Mem... | Member member3 |

The configuration has three members, two books and one loan.

We can validate this by right-clicking on the **Library** instance node, and left-clicking to **Validate** Library and all its children, as done below.



As the model instance is simple, no errors found after Validation.

We now create two further identical loans of book2 by member3 by copy-pasting the current loan under the **Library** model instance. We should have now three loans in total.

The Validation as above will still be successful, although it is **clearly wrong** for the two copies of book2 to be under three loans.

### 2.3 Enrich the Ecore model with OCL using the OCLinEcore text editor

Thus, the semantic constraint that a book cannot be borrowed more times than the copies it has is an example of a constraint that cannot be expressed by simple multiplicities – a more powerful capability is required. The Object Constraint Language provides this capability.

The constraint can be realized as an invariant on a book that specifies that the ***size(loans involving a book)*** is less than or equal to the number of copies of that book.

Note: Close the **Tutorial.xmi** editor before modifying its meta-model. Else you might end up in errors or breaking the model instance.

We need to modify the Book class in the original meta-model so that the condition holds in the instance models. So, the new Book class in the **Library.ecore** will look like below:

```
class Book
{
          invariant SufficientCopies:
                     library.loans->select(book=self)->size() <= self.copies;
          attribute name : String[?];
          attribute copies : Integer[?];
          property library#books : Library[?];
}
```

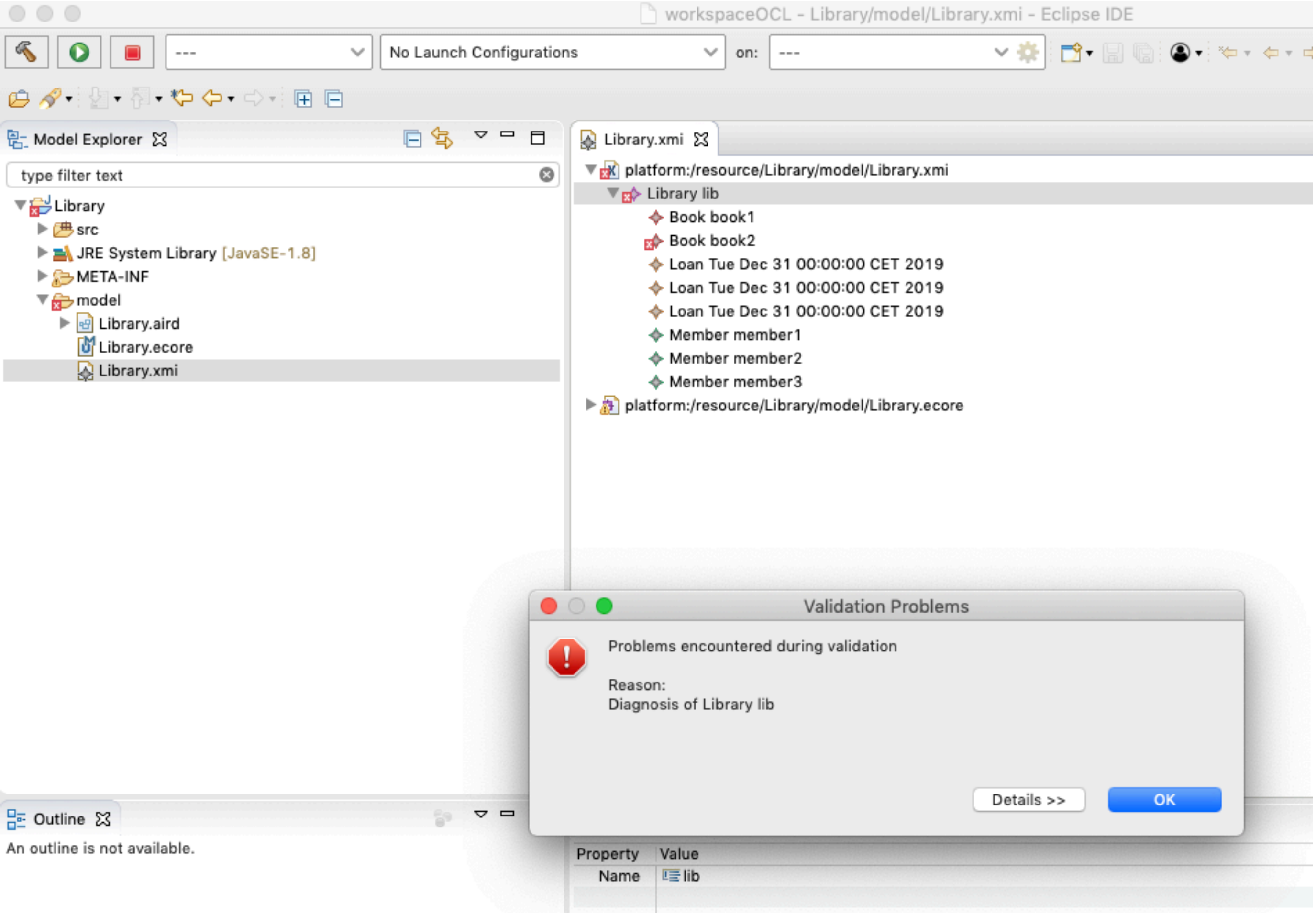
Where:

- an invariant is defined with name SufficientCopies
- within the invariant on a Book, self is the instance of Book being validated.
- library.loans, which is short for self.library.loans, navigates to the library and then to all loans in the library.
- ->select(...) is a collection iteration over the loans. It selects each loan for which its argument expression is true
- book=self, which is short for aLoan : Loan | aLoan.book = self, uses the aLoan iterator over each loan to select those for which the book is the book being validated
- ->size() is a collection operation that just counts the number of selected loans
- <= copies, which is short for <= self.copies converts the count to true if it is consistent, or false if inconsistent.

### 2.4 Validate the model and observe the OCL enrichments

A model instance to be valid, the **SufficientCopies** invariant must always be **true**.

Now, if we validate the **Library.xmi** model instance as before, it will **fail**. Open the model instance **Library.xmi** as **Open With -> Sample Reflective Ecore Model Editor**



```
▼ Library
  ▶ src
  ▶ JRE System Library [JavaSE-1.8]
  ▶ META-INF
  ▼ model
    ▶ Library.aird
      Library.ecore
      Library.xmi
```

```
▼ platform:/resource/Library/model/Library.xmi
  ▼ Library lib
      Book book1
      Book book2
      Loan Tue Dec 31 00:00:00 CET 2019
      Loan Tue Dec 31 00:00:00 CET 2019
      Loan Tue Dec 31 00:00:00 CET 2019
      Member member1
      Member member2
      Member member3
▶ platform:/resource/Library/model/Library.ecore
```

Validation Problems

Problems encountered during validation

Reason:
Diagnosis of Library lib

An outline is not available.

| Property | Value |
|---|---|
| Name | lib |

The **Details** identifies that the **SufficientCopies** invariant is not satisfied for the book2.

If we change the first loan so that book1 is borrowed (instead of book2) and then validate again, the problem is resolved. It is all right for member3 to borrow the one copy of book1 and the two copies of book2.

### 2.5 Use the Interactive OCL Console to execute the OCL enrichments

The OCL Console supports interactive execution of an OCL expression ***in the context of a model instance***.

First, make the OCL Console visible. Go to **Window -> Show View -> Console**. Then right click on the **Open Console** and left click on **Interactive Xtext OCL**.

The **Interactive Xtext OCL** console comprises two main text panes. The upper pane displays results. The lower pane supports entry of queries.

Note: To put the correct context the target instance model must be selected.

To test the OCL Console, select the Library instance model, and then type books or members or loans followed by an Enter. This will show the list of books, list of members, or list of loans in the model instance.

The inputs and outputs are like below:

```
Evaluating:
books
Results:
Library lib::Book book2
Library lib::Book book1

Evaluating:
members
Results:
Library lib::Member member2
Library lib::Member member1
Library lib::Member member3

Evaluating:
loans
Results:
Library lib::Loan Tue Dec 31 00:00:00 CET 2019
Library lib::Loan Tue Dec 31 00:00:00 CET 2019
Library lib::Loan Tue Dec 31 00:00:00 CET 2019
```

We can see the number of loans for each Book by selecting them one by one and executing the `library.loans->select(book=self)->size() <= scopies` OCL constraint in the OCL Console, which will be evaluated to true.

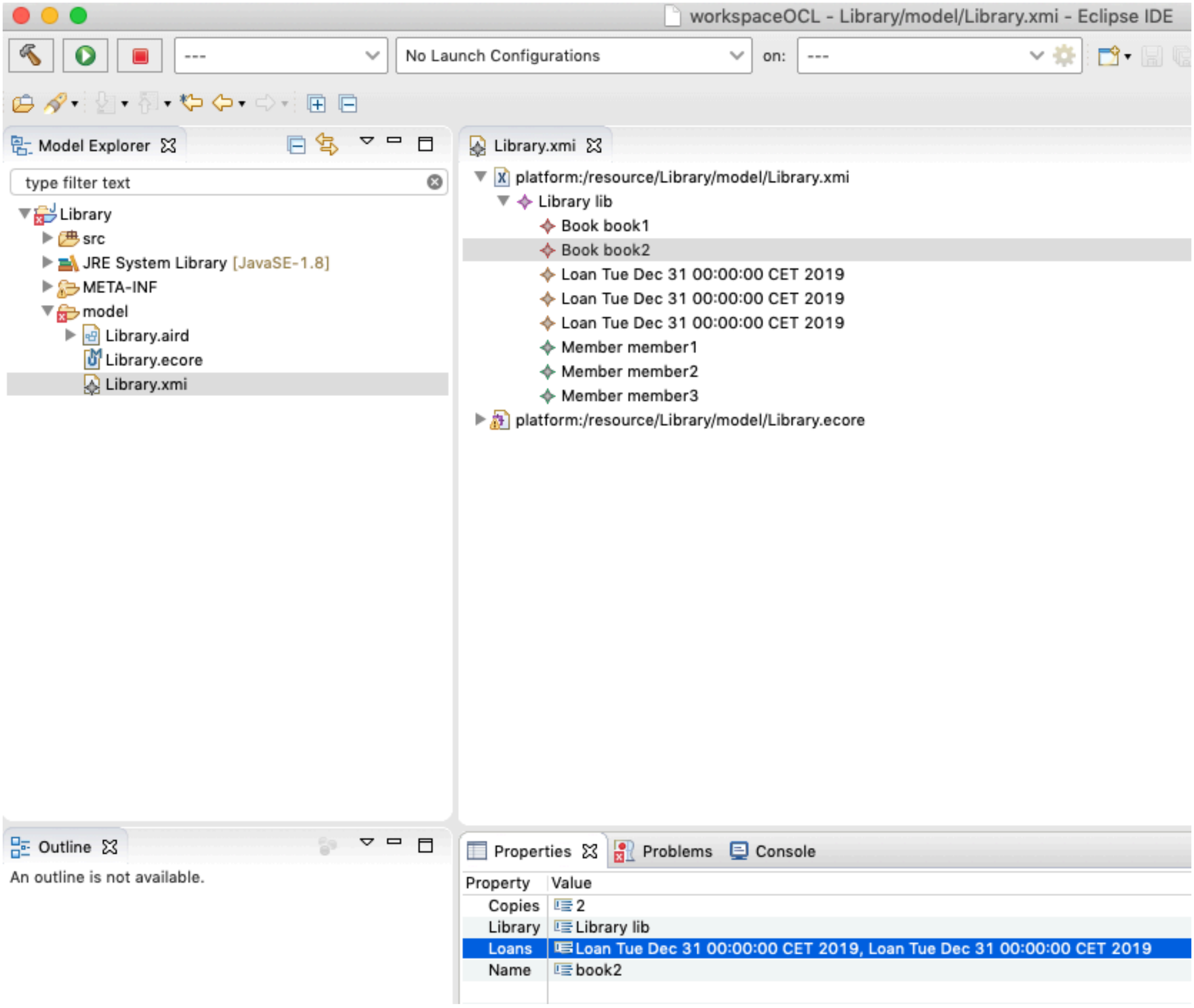
## Helper Features and Operations

We can use helper attributes and operations to make the OCL clearer and provide a richer meta-model. Replace the **Book** class in the **Library.ecore** meta-model by the following:

```
class Book
{
        attribute name : String[?];
        attribute copies : Integer[?];
        property library#books : Library[?];
        property loans : Loan[*] { derived,volatile }
        {
                derivation: library.loans->select(book=self);
        }
        operation isAvailable() : Boolean[?]
        {
                body: loans->size() < copies;
        }
        invariant SufficientCopies:
                library.loans->select(book=self)->size() <=
self.copies;
}
```

Note: The derived property must be volatile to avoid problems when a model is loaded but has no content.

If you open the **Library.xmi** model instance, it will be reloaded with the derived property.



```
type filter text
▼ Library
   ▶ src
   ▶ JRE System Library [JavaSE-1.8]
   ▶ META-INF
   ▼ model
      ▶ Library.aird
        Library.ecore
        Library.xmi
```

An outline is not available.

```
▼ platform:/resource/Library/model/Library.xmi
   ▼ Library lib
        Book book1
        Book book2
        Loan Tue Dec 31 00:00:00 CET 2019
        Loan Tue Dec 31 00:00:00 CET 2019
        Loan Tue Dec 31 00:00:00 CET 2019
        Member member1
        Member member2
        Member member3
▶ platform:/resource/Library/model/Library.ecore
```

| Property | Value |
|---|---|
| Copies | 2 |
| Library | Library lib |
| Loans | Loan Tue Dec 31 00:00:00 CET 2019, Loan Tue Dec 31 00:00:00 CET 2019 |
| Name | book2 |

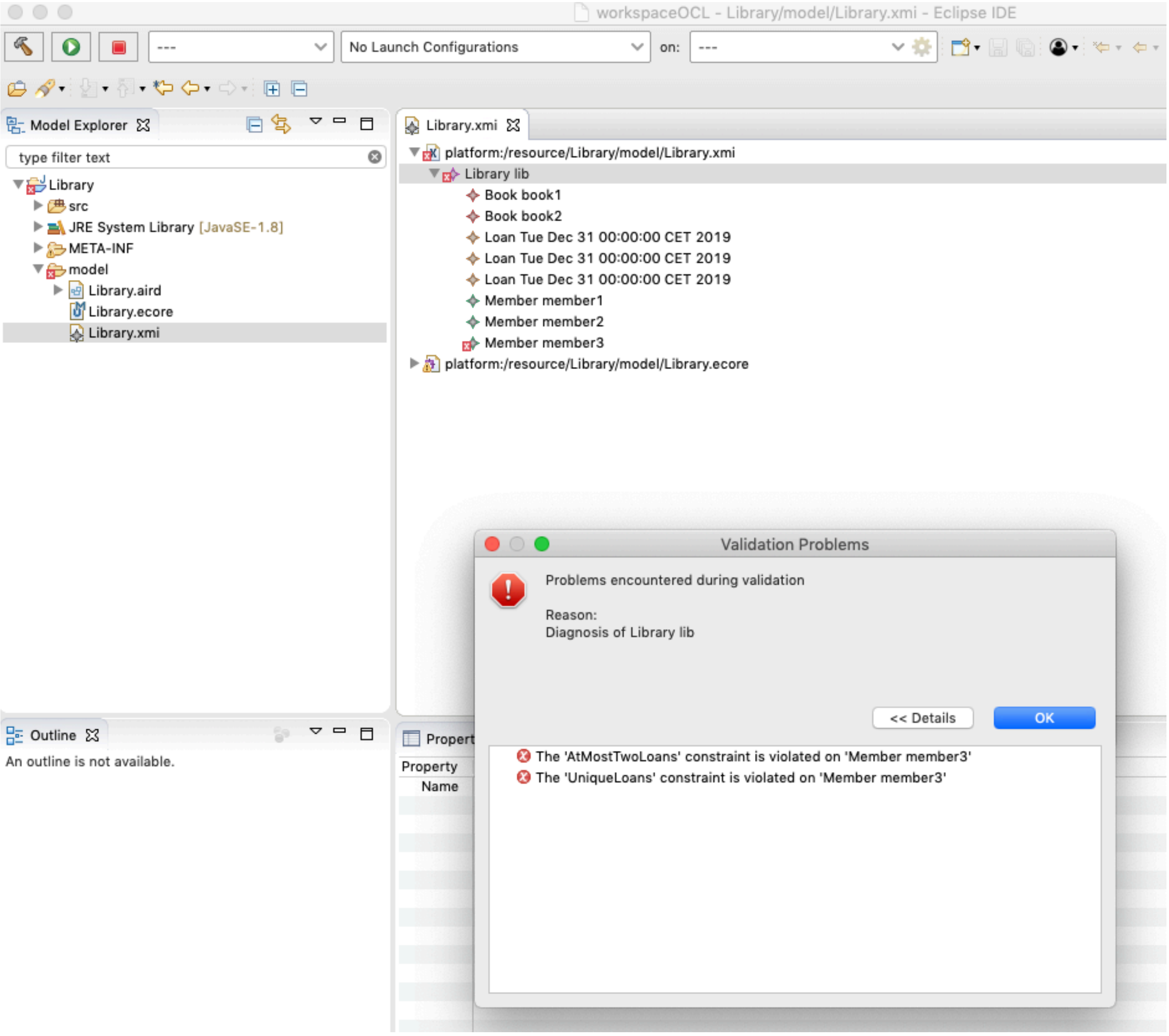
Then, the helper operation can be evaluated in the **OCL Console** view, after closing and reopening, by selecting Book book2 and typing **isAvailable()** for execution. This will return **false** since two copies of book2 are already in load, so book2 is not available now.

We will now add further helpers and constraints to enforce that “at most two loans per member policy” and “to require loans to be unique”.

To do this, we need to replace the Member class in **Library.ecore** meta-model by:

```
class Member
{
        attribute name : String[?];
        property library#members : Library[?];
        property loans : Loan[*] { derived volatile }
        {
                initial: library.loans->select(member=self);
        }
        property books : Book[*] { !unique derived volatile }
        {
                initial: loans->collect(book);
        }
        invariant AtMostTwoLoans:
                loans->size() <= 2;
        invariant UniqueLoans:
                loans->isUnique(book);
}
```

After we validate the **Library.xmi** instance model again, we have a couple of errors since **member3** has **more than 2 loans** and he has **multiple copies** of **book2**:


workspaceOCL - Library/model/Library.xmi - Eclipse IDE
No Launch Configurations
on:
Model Explorer
type filter text
Library
src
JRE System Library [JavaSE-1.8]
META-INF
model
Library.aird
Library.ecore
Library.xmi
Library.xmi
platform:/resource/Library/model/Library.xmi
Library lib
Book book1
Book book2
Loan Tue Dec 31 00:00:00 CET 2019
Loan Tue Dec 31 00:00:00 CET 2019
Loan Tue Dec 31 00:00:00 CET 2019
Member member1
Member member2
Member member3
platform:/resource/Library/model/Library.ecore
Outline
An outline is not available.
Propert
Property
Name
Validation Problems
<< Details
OK

Problems encountered during validation

Reason:
Diagnosis of Library lib

The 'AtMostTwoLoans' constraint is violated on 'Member member3'
The 'UniqueLoans' constraint is violated on 'Member member3'

**Now, how can we correct that?**