

Arbeitsblatt (24.11.2015)

Sequenzdiagramme

Einleitung:

Gegeben ist auf Seite 2 ein noch unvollständiges Sequenzdiagramm für das Erstellen eines Elements im MovieManager, welches mit Hilfe des Frameworks EMF Client Platform (ECP) programmiert wurde. In diesem Sequenzdiagramm fehlen jedoch noch wichtige Teile. Die Code-Ausschnitte von ECP zum Sequenzdiagramm sind auf Seite 3 und 4 gegeben.

Aufgabe 1: Code verstehen

5 Minuten Zeit

Machen Sie sich mit dem gegebenen Code zum Sequenzdiagramm vertraut. Die umgesetzte Funktionalität und der grobe Ablauf sind im folgenden beschrieben.

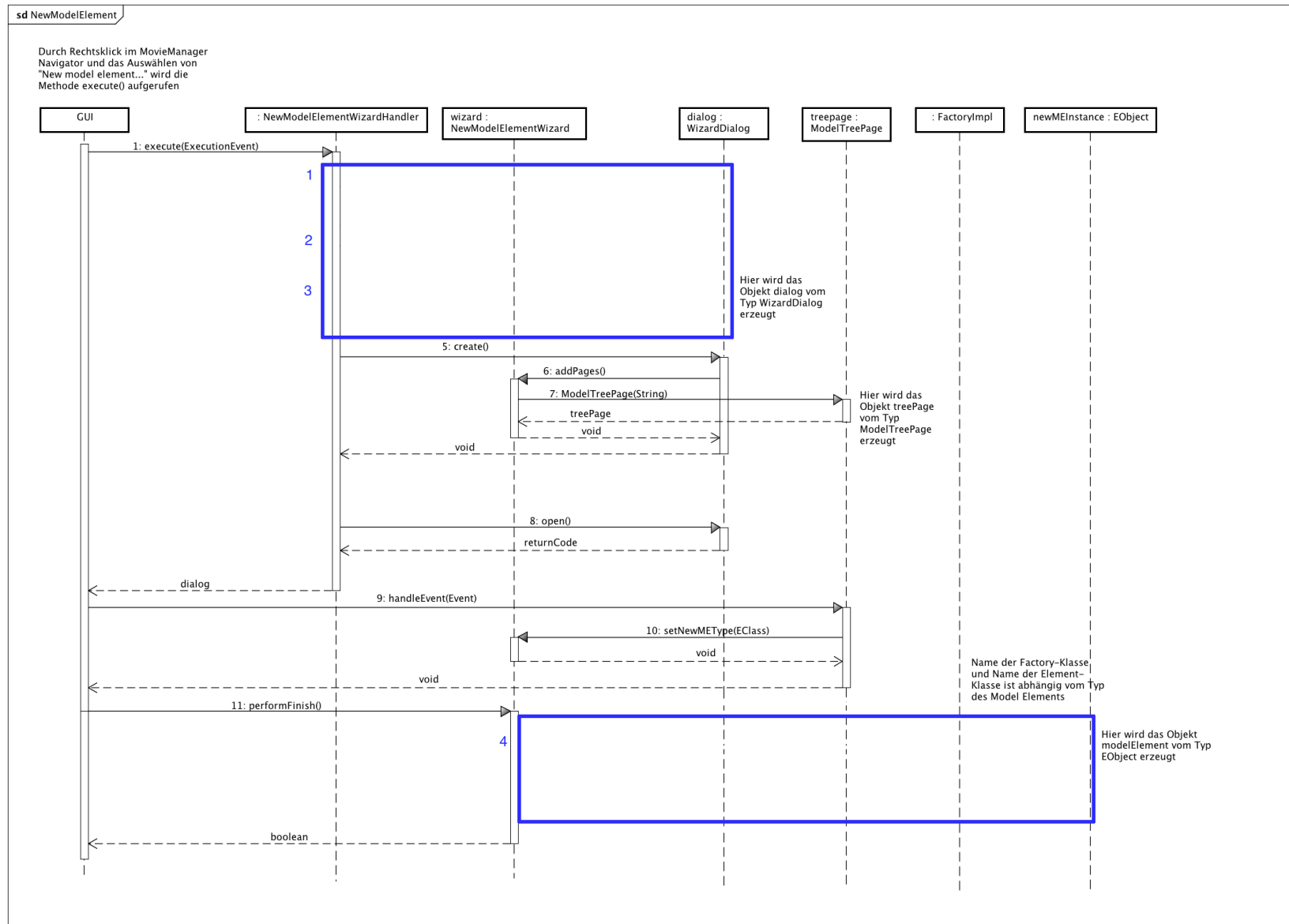
Durch einen Rechtsklick im MovieManager Navigator und das Auswählen von "New model element" wird ein Dialog geöffnet, in welchem man den Typ des neuen Elements auswählen kann. Das Element wird dann mit dem ausgewählten Typ erstellt.

Im Code wird durch das Auswählen von "New model element" die Methode `execute()` der Klasse `NewModelElementWizardHandler` aufgerufen. Von der Methode `execute()` werden Aufrufe an die Klassen `NewModelElementWizard` und `WizardDialog` gerichtet. In der Klasse `NewModelElementWizard` werden weitere Methoden ausgeführt, z.B. `performFinish()`.

Aufgabe 2: Sequenzdiagramm vervollständigen

Bilden Sie 2er Gruppen, 10 Minuten Zeit

1. Vervollständigen Sie mithilfe der gegebenen Code-Ausschnitte das Sequenzdiagramm an den markierten Stellen 1 – 4.
2. Markieren Sie im Code jeweils den Aufruf, der einer der fehlenden Stellen entspricht.



NewModelElementWizardHandler:

```
package org.eclipse.emf.ecp.navigator.handler;

import ...;

/**
 * This is the handler for "Add new model element" context menu command. The command is only shown on
 * LeafSections. The handler initializes and shows the NewModelElementWizard
 */
public class NewModelElementWizardHandler extends AbstractHandler implements IHandler {

    private static final String WIZARD_TITLE = "Add new model element";

    public Object execute(final ExecutionEvent event) throws ExecutionException {
        NewModelElementWizard wizard = new NewModelElementWizard();
        <... code removed ...>
        WizardDialog dialog = new WizardDialog(HandlerUtil.getActiveShell(event), wizard);
        wizard.setWindowTitle(WIZARD_TITLE);
        dialog.create();
        dialog.open();
        return null;
    }
}
```

NewModelElementWizard:

```
package org.eclipse.emf.ecp.navigator.wizards;

import ...;

/**
 * The New Model Element wizard is shown through "Add new model element..." command in the context menu of the
 * Navigator
 * (only on right click on LeafSection). The wizard shows a tree of model packages and their classes.
 * The user can select a Model Element type in this tree and on finish the model element is created and opened for
 * editing.
 */
public class NewModelElementWizard extends Wizard implements IWorkbenchWizard {

    private EObject selectedEObject;
    private EClass newMEType;

    <... code removed ...>

    /**
     * ({@inheritDoc}) This method creates a model element instance from selected type, adds it to Leaf- or
     * CompositeSection, and opens it.
     */
    @Override
    public boolean performFinish() {
        final EObject newMEInstance;
        if (selectedEObject != null && newMEType != null) {
            EPackage ePackage = newMEType.getEPackage(); // geht an Eclipse, ist nicht zu modellieren
            newMEInstance = ePackage.getEFactoryInstance().create(newMEType);
            <... code removed ...>
            ActionHelper.openModelElement(newMEInstance, this.getClass().getName()); // ist nicht zu modellieren
        }
        return true;
    }
}
```