

Studienarbeit

Bewertung von Programmier- und Entwicklungsassistenten
am Beispiel eines Eclipse-Plug-Ins für den Entwurf
verteilter Systeme mit der EventChannelNetwork-
Kommunikationsinfrastruktur

Mohammad Athar Januar

Verantw. Betreuer : Prof. Dr. Walter F. Tichy

Betr. Mitarbeiter : Dipl.-Inform. Marc Schanne



Inhalt

- Motivation
- Grundlagen
- Entwicklung eines Entwicklungsassistenten am Beispiel des ECN-Plug-Ins
- Implementierung
- Ergebnisse & Fazit

Motivation

- Unterstützung bei der Programmierung durch eine IDE

- Eclipse als IDE, Erweiterbarkeit durch Plug-In-Komponente



- Verteilte Systeme mit Nachrichten-kommunikation (EventChannelNetwork)

- Ziel der Studienarbeit :

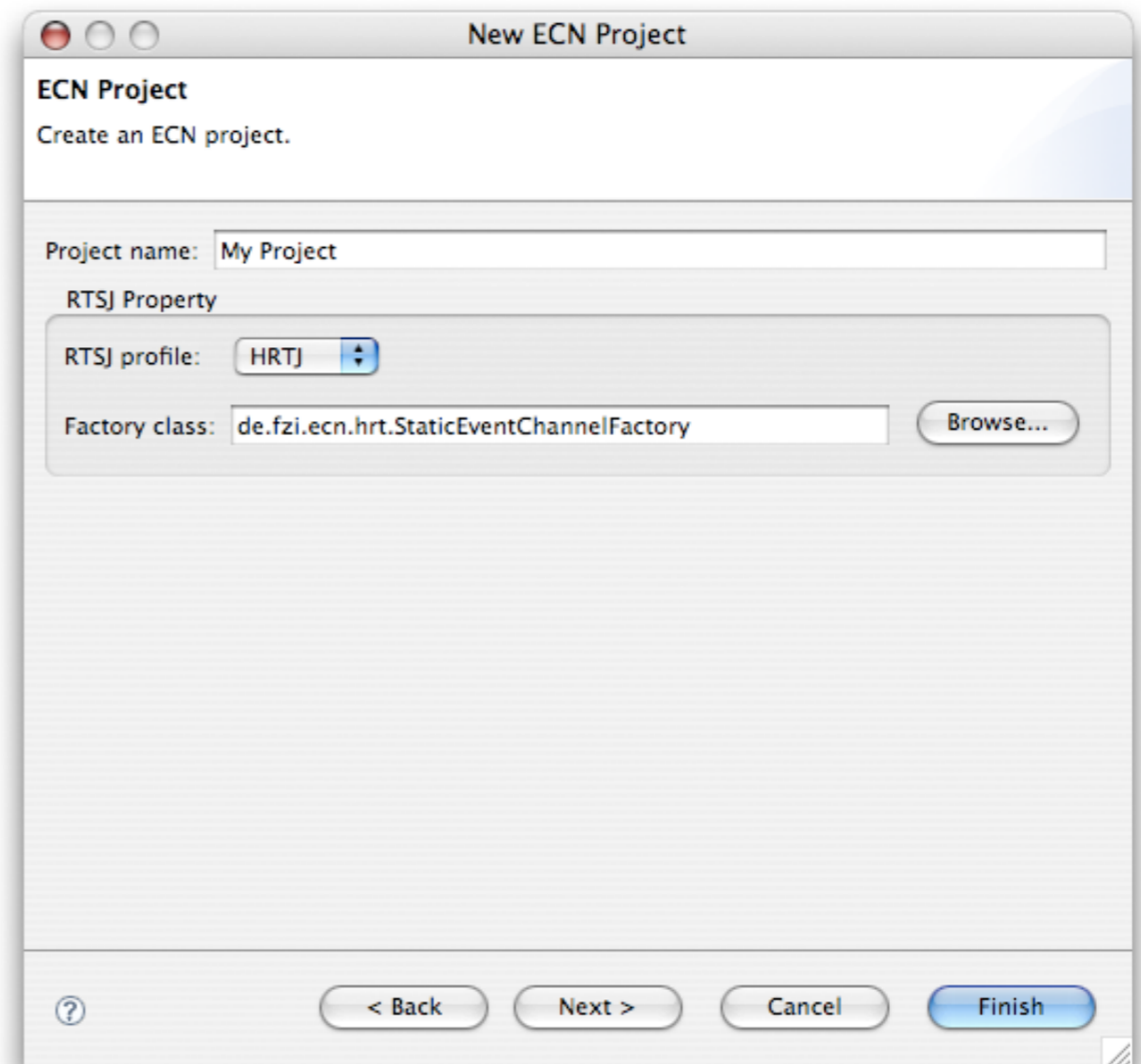


1. Kriterien für die Bewertung von Programmierer- und Entwicklungsassistenten (PEA)
2. Eclipse-Plug-In für EventChannelNetwork (Java, XML)

Grundlagen

Entwicklungsassistent für Softwareentwicklungsprozess

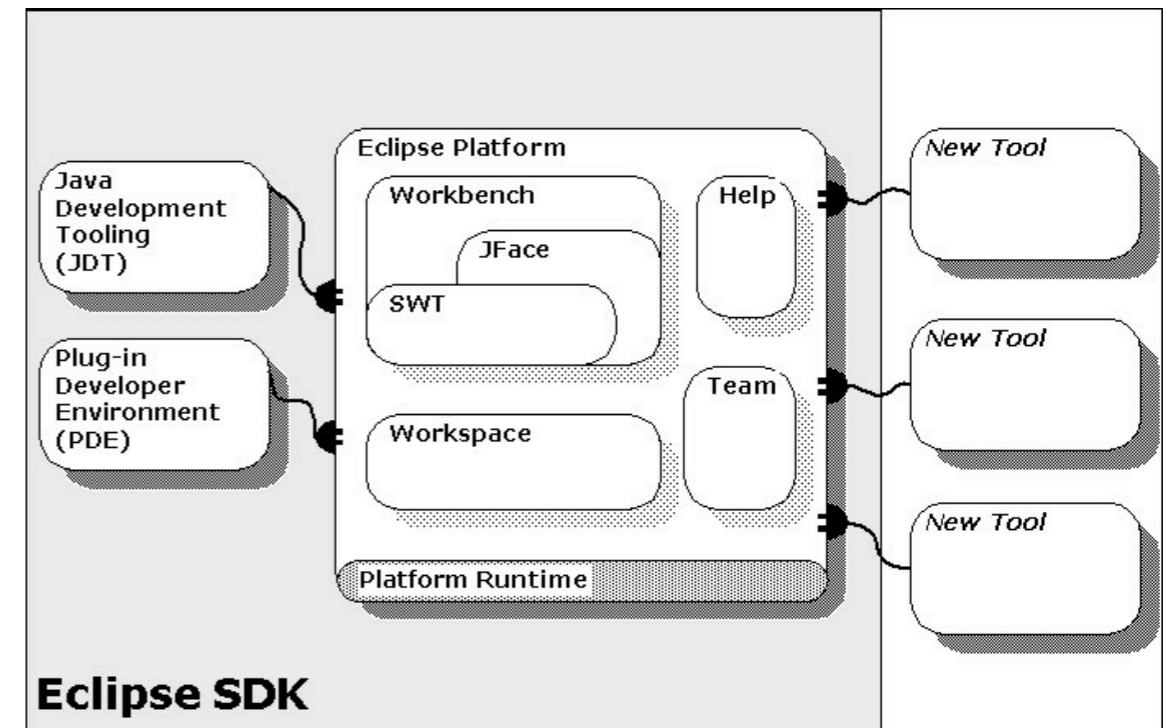
- Eingabe von Informationen für die automatische Erzeugung von Klassen und Beschreibungsdateien



Grundlagen

Eclipse

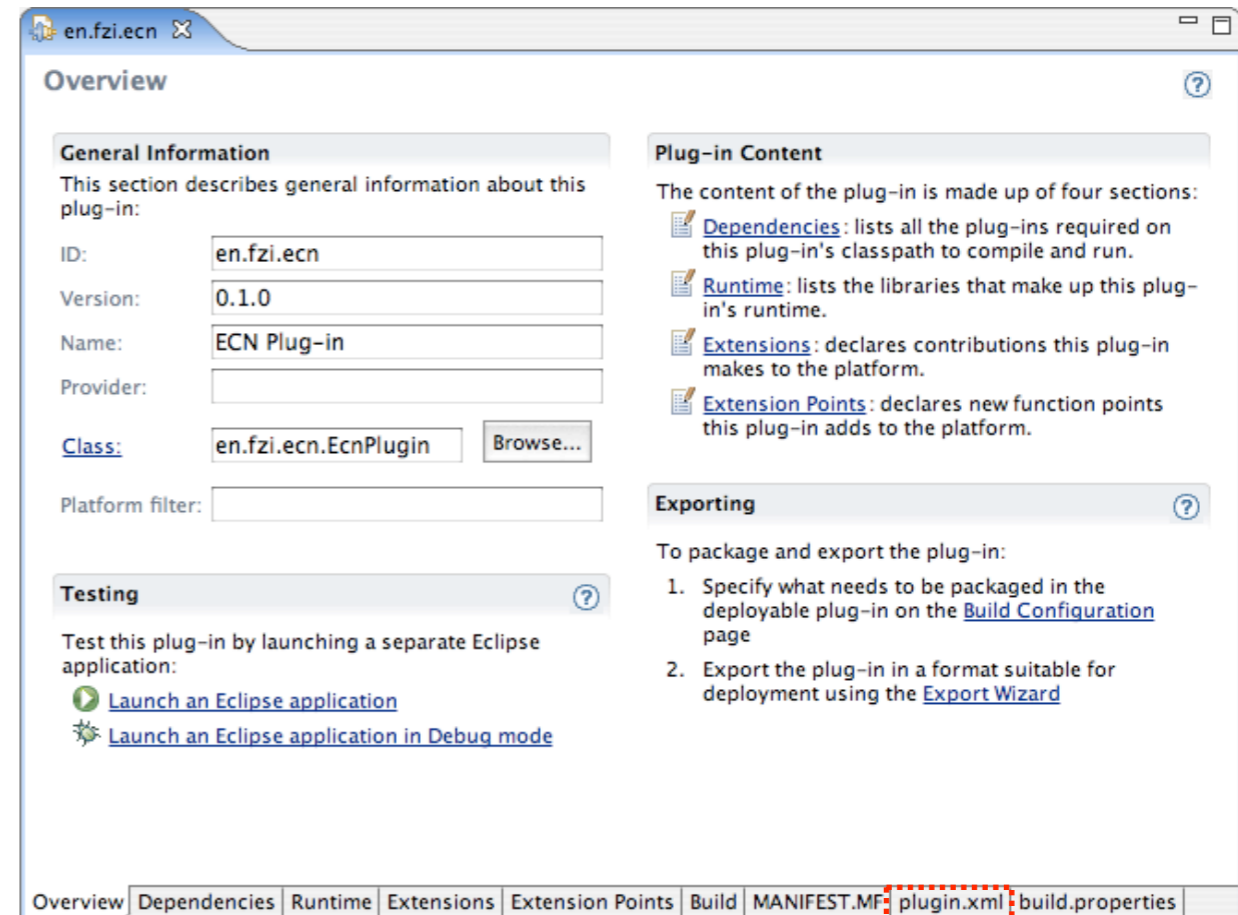
- Eclipse ist für alles und nichts im Besonderen
- Erweiterbarkeit durch Plug-In-Komponenten
- Übliche Eclipse-Plug-Ins : JDT, PDE, CDT, VE, usw.



Grundlagen

Eclipse

- Eclipse-Plug-In ist die kleinste Einheit der Eclipse-Plattform
- PDE ist ein Werkzeug in Eclipse, um Plug-In-Komponente zu entwickeln
- Eine Plug-In-Komponente ist in `plugin.xml` beschrieben.

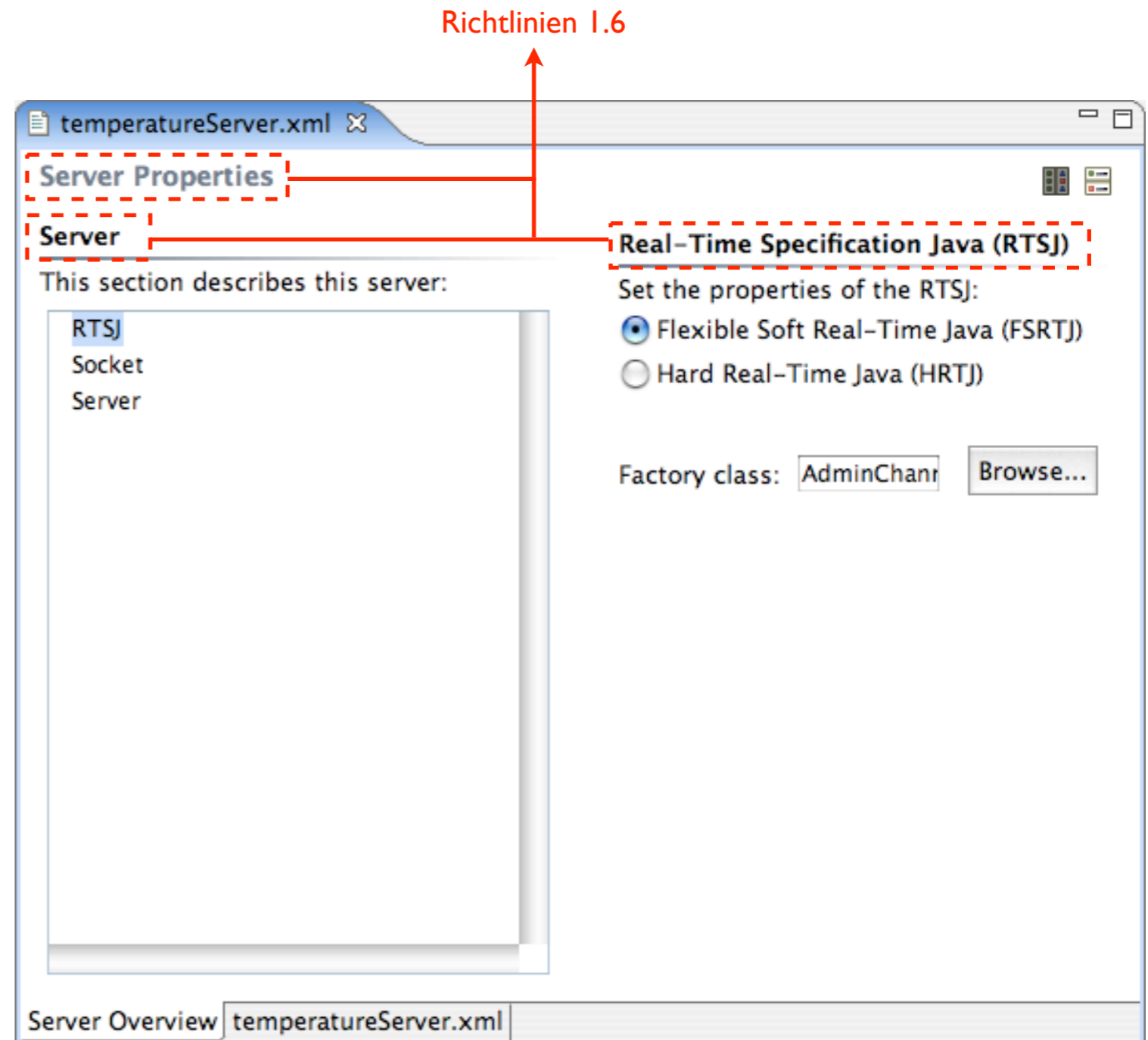


```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.0"?>
<plugin>
  <extension
    id="Test"
    name="NewWizard"
    point="org.eclipse.ui.newWizards">
  </extension>
</plugin>
```

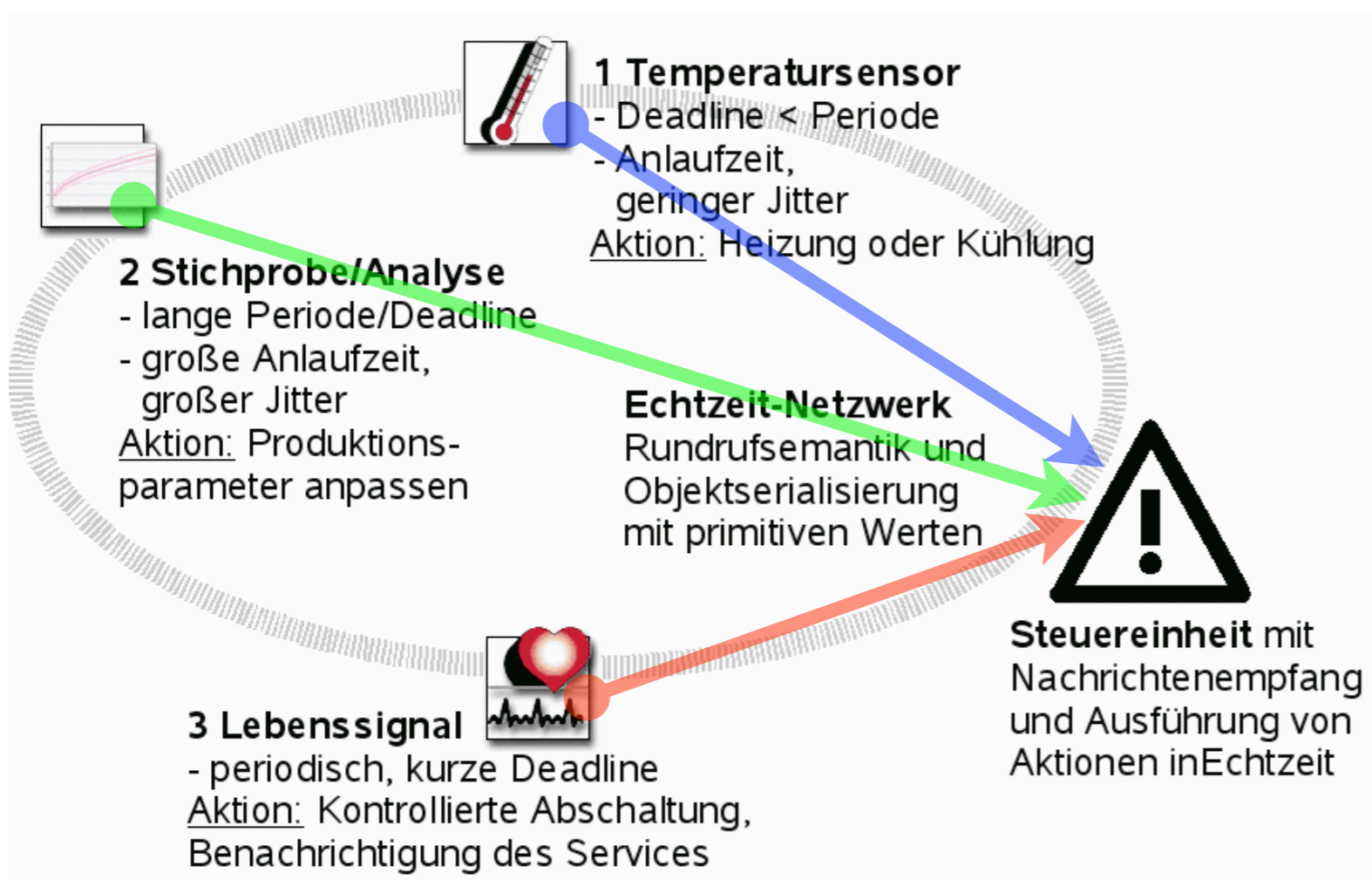
Grundlagen

Designgrundlagen GUI

- Die neue Plug-In-Komponente muss die Konsistenz von Eclipse gewährleisten. Dafür gibt es Eclipse-Benutzerschnittstelle-Richtlinien.
- Eclipse Forms: Glatt aussehender Editor



Entwicklung eines Entwicklungsassistenten am Beispiel des ECN-Plug-Ins



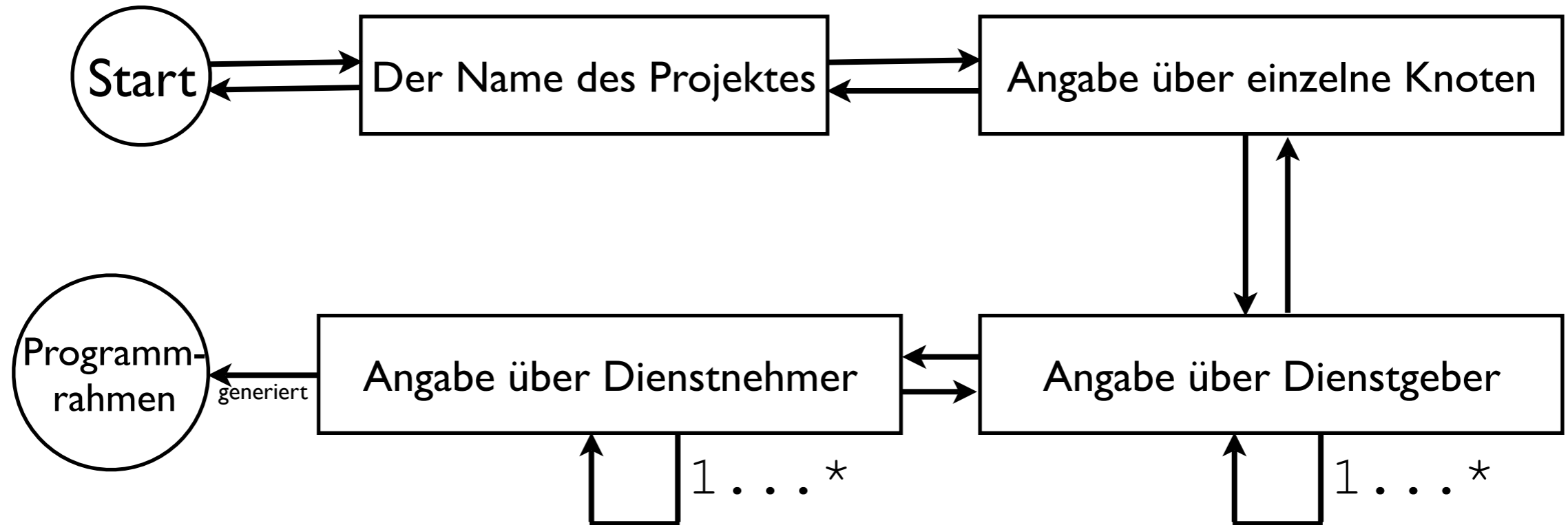
- Nachrichtenkommunikation nach Publiziere/Abonnire-Muster
- Der Dienstgeber schickt Nachrichten innerhalb eines Kanals
- Der Dienstnehmer empfängt sie und handelt durch eine Aktion (*Handler*), in einer Java-Klasse geschrieben

Entwicklung eines Entwicklungsassistenten am Beispiel des ECN-Plug-Ins

- XML-Dateien verführen den Programmierer *Copy-Paste* zu verwenden, um neue Knoten zu erzeugen. Dies führt zu Schwierigkeiten, wenn man falsche Anpassung an Codes durchführt.
- Die Idee ist ein Programmier- und Entwicklungsassistent zu entwerfen, der die Dienstgeber und Dienstnehmer mit gewünschten Werten generiert (Wizard-Bibliothek) und einen einfachen Vorgang zum Ändern der Werte bietet (Forms-Bibliothek).

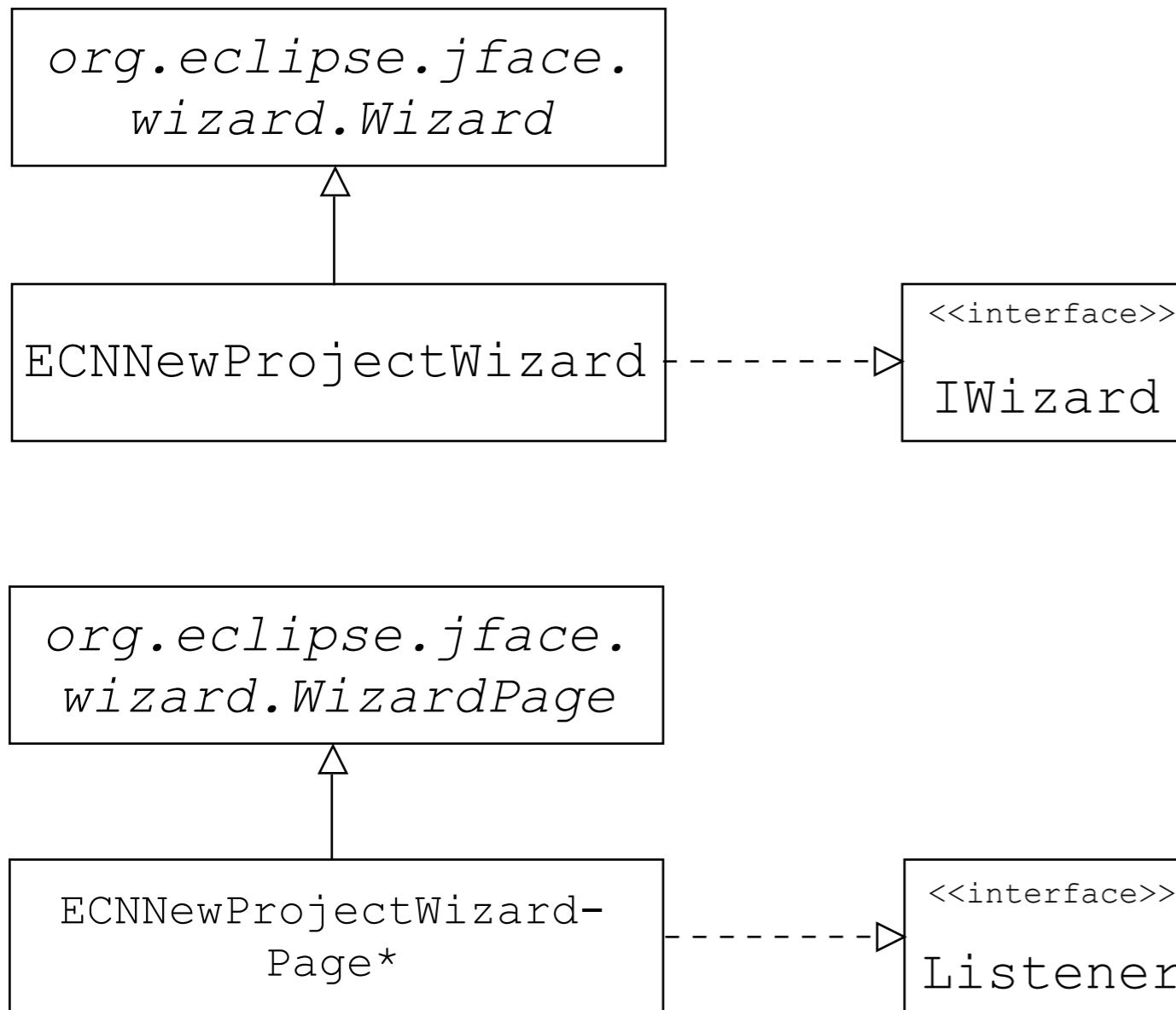
Entwicklung eines Entwicklungsassistenten am Beispiel des ECN-Plug-Ins

- Das ECN-Plug-In:



Implementierung

Der verwendete Erweiterungspunkt : `org.eclipse.ui.newWizards`



```
public class ECNNewProjectWizard extends Wizard implements INewWizard {
    private ECNNewProjectWizardPageOne pageOne;
    private ECNNewProjectWizardPageTwo pageTwo;
    // ...
    /**
     * Adding the page to the wizard.
     */
    public void addPages() {
        pageOne = new ECNNewProjectWizardPageOne();
        addPage(pageOne);
        pageTwo = new ECNNewProjectWizardPageTwo();
        addPage(pageTwo);
        // ...
    }
}
```

```
public class ECNNewProjectWizardPageOne extends WizardPage
    implements Listener{

    // Widgets
    private Label label;
    private Text containerText, factoryClassText;
    private Button browseFactoryClassButton;
    private Combo profileCombo;
    // ...

    public void createControl(Composite parent) {
        Composite composite = new Composite(parent, SWT.NULL);
        GridLayout GridLayout = new GridLayout();
        GridLayout.numColumns = 3;
        composite.setLayout(GridLayout);
        // ...
    }
}
```

Ergebnisse & Fazit

- Gute Einarbeitung in Eclipse
- Arbeitsschritte:
 1. Erweiterungspunkte (*extension points*) finden und erweitern
 2. Java-Klassen schreiben
- Bewertung von PEA:
 1. Wiederverwendbarkeit der (Eclipse-)API-Bibliotheken
 2. Saubere Implementierung durch Code-Trennung nach Model und Benutzeroberfläche (MVC)
 3. Die (Eclipse-)Benutzerschnittstelle-Richtlinien befolgen