



# **Rich Client Platform (RCP)**

## Eine Einführung

Ingo Mohr  
Institut für Informatik  
05. Juni 2008

1. Motivation
2. RCP Konzepte
3. RCP Applikations-Struktur
4. Ein Beispiel
5. Plug-in Deployment mit UpdateSites

1. Motivation
2. RCP Konzepte
3. RCP Applikations-Struktur
4. Ein Beispiel
5. Plug-in Deployment mit UpdateSites

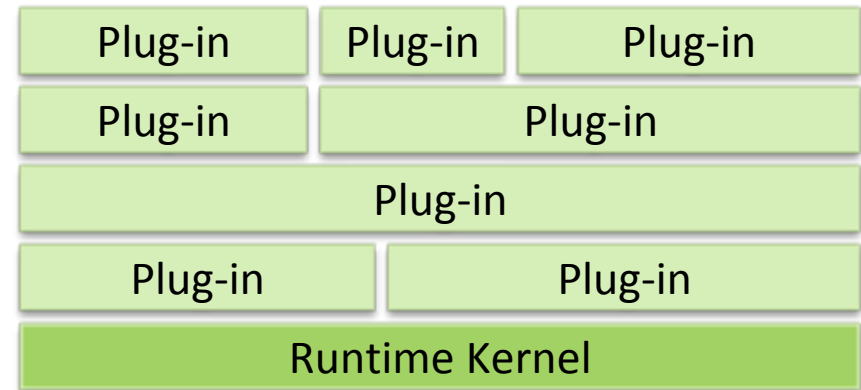
- (Einschub: Eclipse)
  - Urprünglich von IBM als Integrationsplattform für Entwicklungswerkzeuge entworfen
  - Momentan die am meisten genutzte Java IDE
  - Open-Source
  - Wird von der *Eclipse Foundation* (seit 2004) weiterentwickelt
    - Ziel: “... independent, not-for-profit corporation to act as the steward of the Eclipse community.” [2]
    - Mit Beteiligung von IBM, Borland, Intel, BEA, ...
  - Basiert seit Version 3.0 auf RCP (Juni 2004) (“Bruch” mit reiner IDE)
    - 3.1 mit Unterstützung für Java 5 (Juni 2005)
    - 3.2 mit Unterstützung für Java 6 (Juni 2006)
    - 3.3 u.a. SWT-Unterstützung für Windows Vista (Juni 2007)
    - 3.4 mit Unterstützung für 64Bit-Java -Distributionen (geplant: Juni 2008)
  - Seit 3.2 “simultane” Releases (Callisto, Europa, Ganymede)

- Rich Clients
  - Desktop-Applikationen
  - “Rich-UI”
    - z.B. Graphische Editoren “ohne Grenzen”
  - Offline verfügbar
  - Lokale Daten
  - Beinhalten Geschäftslogik

- RCP
  - RCP = Rich Client Platform
  - Mit Eclipse 3.0 erschienen
    - RCP != Eclipse
      - Wikipedia liegt **falsch**
        - “... ist Eclipse selbst nur der Kern, der die einzelnen Plug-ins lädt...”  
(wikipedia.de, [http://de.wikipedia.org/wiki/Eclipse\\_\(IDE\)](http://de.wikipedia.org/wiki/Eclipse_(IDE)), 29.05.2008)
        - Eclipse ist eine RCP-Applikation
        - Aber: RCP ist unabhängig von Eclipse
  - Mächtiges Rahmenwerk zum Entwickeln von Rich Client Applikationen
  - Kompletter Source-Code verfügbar
  - Eclipse Public Licence
  - Große, aktive Community

- RCP

- “Extendable framework”
- Basiert auf Schichten aus Plug-ins



- Vorteile

- Code in lose gekoppelte Einheiten teilbar
- Applikation inkrementell erweiterbar
- Vorhandene Plug-ins wiederverwendbar
- Kunden / Benutzern können eigene Erweiterungen erlaubt werden

1. Motivation
2. RCP Konzepte
3. RCP Applikations-Struktur
4. Ein Beispiel
5. Plug-in Deployment mit UpdateSites



- Plug-in
  - Basis-Komponente
  - Mit Abhängigkeiten zu anderen Plug-ins
  - Kann Extension-Points anderer Plug-ins nutzen (durch Extensions)
  - Kann eigene Extension-Points für andere Plug-ins definieren
- Extensions, Extension-Points
  - Deklarative “Verdrahtung” zwischen Plug-ins und dem RCP Framework
- Feature
  - Gruppierung von Plug-ins
  - (wie auch bei Plug-ins) mit Versions-Nummer
- Fragment
  - “Plug-in ohne Funktionalität”
    - Beispielsweise für Internationalisierung

- Product
  - Applikation mit Branding
- Update-Site
  - Verzeichnis (lokal oder auf Server) zum Installieren oder Verwalten von Features

- Equinox
  - Equinox = OSGi-Implementierung für RCP
  - Basis von Eclipse 3.x
  - OSGi = Open Service Gateway Initiative
    - Service Platform
    - Definiert Service-Bundles
    - “hot plugging”
      - dynamisches Installieren/Deinstallieren/Aktualisieren von Bundles
    - Security, Logging, Remote Configuration API
    - Deklarative Services

- Plug-in
  - Basis-Komponente in einer RCP-Architektur
  - Plug-in in RCP = Service Bundle in OSGi
  - Kann ein oder mehrere Plug-ins erweitern (Extensions)
  - Kann Extension-Points zur eigenen Erweiterung definieren
  - Kann Abhängigkeiten zu anderen Plug-ins definieren
  - Führt Sichtbarkeits-Einschränkungen für Java-Packages ein
  - Jedes Plug-in hat seinen eigenene ClassLoader-Instanz
  - Erlaubt Lazy-Start
  - UI-Plug-ins / Non-UI-Plug-ins
- Definiert durch `META-INF/MANIFEST.MF`, `plugin.xml` und `build.properties`

- MANIFEST.MF – Beispiel (Eclipse 3.4)

```
Manifest-Version: 1.0
Bundle-ManifestVersion: 2
Bundle-Name: IMore Mail Core Plug-in
Bundle-SymbolicName: de.imore.mail.core
Bundle-Version: 1.0.0
Bundle-Activator: de.imore.mail.core.MailCoreActivator
Bundle-Vendor: IMore Soft
Require-Bundle: org.eclipse.core.runtime
Bundle-RequiredExecutionEnvironment: JavaSE-1.6
Bundle-ActivationPolicy: lazy
Export-Package: de.imore.mail.core
```

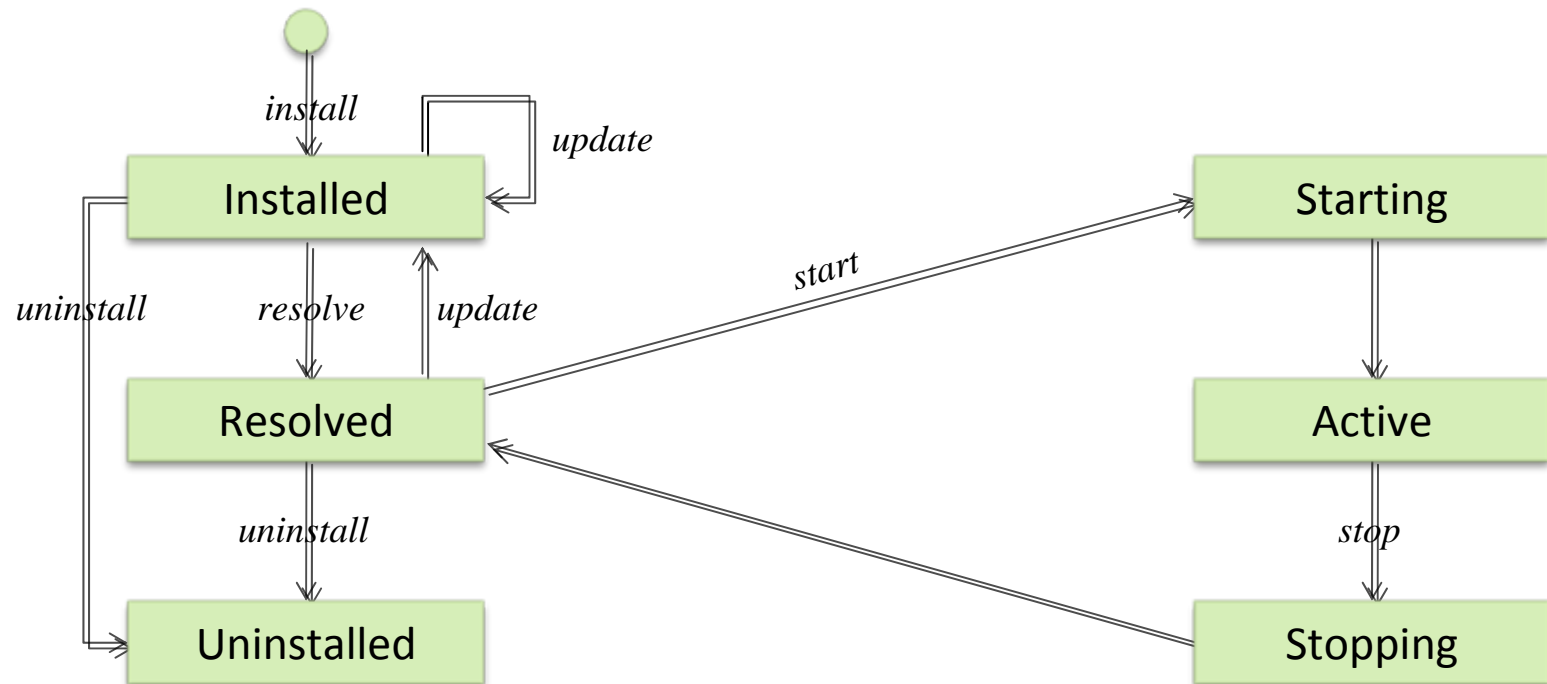
- **plugin.xml – Beispiel** (Eclipse 3.4)
  - Definiert Extensions und Extension-Points (ist optional)

```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.2"?>
<plugin>

  <extension
    id="application"
    point="org.eclipse.core.runtime.applications">
    <application>
      <run
        class="de.imore.mail.ui.Application">
      </run>
    </application>
  </extension>
  <extension
    point="org.eclipse.ui.perspectives">
    <perspective
      name="RCP Perspective"
      class="de.imore.mail.ui.Perspective"
      id="de.imore.mail.ui.perspective">
    </perspective>
  </extension>

  ...
```

- Plug-in Lifecycle



1. Motivation
2. RCP Konzepte
3. RCP Applikations-Struktur
4. Ein Beispiel
5. Plug-in Deployment mit UpdateSites



- Aufbau

Runtime

Equinox

- Equinox: OSGi Implementierung für RCP
- Plug-in Model verwaltet Abhängigkeiten und Plug-in Aktivierung
- Extension-Registry verwaltet Extensions und Extension-Points

- Aufbau



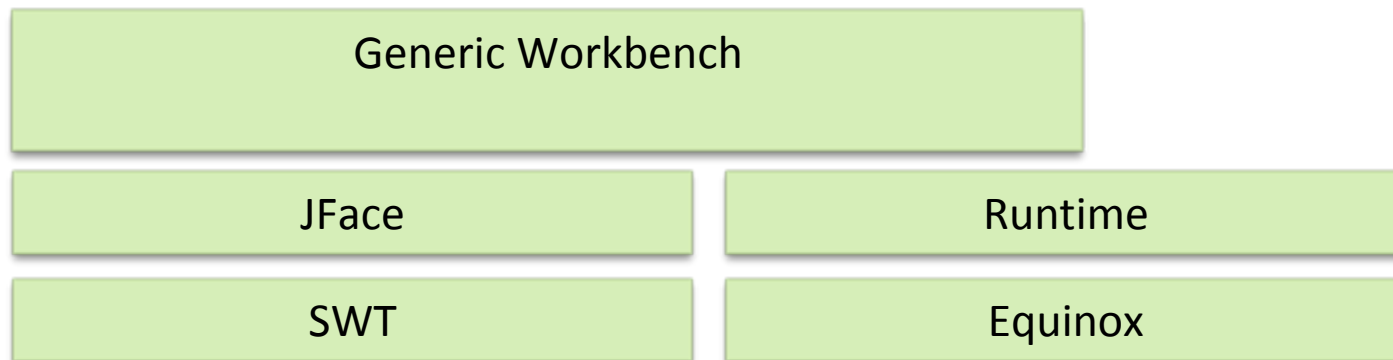
- SWT: Standard Widget Toolkit
  - Platform-abhängige (!) UI-Bibliothek
  - Natives Look & Feel; kann standalone benutzt werden

- Aufbau



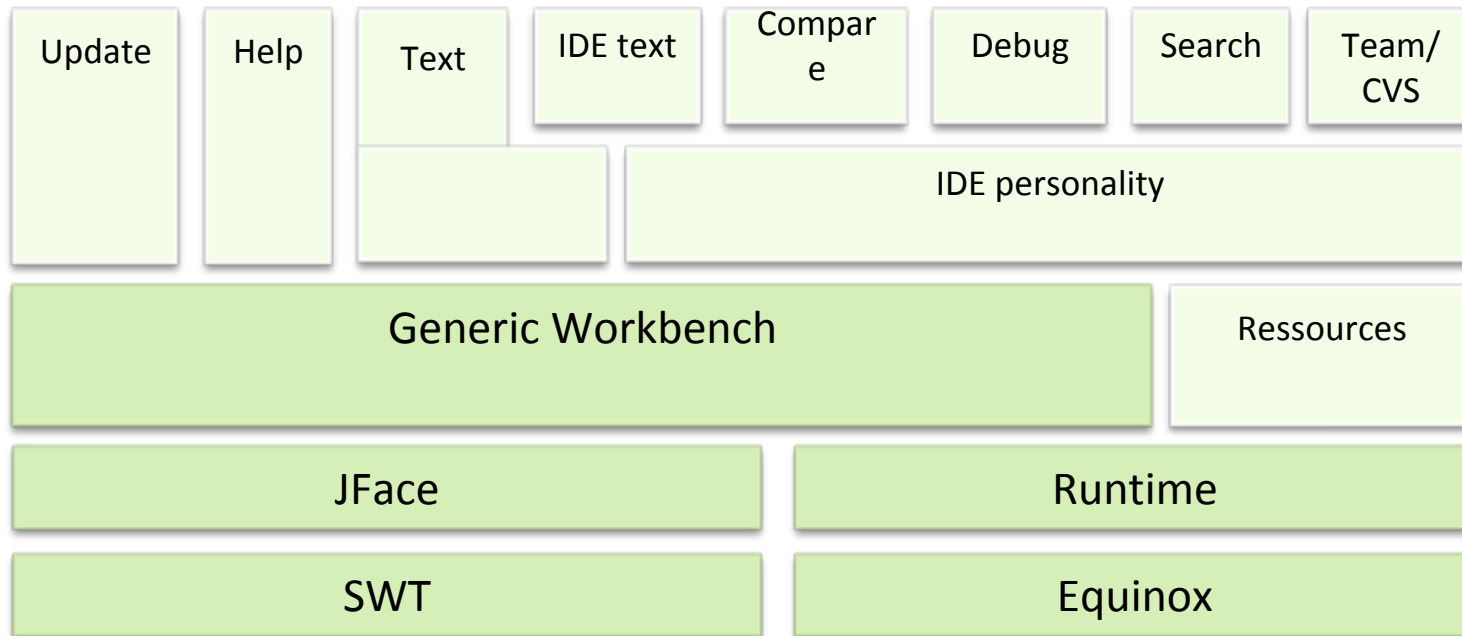
- JFace
  - Baut auf SWT auf
  - Führt Viewer ein (MVC), Dialoge, Wizards, Actions, Resource-Registries, ...

- Aufbau



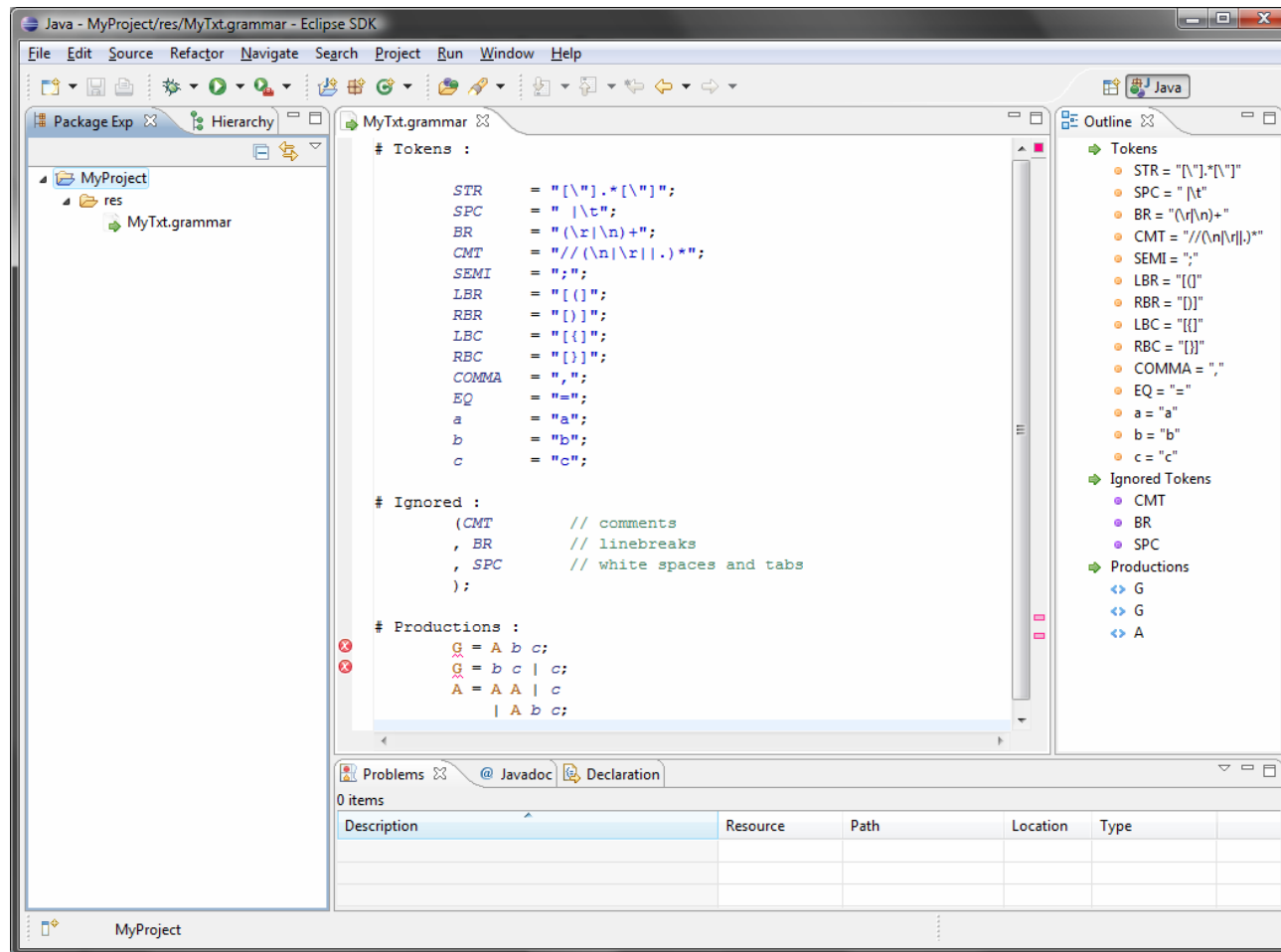
- Generic Workbench
  - Führt Perspectives ein, Views, Editors, action contribution, ...

- Aufbau

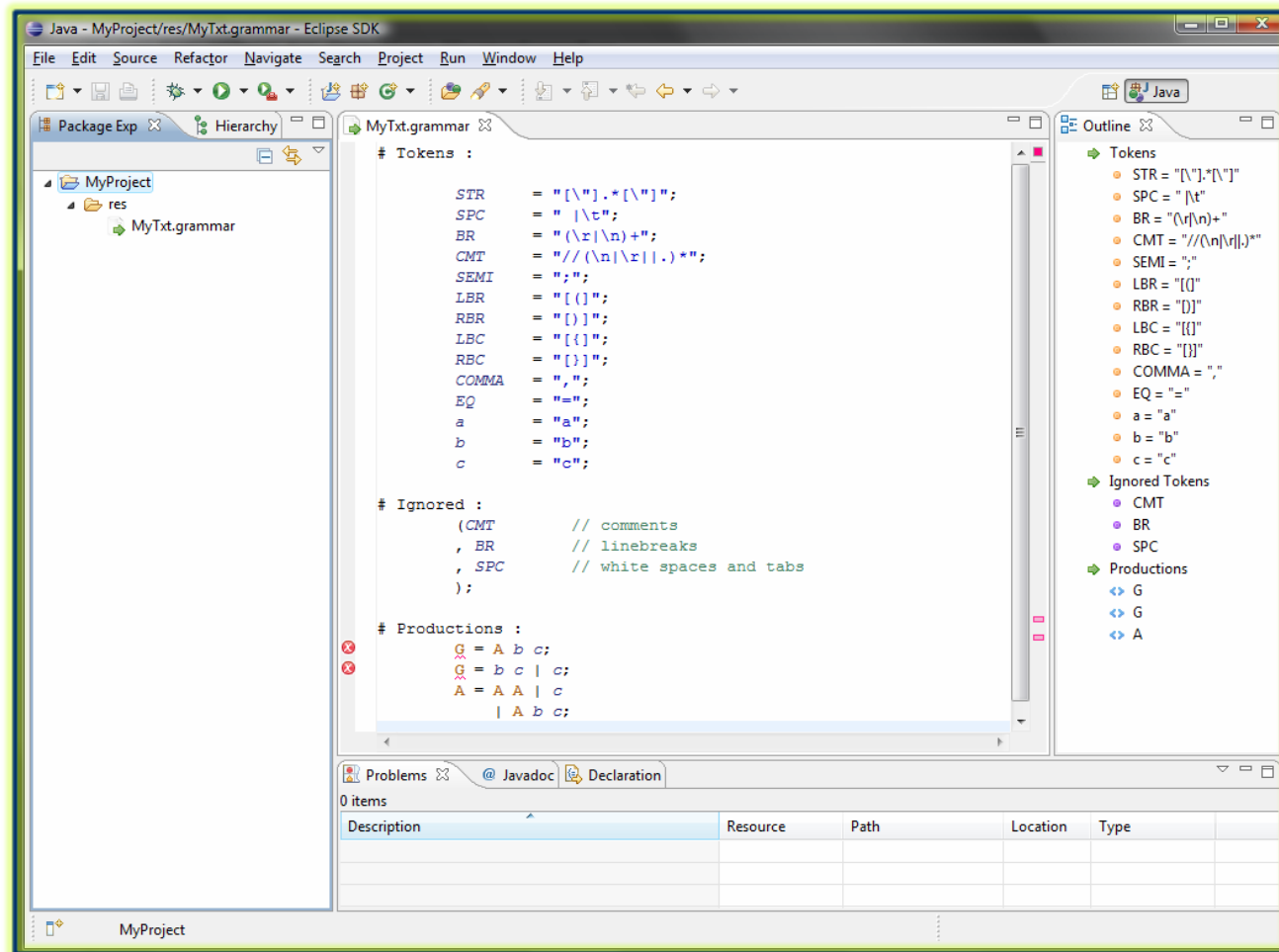


- Exemplarischer Aufbau: Eclipse 3.x

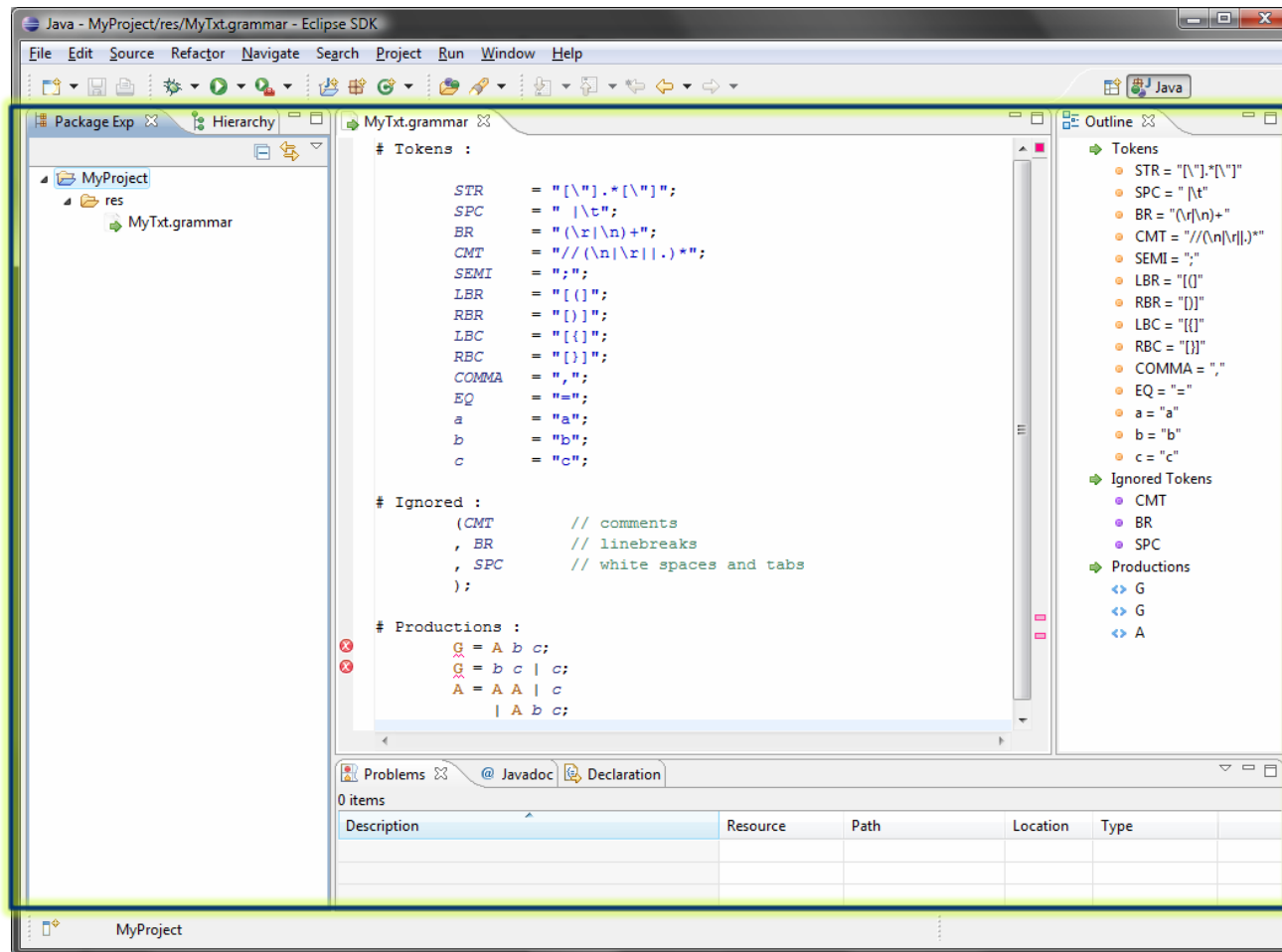
- Workbench-Struktur



- Workbench-Struktur: Workbench Window

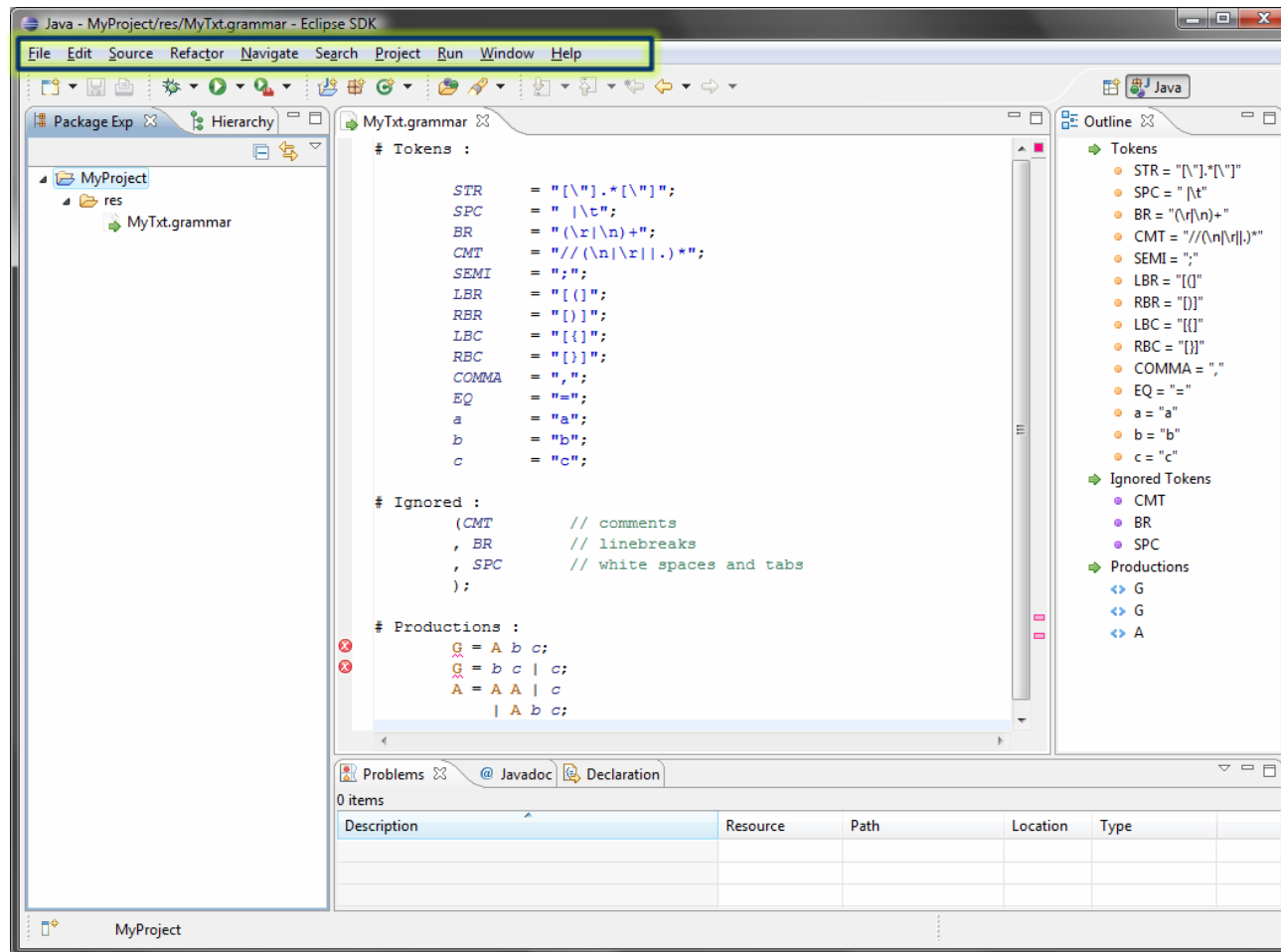


- Workbench-Struktur: Page mit aktiver Perspektive

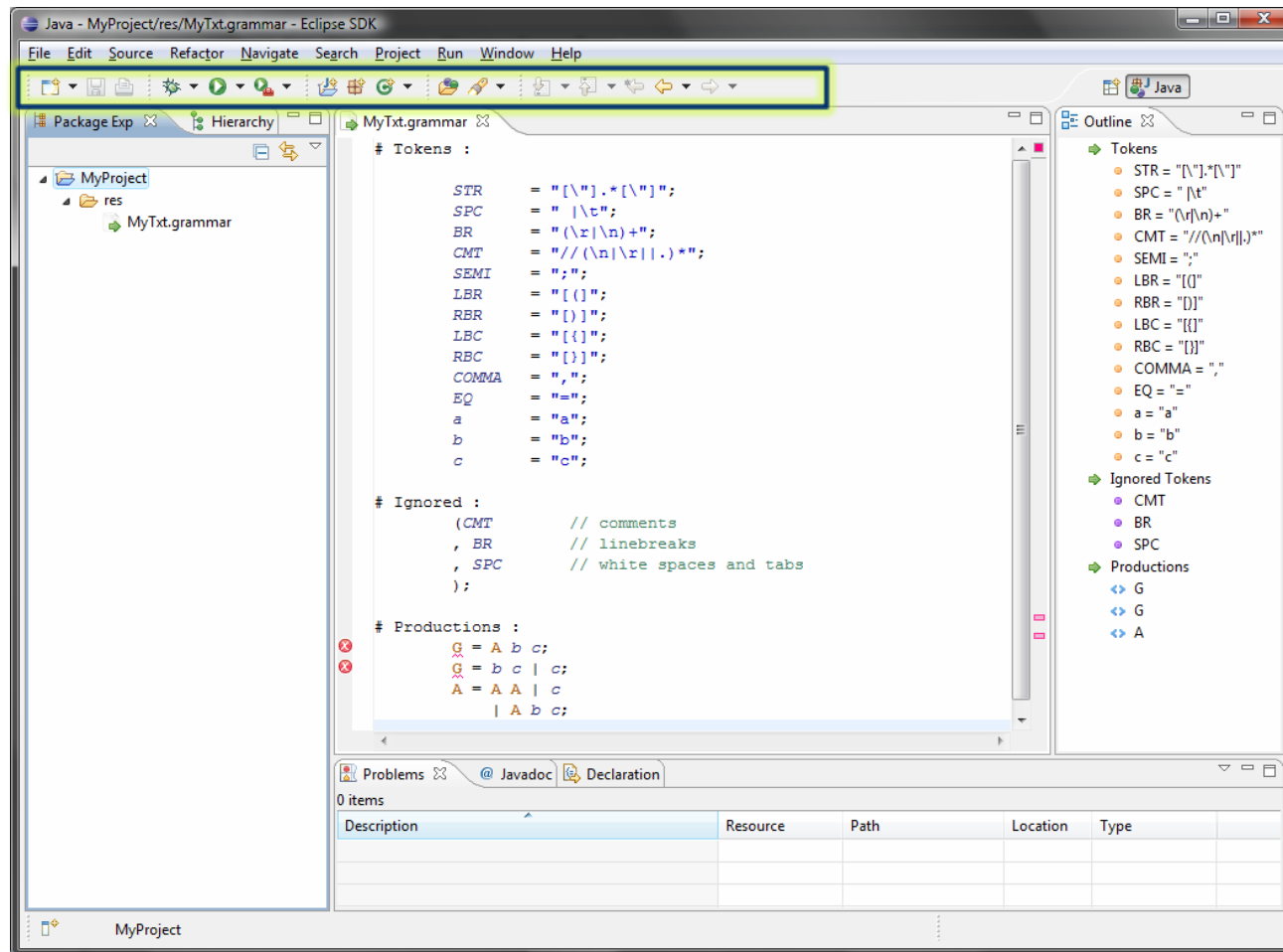




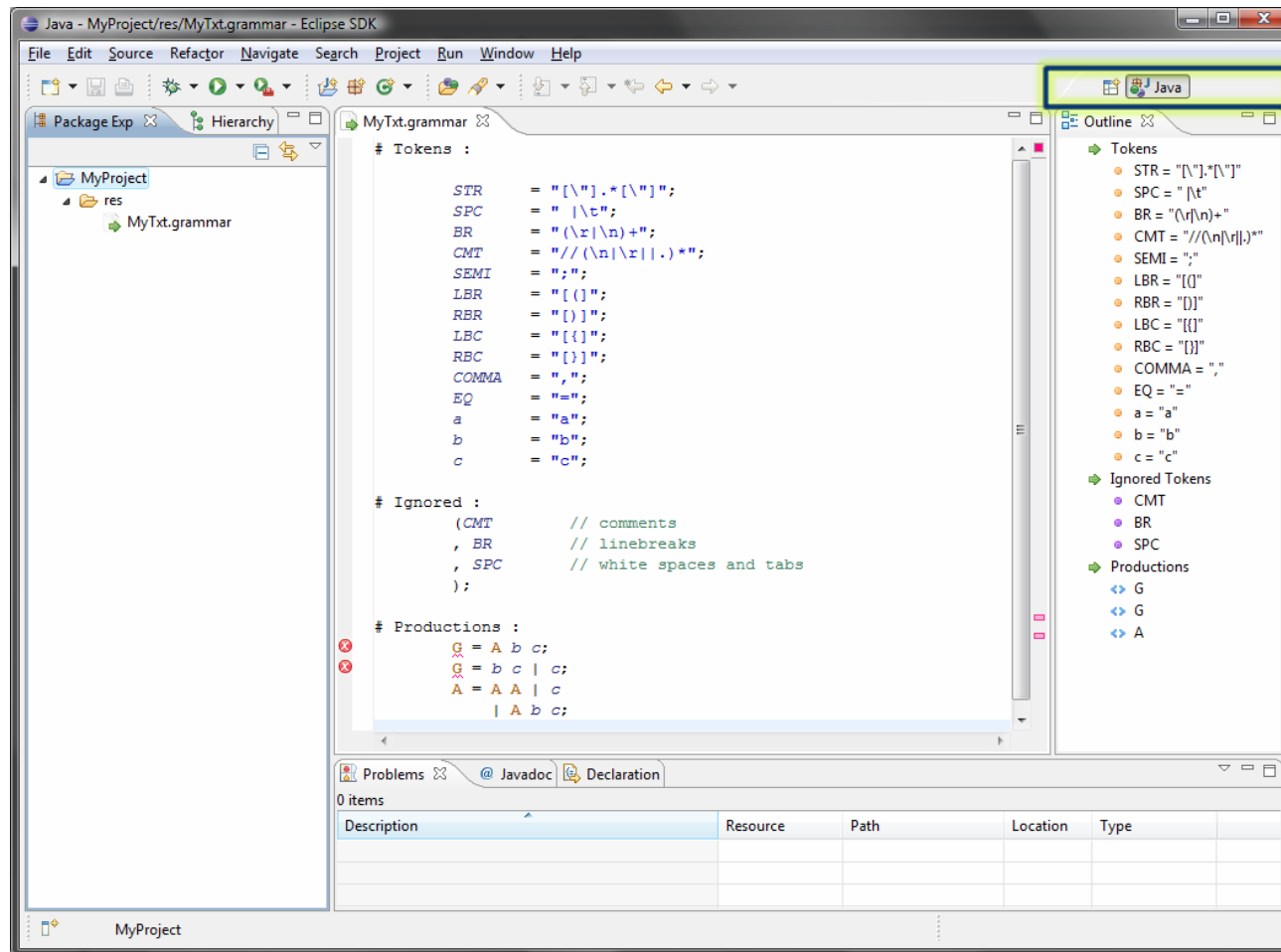
- Workbench-Struktur: Menu Bar



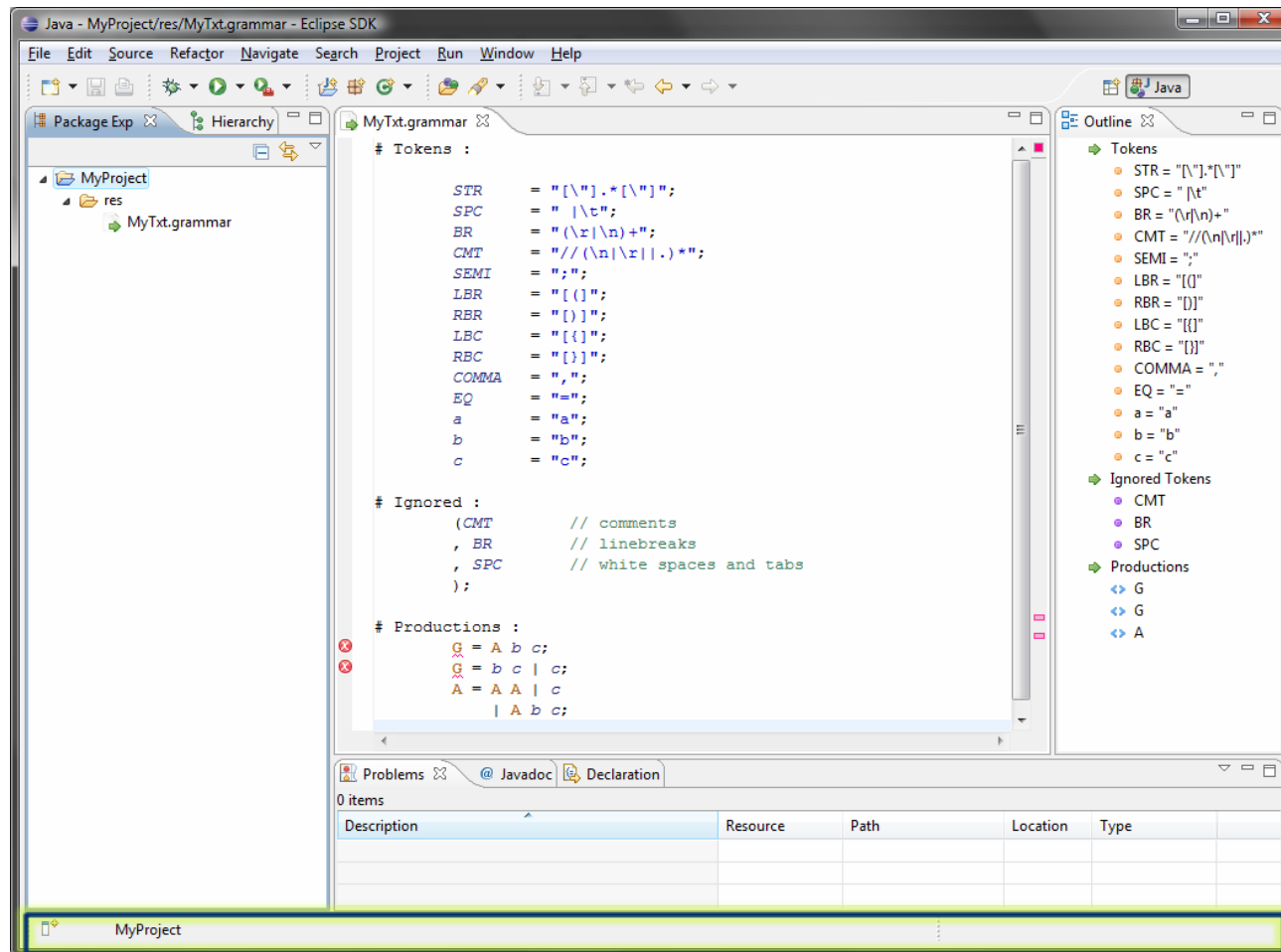
- Workbench-Struktur: Tool Bar



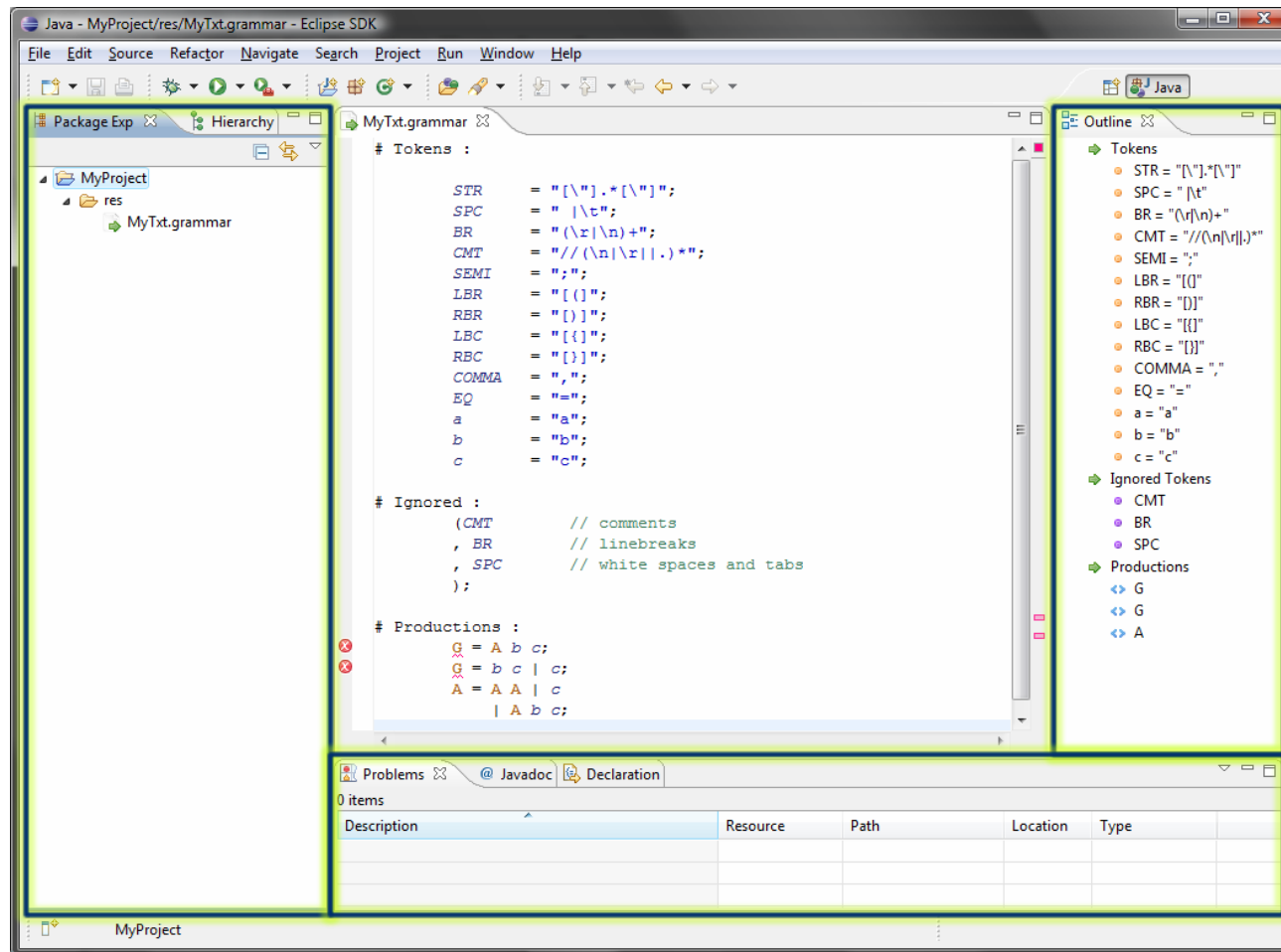
- Workbench-Struktur: Perspective Bar



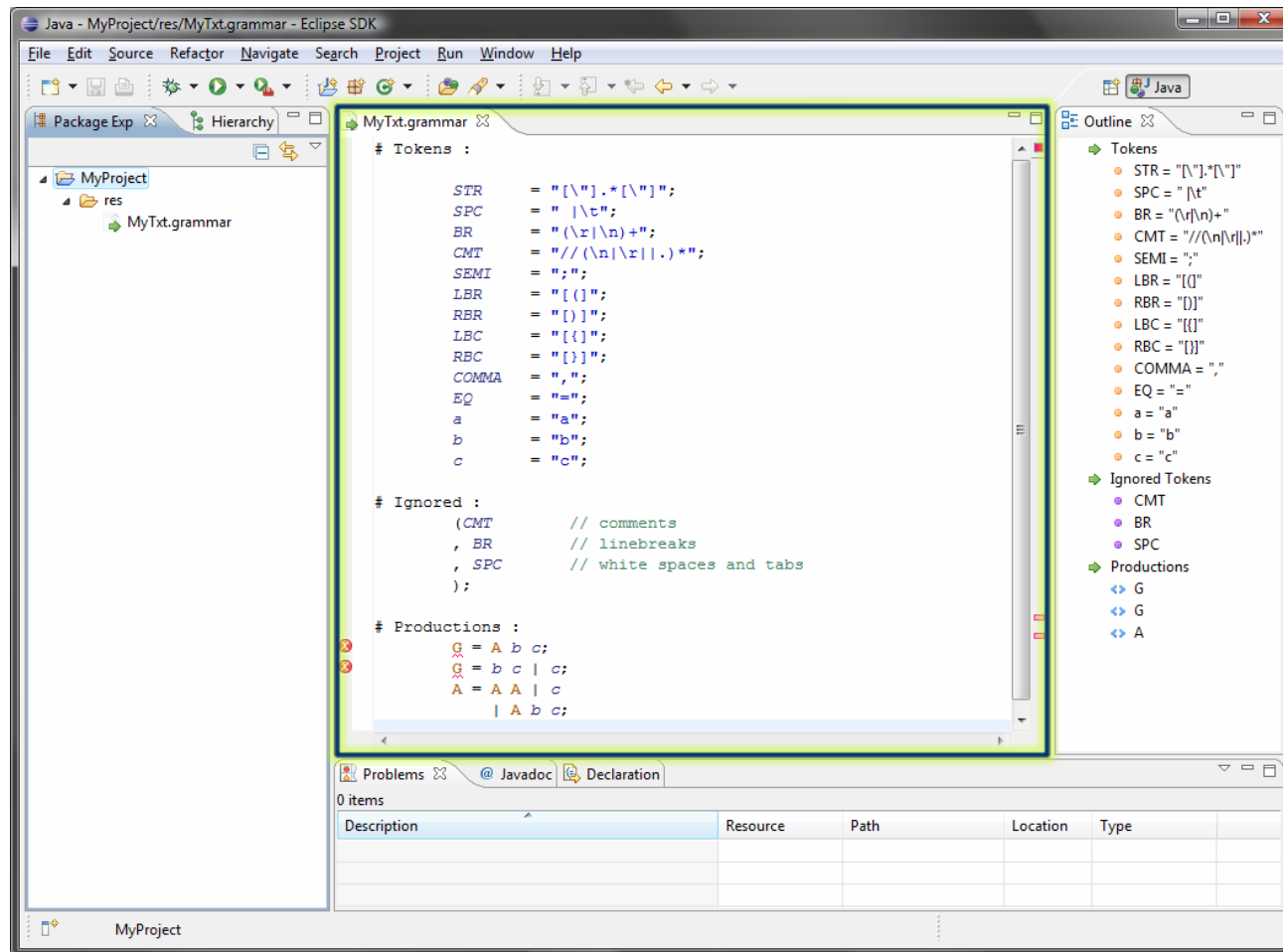
- Workbench-Struktur: Status Bar



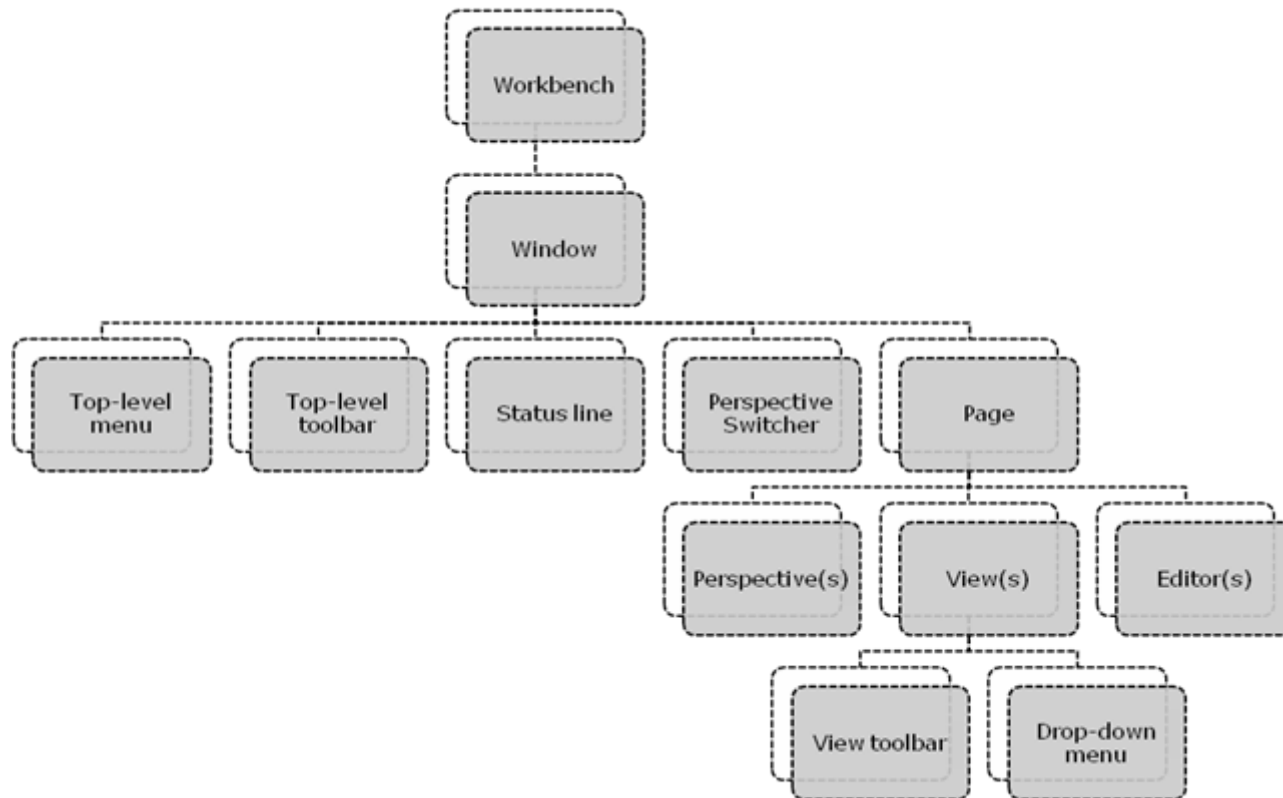
- Workbench-Struktur: Views



- Workbench-Struktur: Editor Area (mit aktivem Editor)



- Workbench-Struktur: Verwaltungs-Hierarchie



1. Motivation
2. RCP Konzepte
3. RCP Applikations-Struktur
4. Ein Beispiel
5. Plug-in Deployment mit UpdateSites



# Beispiel...

1. Motivation
2. RCP Konzepte
3. RCP Applikations-Struktur
4. Ein Beispiel
5. Plug-in Deployment mit UpdateSites

- Plug-ins können direkt aus Eclipse heraus gebaut und exportiert werden (z.B. als JAR)
  - ... und manuell in die Plug-in Struktur der Zielapplikation kopiert werden
  - Nachteile
    - Plug-ins können nicht automatisch aktualisiert werden
    - Plug-in Versionen müssen selbst verwaltet werden
    - Manuelles Kopieren umständlich
    - Abhängigkeiten beim "Installieren" werden nicht berücksichtigt

- Besser:
  - Erstellen und exportieren einer UpdateSite
    - Kann lokal, in ein Verzeichnis auf einem Server oder auf einen Webserver kopiert werden
    - Beinhaltet Features (mit deren Plug-ins und Fragments)
      - Jeweils mit verschiedenen Versionen
    - RCP-Applikationen (mit entsprechendem Update-Plug-in) können ihre Features durch UpdateSites verwalten
      - Install, Update, Deinstall

# Plug-in Deployment mit UpdateSites

- Vorgehen:
  - UpdateSite-Projekt erstellen
  - Feature-Projekt erstellen
    - Plug-ins und Fragments zuweisen
  - Im UpdateSite-Projekt
    - Feature-Versionen (gegebenenfalls) synchronisieren
    - "BuildAll" ausführen
      - UpdateSite wird im Projekt erstellt
    - In das Zielverzeichnis kopieren
  - Fertige UpdateSite in RCP-Applikation anmelden
  - Features auswählen und installieren
  - Fertig

Vielen Dank!

- [1] Eclipse
  - [www.eclipse.org](http://www.eclipse.org)
  - [www.eclipse.org/eclipse](http://www.eclipse.org/eclipse)
- [2] About the Eclipse Foundation
  - <http://www.eclipse.org/org/>
- [3] EclipsePedia
  - [http://wiki.eclipse.org/index.php/Rich\\_Client\\_Platform](http://wiki.eclipse.org/index.php/Rich_Client_Platform)
- [4] OSGi Alliance
  - [www.osgi.org](http://www.osgi.org)
- [5] Developing for the Rich Client Platform, EclipseCon
  - [http://www.eclipsecon.org/2005/presentations/EclipseCon\\_2005\\_Tutorial26.pdf](http://www.eclipsecon.org/2005/presentations/EclipseCon_2005_Tutorial26.pdf)
  - [http://wiki.eclipse.org/images/d/d9/EclipseCon\\_RCP\\_Tutorial\\_2006.pdf](http://wiki.eclipse.org/images/d/d9/EclipseCon_RCP_Tutorial_2006.pdf)