# Eclipse RCP Part III

Automotive Financial Services Insurance Life Science & Healthcare Public Sector Telecommunications & Media Travel & Logistics Utilities Automotive Financial Services Insurance Life Science & Healthcare Public Sector Telecommunications & Media Travel & Logistics Utilities Automotive Financial Services Insurance Life Science & Healthcare Public Sector Telecommunications & Media Travel & Logistics Utilities Automotive Financial Services Insurance Life Science Healthcare Public Sector Telecommunications & Media Travel & Logistics Utilities Automotive Financial Services Life Science & Healthcare Public Sector Telecommunications & Media Travel & Logistics Utilities Automotive Financial Services Insurance Life Science & Healthcare Public Sector Telecommunications & Media Travel & Logistics Utilities Automotive Financial Services Insurance Life Science & Healthcare Telecommunications & Media Travel & Logistics Utilities Automotive Financial Services Insurance Life Science & Healthcare Public Sector Telecommunications & Media Travel & Logistics Utilities Automotive Financial Services Insurance Life Science & Healthcare Public Sector Telecommunications & Media Travel & Logistics Utilities Automotive Financial Services Insurance Life Science & Healthcare Public Sector

.consulting .solutions .partnership

msg systems

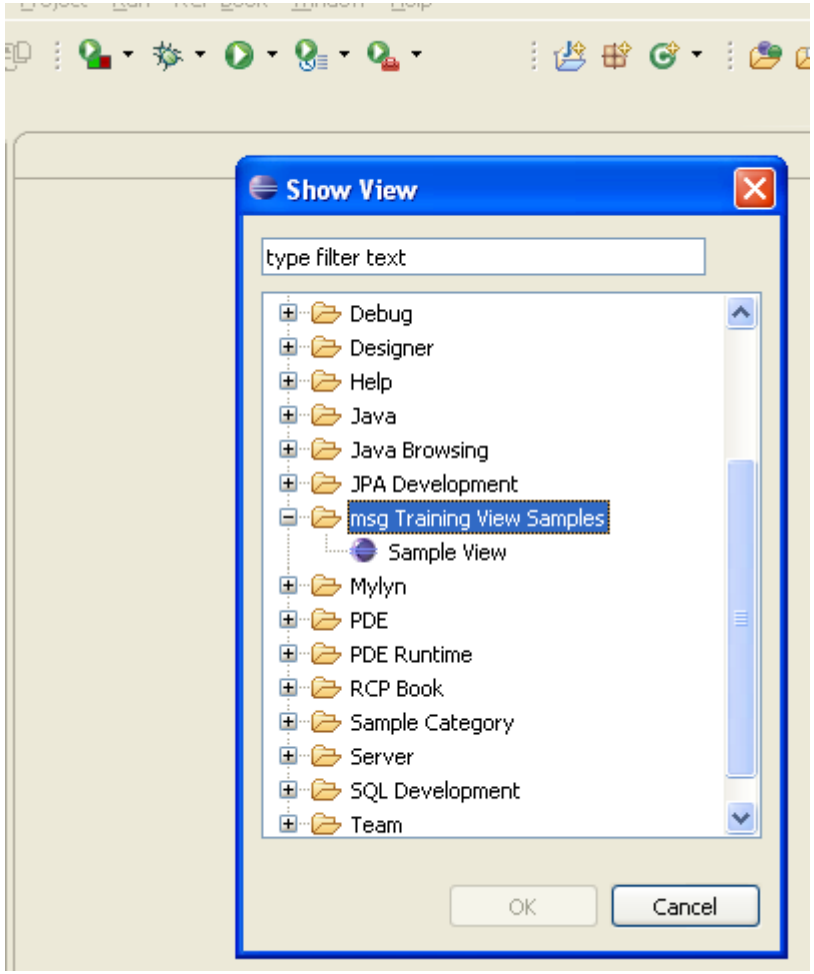
## Objective

- Building your first view
- Understanding extension points

# Views

- A view is like a window in the Eclipse workbench.
  - Views can be opened, closed, minimized and maximized by the user. This can also be done programmatically
- A view has an own toolbar with (optional) actions
- There is always one active view/editor at a time
- Views can be tabbed in a viewstack:

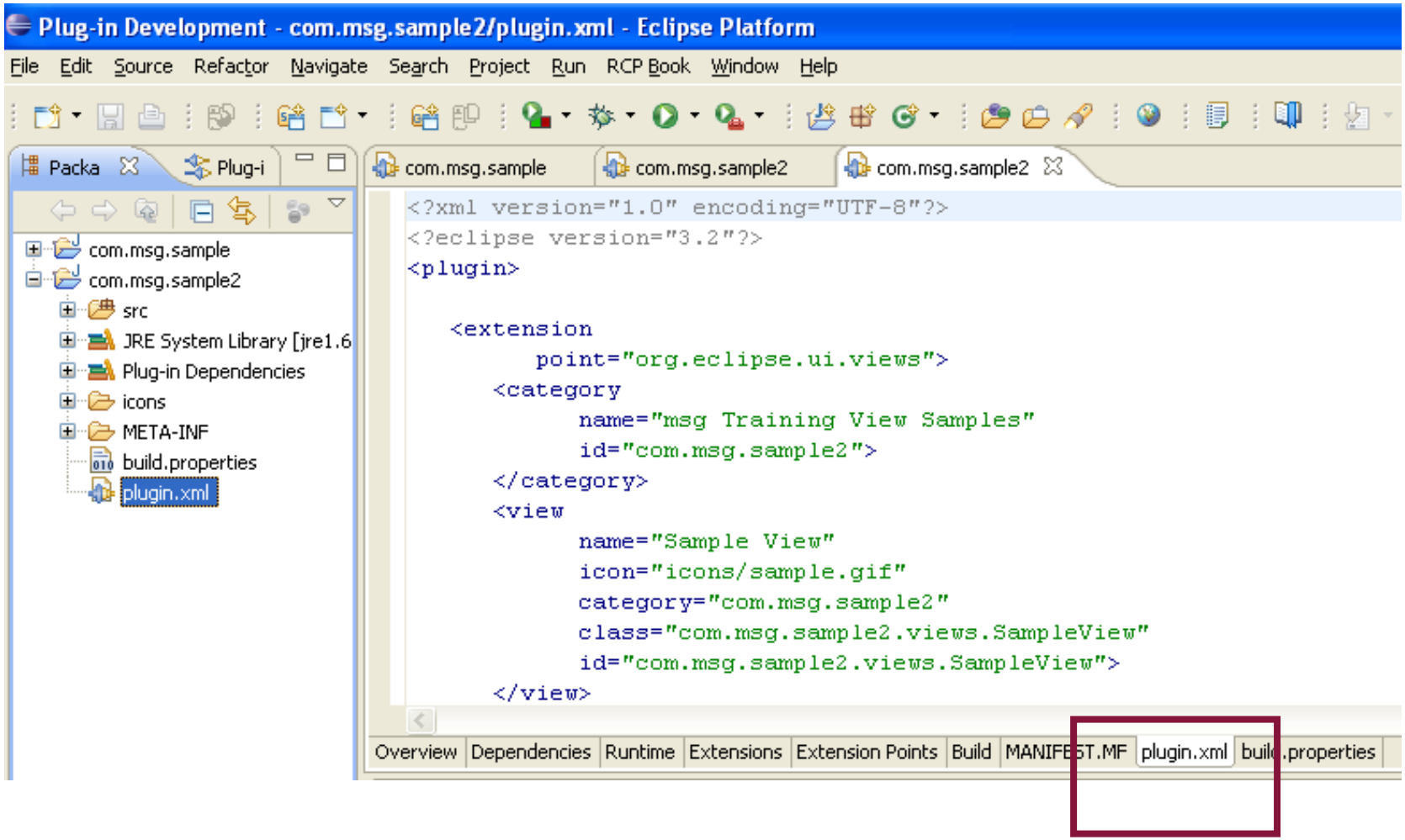

# Example



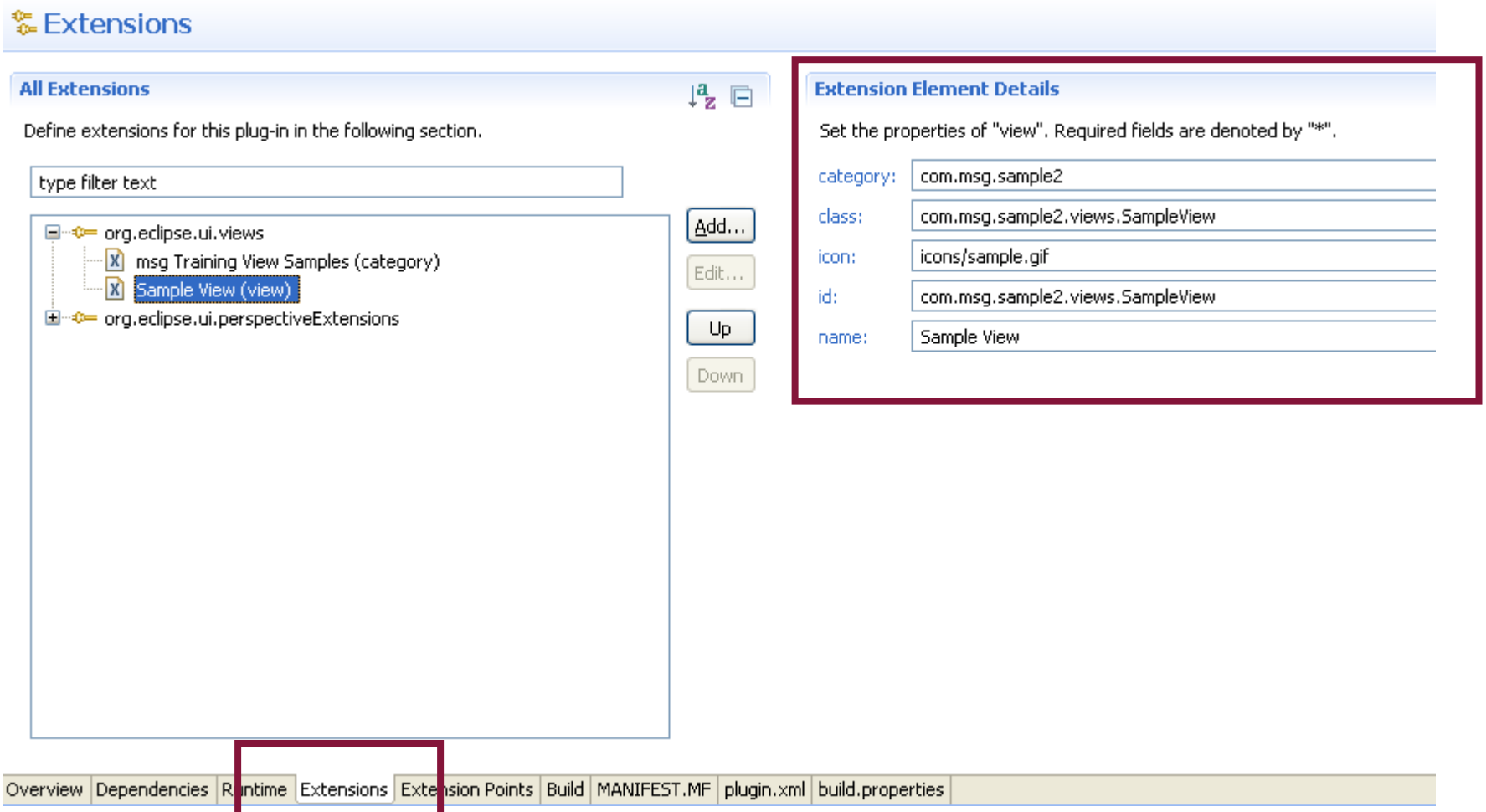

## The information is available without loading the plug-ins

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

   <extension
         point="org.eclipse.ui.views">
      <category
            name="msg Training View Samples"
            id="com.msg.sample2">
      </category>
      <view
            name="Sample View"
            icon="icons/sample.gif"
            category="com.msg.sample2"
            class="com.msg.sample2.views.SampleView"
            id="com.msg.sample2.views.SampleView">
      </view>
   </extension>
   <extension
```

# plugin.xml

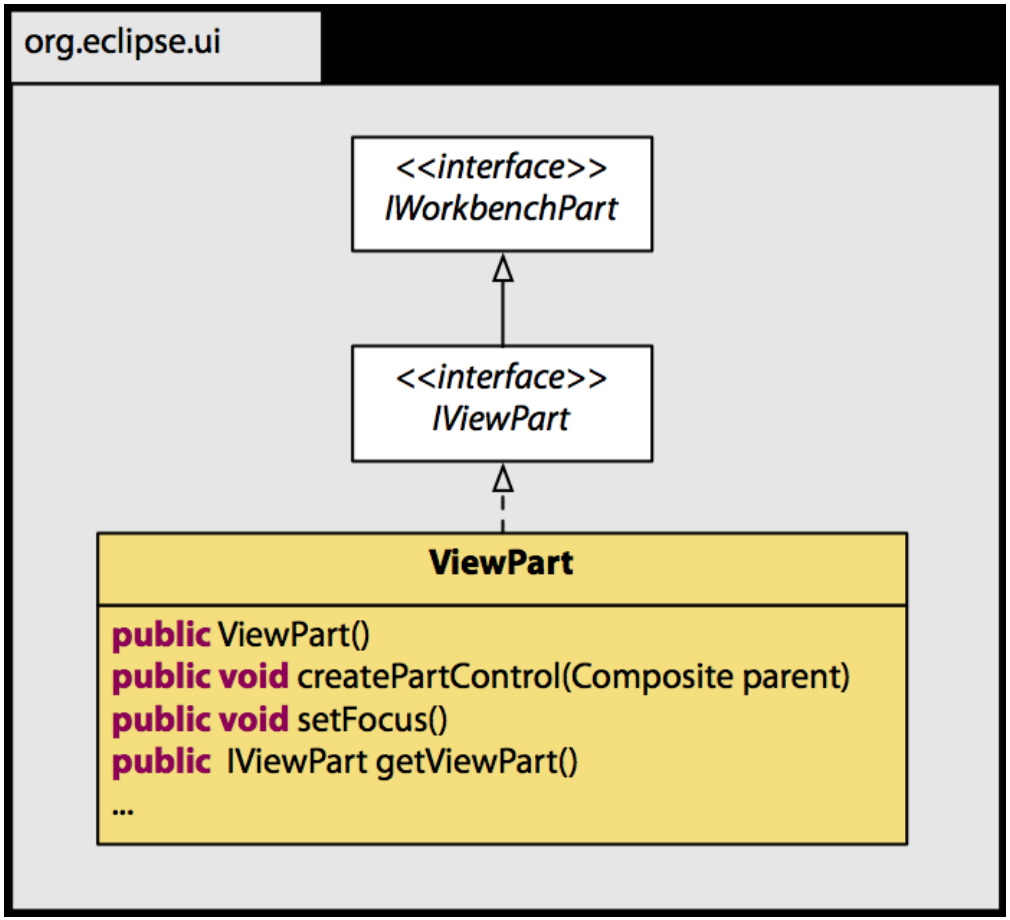


# Extensions



# ViewPart

- The specified view class is implemented using ViewPart

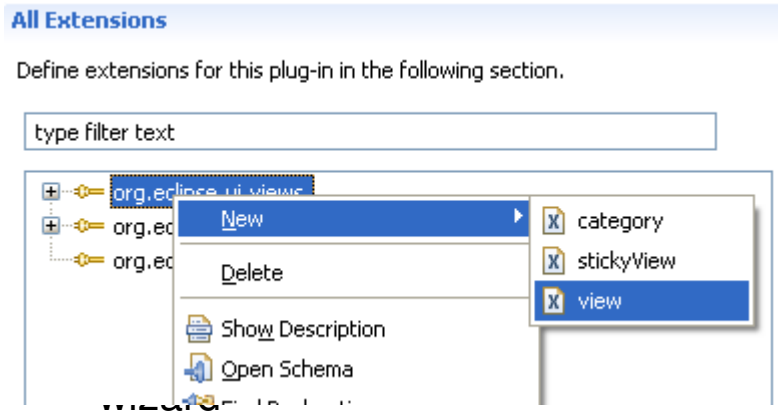


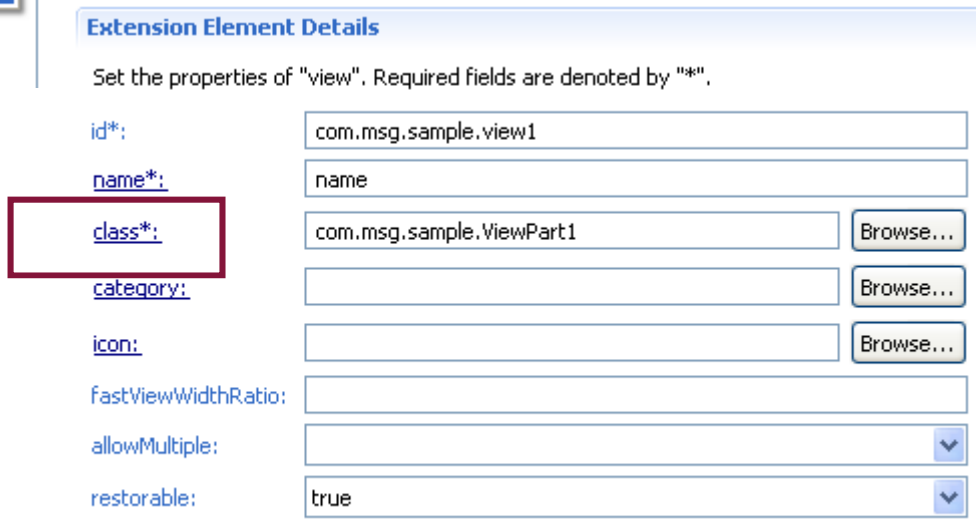
# Sample view implementation

```
public void createPartControl(Composite shell) {

shell.setLayout(new FillLayout());


DateTime calendar = new DateTime (shell, SWT.CALENDAR);
calendar.addSelectionListener (new SelectionAdapter () {
public void widgetSelected (SelectionEvent e) {
System.out.println ("calendar date changed");
}
});

DateTime time = new DateTime (shell, SWT.TIME);
time.addSelectionListener (new SelectionAdapter () {
public void widgetSelected (SelectionEvent e) {
System.out.println ("time changed");
}
});

}
```

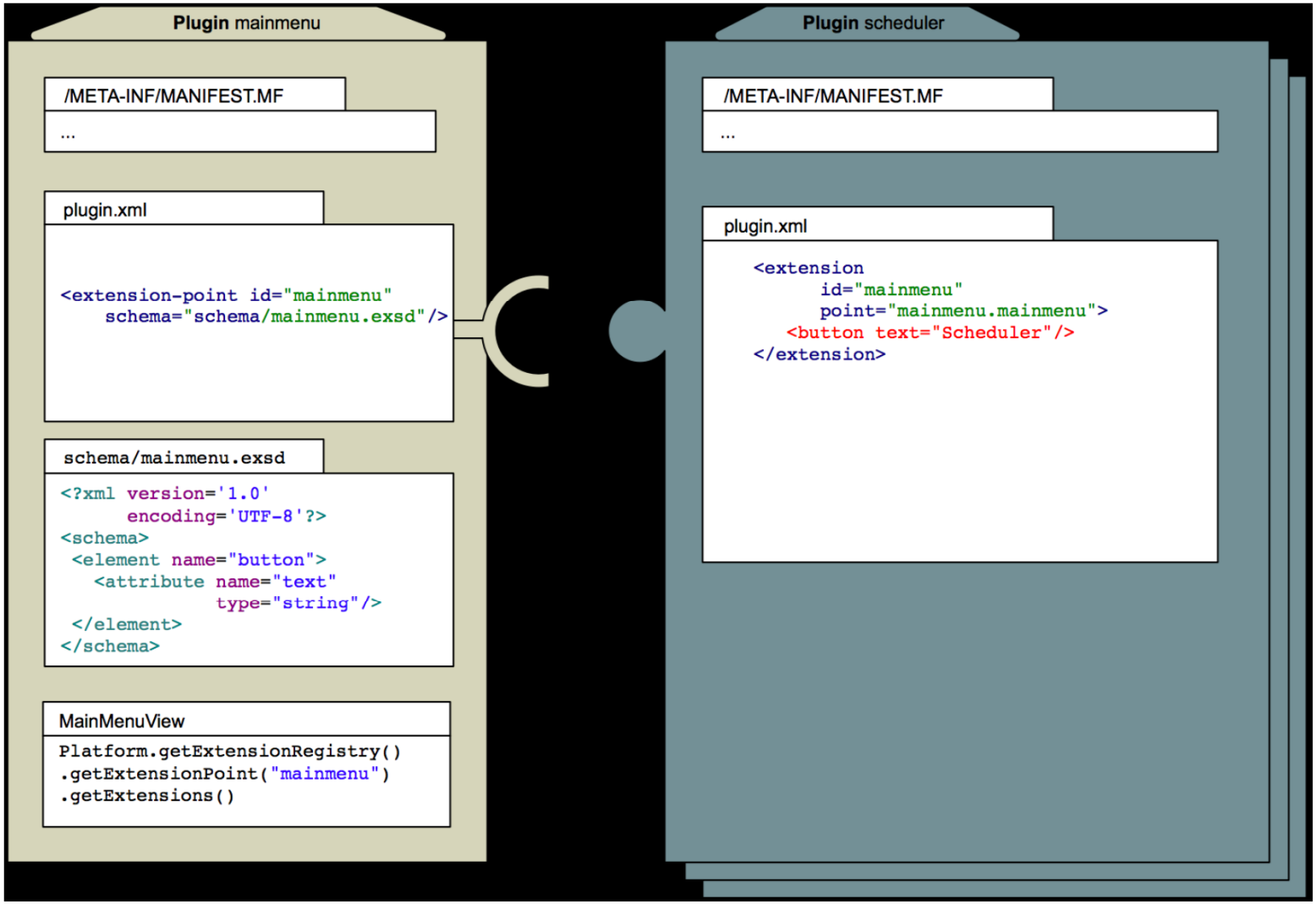
# Lab

- Add another view to the extension **“org.eclipse.ui.views”**
  → right click on org.eclipse.ui.views

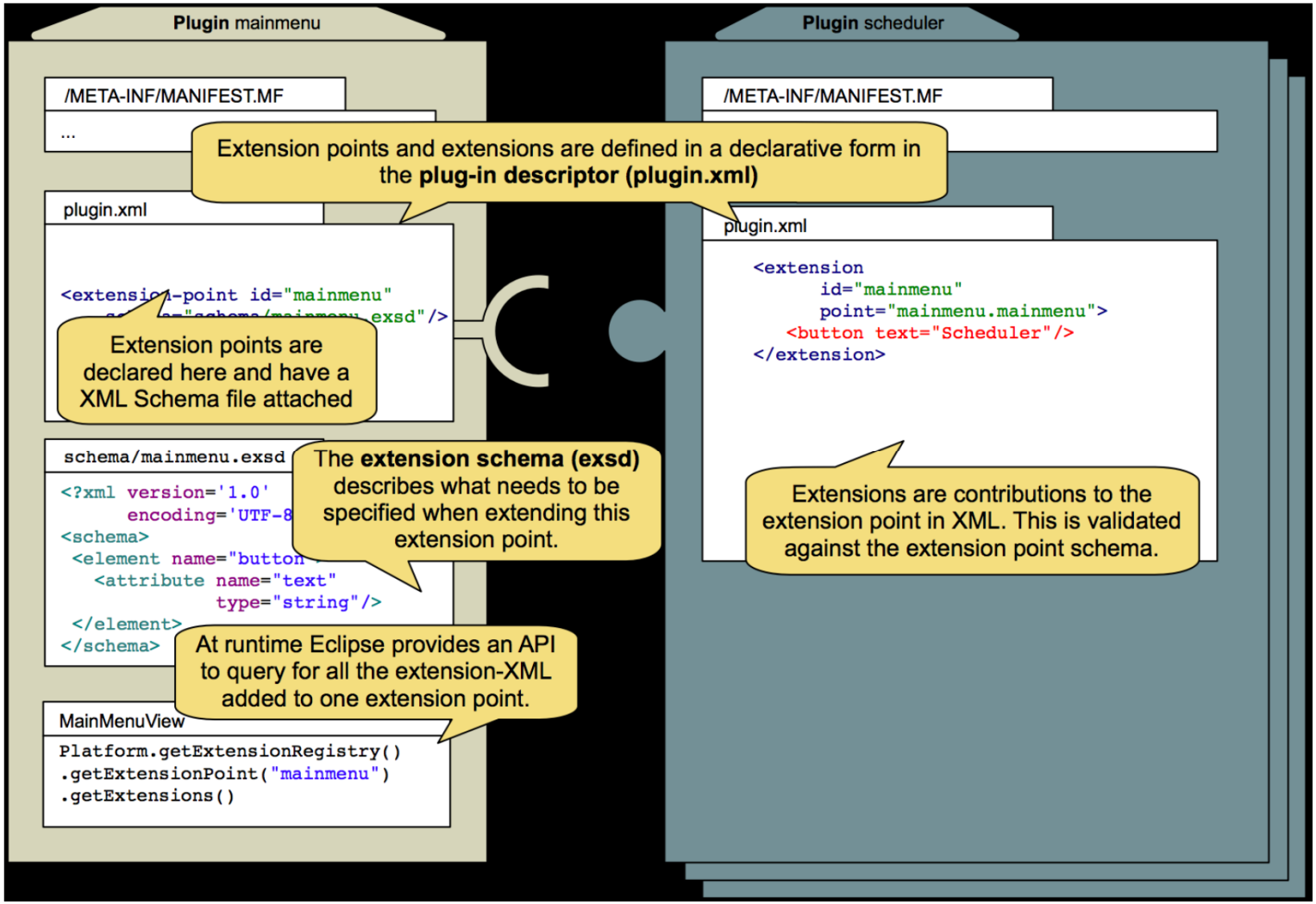


# plug-in concept



# how it fits together

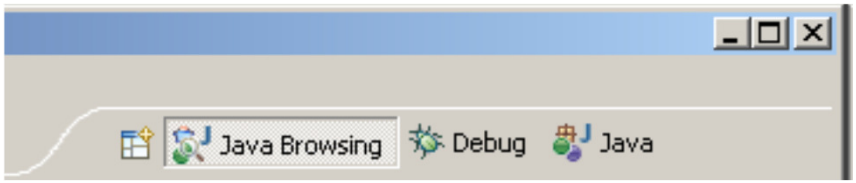


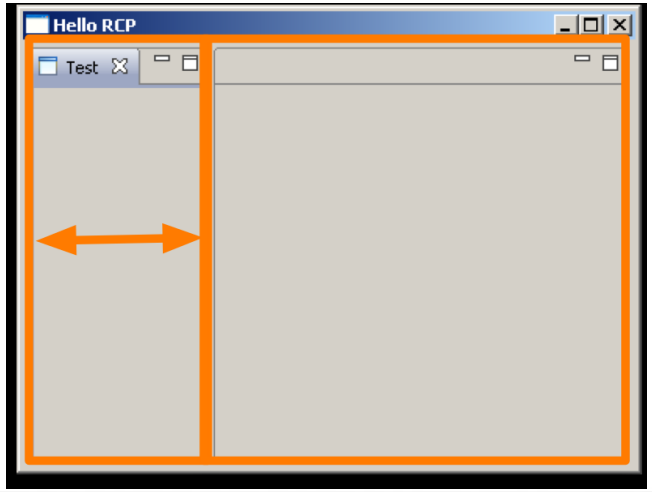
## Workbench Extension Point

- Dozens of extension points are provided by the Eclipse platform
- plug-ins and can be used to extend Eclipse and its workbench
- Examples for workbench extension points:
  - org.eclipse.ui.perspectives
  - org.eclipse.ui.views
  - org.eclipse.ui.editors
  - org.eclipse.ui.newWizards
- Full extension point reference: Help > Platform Plug-in Developer Guide > Reference > Extension Points Reference
- Declare new extension points to make your plug-ins extensible

# Perspective

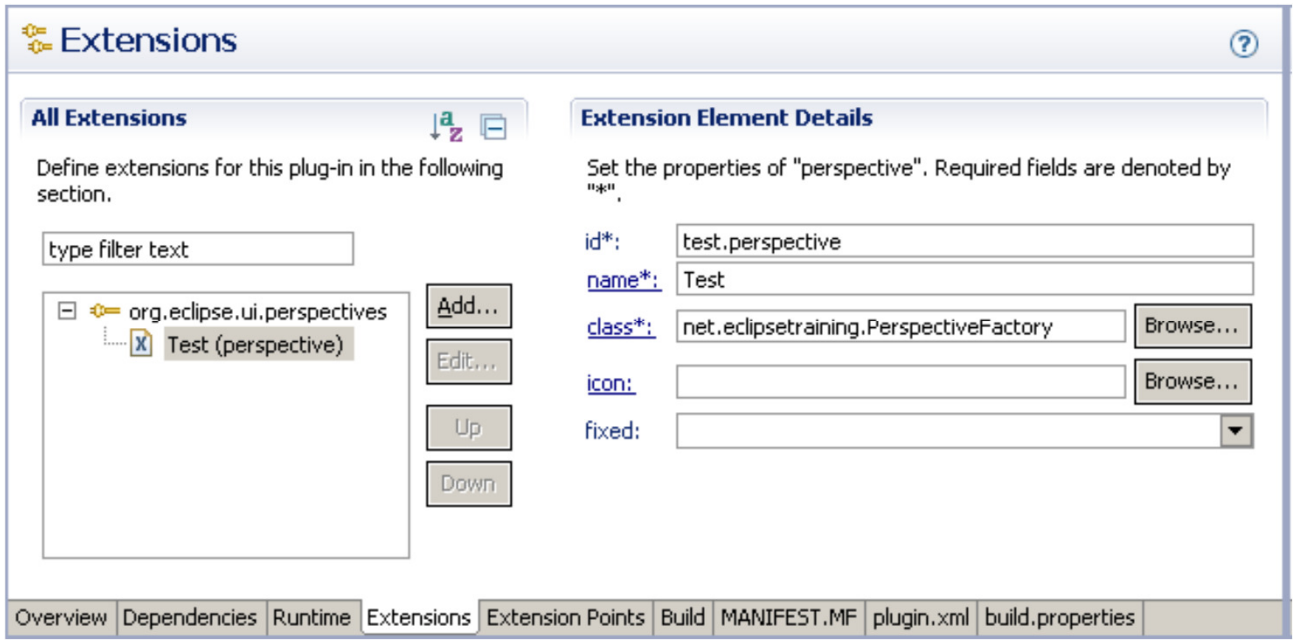
- A perspective defines the arragement (size and position) of workbench elements like views and editors
- It can define items for the menu and toolbar
- Perspectives can be switched by the user, for example using the PerspectivesBar
- Typically used for "bundling" of application- or role-specific functionalities, e.g.: end user mode, administration mode, report mode etc.



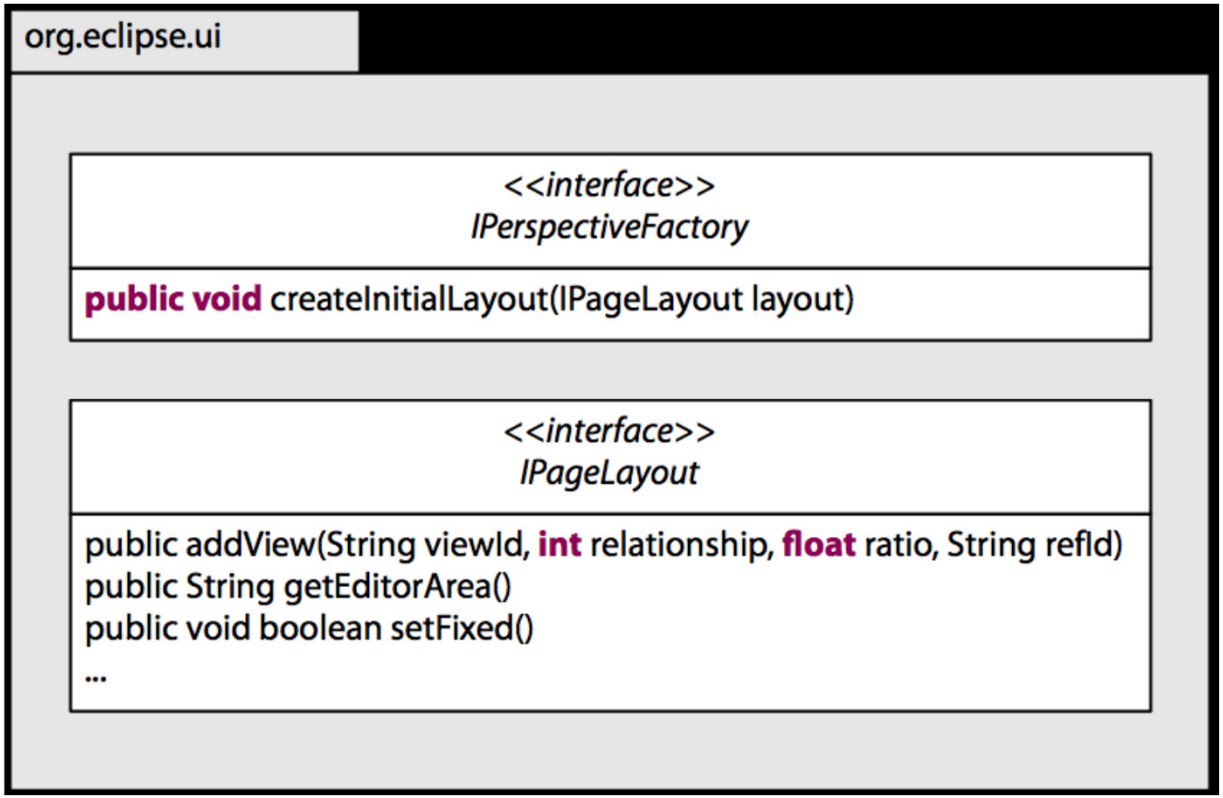
# Perspective

- Perspectives can be contributed using the extension point **org.eclipse.ui.perspectives**



## Perspective factory

- The specified PerspectiveFactory class configures the initial layout of the perspective



## Sample

```
public class Perspective implements IPerspectiveFactory {

public void createInitialLayout(IPageLayout layout) {
String editorArea = layout.getEditorArea();
layout.setEditorAreaVisible(false);

layout.addView(NavigationView.ID,  IPageLayout.LEFT, 0.25f,
    editorArea);

IFolderLayout folder = layout.createFolder("messages",
    IPageLayout.TOP, 0.5f, editorArea);
folder.addPlaceholder(View.ID + ":*");
folder.addView(View.ID);

layout.getViewLayout(NavigationView.ID).setCloseable(false);
}
}
```

## Lab

- Create a RCP Application
- Add two (or more) views to your application
- Add two perspectives to your application and implement the layout in the perspective factory
- Enable the perspective tool bar (see snippets)
- Try to configure your perspective using the code-snippets

# Vielen Dank für Ihre Aufmerksamkeit


.consulting .solutions .partnership

.msg systems