

Tom Schindl

Tom.schindl@bestsolution.at
<http://www.bestsolution.at>
<http://tomsondev.bestsolution.at>

Innsbruck, Austria

e4 – Adopt the programming model

Eclipse Summit Europe 2010
Thursday, November 4th , 2010

e4 – About Me

- **Founder and Owner of BestSolution.at**
- **Eclipse Committer**
 - e4
 - Platform UI
 - EMF
- **Projectlead**
 - Nebula
 - UFaceKit

e4 – The target

- **Target**

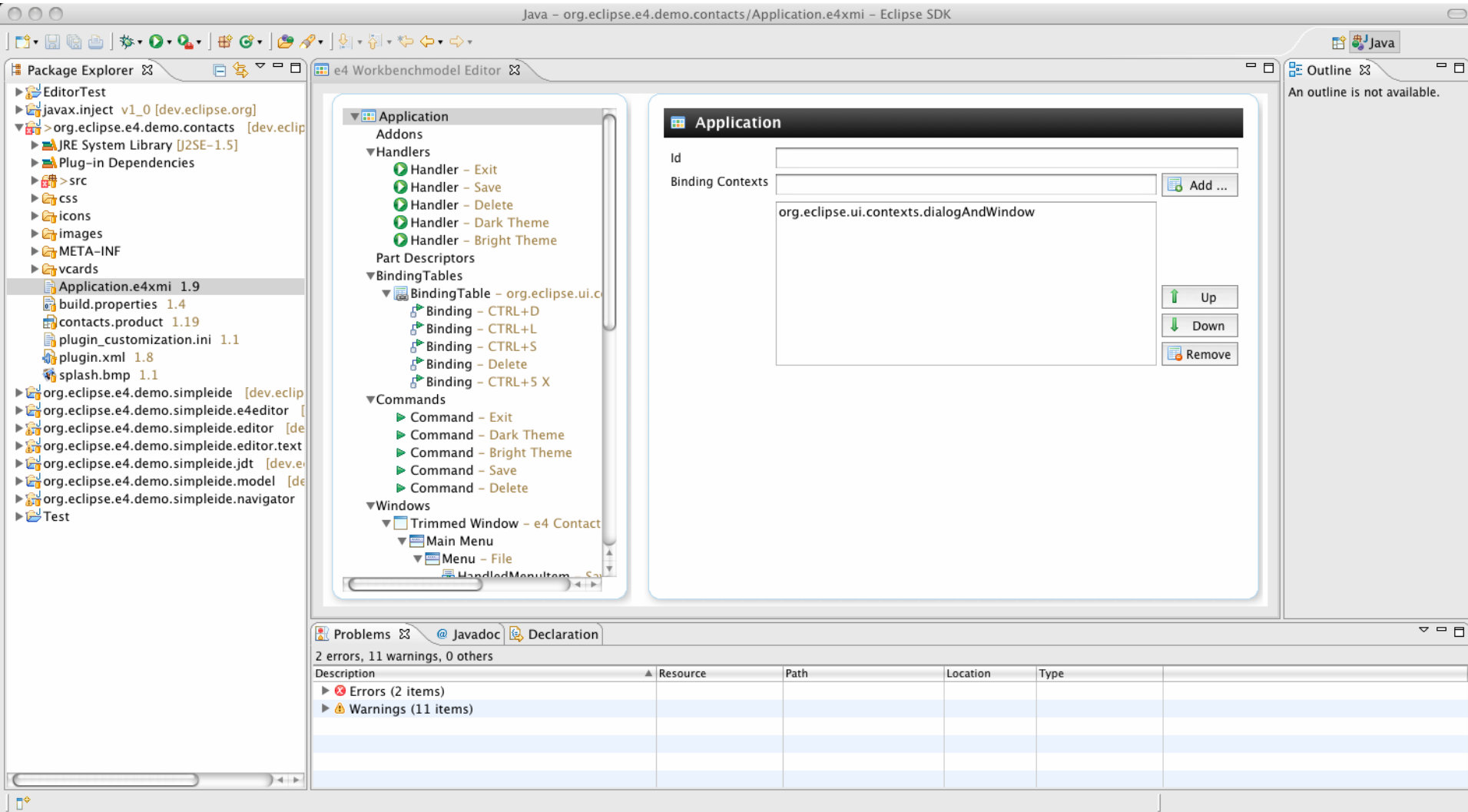
- Maintain one piece of code with runs in 3.x and 4.x SDK and ideally in any application built ontop of the Eclipse 4.0 Application Platform

- **Example**

- The Workbench Modeleditor shipping with Eclipse 4.0



E4 - ModelEditor



E4 - ModelEditor

Java - org.eclipse.e4.demo.contacts/Application.e4xmi - Eclipse SDK

Package Explorer

- EditorTest
- javax.inject v1_0 [dev.eclipse.org]
- org.eclipse.e4.demo.contacts [dev.eclipse.org]
 - JRE System Library [J2SE-1.5]
 - Plug-in Dependencies
 - src
 - css
 - icons
 - images
 - META-INF
 - vcards
 - Application.e4xmi 1.9**
 - build.properties 1.4
 - contacts.product 1.19
 - plugin_customization.ini 1.1
 - plugin.xml 1.8
 - splash.bmp 1.1
- org.eclipse.e4.demo.simpleide [dev.eclipse.org]
 - org.eclipse.e4.demo.simpleide.e4editor [dev.eclipse.org]
 - org.eclipse.e4.demo.simpleide.editor [dev.eclipse.org]
 - org.eclipse.e4.demo.simpleide.editor.text [dev.eclipse.org]
 - org.eclipse.e4.demo.simpleide.jdt [dev.eclipse.org]
 - org.eclipse.e4.demo.simpleide.model [dev.eclipse.org]
 - org.eclipse.e4.demo.simpleide.navigator [dev.eclipse.org]
 - Test

e4 Workbenchmodel Editor

Application

- Addons
 - Handlers
 - Handler - Exit
 - Handler - Save
 - Handler - Delete
 - Handler - Dark Theme
 - Handler - Bright Theme
 - Part Descriptors
- Binding Tables
 - BindingTable - org.eclipse.ui.com
 - Binding - CTRL+D
 - Binding - CTRL+L
 - Binding - CTRL+S
 - Binding - Delete
 - Binding - CTRL+S X
- Commands
 - Command - Exit
 - Command - Dark Theme
 - Command - Bright Theme
 - Command - Save
 - Command - Delete
- Windows
 - Trimmed Window - e4 Contacts
 - Main Menu
 - Menu - File
 - HandledMenuItem - Save
 - HandledMenuItem - Delete

Application

Id:

Binding Contexts: **Add ...**

org.eclipse.ui.contexts.dialogAndWi

Up **Down** **Remove**

Outline
An outline is not available.

Problems @ Javadoc Declaration

Description	Resource	Path	Location	Type
Errors (2 items)				
Warnings (11 items)				

Application.e4xmi - org.eclipse.e4.demo.contacts

e4 – 2 ways of SingleSourcing

- **Write 3.x ViewParts, EditorParts using only public APIs**
 - Drawback: You'll always need to run with the Compat Layer in 4.0
- **Adopt the Eclipse 4.0 Programing model**
 - Advantage: You create reuseable components for Eclipse 4.0 and Eclipse 3.x



e4 – 4.0 techs for 3.x

- **Many of the things worked on in the 4.0 stream work also in 3.x**
 - CSS-styling and theming of internal Part contents works in 3.6
 - DI works in 3.x
 - XWT works in 3.x
 - ...



e4 – 4.0 techs for 3.x

- **Hard to port from 4.0**
 - Handler stuff (@CanExecute)
- **Impossible to port from 3.x**
 - Modeled Workbench
 - Totally flexible rendering



e4 – How to single source

- **View/Editor content is part of a POJO**
- **No reference on 3.x nor 4.x API**
 - e.g. No direct reference to the selection service
 - e.g. No direct reference to ViewSite (e.g. to access workbench services)
- **Use broker services which are injected into your POJO**

e4 – Bundle Structuring

- **com.mycompany.views:**
 - A bundle with no **org.eclipse.ui** dependencies. Allowed dependencies:
 - org.eclipse.core.*
 - javax.inject
 - org.eclipse.e4.ui.di
- **com.mycompany.views.e3:**
 - A bundle to register POJOs in a 3.x world
 - e.g. usage of views-Extension Point, editors-Extension Point

e4 – Workbench Parts

- **ViewPart**

- Easy:

- `createPartControl()`
 - `setFocus()`

- Medium:

- Selection propagation



e4 – Workbench Parts

- **EditorPart**

- Easy

- createPartControl()
 - setFocus()

- Medium

- Selection propagation

- Advanced

- Abstraction of the IEditorInput
 - Dirty Lifecycle



e4 – ViewParts

▪ A view in 3.x

```
public class SampleView extends ViewPart {  
  
    private TableViewer viewer;  
  
    @Override  
    public void createPartControl(Composite parent) {  
        viewer = new TableViewer(parent, SWT.BORDER);  
    }  
  
    @Override  
    public void setFocus() {  
        viewer.getControl().setFocus();  
    }  
}
```

e4 – ViewParts

▪ A view in 4.0

```
public class SamplePOJOView {  
    private TableViewer viewer;  
  
    @Inject  
    public SamplePOJOView(Composite parent) {  
        viewer = new TableViewer(parent, SWT.BORDER);  
    }  
  
    @Focus  
    public void setFocus() {  
        viewer.getControl().setFocus();  
    }  
  
    public static void main(String[] args) {  
        // Create Shell  
    }  
}
```

e4 – ViewParts

▪ Use the Eclipse 4.0 POJO in 3.x

```
public class SampleView extends ViewPart {  
  
    private SamplePOJOView pojo;  
  
    @Override  
    public void createPartControl(Composite parent) {  
        pojo = new SamplePOJOView(parent);  
    }  
  
    @Override  
    public void setFocus() {  
        pojo.setFocus();  
    }  
}
```

e4 – EditorParts

- **Similar to ViewPart**

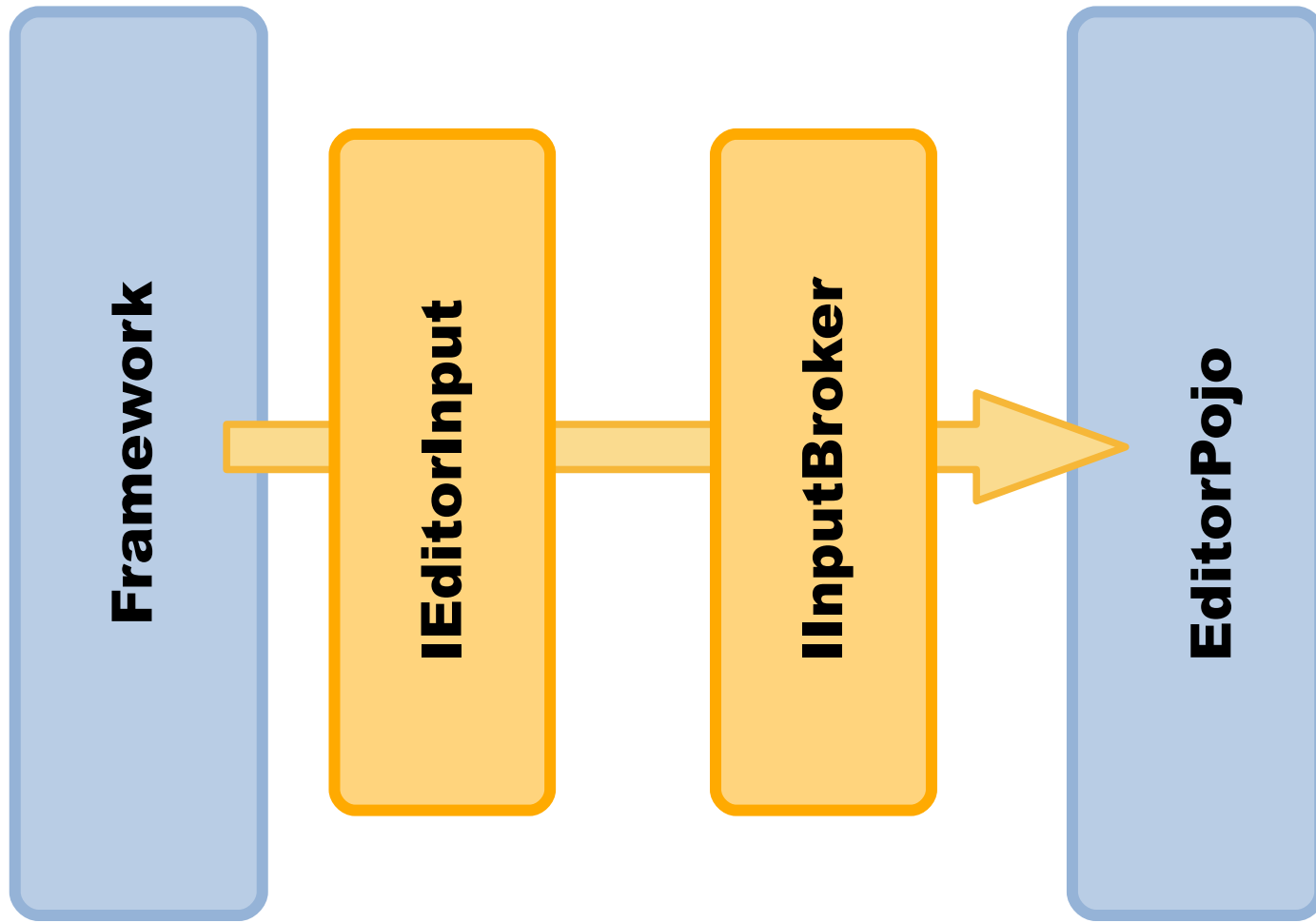
- createPartControl()
- setFocus()

- **Problem**

- IEditorInput: Would give us a dependency on org.eclipse.ui – Need to define a broker services



e4 – EditorParts



e4 – Hiding IEditorInput

▪ Example for EMF-Resource

```
public interface IEMFResource {  
    public Resource getResource();  
}  
  
public class EMFEditorResource implements IEMFResource {  
    private URI uri;  
  
    public EMFEditorResource(IEditorInput input) {  
        uri = Util.getURI(input); // from emf.edit  
    }  
  
    public Resource getResource() {  
        ResourceSet set = new ResourceSetImpl();  
        return set.createResource(uri);  
    }  
}
```

e4 – Hiding IEditorInput

▪ Example for EMF-Resource

```
public class TestEditor extends EditorPart {  
    private EMFEditorResource resource;  
  
    @Override  
    public void init(IEditorSite site, IEditorInput input)  
        throws PartInitException {  
        resource = new EMFEditorResource(input);  
    }  
  
    @Override  
    public void createPartControl(Composite parent) {  
        SamplePOJOView view = new SamplePOJOView(parent, resource);  
    }  
}
```

e4 – The forward compat layer

- **Gives you access to**
 - DI of OSGi-Services
 - Brokers for
 - Clipboard
 - DirtyState
 - Selection
- **Part of e4 Tooling download**
 - `org.eclipse.e4.tools.compat`
 - `org.eclipse.e4.tools.services`



e4 – The forward compat layer

▪ **How to adopt it**

```
public class E4WorkbenchModelEditor extends DIEDitorPart<ApplicationModelEditor> {  
    public E4WorkbenchModelEditor() {  
        super(ApplicationModelEditor.class, COPY|CUT|PASTE);  
    }  
}
```

▪ **Example**

- The e4 workbench model editor



e4 – The forward compat layer

▪ Make it available through Injection

```
public class XMIResourceFunction extends ContextFunction {  
    public Object compute(final IEclipseContext context) {  
        final IEditorInput input = context.get(IEditorInput.class);  
  
        URI resourceURI = EditUIUtil.getURI(input);  
        XMIModelResource resource = new XMIModelResource(resourceURI);  
  
        return resource;  
    }  
}
```

e4 – The forward compat layer

- **Things proposed for 4.1**
 - Support for EventBroker
 - Support for Memento stuff
 - Support for Handlers (@Execute)
 - Access to the „20 Things“



E4 - Links

- **My Blog**
<http://tomsondev.bestsolution.at>
- **e4 wiki**
<http://wiki.eclipse.org/E4>
- **Mail**
tom.schindl@bestsolution.at



**THE
END**