

Java UI's mit weniger Code

- Reduktion der UI Entwicklungs- und Wartungskosten
- Reduktion des UI Technologiekodes
- Vermeidung der Zersplitterung der UI Logik auf verteilte Codestellen
- Reduzierung der Testkosten (JUnit Testbarkeit des UIs)

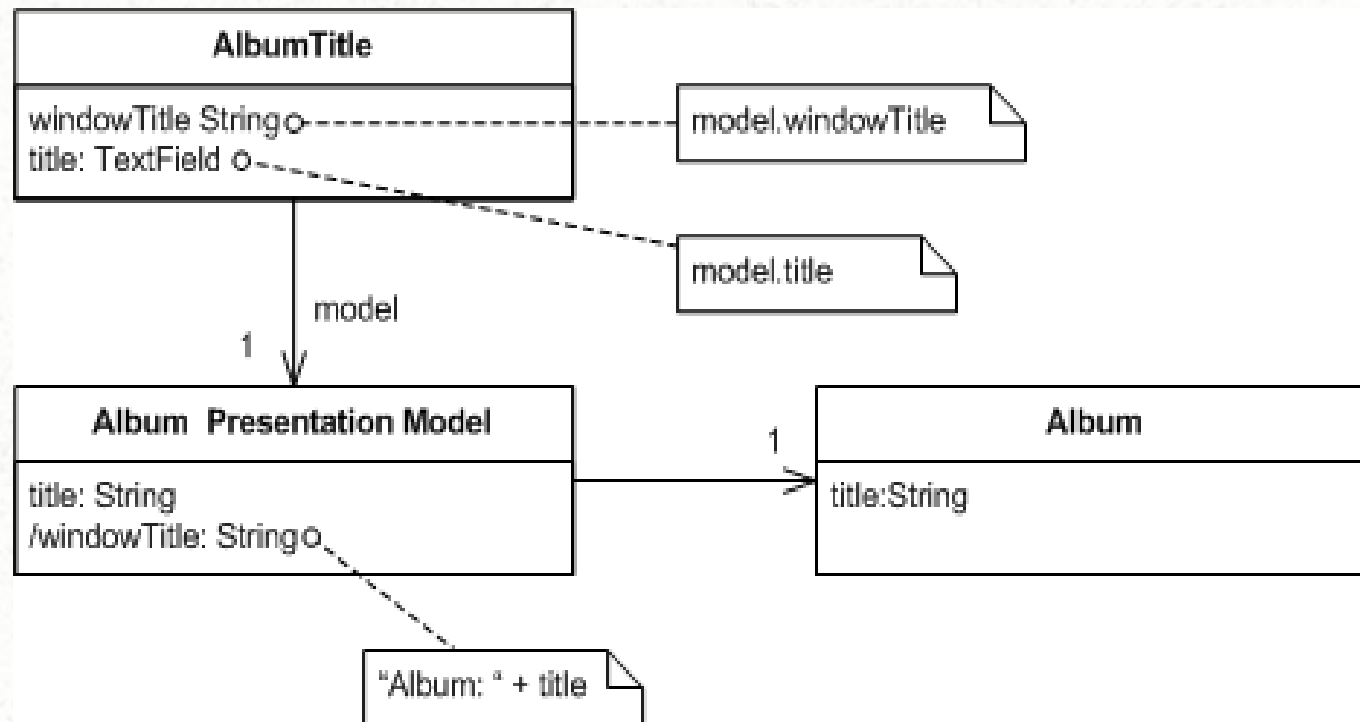
Aktuelles Technologiefeld

- Mit JPA, Spring etc. lassen sich Standardaufgaben unteren Anwendungsebenen effizient umsetzen.
- Standards der UI Entwicklung sind weniger gut unterstützt.
- Das UI ist deshalb oft ein unverhältnismäßig hoher Kostenfaktor.

Presentation Model (PM)

- In der View Technologie Schicht werden nicht die nackten Datenobjekte genutzt.
- Stattdessen: Reichhaltige UI Objekte - Präsentationsmodelle.
- Auf diese UI Logik Implementierung setzt dann nur noch eine schmale Layout Implementierung auf.

PM (Diagramm von Martin Fowler)



Aufgaben des PMs eines UI Controls

- Ein-Ausgabe des Wertes (Data Binding)
- Internationalisierte Titel, Tooltips, Icons
- Bereitstellung der Optionen und Kommandos für Popups etc.
- Sichtbarkeits- und Editierbarkeitssteuerung
- Konvertierung, Validierung
- Fehler- bzw. Informationsstatusbereitstellung (Zustandsinfo)
- Änderungsinformation (Event Binding)
- Caching, Eingabewertpufferung
- Backend Integration (Dependency Injection, EL)

Zutaten für ein Formular

- Eine Pojo Datenklasse
- Ein PM für die UI Logik
- Ein View für das Layout

Ein PM

```
@PmBeanCfg(beanClass=DemoBean.class, autoCreateBean=true)
public class BasicDemoElementPm extends PmBeanBase<DemoBean> {

    @PmAttrStringCfg(maxLen=10) @PmAttrCfg(required=true, defaultValue="...")
    public final PmAttrString      textField = new PmAttrStringImpl(this);

    public final PmAttrDate        dateField = new PmAttrDateImpl(this);

    public final PmAttrEnum<Color> color = new PmAttrEnumImpl<Color>(this, Color.class);

    public final PmCommand          cmdSave = new PmCommandImpl(this) {
        protected void doItImpl() {
            ...
        }
    };
};
```

Resources_de.properties

```
basicDemoElementPm=Vorstellung von myUi Basisfunktionen
basicDemoElementPm.textField=Kurzer Text
basicDemoElementPm.dateField=Ein Datum
basicDemoElementPm.dateField.format=d.M.yy;dd.MM.yyyy
basicDemoElementPm.color=Farbe
basicDemoElementPm.color.tooltip=Sie können hier eine Farbe für... auswählen.
basicDemoElementPm.Color.RED=Rot
basicDemoElementPm.Color.GREEN=Grün
```

Ein SWT View

Anbindung an einen per GUI Builder hergestellten View:

```
public class MyCompositeBuilder extends PvCompositeBase<MyComposite, BasicDemoElementPm> {  
  
    @Override  
    protected void bindImpl(MyComposite view, BasicDemoElementPm pm) {  
  
        bindWidget(view.getMyName() , view.getNameLabel() , pm.textField);  
        bindWidget(view.getTextArea(), view.getTextAreaLabel(), pm.textFieldLong);  
        bindButton(view.getButton1() , pm.cmdSave);  
  
    }  
}
```

*Üblicherweise nur eine Zeile je Control Binding
(incl. Sichtbarkeits-Enablement Steuerung, Statusvisualisierung etc.)*

Noch weniger bei Nutzung generischer Views (Trees, Tables, Standardforms)

Ein JSF View

```
<m:form formId="f" pm="#{basicDemoElementPm}">
  <table>
    <m:tr pm="#{pm.textField}" >      <m:inputText />      </m:tr>
    <m:tr pm="#{pm.textFieldLong}" >  <m:inputTextarea />  </m:tr>
    <m:tr pm="#{pm.color}" >          <m:selectOneMenu />  </m:tr>
  </table>

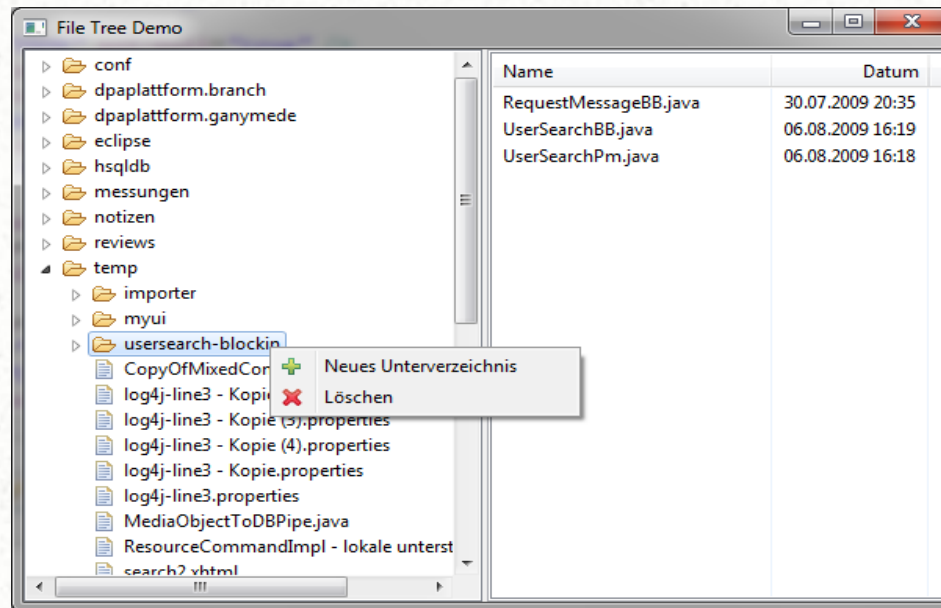
  <m:commandButton pm="#{pm.cmdSave}" />
  <m:commandButton pm="#{pm.cmdBack}" />
</m:form>
```

Fokussiert auf das Layout.

Logik liegt im PM (Sichtbarkeit, Validierung etc.)

*Üblicherweise nur ein Tag je Control Binding
(incl. Sichtbarkeits-Enablement Steuerung, Statusvisualisierung etc.)*

Ein File Browser



ca. 100 Zeilen UI Logik Code

7 Zeilen View Code (SWT)

Zusammenfassung

- UI Technologiecode entfällt in Standardanwendungen zu 90%.
Viele 'boiler plate' Codeblöcke entfallen.
- UI Logik gut wartbar und testbar an einem Ort.
- UI Kosten und Umsetzungszeiten deutlich reduziert (bei dpa/mecom).
- Goodie: Gute Basis für Web- und Rich Client Entwicklung.

?

- Ist der Ansatz gut / interessant?
- Lohnt eine Open Source Veröffentlichung?
- Aktuelle Chancen
 - Vorstellung bei Otto
 - Förderung durch den Pro-Ideen Fonds



Ende