Are you still manually coding UIs?

Maximilian Koegel
mkoegel@eclipsesource.com
Data conforms to a domain-specific data model
Data is often viewed/edited in a form-based UI
Data model needs to be mapped to UI
Modeling the UI

1. Data Binding with Controls
2. Modeling Structure with Layouts
Modeling UI Data Binding with Controls

Domain Model

View Model

UI

Domain Model References
Modeling UI Structure with Layouts

View Model

- Domain Model Instances
  - User View Model
    - UserView
      - HorizontalLayout
        - VerticalLayout Left Column
          - Control firstName
          - Control weight
          - Control nationality
          - Control timeOfRegistration
          - Control active
        - VerticalLayout Column Right
          - Control lastName
          - Control heigh
          - Control gender
          - Control dateOfRegistration
          - Control eMails

UI

- Maximilian Kögel [User]
  - First Name: Maximilian
  - Weight
  - Gender: Male
  - Date Of Birth: dd.MM.yyyy
  - Nationality: German
  - Time Of Registration: 27.10.2013
  - EMail: ""
Extensibility and Adaptability

- Custom Controls
  - e.g. Email-Control

- Custom View Model Elements and Renderers
  - e.g. Group
Custom Control: Email

![Custom Control Email](image)
Custom View Model Element

- Define new view model element in Ecore

- Generate model and edit plugins
- Create a Renderer
Exchangeable UI Technology

- UI Model is independent of Renderers
- Only renderers are UI technology dependent
- Renderers can be replaced
- Allows for parallel use of different UI technologies
- Existing renderers:
  - RCP/SWT
  - Web based on RAP
  - JavaFX (demo only)
Web-Renderer based on RAP
When to use/not use of UI Modeling

- Large Domain Model
- Many different Views
- Frequent Domain Model changes
- Homogenous UI
- UI Technology Independence
- Improved Customer Involvement
  - Fast Turnaround + Rapid Prototyping
  - Easy-to-grasp UI concepts
More Information

• Current Release 1.1: Minimal public view model API
• Roadmap:
  • Expose more view model API
  • JavaFX Renderer
  • Tooling

• EMF Client Platform: http://eclipse.org/emfclient

• Twitter: #emfcp https://twitter.com/EMFCP
# JavaFX Renderer

![JavaFX Renderer Image](image-url)

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Name</td>
<td>Male</td>
</tr>
<tr>
<td>Last Name</td>
<td></td>
</tr>
<tr>
<td>Heighth</td>
<td>0</td>
</tr>
<tr>
<td>Weight</td>
<td>0</td>
</tr>
<tr>
<td>Time Of Registration</td>
<td>Bla</td>
</tr>
<tr>
<td>Date Of Birth</td>
<td></td>
</tr>
<tr>
<td>Gender</td>
<td>Male</td>
</tr>
<tr>
<td>Nationality</td>
<td>German</td>
</tr>
<tr>
<td>Active</td>
<td></td>
</tr>
<tr>
<td>EMails</td>
<td></td>
</tr>
</tbody>
</table>
Embedding the editor
What is EMF?

“The EMF project is a **modeling framework** and **code generation** facility for building tools and other applications based on a **structured data model**. From a model specification described in XMI, EMF provides tools and runtime support to produce a set of **Java classes** for the model, along with a set of adapter classes that enable **viewing** and **command-based editing** [...]”

Source: http://www.eclipse.org/emf
What is EMF Client Platform?

- Navigator
- Editor
- Validation View
- Model Workspace
- Dialogs