EMFForms:
Are you still manually coding UIs?
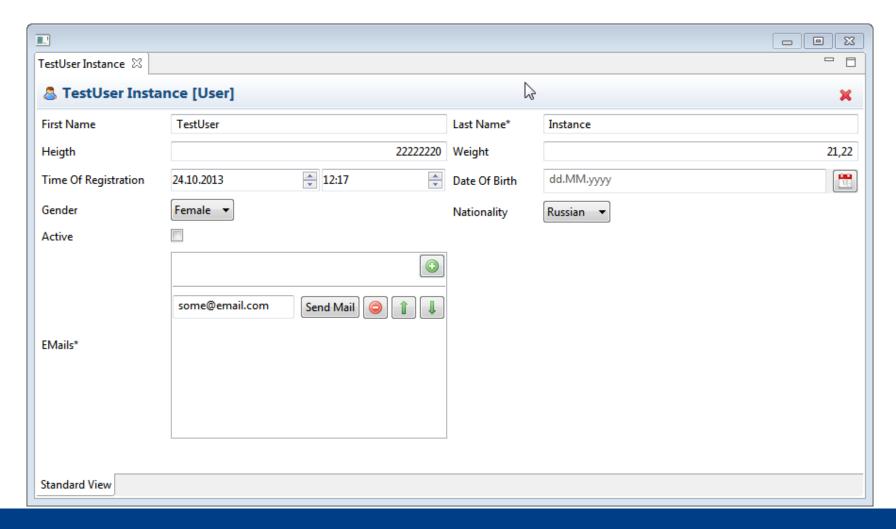
Maximilian Kögel
mkoegel@eclipsesource.com
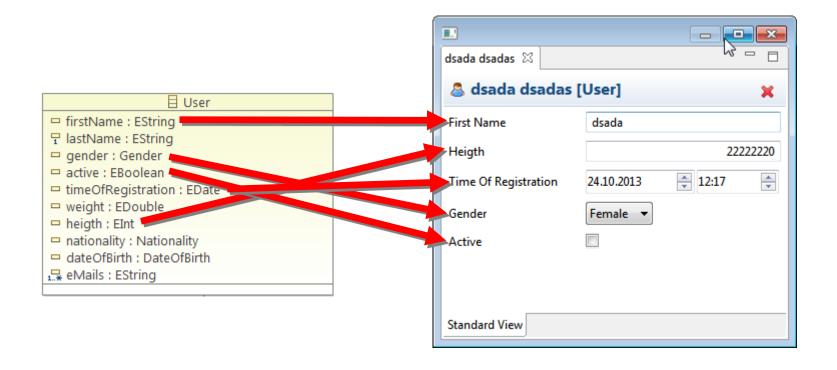
Data is often viewed/edited in a form-based UI
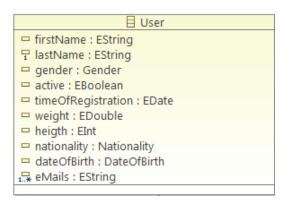
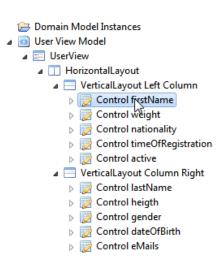# Data model needs to be mapped to UI

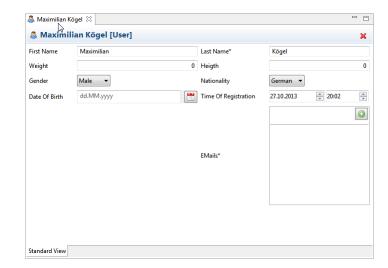# Modeling the UI

Renderer

Domain Model (XSD, Ecore) + **View Model** = **User Interface**

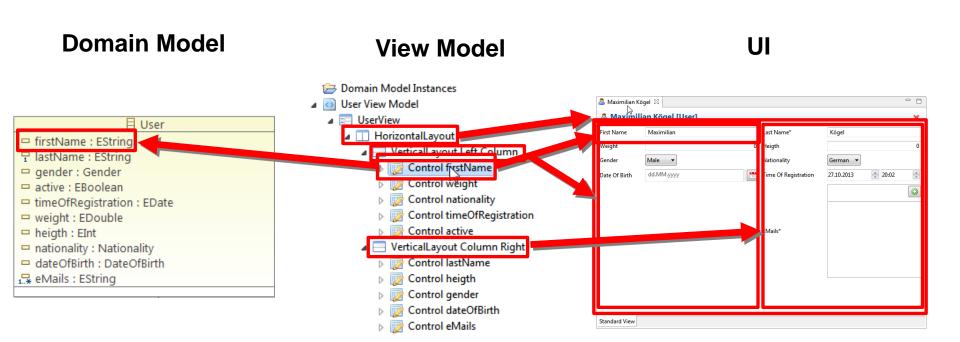# Modeling the UI

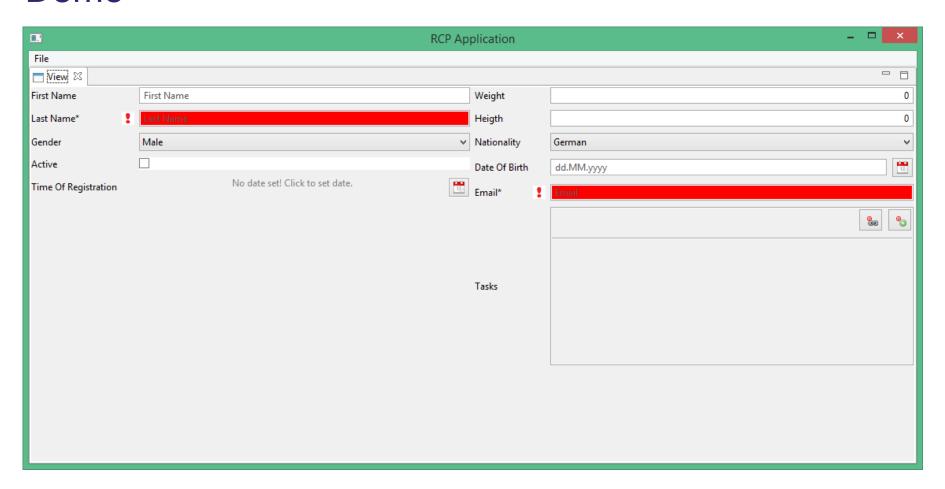**Domain Model** **View Model** **UI**

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

Demo

EMF Forms Features
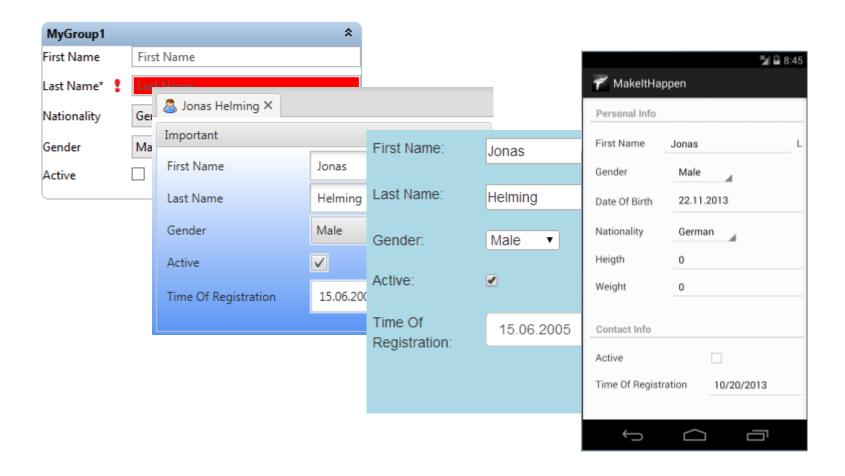
- Controls to view and edit data
- Layouts to structure UI
- Validation
- View Model Tooling
- Rule-based visibility and enablement of controls
- Exchangeable UI Technology:
  - Desktop:
    - SWT (Production)
    - JavaFX (Development)
  - Browser/Web based on RAP (Production)
  - Mobile based on Tabris (Demo)

First Name | Maximilian
Last Name* | ❗ Last Name
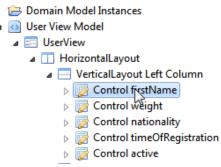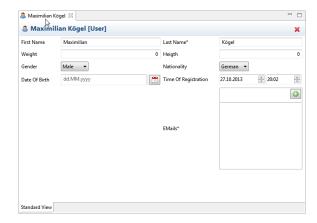
Rendering technologies

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 1.3: View Model Tooling
- Roadmap:
    - Improve Tooling
    - Expose more API
    - JavaFX Renderer

- **EMF Forms: http://emfforms.org**

- Twitter: #emfcp https://twitter.com/EMFCP