Building Business UIs with EMF Forms

Maximilian Koegel
mkoegel@eclipsesource.com
Data is often viewed/edited in a form-based UI
Data model needs to be mapped to UI
Manual coding of these UIs is a huge effort

Shell shell = new Shell (display);
GridLayout gridLayout = new GridLayout (5, false);
shell.setLayout(gridLayout);

Label Vorname = new Label (shell, SWT.NONE);
Vorname.setText("Vorname");

Text VornameField = new Text (shell, SWT.BORDER);
GridData data = new GridData (SWT.FILL, SWT.CENTER, true, false);
VornameField.setLayoutData (data);

Shell shell = new Shell (display);
GridLayout gridLayout = new GridLayout (5, false);
shell.setLayout(gridLayout);

Label Vorname = new Label (shell, SWT.NONE);
Vorname.setText("Vorname");

Text VornameField = new Text (shell, SWT.BORDER);
GridData data = new GridData (SWT.FILL, SWT.CENTER, true, false);
VornameField.setLayoutData (data);

Label Trennlinie = new Label (shell, SWT.SEPARATOR);
Trennlinie.setLayoutData (data);

Text Trennlinie = new Text (shell, SWT.BORDER);
GridData data = new GridData (SWT.FILL, SWT.CENTER, false, false, 1, 2);
Trennlinie.setLayoutData (data);

Label Nachname = new Label (shell, SWT.NONE);
Nachname.setText("Nachname");

Text NachnameField = new Text (shell, SWT.BORDER);
GridData data = new GridData (SWT.FILL, SWT.CENTER, true, false);
NachnameField.setLayoutData (data);

Label Geschlecht = new Label (shell, SWT.NONE);
Geschlecht.setText("Geschlecht");

Text GeschlechtField = new Text (shell, SWT.BORDER);
GridData data = new GridData (SWT.FILL, SWT.CENTER, false, false, 1, 2);
GeschlechtField.setLayoutData (data);

shell.pack ();

Shell shell = new Shell (display);
GridLayout gridLayout = new GridLayout (5, false);
shell.setLayout(gridLayout);

Label Vorname = new Label (shell, SWT.NONE);
Vorname.setText("Vorname");

Text VornameField = new Text (shell, SWT.BORDER);
GridData data = new GridData (SWT.FILL, SWT.CENTER, true, false);
VornameField.setLayoutData (data);

Label Trennlinie = new Label (shell, SWT.SEPARATOR);
Trennlinie.setLayoutData (data);

Text Trennlinie = new Text (shell, SWT.BORDER);
GridData data = new GridData (SWT.FILL, SWT.CENTER, false, false, 1, 2);
Trennlinie.setLayoutData (data);

Label Nachname = new Label (shell, SWT.NONE);
Nachname.setText("Nachname");

Text NachnameField = new Text (shell, SWT.BORDER);
GridData data = new GridData (SWT.FILL, SWT.CENTER, true, false);
NachnameField.setLayoutData (data);

Label Geschlecht = new Label (shell, SWT.NONE);
Geschlecht.setText("Geschlecht");

Text GeschlechtField = new Text (shell, SWT.BORDER);
GridData data = new GridData (SWT.FILL, SWT.CENTER, false, false, 1, 2);
GeschlechtField.setLayoutData (data);

shell.pack ();
Modeling the UI

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

- Domain Model Instances
- User View Model
  - UserView
    - HorizontalLayout
      - VerticalLayout Left Column
        - Control firstName
        - Control weight
        - Control nationalit
        - Control timeOfRegistration
        - Control active
      - VerticalLayout Column Right
        - Control lastName
        - Control heigth
        - Control gender
        - Control dateOfBirth
        - Control eMails

- User Interface Example
Modeling the UI

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

RCP Application

First Name
Last Name*
Gender
Active
Time Of Registration

Weight
Height
Nationality
Date Of Birth
Email*

Tasks

No data set! Click to set date.
Example: Embedding EMF Forms in a SWT UI

- EMF Forms can be embedded on every Composite
- ECPSWTViewRenderer.INSTANCE.render(Composite parent, EObject domainObject);
EMF Forms Features

• Controls to edit data + Layouts to structure UI
• IDE-Tooling
• Many additional view model elements
• Live-Validation
• Rule-based visibility and enablement of controls
• Exchangeable UI Technology:
  • SWT (Production)
  • JavaFX (Development)
  • Browser/Web based on RAP (Production)
  • Browser/Web based on AngularJS (Development)
  • Mobile based on Tabris (Demo)
Rendering technologies
When to use 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

- Next Release 1.7 Mars SR1
- Roadmap:
  - API/SPI improvements
  - Specific Ecore Edit Support
  - Work towards a new Domain Model Reference API
  - Web Renderer based on AngularJS
    => http://JSONForms.org

- EMF Forms: http://emfforms.org
  => Getting started Tutorial

- Twitter: #emfforms @EMFFForms
Evaluate this session at www.eclipsecon.org

WITH

+1 0 -1