Most of the Eclipse RCP developers need to design GUIs

Thank you
Captain Obvious!
Fortunately, the Eclipse community can help you!

The very large ... and very prolific community
Many presentation frameworks are proposed to Eclipse RCP developers...

Therefore a **choice** must be done...
Lost?

Ultimately, this presentation will help you!
This presentation is...

* A guide to assist you in choosing a presentation framework

  - Focused on the features
  - Structured with the same sample application
  - Highlighting the strengths of the frameworks
This presentation is not...

- A checklist of what the frameworks do or do not
- A comparative table of all the features you can imagine for a GUI
- A fight
First technology
Sapphire
Sapphire

Creation: October 20th, 2010
Site: http://www.eclipse.org/sapphire/
Status: Incubation
Current release: 0.7.1
Project lead: Konstantin Komissarchik (Oracle)
Outline

An GUI model

*Designed with java interfaces*

*Annotated with dedicated Sapphire annotations*

Sapphire generates the implementations of the interfaces

*With a specific compiler*

*Hooked on the JDT compiler*

Generated GUIs are rendered into Sapphire graphical hosts

*Wizards, Editors...*
Outline (II)

User doesn't take care of GUI implementations details
- Sapphire choose widgets, layouts...
- Choice based on the model annotations

GUIs are well integrated in Eclipse Workbench
- JFace wizards or dialogs
- Integration with WTP XML editor

Automatic Model ↔ GUI bidirectional mapping
- GUI data are stored into the generated model
- Sapphire annotations can be used to define XML mapping
For the sample application
But what is this sample application?

I would like...

- A dedicated Eclipse editor to edit information about conference

I would like...

- A form editor,
  - with several pages (participants, sessions, ...)

I would not like...

- Waste too much time on the GUI design of course :-(
Back to sapphire...

... I can easily create a model defining my conference

The most complicated part of this step was to set up the project...
... and then create the sapphire descriptor file

With a nice editor :-)}
Finally, after some questions on the forum...

TaDaaa, I've got my conference editor
Summary

😊 What I liked...
+ Good getting started tutorial
+ Good framework architecture
+ Good looking of the resulting GUI

😢 What I found difficult...
- Lack of a project initialization wizard
- Lack of documentation for some cool features
- Defining a model conforming to sapphire conventions could be a bit technical
Second technology

Extended Editing Framework (EEF)

Disclaimer: The speaker may be the project leader of this technology
Creation: April 8th, 2009
Site: http://www.eclipse.org/eef/
Status: Mature
Current release: 1.3.0
Project lead: Goulwen Le Fur (Obeo)
Outline

GUI modeling via EMF models
  Ecore for the business model
  View model for the graphical parts
  Binding model mapping Ecore concepts with views
Code generation from the GUI models
  Generating all GUI artefacts required for the application
  An extensible architecture to customize the default behavior
Code well integrated with Eclipse Workbench
  Easy integration with properties views, wizards, editors.
Focused on parametrization
  *User can define the widgets used, the view layout...*
  *He can also use its own specific widgets*
  *View and mapping reuse*

Easy to use
  *GUI generated with only a few clicks*
  *Incremental design process*
Back to my conference editor...

First, I must define my conference model with Ecore

Thanks to the sirius based Ecore Tools(*)

(*)Interested? Meet Cedric Wednesday @10:00 in this room
Ok, then I can use the EEF initializer to create models

Here we use the EEF Architecture Initializer
We have a second set of models to produce

We have another initializer for these models: the Editor model initializer
I'm ready to launch the EEF generators...

And here we have a second conference editor
Summary

😊 What You will like...
+ Good getting started guide
+ Extensible architecture facilitating views customisation
+ Good looking of the resulting GUI

😢 What You can find difficult...
- Lack of documentation for some cool features
- No way to use EEF without code generation in V1
About EEF v2

The EEF Team prepares a brand new version of EEF for two years now.

This fully redesigned version will be highly modular and will no longer require code generation.

The runtime will allow you to plug in renderer for Eclipse 3/4, SWT, JavaFX...
Third technology

EMF Client Platform
EMF Client Platform

Creation: August 25th, 2010
Site: http://eclipse.org/ecp/
Status: Mature
Current release: 1.2.0
Project leads: Jonas Helming (Eclipse Source) Maximilian Koegel (Eclipse Source)
But... Let's be more specific
This session deals with EMF Forms...

A subcomponent of ECP dedicated to model editing

ECP provides other components for various features:
  · Collaborative editing
  · ...

Regarding to the topic of this session, I focused on EMF Forms
Outline

EMF Forms runtime provides editing GUI for EMF models

- No configuration required
  - Default strategy defined for each Ecore concept

User can customize the editing views

- By providing a view description model
  - Customization mainly focused on the view layout
Outline (II)

This can be used directly inside ECP

*Using the provided RCP application*

*Requires to include the application in ECP*

Result can be integrated in external applications

*API provided to render an EMF element on a Composite*

*User can choose the configuration to use for rendering*
Let's make another conference editor;)

Good news, I can reuse my Ecore model!

Good news... for me:)
The default views don't match with my expectations

Let's customize them

Hopefully I've got only some classes
It's still lacking the editor

EMF Forms lets me render view on an SWT composite...

... but I want a multipage editor, with master/details for participants and sessions page

Ok, let's do some code
After some lines of code

Here we are, my third conference editor
Summary

😊 What I liked...
+ The default view production strategy
+ Good integration with EMF.edit

😊 What I found difficult...
- Configuration can be verbose (1 model/concept!)
- Limited Integration with external application
- Not a perfect graphical rendering all the time
Fourth technology

Eclipse Scout
Eclipse Scout

Creation: April 7th, 2010
Site: http://eclipse.org/scout/
Status: Mature
Current release: 3.9.0
Project leads: Andreas Hoegger (BSI)
Matthias Zimmermann (BSI)
Outline

Scout is an application development framework
Not only focused on the GUI part
Addressing other concerns like persistence, security...

GUI part is structured by outlines
An outline presents a set of data
Forms can be attached to these outlines

Scout SDK compute a model of the application
Easy GUI customization through assistants
Very traditional way to model GUI
Scout natively produce GUI with multiple renderers

    SWT, Swing, RAP...

Clients generated by Scout have the same logic

    All necessary elements for GUI alignment are provided by Scout runtime
We're reaching the limit of the topic!
Scout is more than a presentation framework

Basically, GUIs produced by Scout are intended to be used with the back ends produced by Scout

I started the conference application but I didn't led the experimentation till the end – I focused only on the client side
Example of Scout editing form

For a conference participant
Summary

😊 What I liked...
+ 1 model for multiple renderers
+ Good tutorials to start
+ Good management of i18n

😡 What I found difficult...
- Many configurations that could be difficult to leverage
- Resulting architecture is constrained by the framework
Fifth technology

Wazaabi
Creation: January 4th, 2012
Site: http://eclipse.org/wazaabi/
Status: Incubation
Current release: ???
Project lead: Olivier Moises (Generic Concept)
Due to a lack of release, the wazaabi experimentation was not performed.
Wazaabi is a project undergoing a major restructuring, I'm confident that it will be soon released With all the necessary documentation
Summary

😊 What I liked...
+ a clean site

😔 What I found difficult...
- No release
- No documentation
Recap

- The Eclipse foundation hosts many presentation frameworks
- They have different maturity levels
- The choice relies on the context of your project

*Always an eye on us*

*Captain Obvious!*
Recap (II)

- Projects that rely on EMF to handle data should use EEF or ECP
- EEF (v1) has a generative approach
  - It generates a default configuration that user customize
  - The generated GUIs are intended to be integrated into existing applications
- ECP (EMF Forms) has an interpreted approach
  - It has a default rendering strategy that user can customize sporadically
  - The GUI development is more efficient if the views are used inside the ECP platform
Recap (III)

Projects out of the EMF world can use Sapphire or Scout.

Sapphire uses a high level GUI model.
→ It generates code and takes the responsibility of the technical details.
→ It is also intended to be included into existing (Eclipse) applications.

Scout is a global application framework.
→ It has a very technical model.
→ It manages all the aspects of an application.
Thanks!

Goulwen Le Fur – goulwen.lefur@obeo.fr
Evaluate This Session

1. Sign-in: www.eclipsecon.org

2. Select session from schedule

3. Evaluate: +1 0 -1
Credentials

Titlepage: Sennett, Tomas
P5: Lori Greig - http://www.flickr.com/photos/39585662@N00/5331407245/
P8: http://en.wikipedia.org/wiki/Yogo_sapphire
P12: Cellular Immunity - http://www.flickr.com/photos/58534808@N00/323527111/
P18: arquera - http://www.flickr.com/photos/34145688@N00/90120985/
P28: mike baird - http://www.flickr.com/photos/72825507@N00/3898808431
P30: Kaysse - http://www.flickr.com/photos/29862505@N08/3129064412/
P39: Ron Cogswell - http://www.flickr.com/photos/22711505@N05/7444596326/
P43: joiseyshowaa- http://www.flickr.com/photos/30201239@N00/2179747557/
P47: http://pixabay.com/fr/wasabinuesse-%C3%A9crous-wasabi-noyaux-141544/
P49: Alan Klim - http://www.flickr.com/photos/61203681@N08/8149383268/
P52: eamoncurry123 - http://www.flickr.com/photos/7159257@N04/6226070528/

Icons at pages 1, 3, 6, 7, 13, 17, 26, 38, 46, 51, 53, 54 and 55 come from The noun project: http://www.thenounproject.org

Thanks to Gabriele Fumero, Katie M Westbrook, Roger Cline, Quan Do, Ilsur Aptukov, Austin Condiff, Brennan Novak, Brian Gonzalez, Sarah Cavins and SuperAtic LABS.