



Best Practices For Efficient Development Of JavaFX Applications

Grzegorz Kruk, CERN, Beams Department, Controls Group

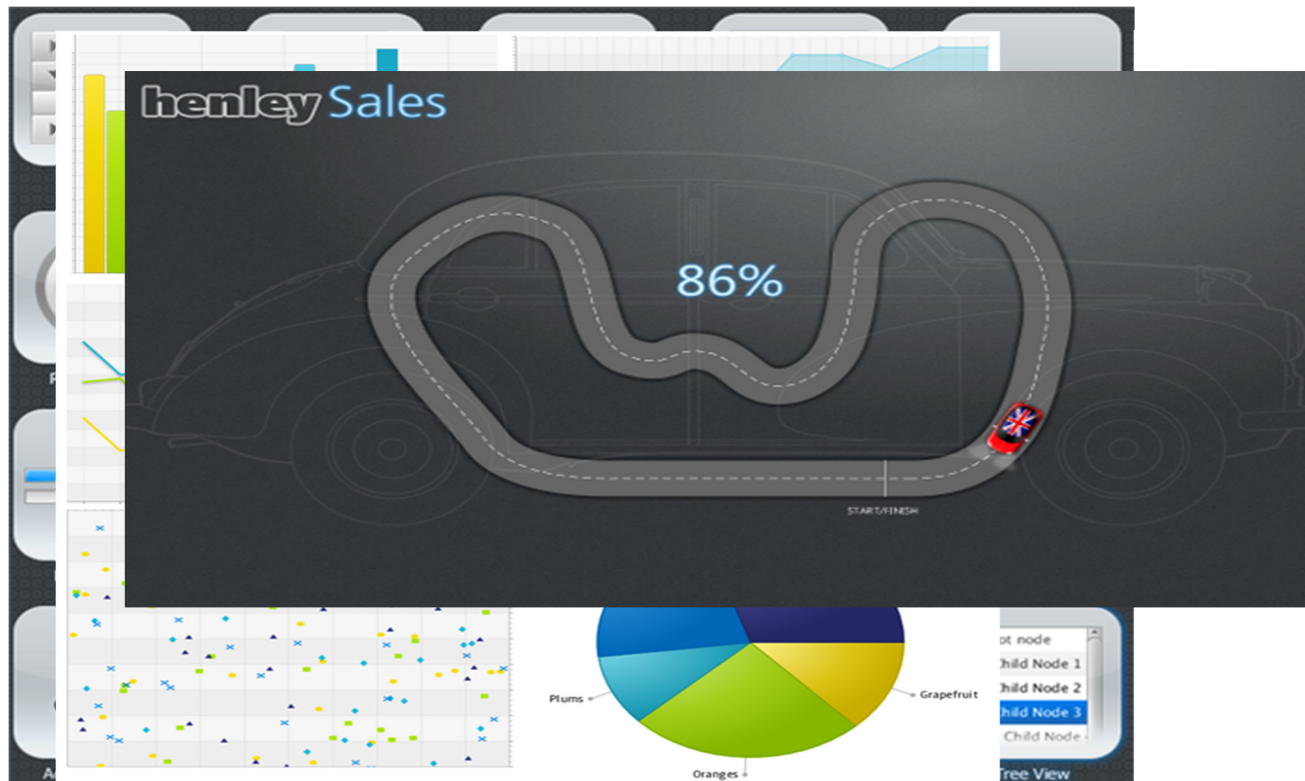
What is JavaFX?



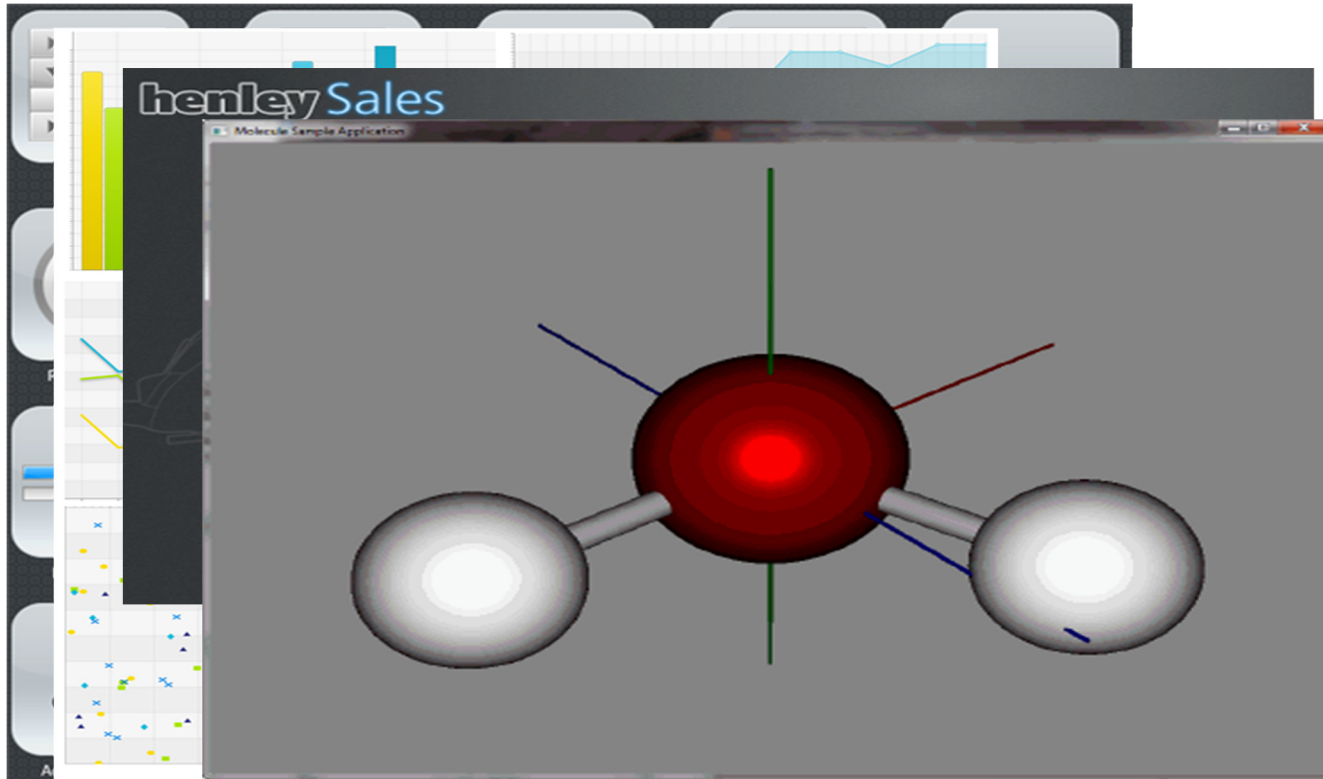
What is JavaFX?



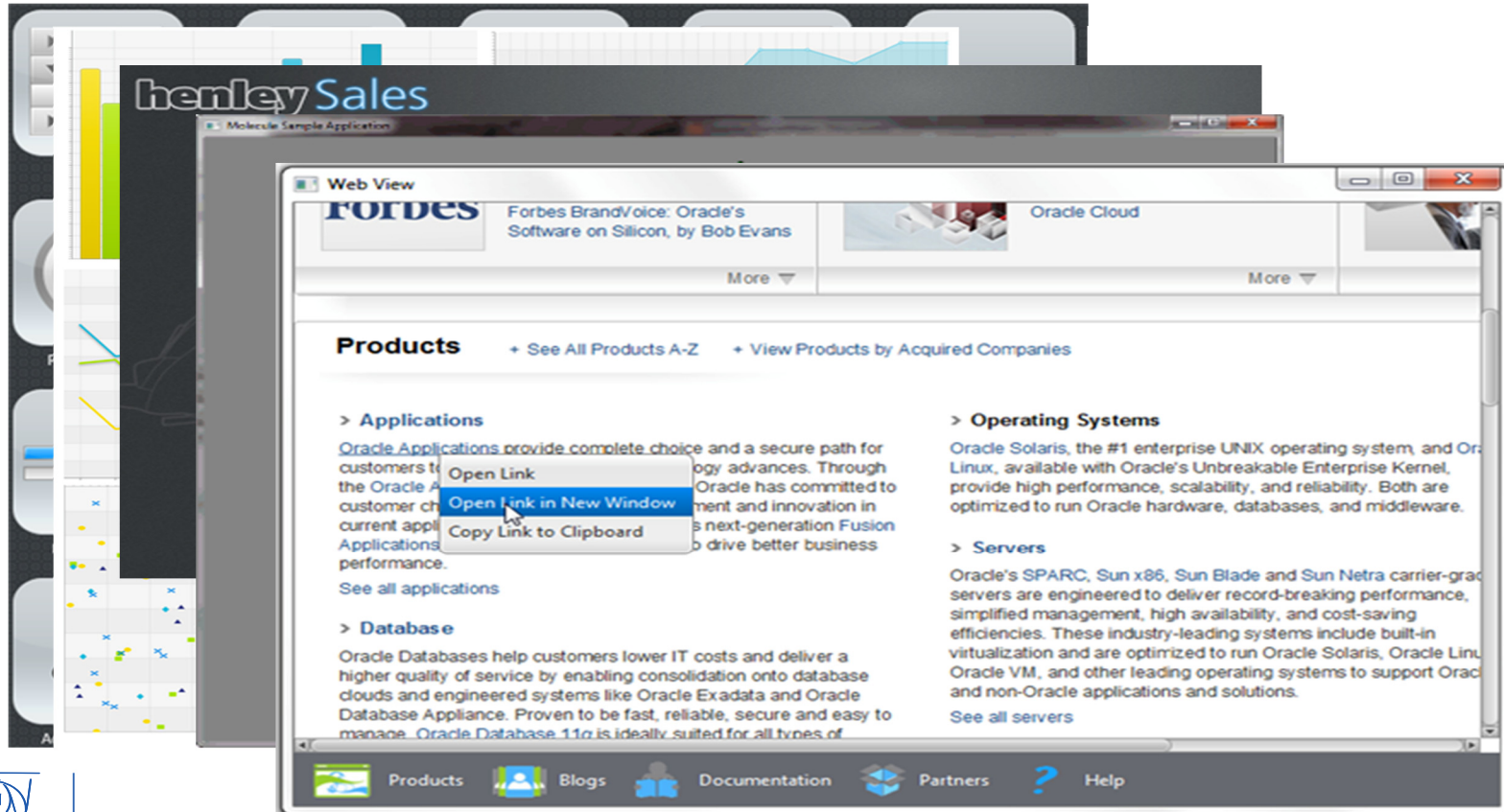
What is JavaFX?



What is JavaFX?



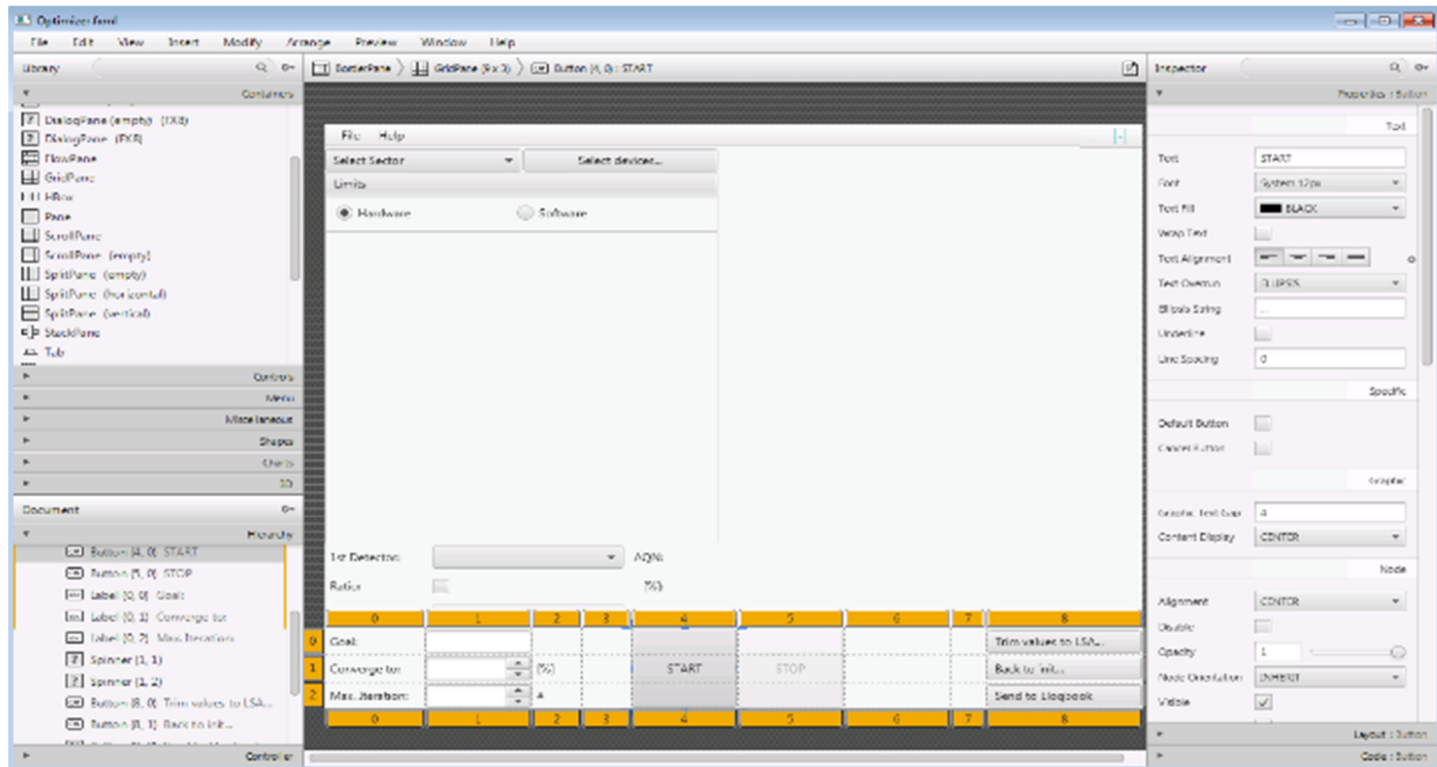
What is JavaFX?



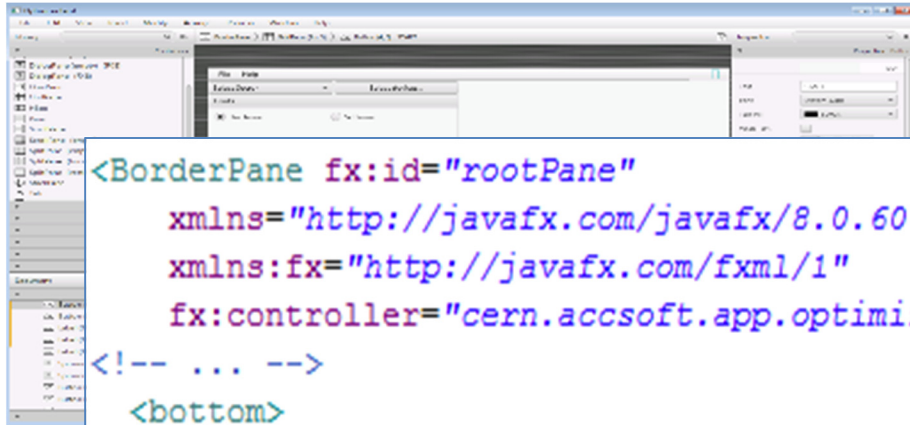
What is JavaFX?



Separation Of Concerns



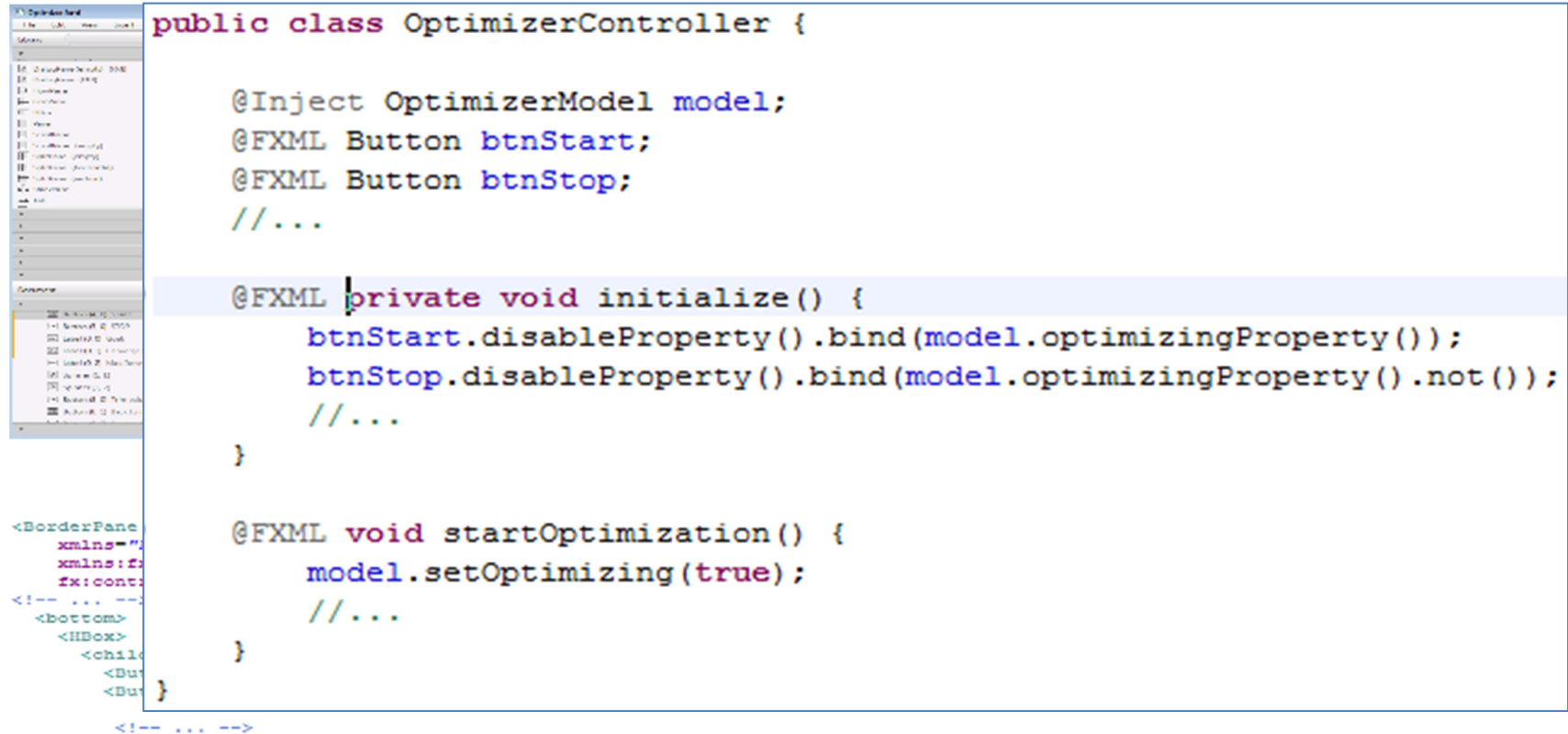
Separation Of Concerns



```
<BorderPane fx:id="rootPane"
  xmlns="http://javafx.com/javafx/8.0.60"
  xmlns:fx="http://javafx.com/fxml/1"
  fx:controller="cern.accsoft.app.optimizer.OptimizerController"
<!-- ... -->
<bottom>
  <HBox>
    <children>
      <Button fx:id="btnStart" text="START" onAction="#startOptimization"/>
      <Button fx:id="btnStop" text="STOP" onAction="#stopOptimization"/>

    <!-- ... -->
  </children>
</HBox>
</bottom>
</BorderPane>
```

Separation Of Concerns



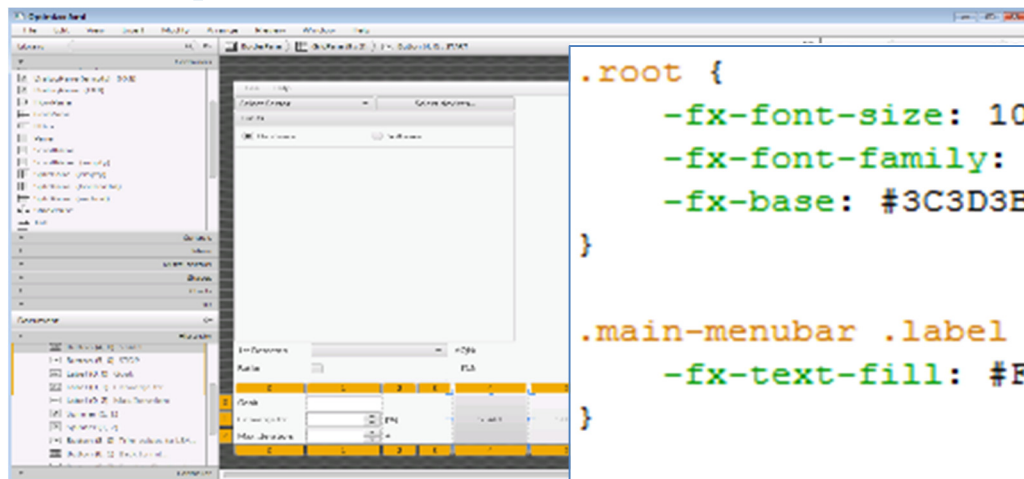
The screenshot shows an IDE with two panels. The top panel displays the source code of the `OptimizerController` class, which is a JavaFX controller. The bottom panel shows an XML snippet for a `BorderPane` layout, which is part of the FXML file associated with the controller. The code in the top panel is as follows:

```
public class OptimizerController {  
  
    @Inject OptimizerModel model;  
    @FXML Button btnStart;  
    @FXML Button btnStop;  
    //...  
  
    @FXML private void initialize() {  
        btnStart.disableProperty().bind(model.optimizingProperty());  
        btnStop.disableProperty().bind(model.optimizingProperty().not());  
        //...  
    }  
  
    @FXML void startOptimization() {  
        model.setOptimizing(true);  
        //...  
    }  
}
```

The XML snippet in the bottom panel is as follows:

```
<BorderPane  
    xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml" fx:controller="OptimizerController">  
    <!-- ... -->  
    <bottom>  
        <HBox>  
            <children>  
                <Button fx:id="btnStart" text="Start" />  
                <Button fx:id="btnStop" text="Stop" />  
            </children>  
        </HBox>  
    </bottom>  
</BorderPane>
```

Separation Of Concerns



```
<BorderPane fx:id="rootPane"
  xmlns="http://javafx.com/javafx/8.0.60"
  xmlns:fx="http://javafx.com/fxml/1"
  fx:controller="cern.acsoft.app.optimized"
  <!-- ... -->
  <bottom>
    <HBox>
      <children>
        <Button fx:id="btnStart" text="START" />
        <Button fx:id="btnStop" text="STOP" />
      </children>
    </HBox>
  </bottom>
  <!-- ... -->
```

```
.root {
    -fx-font-size: 10.0pt;
    -fx-font-family: "Carlito";
    -fx-base: #3C3D3B;
}

.main-menubar .label {
    -fx-text-fill: #FFFFFF;
}

.radio-button {
    -fx-text-fill: #FFFFFF;
}

.about-style > *.button-bar {
    -fx-background-color: black;
}
```

```
roller {
    model;

    initialize() {
        property().bind(model.optimizingProperty());
        property().bind(model.optimizingProperty().not());
    }

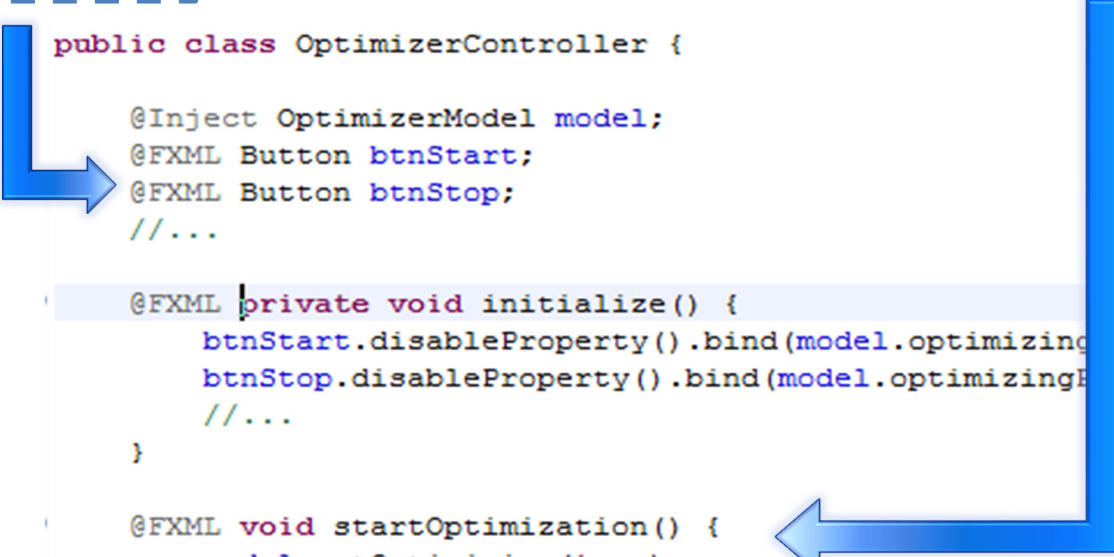
    action() {
        if (true) {

```

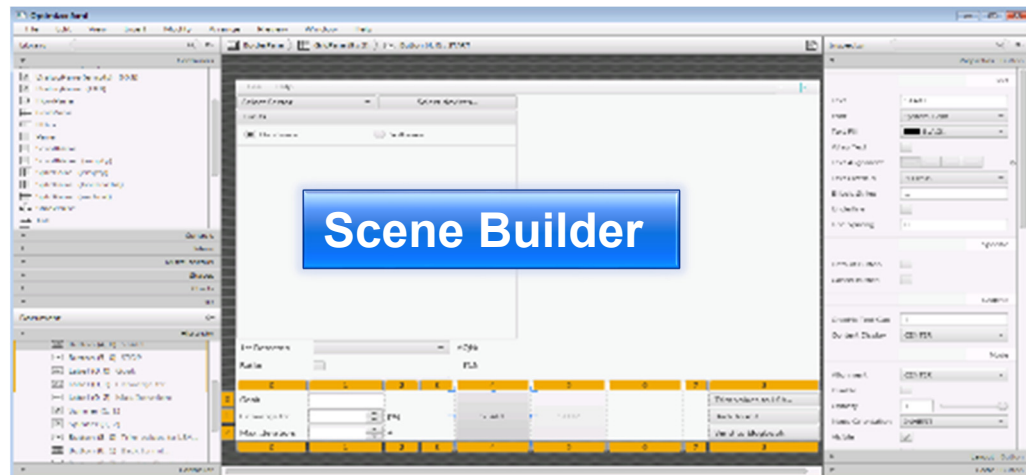
Separation Of Concerns

```
<Button fx:id="btnStart" text="START" onAction="#startOptimization"/>  
<Button fx:id="btnStop" text="STOP" onAction="#stopOptimization"/>
```

```
public class OptimizerController {  
  
    @Inject OptimizerModel model;  
    @FXML Button btnStart;  
    @FXML Button btnStop;  
    //...  
  
    @FXML private void initialize() {  
        btnStart.disableProperty().bind(model.optimizingProperty());  
        btnStop.disableProperty().bind(model.optimizingProperty().not());  
        //...  
    }  
  
    @FXML void startOptimization() {  
        model.setOptimizing(true);  
        //...  
    }  
}
```



Separation Of Concerns



```
<BorderPane fx:id="rootPane"
  xmlns="http://javafx.com/javafx/8.0.60"
  xmlns:fx="http://javafx.com/fxml/1"
  fx:controller="com.example.optimizer.OptimizerController">
  <!-- ... -->
  <bottom>
    <HBox>
      <children>
        <Button fx:id="btnStart" text="START" onAction="#startOptimization"/>
        <Button fx:id="btnStop" text="STOP" onAction="#stopOptimization"/>
      </children>
    </HBox>
  </bottom>
</!-- ... -->
```

FXML

```
.root {
  -fx-font-size: 10.0pt;
  -fx-font-family: "Carlito";
  -fx-base: #3C3D3B;
}

.main-menubar .label {
  -fx-text-fill: #FFFFFF;
}

.about-style > *.button-bar {
  -fx-background-color: black;
}

.button {
  -fx-font-family: "Carlito";
  -fx-base: #64C8C8;
}
```

CSS

```
public class OptimizerController {
```

```
    @Inject OptimizerModel model;
    @FXML Button btnStart;
    @FXML Button btnStop;
    //...
```

```
    @FXML private
    btnStart
    btnStop
    //...
```

```
    @FXML void startOptimization() {
        model.setOptimizing(true);
        //...
    }
}
```

Controller

Separation Of Concerns

MINI5P

Hardware

BR2.DH250

MIN: -10.0
CCV: -2.17 AON: -2.17
MAX: 10.0

BR2.DH270

MIN: -10.0
CCV: 0.99 AON: 0.00
MAX: 10.0

BR2.DVT50

MIN: -10.0
CCV: -0.76 AON: -0.76
MAX: 10.0

BR2.DVT70

MIN: -10.0
CCV: 0.92 AON: 0.91
MAX: 10.0

1st Detector: BR2.DH250 AON: 16.4.60...
Ratio: [] [%]
2nd Detector: BR2.DH270 AON: []
Goal: 160 E10
Converge to: 10 [%]
Max. Iteration: 100 #

START STOP

Trim values to USA...
Back to Init...
Send to Flightbook

Converged successfully - Optimization finished.

Current [A]

Time

DR2.DVT10

Time

size: 10.0pt;
family: "Carlito";
#3C3D3B;

.label {
fill: #FFFFFF;
S

.button-bar {
round-color: black;

family: "Carlito";
#64C8C8;

ler

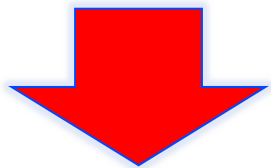
mixingProperty():
izingProperty().not();

Courtesy of E.Piselli and A.Akroh: TUPHA120

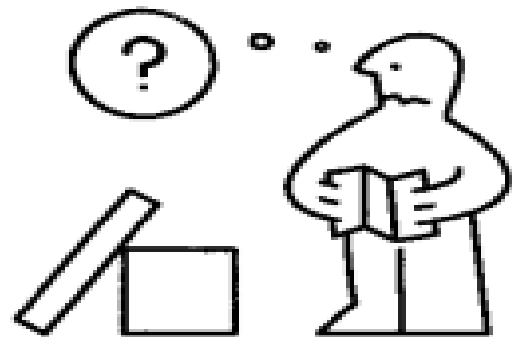
12/10/2017

Architectural Guidelines Missing

- Every team/developer works out its own style
- Every application has its own structure



- x Sharing development
- x Sharing components
- x Applications take over
- x Maintenance



How to deal with this?

Structure

+

Tools

Project Structure



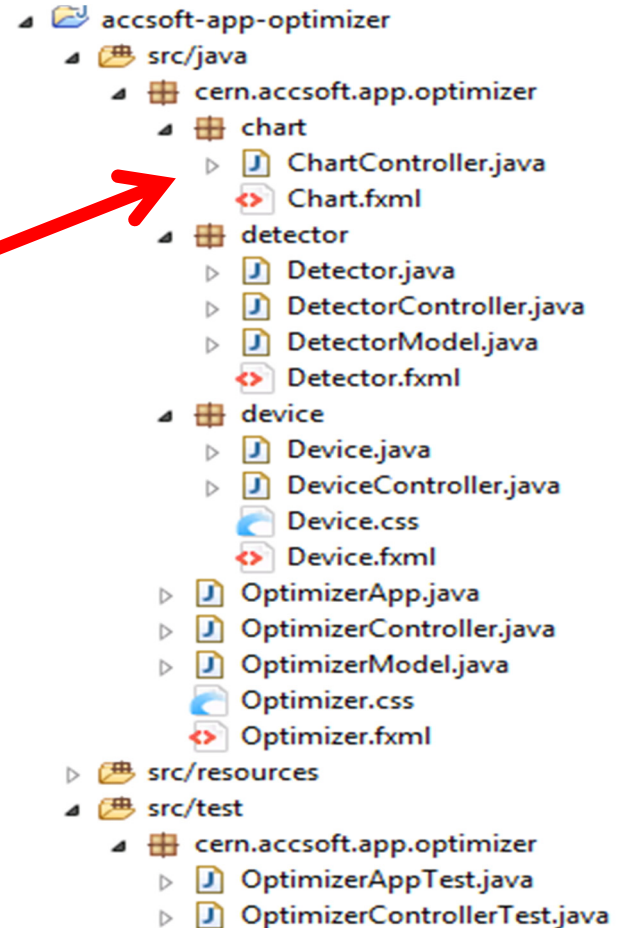
- accsoft-app-optimizer
 - src/java
 - cern.accsoft.app.optimizer
 - chart
 - ChartController.java
 - Chart.fxml
 - detector
 - Detector.java
 - DetectorController.java
 - DetectorModel.java
 - Detector.fxml
 - device
 - Device.java
 - DeviceController.java
 - Device.css
 - Device.fxml
 - OptimizerApp.java
 - OptimizerController.java
 - OptimizerModel.java
 - Optimizer.css
 - Optimizer.fxml
 - src/resources
 - src/test
 - cern.accsoft.app.optimizer
 - OptimizerAppTest.java
 - OptimizerControllerTest.java

Project Structure

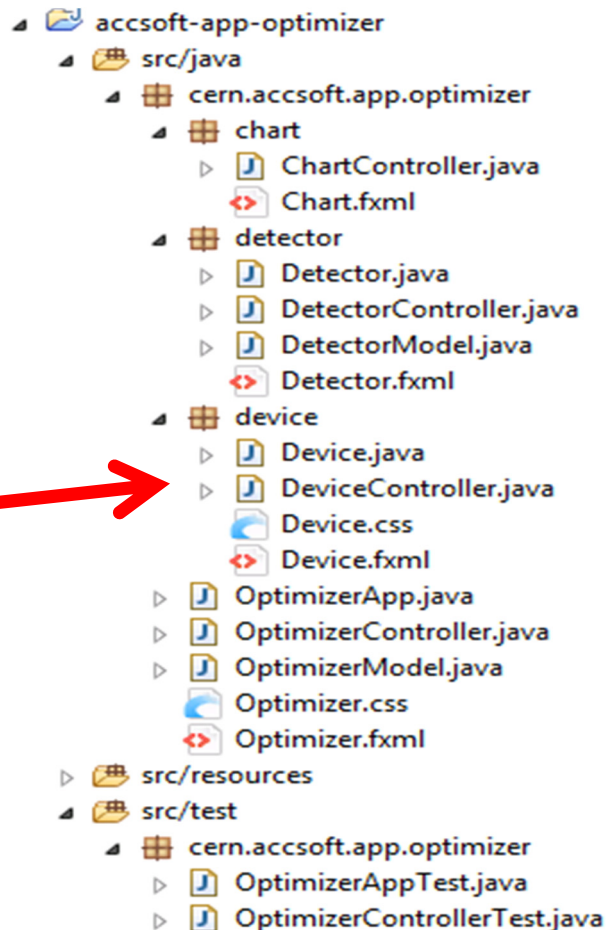


- accsoft-app-optimizer
 - src/java
 - cern.accsoft.app.optimizer
 - chart
 - ChartController.java
 - Chart.fxml
 - detector
 - Detector.java
 - DetectorController.java
 - DetectorModel.java
 - Detector.fxml
 - device
 - Device.java
 - DeviceController.java
 - Device.css
 - Device.fxml
 - OptimizerApp.java
 - OptimizerController.java
 - OptimizerModel.java
 - Optimizer.css
 - Optimizer.fxml
 - src/resources
 - src/test
 - cern.accsoft.app.optimizer
 - OptimizerAppTest.java
 - OptimizerControllerTest.java

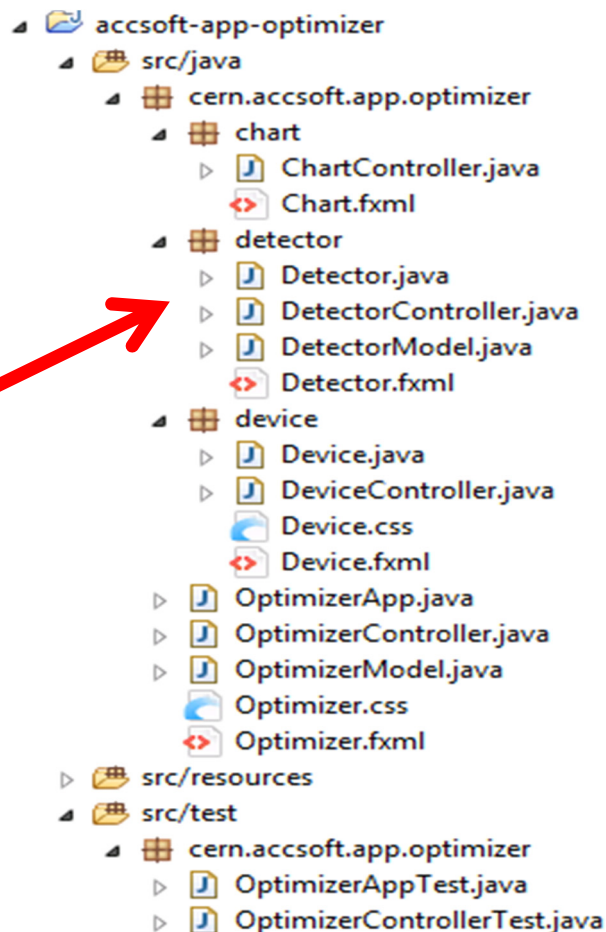
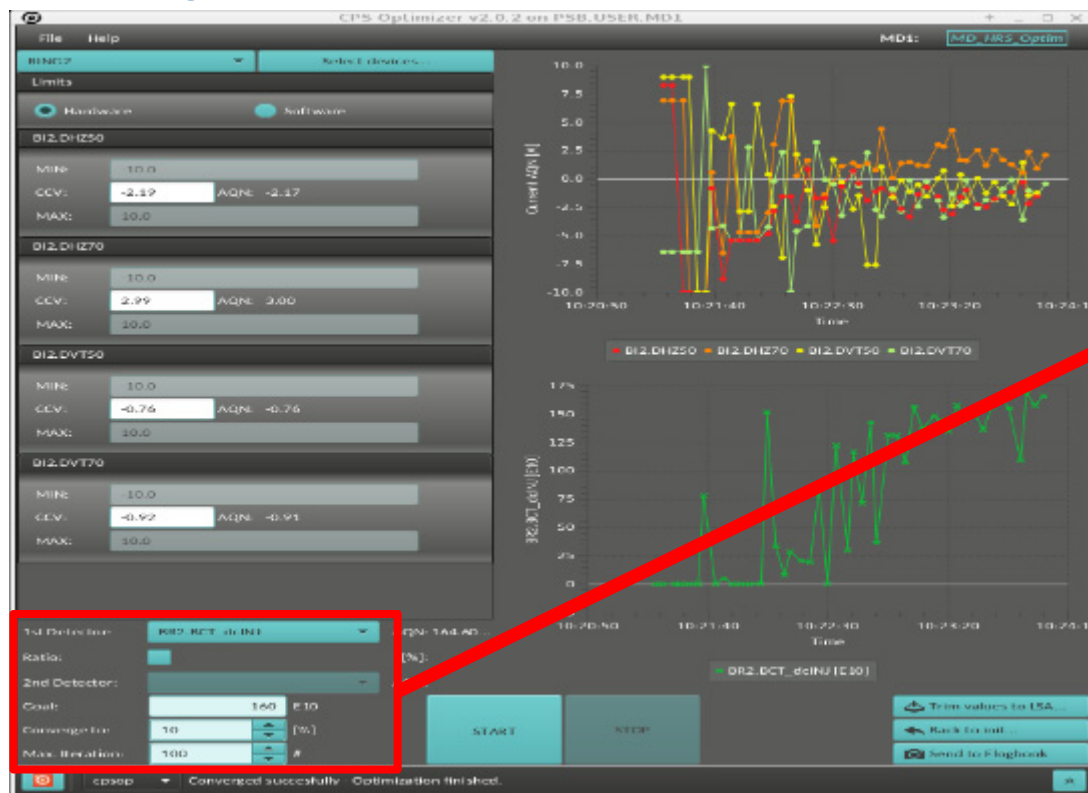
Project Structure



Project Structure

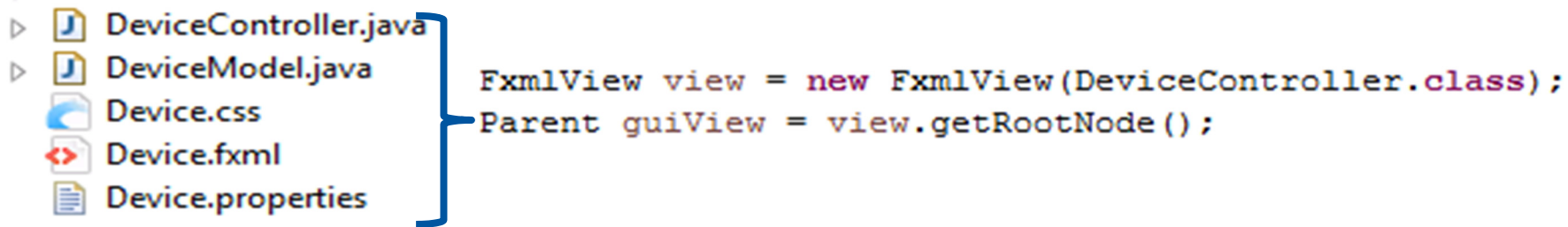


Project Structure



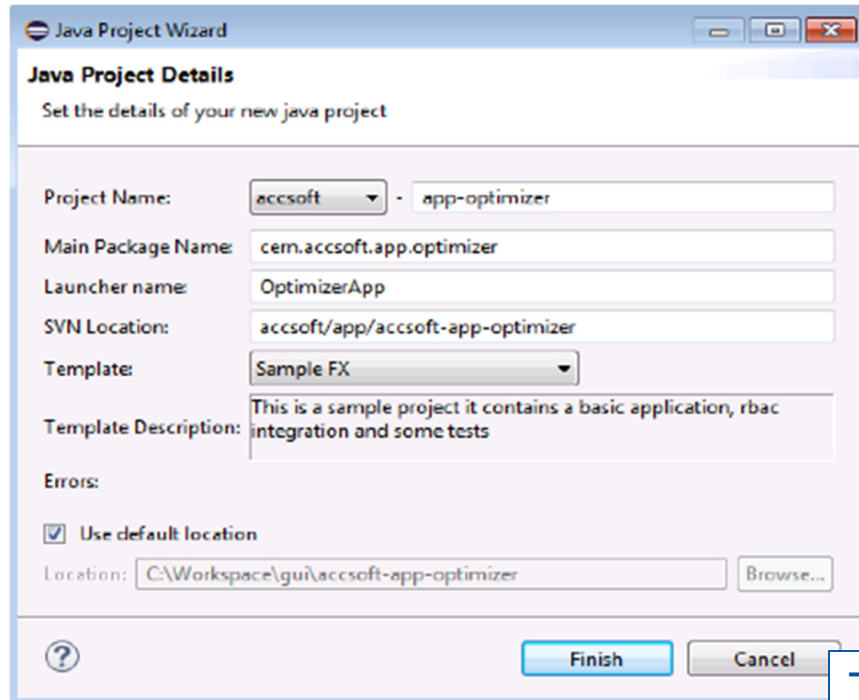
FxmlView

- Convention over Configuration
 - Inspired by Afterburner.fx library



- Small and simple to use
- Simplifies instantiation of views
- Built-in *@Inject* Dependency Injection (DI)
 - Models, Services, Properties
 - Plug-in external DI framework e.g. Spring or Google Guice

Project Generator



Java Project Wizard

Set the details of your new java project

Project Name: accsoft - app-optimizer

Main Package Name: cern.accsoft.app.optimizer

Launcher name: OptimizerApp

SVN Location: accsoft/app/accsoft-app-optimizer

Template: Sample FX

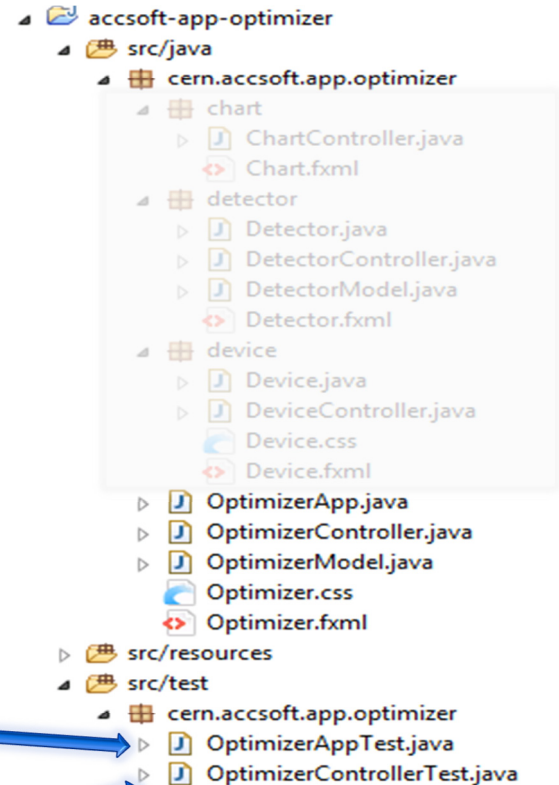
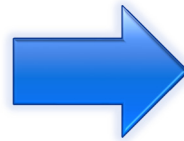
Template Description: This is a sample project it contains a basic application, rbac integration and some tests

Errors:

☒ Use default location

Location: C:\Workspace\gui\accsoft-app-optimizer

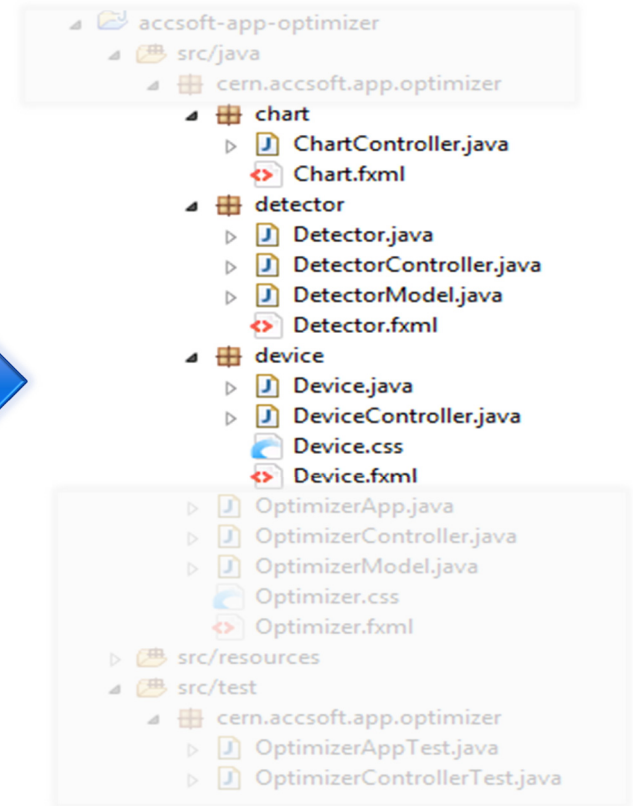
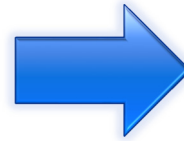
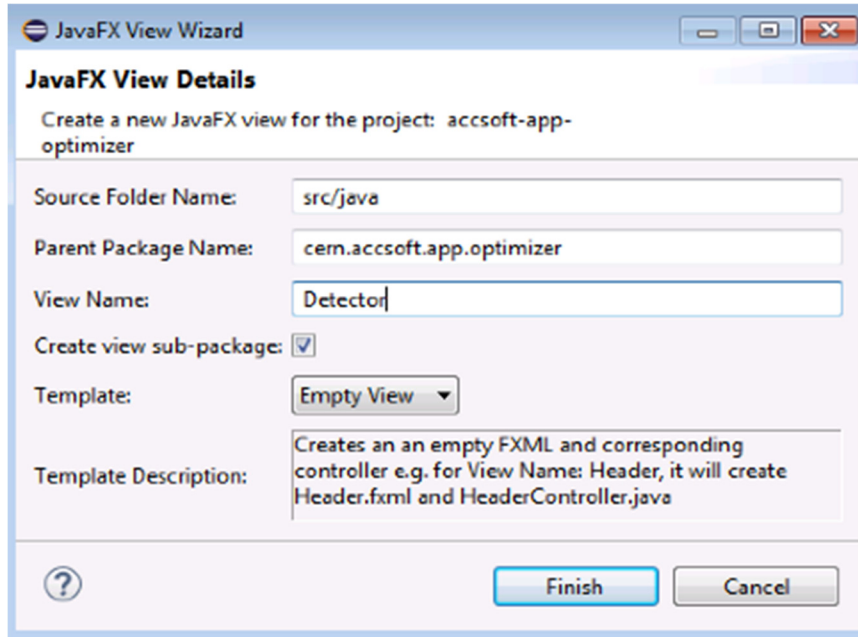
Finish Cancel



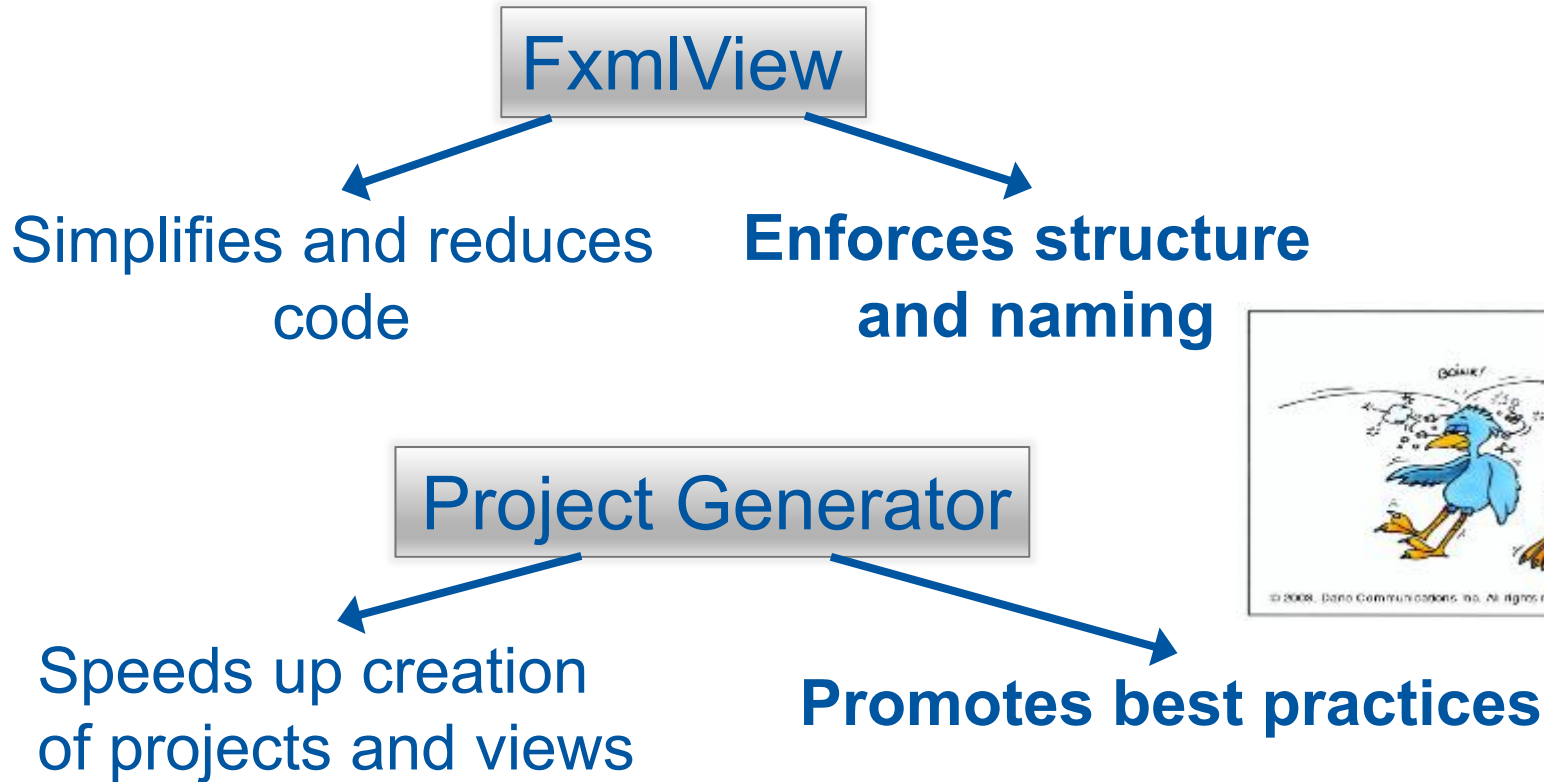
TestFX Test

JUnit Test

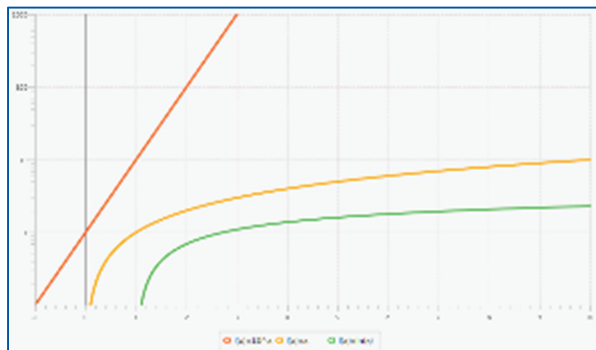
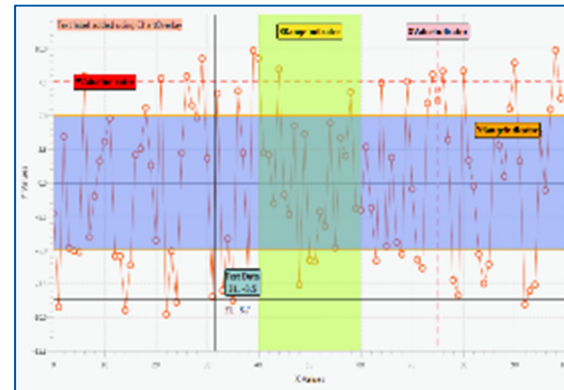
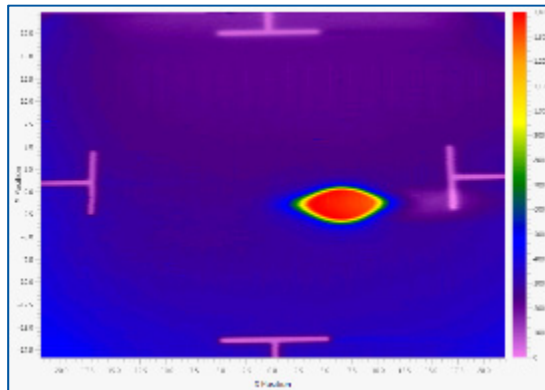
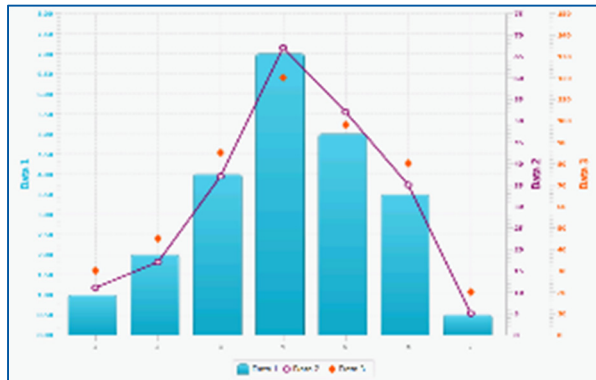
View Generator



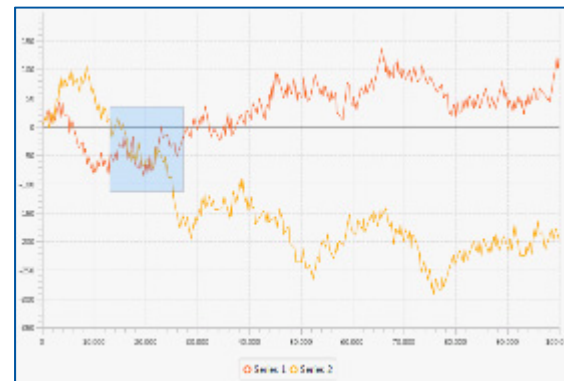
Twofold Advantage



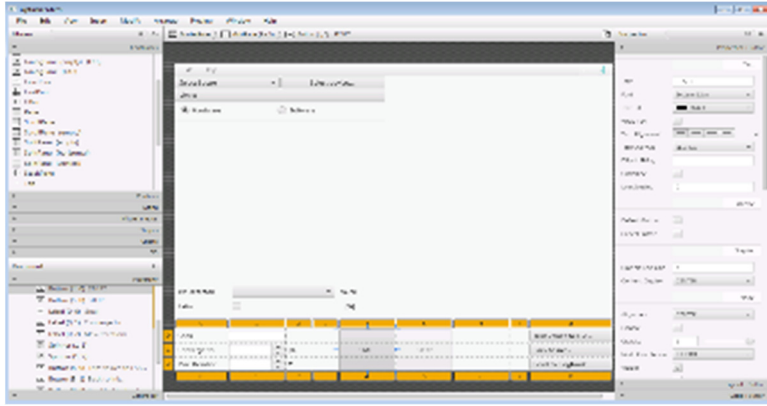
JavaFX Charting Extensions



TUPHA186



Keep FXML and Controller in Sync



```
<Button fx:id="btnStart" text="START" onAction="#startOptimization"/>
<Button fx:id="btnStop" text="STOP" onAction="#stopOptimization"/>
```

```
public class OptimizerController {
    @Inject OptimizerModel model;

    @FXML Button btnStart;
    @FXML Button btnStop;

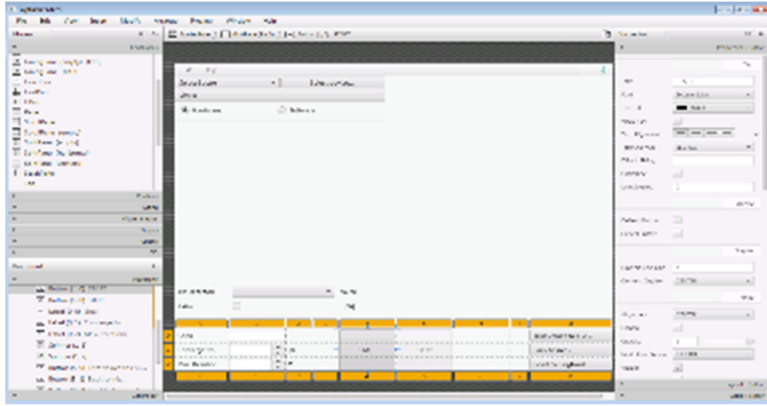
    @FXML void startOptimization() {
        model.setOptimizing(true);
        //...
    }

    @FXML void stopOptimization() {
        model.setOptimizing(false);
        //...
    }

    //...
}
```



Keep FXML and Controller in Sync



```
<Button fx:id="btnStart" text="START" onAction="#startOptimization"/>
<Button fx:id="btnStop" text="STOP" onAction="#stopOptimization"/>
<Button fx:id="btnRefresh" text="Refresh" onAction="#refreshView"/>
```

```
public class OptimizerController {
    @Inject OptimizerModel model;

    @FXML Button btnStart;
    @FXML Button btnStop;

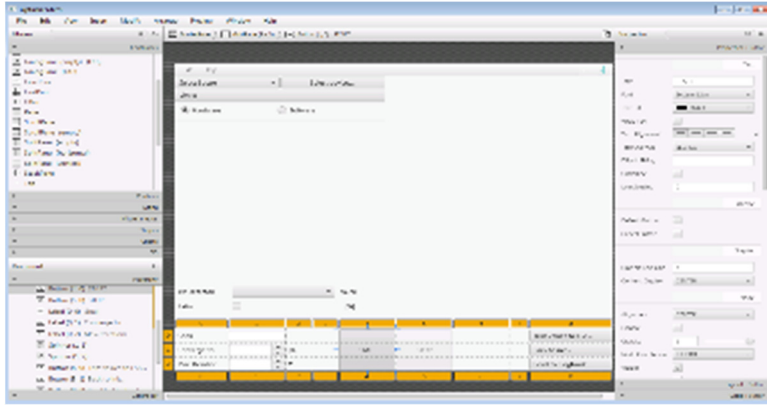
    @FXML void startOptimization() {
        model.setOptimizing(true);
        //...
    }

    @FXML void stopOptimization() {
        model.setOptimizing(false);
        //...
    }

    //...
}
```



Keep FXML and Controller in Sync



```
<Button fx:id="btnStart" text="START" onAction="#startOptimization"/>
<Button fx:id="btnStop" text="STOP" onAction="#stopOptimization"/>
<Button fx:id="btnRefresh" text="Refresh" onAction="#refreshView"/>
```



Generate Missing FXML Fields and Handlers

```
public class OptimizerController {
    @Inject OptimizerModel model;

    @FXML Button btnStart;
    @FXML Button btnStop;

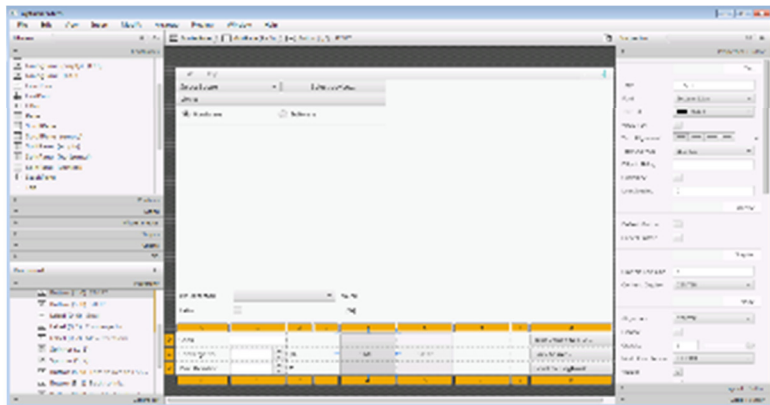
    @FXML void startOptimization() {
        model.setOptimizing(true);
        //...
    }

    @FXML void stopOptimization() {
        model.setOptimizing(false);
        //...
    }

    //...
}
```



Keep FXML and Controller in Sync



```
<Button fx:id="btnStart" text="START" onAction="#startOptimization"/>
<Button fx:id="btnStop" text="STOP" onAction="#stopOptimization"/>
<Button fx:id="btnRefresh" text="Refresh" onAction="#refreshView"/>
```



Generate Missing FXML Fields and Handlers



```
public class OptimizerController {
    @Inject OptimizerModel model;

    @FXML Button btnStart;
    @FXML Button btnStop;
    @FXML Button btnRefresh;

    @FXML void startOptimization() {
        model.setOptimizing(true);
        //...
    }

    @FXML void stopOptimization() {
        model.setOptimizing(false);
        //...
    }

    @FXML void refreshView() {
        //...
    }
    //...
}
```



Summary

JavaFX: rich, mature and productive

...but doesn't provide structural guidelines

Convention
Tools
Templates



- ✓ Don't waste time on repeatable things
- ✓ Follow examples
- ✓ Focus on specific logic