

Functional Testing Swing User Interfaces with Frankenstein

Vivek Prahlad
Senior Developer, ThoughtWorks

Agenda

- Why a new tool?
- Introduction to Frankenstein
- Demo
- Q & A

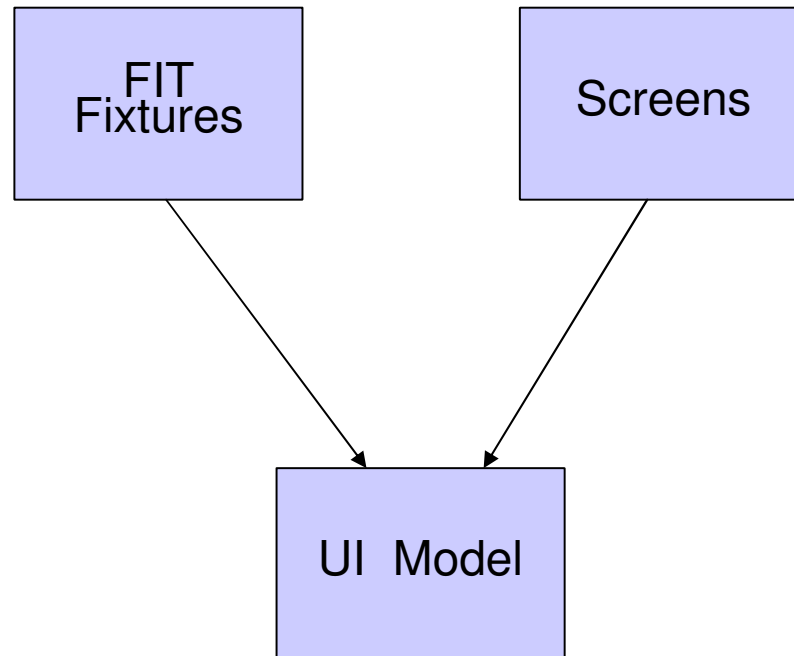
Why?

Swing Client, pre-existing
FIT scripts

Needed to write fixtures for
new screens

Did not test actual UI

Tested UI model layer



No Swing driver

Duplication

Enter UserName foo

Enter Password bar

Initiate Login

Navigate Create>Blog Entry

...

Enter UserName foo

Enter Password bar

Initiate Login

Navigate Edit>Blog Entry

...

Not easy to create fixtures

Writing fixtures slowed down
development

Additional features

Record & Play

Multithreaded apps?

Swing is single threaded

but

*Serious Swing apps use
worker threads*



Project - com.thoughtworks.frankenstien.record...

Project Packages

- DefaultRecorder
- DialogRecorder
- EventList
- EventRecorder
- FrankensteinListener
- InternalFrameRecorder
- KeystrokeRecorder
- ListSelectionRecorder
- MenuNavigationRecorder
- RadioButtonRecorder
- Recorder
- SelectDropDownRecorder
- SelectTreeRecorder
- TableEditRecorder
- TabSwitchRecorder
- TextFieldRecorder
- WindowActivationRecorder

script

ui

test

- build.xml
- build.zip
- close_window
- cobertura.ser

```

11
12  /**
13   * Understands recording tree selection events.
14   */
15  public class SelectTreeRecorder extends AbstractComponentRecorder implements TreeSelectionListe
16
17      public SelectTreeRecorder(EventRecorder recorder, NamingStrategy namingStrategy) {
18          super(recorder, namingStrategy, JTree.class);
19      }
20
21  void componentShown(Component component) {
22      tree(component).addTreeSelectionListener(this);
23  }
24
25  private JTree tree(Component component) {
26      return (JTree) component;
27  }
28
29  void componentHidden(Component component) {
30      tree(component).removeTreeSelectionListener(this);
31  }
32
33  public void valueChanged(TreeSelectionEvent e) {
34      TreePath path = e.getNewLeadSelectionPath();
35      if (path == null) return;
36      recorder.record(new SelectTreeEvent(componentName((Component) e.getSource()), treePath

```

Run - All tests

Done: 3/17 of 317(5.162 s)

com.thoughtworks.frankenstien

All Tests Passed

Output

```

/opt/sun-jdk-1.5.0.06/bin/java -Dawt.headless=true -Didea.launcher.port=7541 -Didea.launcher.bin.path=/usr/local
52 test classes found in package "com.thoughtworks.frankenstien"

Playing: SelectDropDownEvent: Combo: comboName2, Choice: choice2

Process finished with exit code 0

```

All tests Run TODO

The screenshot displays the LimeWire application window. The title bar reads "LimeWire" and the menu bar includes "File", "Navigation", "Tools", and "Help". Below the menu bar are navigation buttons for "Search", "Monitor", "Connections", "Community", and "Library".

The main area is titled "Connected Hosts" and contains a table with the following data:

Host	Status	Messages (I/O)	Bandwidth (I/O)	Dropped (I/O)	Hosts
63.227.23.53:6346	Outgoing	3251 / 2789	0.974 / 0.272 KB/s	20% / 0%	26
169.237.41.142:6346	Outgoing	5108 / 3446	2.028 / 0.671 KB/s	34% / 0%	370
149.119.3.237:110	Outgoing	12684 / 9806	0.943 / 0.455 KB/s	14% / 0%	450
128.2.53.200:6346	Outgoing	11908 / 8256	0.784 / 1.004 KB/s	12% / 0%	547

Below the table are controls for "Remove" and "Add" (with a text input field containing "6346"). A status line indicates "Keep 4 connections up, with no more than 2 incoming" and a "Freeloaders..." button.

The "AutoConnect Hosts" section has an input field and "Add" and "Remove" buttons. The list below contains:

- router.limewire.com:6346
- gnutellahosts.com:6346

The "Discovered Hosts" section has a list of IP addresses and ports, with "Connect", "Remove", "Get More", and "Clear" buttons:

- 216.46.80.253:6346
- 62.4.252.87:6346
- 195.30.1.44:6346
- 158.130.65.144:6346
- 128.253.120.150:3256
- 141.217.66.171:6346

A checkbox is checked: "Prefer AutoConnect hosts instead of discovered hosts (recommended)".

The bottom status bar shows a green status indicator, a "Disconnect" button, and the text "1K hosts / 296K files / 16 TB available". It also includes "Sharing 21 files" and a "Feedback..." button.

What 's the problem?

Tests with arbitrary waits


```
click_button("connect")  
wait(20) //Pray!  
do_something_else
```

Fragile, slow tests

Explicit Synchronization

```
wait_for_text("done")
```

Verbose tests

```
upload_file("file")  
wait_for_text("done")
```

Obscures intent

Custom components?

Evaluated Abbot, Marathon

Didn't quite have what we'd
wanted

'Let's write it!'
-Chris Stevenson

Introducing Frankenstein



FIT like scripts

Navigate Create>Blog

EnterText post *my blog text*

SelectDropDown time now

ClickButton Post

Plain text

Easy to write,
Human Readable





Understood (and shared) by
the whole team

Record & Playback

..useful for reproducing
bugs

No arbitrary waits

Works with JDK 1.4+

Code changes?

Name components with
`component.setName()`

Works with applications
where codebase cannot be
modified

...with a few limitations

Not easy record custom
components

i18n

Test multithreaded apps via
WorkerThreadMonitor

```
public interface WorkerThreadMonitor {  
    public void start();  
    public void waitForIdle(); //blocking  
}
```




Project - com.thoughtworks.frankenstien.record...

Project Packages

- DefaultRecorder
- DialogRecorder
- EventList
- EventRecorder
- FrankensteinListener
- InternalFrameRecorder
- KeystrokeRecorder
- ListSelectionRecorder
- MenuNavigationRecorder
- RadiobuttonRecorder
- Recorder
- SelectDropDownRecorder
- SelectTreeRecorder
- TableEditRecorder
- TabSwitchRecorder
- TextFieldRecorder
- WindowActivationRecorder

script

ui

test

- build.xml
- build.zip
- close_window
- cobertura.ser

```
DefaultWindowContext.java SocketListener.java CheckBoxRecorder.java CheckTextRecorder.java TabSwitchRecorder.java
SelectTreeRecorder.java SimpleWorkerThreadMonitor.java DefaultRecorder.java EventList.java SocketListenerTest.java

11
12 /**
13  * Understands recording tree selection events.
14  */
15 public class SelectTreeRecorder extends AbstractComponentRecorder implements TreeSelectionList
16
17     public SelectTreeRecorder(EventRecorder recorder, NamingStrategy namingStrategy) {
18         super(recorder, namingStrategy, JTree.class);
19     }
20
21     void componentShown(Component component) {
22         tree(component).addTreeSelectionListener(this);
23     }
24
25     private JTree tree(Component component) {
26         return (JTree) component;
27     }
28
29     void componentHidden(Component component) {
30         tree(component).removeTreeSelectionListener(this);
31     }
32
33     public void valueChanged(TreeSelectionEvent e) {
34         TreePath path = e.getNewLeadSelectionPath();
35         if (path == null) return;
36         recorder.record(new SelectTreeEvent(componentName((Component) e.getSource()), treePath
37
```



Run - All tests

Done: 317 of 317(5.162 s)

com.thoughtworks.frankenstien

- All Tests Passed

Output Statistics

```
/opt/sun-jdk-1.5.0.06/bin/java -Dawt.headless=true -Didea.launcher.port=7541 -Didea.launcher.bin.path=/usr/local
52 test classes found in package "com.thoughtworks.frankenstien"

Playing: SelectDropDownEvent: Combo: comboName2, Choice: choice2

Process finished with exit code 0
```

How is the
WorkerThreadMonitor used?

Wait for all Swing events to
be completed

Wait for all worker threads
to complete

Easy to extend test
vocabulary

Add assertions

Easy integration

```
java com.yourapp.MainClass args...
```



```
java  
com.thoughtworks.frankenstein.application.PipingMain  
com.yourapp.MainClass args...
```

```
..or write your own main  
class
```

Custom recorders for custom components

Ruby driver

Variables, functions

Keep tests DRY

Other language drivers easy
to add

~ 1500 LOC

Packages

- [All](#)
- [com.thoughtworks.frankenstien.application](#)
- [com.thoughtworks.frankenstien.common](#)
- [com.thoughtworks.frankenstien.events](#)
- [com.thoughtworks.frankenstien.naming](#)
- [com.thoughtworks.frankenstien.playback](#)
- [com.thoughtworks.frankenstien.recorders](#)
- [com.thoughtworks.frankenstien.script](#)
- [com.thoughtworks.frankenstien.ui](#)

Coverage Report - All Packages

Package /	# Classes	Line Coverage	Branch Coverage	Complexity
All Packages	88	98%	100%	1.427
com.thoughtworks.frankenstien.application	9	92%	100%	1.326
com.thoughtworks.frankenstien.common	6	100%	100%	2.5
com.thoughtworks.frankenstien.events	23	99%	100%	1.387
com.thoughtworks.frankenstien.naming	11	100%	100%	1.552
com.thoughtworks.frankenstien.playback	4	99%	100%	2.381
com.thoughtworks.frankenstien.recorders	28	98%	100%	1.256
com.thoughtworks.frankenstien.script	3	100%	100%	1.167
com.thoughtworks.frankenstien.ui	4	100%	100%	1.833

Report generated by [Cobertura](#) 1.7 on 5/6/06 11:29 AM.

All Packages

Classes

- [AbstractCheckRecorder \(100%\)](#)
- [AbstractComponentNamingStrategy \(100%\)](#)
- [AbstractComponentRecorder \(100%\)](#)
- [AbstractFrankensteinEvent \(100%\)](#)
- [ApplicationLauncher \(N/A\)](#)
- [ButtonClickRecorder \(100%\)](#)
- [ButtonNamingStrategy \(100%\)](#)
- [CancelTableEditEvent \(100%\)](#)
- [CheckBoxRecorder \(100%\)](#)
- [CheckTextEvent \(100%\)](#)
- [CheckTextRecorder \(100%\)](#)
- [ClickButtonEvent \(100%\)](#)
- [ClickCheckboxEvent \(100%\)](#)
- [ClickRadioButtonEvent \(100%\)](#)
- [ComponentFinder \(N/A\)](#)
- [ComponentHierarchyWalker \(100%\)](#)
- [ComponentMatchingRule \(N/A\)](#)
- [ComponentName \(100%\)](#)
- [ComponentNameMatchingRule \(100%\)](#)
- [ComponentNamingStrategy \(N/A\)](#)
- [ComponentPositionComparator \(100%\)](#)
- [ComponentRecorder \(N/A\)](#)
- [ComponentVisibility \(N/A\)](#)
- [CounterBasedNamingStrategy \(100%\)](#)
- [DefaultComponentFinder \(98%\)](#)
- [DefaultComponentVisibility \(100%\)](#)
- [DefaultEventRegistry \(100%\)](#)
- [DefaultFileDialogLauncher \(100%\)](#)
- [DefaultFrankensteinRecorder \(98%\)](#)

Demo

How does it work?

Recorders hook onto Swing
Event Queue

Add component-specific
listeners

Create Events when listeners
are triggered

Events know how to replay
themselves

IOC + PicoContainer

What 's missing?

More assertions

Support for relatively less
used components

Open Sourcing soon

Thank You

Blog at
<http://www.vivekprahlad.com>

Email:
vprahlad@thoughtworks.com