Defining Swing GUIs Declaratively: A Survey

Hans Muller

Java™ Desktop Engineering
Sun Microsystems
http://www.sun.com

TS-7122

There's a Lot Going On

Survey the field and take a close look at the best

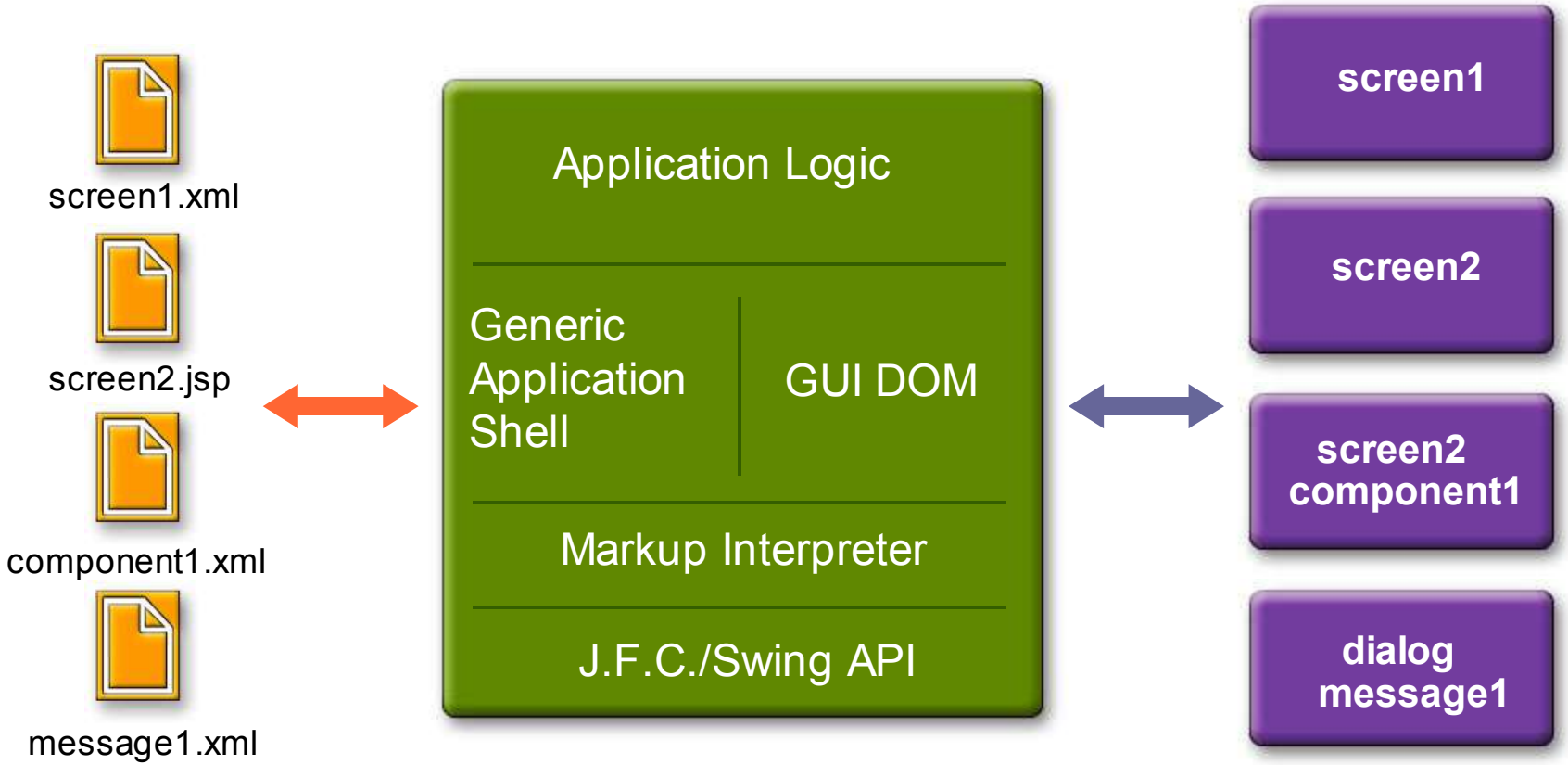
A brief tour of the entire Java™ Foundation Classes (J.F.C.)/**Swing**
XML landscape and a detailed look at some of the best examples; short-term and long-term recommendations for developers

Agenda

Why bother with GUI markup anyway?
A brief history of GUI markup languages
Current non-Java GUI markup standouts
J.F.C./Swing markup survey
Close ups (code and XML gore galore)
Some recommendations

GUI Markup

GUI markup is a special purpose language for defining the appearance and visual behavior of a user interface



Why GUI Markup?

- Dynamic GUI Generation
- Expressive
- Compact
- Convenience
- Increase leverage
- Perspective shift

- Not a substitute for GUI Tools/IDEs

Early GUI Markup

Before 1990

GUI Markup Before 1990

- 1970s Teletext, VideoText,…

- 1980s
  - User Interface Management Systems
  - Sun's DevGuide GIL format (Lisp!)
  - Apple's HyperCard*

UI Markup Before 1990: UIL

- UIL for CDE/Motif
  - Compiled UIL files were interpreted by MRM, the Motif Resource Manager
  - Intended as the "standard" GUI builder archive format
  - Classes/properties same as C Motif API
  - Not widely accepted/liked
- Not widely accepted/liked

```
object hello : XmLabel
{
  arguments {
    XmNlabelString = "Hello, World!";
    XmNx = 50;
    XmNy = 50;
  };
};
```

Hello Globe in UIL


```
module globe

value
  world_colors : color_table (
     background color = ' ',
       color ('black') = '*',
       color ('blue') = '.',
       color ('green') = 'x',
       color ('white') = '=');

 object root : XmLabel {
  arguments {
    XmNlabelType = XmPIXMAP;
    XmNlabelPixmap = world_icon;
    XmNmarginWidth = 10;
    XmNmarginHeight = 10;
  };
};
```

```
world_icon : icon (
  color_table = world_colors,
    '     ******     ',
    '   **.===..**   ',
    '  *xx.==..x..*  ',
    ' *xxx....xxx..* ',
    ' *.xxxxxxxxx.x* ',
    '*.xxxxxx.xxx.xx*',
    '*.xxxxxxxxx...x*',
    '*.xxxxxxxxx...x*',
    '*..xxxxxxxx...x*',
    '*...xxxx..x....*',
    '*....xx.....x..*',
    ' *....xx......* ',
    ' *....xxxxx...* ',
    '  *..xxxxxxx.*  ',
    '   **xxxxxx**   ',
    '     ******     '  );
end
```

Current Swing XML Markup Schemes

- A quick review of
  - Beryl XMLGUI
  - SwixAT
  - SwixNG
  - XOM
  - JFCML
  - …(about a dozen more)
- A close look at SwingML, SwixML

Disclaimer

- I've tried to make a comprehensive survey
  - I've probably overlooked some projects/products
  - My apologies if one of them was yours

Swing GUI Markup

A whirlwind tour

Survey Summary Preview

- There are a lot of projects out there
  - Many are dormant, a handful are dead
  - A few are pretty good!
- Common Themes
  - Purely declarative vs. scripting support
  - Special purpose vs. generic schemas
  - Availability: commercial, free, open source, private

XwingML

http://www.bluestone.com/xml/XwingML/

- The oldest Swing XML schema I know of
  - Freeware intended to leverage sales of an XML server product
  - Emerged in early 1999
- Gone without a trace

XOM

http://sourceforge.net/projects/xom/

- Dead

JXUL

http://jxul.sourceforge.net/

- Dead too

CookSwing

http://cookxml.sourceforge.net/cookswing/

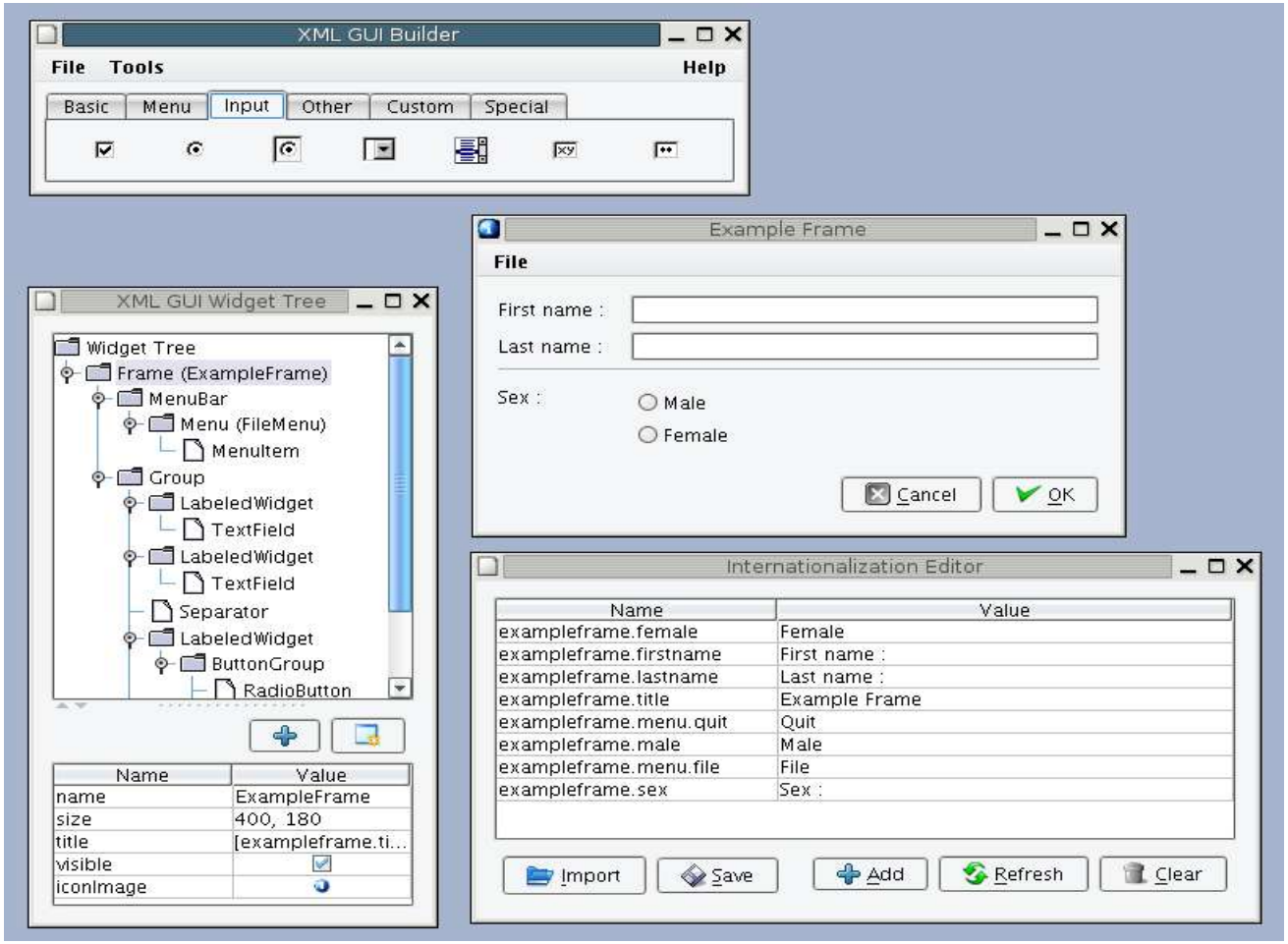
- Similar to SwiXML
  - Support for more layout managers, borders, ..
  - Integral vector data, list/table model support
  - Dubious ActionListener "synchro" feature
- Loads of Web started demos!
- Developer: Heng Yuan (heng@ag.arizona.edu)
- First released (version 1.0) in early August 2004; updated about a month later
- Dead?

Beryl Project XMLGUI

http://xmlgui.tigris.org/

- Archive format for XML GUI builder
  - Part of the Beryl project
  - XML markup schema appears to be simple but it's not specified
- Developer: Jakob Wenzel (wazlaf@tigris.org)
- Two releases, in February and April 2004
  - Demos, IDE work, project is stable
  - Not much documentation

Beryl Project XMLGUI Builder



Purnama Project XUI ("zooey")

http://geekkit.bcit.ca/xui

- Straightforward mapping of Swing on XML
  - Intended as presentation layer for the semantic Web
  - parser supports namespaces, validation
- Developer Arron Ferguson (arron_ferguson@bcit.ca)
  - Comprehensive documentation
  - Working prototype
  - Work stopped in December 2003 (graduation!)

XULUX ("zool looks")

http://xulux.org/

- Based on NYX ("not yet XULUX")
  - "Currently there are no releases of XULUX. Not even an alpha release. Since XULUX currrently still is a moving target, I think it is best to not have any releases, until the API stabelises."
- Developer James Strachan (jastrachan@mac.com)
- Dormant

SwixAT

http://www.swixat.org

- An attempt to integrate SwiXML with
  - Spring framework—dependency injection
  - JXPath—XML data binding
  - BeanShell—scripting
  - Groovy—more scripting
- Developer: Paolo Marrone (paolo@marrone.org)
- Underway since early March 2005
  - Most recent release (0.3) on May 3rd

SwixNG

http://swixng.sourceforge.net/

- A minor incompatible fork of SwiXML
  - Added support for TableLayout
  - Removed various things, including Mac-specific stuff
- Developer: Michael Klaus
- One 0.2 release in February 2005

JFCML

http://jfcml.sourceforge.net/

- Schema is very similar to SwiXML
  - Implementation: fine grained lazy loading of infrastructure, capabilities
  - BeanShell integration in progress
- Developer: Shawn Curry (_shawn_curry_@excite.com)
- Release in September 2004 (some changes since then)
  - Development is active
  - Work on demos and docs

Luxor

http://luxor-xul.sourceforge.net/

- Schema is similar to Mozilla XUL (subset)
  - Java, Python, or Groovy for behavior
- Developer: Gerald Bauer (gerald.bauer@mac.com)
- Since 2001, most recent update February 2005
  - Development continues
  - Extreme Web presence
  - Counter app showcase definitely worth a visit http://xul.sourceforge.net/counter.html

JGraphAppKit

http://www.jgraph.com/

- Just for menus and toolbars, actions
  - Works for applets too (despite the name)
- Developer: Gaudenz Alder (gaudenz@jgraph.com)
- Release as a standalone framework, July 2004
  - Now a part of the JGraph platform
  - Next major release

Proto

http://www.pierlou.com/prototype/

- XML schema, similar to Swing API
  - Scripting with BeanShell, TCL
  - Some JDBC™ support
- Development stopped in 1999
- Has morphed into the Ultrid project

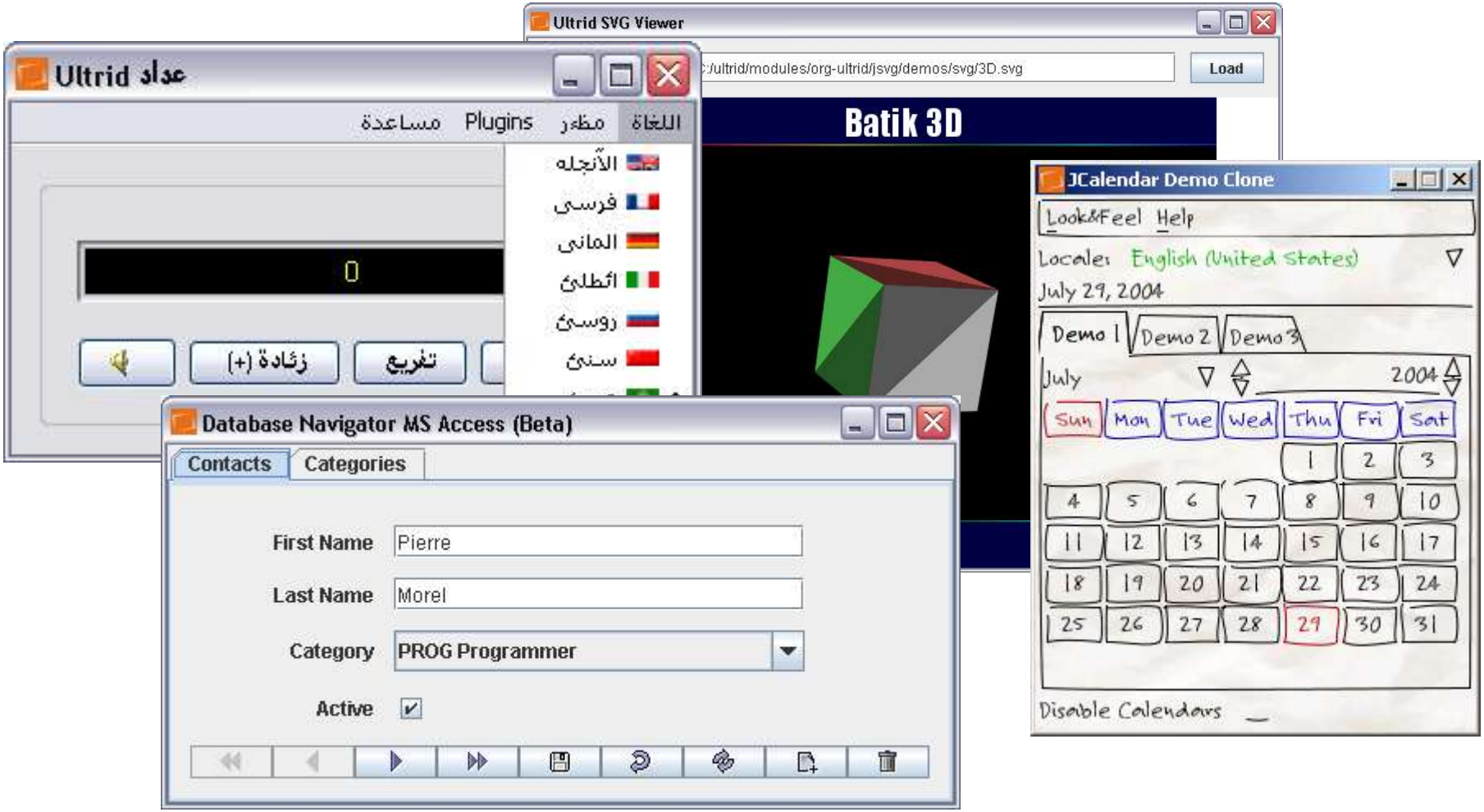
Ultrid

http://www.ultrid.com/

- XML schema is called XPML (eXtensible Programming Markup Language)
  - Closely mirrors Swing API
  - Classes map to tag names
  - Support for creating top level objects, like listeners
  - Layout constraints as attributes or subordinate elements
  - Much more...
- Intended to be the core of a Swing "browser"

Ultrid—Examples

ULTRID



Ultrid

"Ultrid helps you to create Java Cross-Platform application prototypes with disconcerting ease."

- Developer Pierre Morel (pierre@ultrid.com)
- Last release was December 2004
  - Working in an IDE
  - Targeting low-skill developers
- Not a particularly catchy name in English

Ultrid Summary

- Lots of good ideas here
  - Mostly straightforward mapping of the Swing API
  - Support for localization, JavaHelp, preferences
    - Many more custom modules (L&F, Audio, Batik ...)
  - Parameterized XPML parts
  - BSF integration for scripting
  - Simple primitives for extensions
- Weaknesses
  - Large API lacks comprehensive documentation
  - Not open source

XMLFace

http://www.xesoft.com/products/xmlface//

- Commercial product form XEsoft
  - Free for developers
  - Good docs, examples, demos
- XFM schema defines a simplified GUI toolkit
  - More than HTML forms, (much) less than Swing
  - Java/JavaScript™ for behavior
  - Java toolkit API, quite similar to Swing
- Nice logo!

XUI

http://xui.sourceforge.net/

- Open Source project (1.0.4 release January 2005)
  - Commercial version "Carousel" based on XUI
  - Good docs, examples, demos; NetBeans™ support!
  - 2.0 to feature Synth L&F based on SVG (Batik)
- XML schema defines a simplified GUI toolkit
  - Similar to AWT in scale, capabilities
  - Can target Swing or AWT
  - Java for event handling
  - Separate documents for GUI, validation, styles
  - Interesting data binding system

eNode

http://www.enode.com

- eNode UI Markup Language
  - Comprehensive mapping to Swing, even J2SE™
- Core "Object Realizer" now open source
  - http://openmarkup.dev.java.net
  - Implementation was contributed to the JDNC project
- Commercial product
  - Top notch docs and examples
  - Development continues (J2SE 1.5 version)
  - Web site hasn't been updated for some time

Current Swing Markup Schemes

- A quick review of
  - Beryl XMLGUI
  - SwixAT
  - SwixNG
  - XOM
  - JFCML
  - ...
- A close look at SwingML, SwixML

SwingML

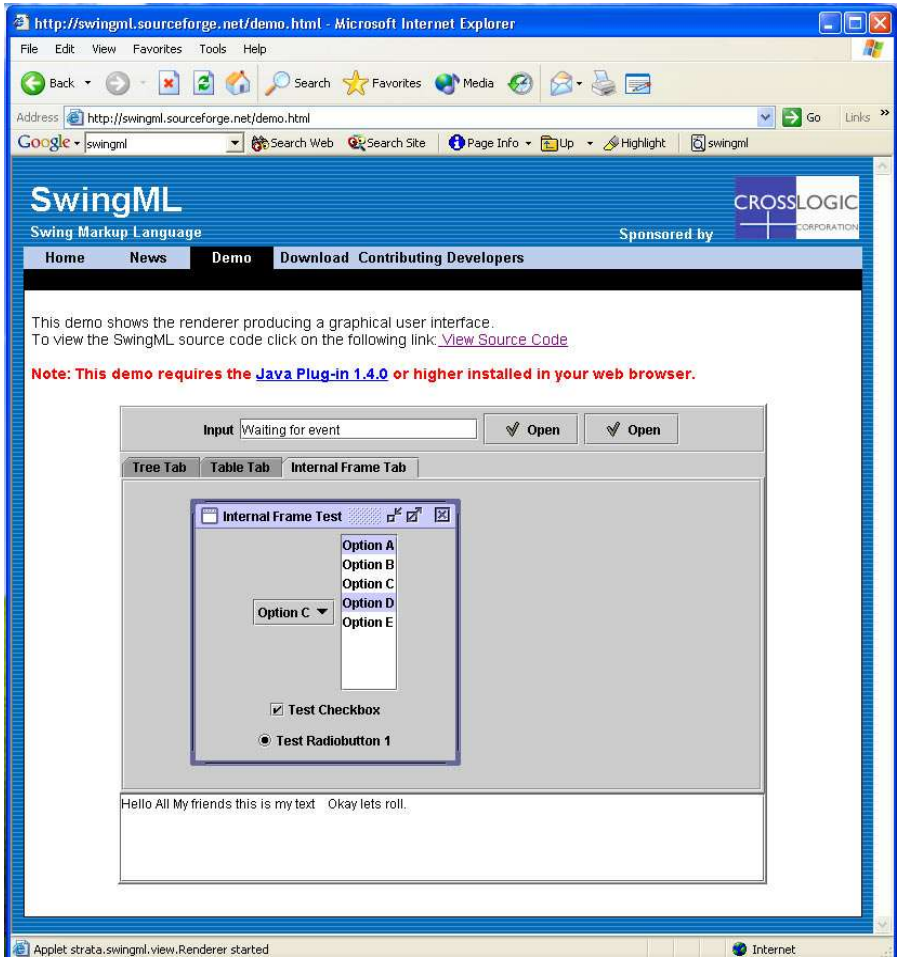
http://swingml.sourceforge.net

SwingML

http://swingml.sourceforge.net

- Characteristics
  - Custom XML schema
  - Supports commonly used Swing features
  - Integral data for trees, tables, etc.
  - Scripting support via Apache's BSF
  - Similar in "weight" to HTML forms
- Current release is 1.1, first released in 2002
- Developers
  - Ezequiel Cuellar, Robert Morris, Bram Stieperaere, Marcelo W. Lopez Cremona

SwingML in Action



http://java.sun.com/products/jfc/tsc/sightings/S12.html

SwingML Component Tags (20)

- label
- textfield
- button
- tree
- table
- list
- combobox
- checkbox
- slider
- radiobutton
- panel
- tabbedpane
- splitpane
- toolbar
- buttongroup
- textarea
- passwordtext
- editorpane
- internalframe
- menubar

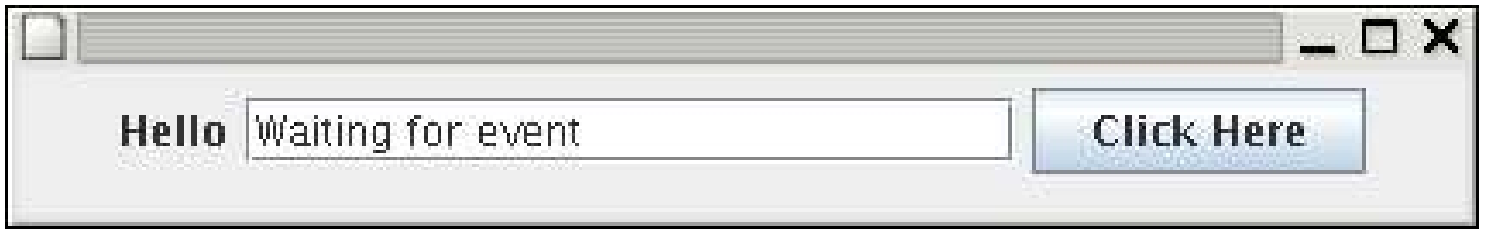
SwingML Shell Application

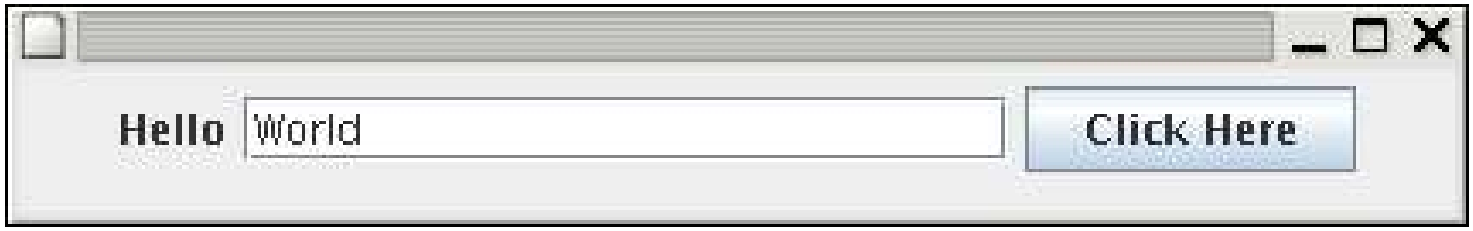
```
public class HelloWorld {
  public void run() throws Exception {
    JFrame frame = new JFrame("Hello World");
    SwingMLRenderer renderer = new SwingMLRenderer();
    renderer.render("HelloWorld.xml", frame);
    frame.pack();
    frame.setVisible(true);
  }
  public static void main(String[] args) {
      Runnable doRunHelloWorld = new Runnable() {
          public void run() { new HelloWorld().run(); }
      };
      SwingUtilities.invokeLater(doRunHelloWorld);
  }
}
```

SwingML HelloWorld Example

```
<panel orientation="center">
  <label name="helloLabel" text="Hello"/>
  <textfield
     name="worldField"
     text="Waiting for event" cols="20"/>
  <button name="worldButton" Text="Click Here">
    <listener event="ActionListener.actionPerformed">
      <action
         component="worldField"
         method="setText"
         types="String" values="World"/>
      </listener>
   </button>
</panel>
```

SwingML HelloWorld Example





SwingML Example: Listeners

```
<button name="worldButton" Text="Click Here">
  <listener event="ActionListener.actionPerformed">
    <action
      component="wordField"
      method="setText"
      types="String" values="World"/>
   </listener>
</button>
```

... the ActionListener is logically equivalent to:

```
new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        wordField.setText("World");
    }
};
```

SwingML Example: Listeners (Cont.)

- SwingML listener actions
  - Define one Swing/AWT listener method like **`MouseListener.mousePressed`**
  - That can call one (arbitrary method) on a previously defined component
  - Can provide one or more parameters
- External-action tag enables invoking a method on an arbitrary class

SwingML Listener external-action tag

```
<listener event="ActionListener.actionPerformed">
  <external-action
        external-class="com.foo.bar.MyActionHandler"
        component="wordField">
    <action-param name="text" value="world"/>
    <action-param name="color" value="red"/>
  <external-action/>
 </listener>
```

- external-action semantics (roughly): call a MyActionHandler method with the wordField component and key/value action-params as parameters

Defining SwingML external-actions

- external-action classes implement must InvokableEvent (a "callback" interface)

```
public interface InvokableEvent {
    public void initialize(Component c, Object[] params);
    public void invoke();
    public void destroy();
}
```

... which is used roughly like this:

```
public void actionPerformed(ActionEvent e) {
    InvokableEvent handler =
        new com.foo.bar.MyActionHandler();
    handler.initialize(incomingComponent, incomingParams)
    handler.invoke();
    handler.destroy();
}
```

SwingML external-action Lifetime: the RetainedInMemory Interface

- external-action classes also typically implement RetainedInMemory:

```
public interface RetainedInMemory {
    int MEM_VOLATILE = 0;
    int MEM_GLOBAL = 1;
    int MEM_DOCUMENT = 2;
    public int retainInMemory();
}
```

SwingML: Support for JavaScript

- Based on Apache Bean Scripting Framework http://jakarta.apache.org/bsf/
  - Support for a variety of scripting languages, including JavaScript (Rhino), JPython, BeanShell, etc...
  - Refer to SwingML components by name
  - Full access to Swing (J2SE) API

SwingML Example: JavaScript

```
<external-action
    component="aTable"
    external-class="swingml.bsf.Script">
  <action-param
    name="script-engine" value="javascript"/>
  <action-param name="script">
    <![CDATA[
      var rows = aTable.getSelectedRows();
      if (rows.length > 0) {
        var row = rows.length - 1;
        var smodel = aTable.getSelectionModel();
        smodel.setSelectionInterval(0, row);
        wordField.setText("Selected " + row + " rows");
      } ]]>
   </action-param>
</external-action>
```

SwingML: Integral Data

- Data for JList, JTable, JTree, JComboBox can be part of the markup (XML) file
- Convenient for browser-style apps

```
<list name="List1" mode="Multiple" dndenabled="True">
  <item text="Item one" />
  <item text="Item two" selected="True" />
  <item text="Item free" />
  <item text="Item four" />
  <item text="Item foo" selected="True"/>

</list>
```

SwingML: HTML Style Table Data

```
<table name="testTable">
 <tc text="Col A" type="Boolean"/>
 <tc text="Col B" type="Combo" items="1,2,3,4" />
 <tc text="Col D" type="String"/>
 <tr>
   <td value="False" editable="True"/>
   <td value="1" editable="True"/>
   <td value="D" editable="True"/>
 </tr>
 <tr>
   <td value="True" editable="True"/>
   <td value="2" editable="True"/>
   <td value="H" editable="True"/>
 </tr>
</table>
```

SwingML Summary

- Strengths
  - Similar in "weight" to HTML forms
  - Supports commonly used Swing features
  - Integral data for trees, tables, etc.
  - Scripting support via Apache's BSF
- Weaknesses
  - New schema learning curve
  - 1.1 Release is still pretty rough
  - Quirky callback machinery
  - Project has gone somewhat dormant

SwiXML

http://www.swixml.org/

SwiXML

http://www.swixml.org

SwiXml

- Simple mapping of Swing on XML schema
  - Comprehensive support for Swing API
  - No markup support for data binding and behavior
- Runtime (swixml.jar) is very small—only 40K bytes!
- Current release is 1.2, first releases in 2003
- Open source project launched by Wolf Paulus

SwiXML Schema

- One tag for most Swing component classes
- One attribute for most component properties
  - Special syntax for common object property types
  - Action attribute names "client object" field
- Nesting means containment
  - Special case for layout constraints

SwiXML Component Tags (41)

- applet
- button
- buttongroup
- checkbox
- checkboxmenuitem
- combobox
- component
- desktoppane
- dialog
- editorpane
- formattedtextfield
- frame
- hbox
- internalframe
- label
- list
- menu
- menubar
- menuitem
- optionpane
- panel
- passwordfield
- popupmenu
- progressbar
- radiobutton
- radiobuttonmenuitem
- scrollpane
- separator
- slider
- spinner
- splitpane
- tabbedpane
- table
- tableheader
- textarea
- textfield
- textpane
- togglebutton
- toolbar
- tree
- vbox

SwiXML Shell (Application)

```
public class HelloWorld {
  public void run() { ... }
  public JTextField tf;
  public Action doHelloWorld = new AbstractAction() {
    public void actionPerformed(ActionEvent e) {
      tf.setText("World");
    }
  };
  public static void main(String[] args) {
    Runnable doRunHelloWorld = new Runnable() {
      public void run() {new HelloWorld().run();}
    };
    SwingUtilities.invokeLater(doRunHelloWorld);
  }
}
```

SwiXML Shell Application (Cont.)

```
public class HelloWorld {
  public void run() throws Exception {
    SwingEngine engine = new SwingEngine(this);
    JFrame frame =
      (JFrame)(engine.render("HelloWorld.xml"));
    frame.pack();
    frame.setVisible(true);
  }
  public JTextField tf;
  public Action doHelloWorld = ...;
  public static void main(String[] args) { ... }
}
```

SwiXML HelloWorld Example

```
<frame size="640,480" title="Hello SWIXML World">
  <panel constraints="BorderLayout.CENTER">
    <label
      LabelFor="tf"
      Foreground="blue" text="Hello"/>
    <textfield id="tf"
      Columns="20" Text="Swixml"/>
    <button
      Text="Click Here"
      Action="doHelloWorld"/>
  </panel>
</frame>
```

SwiXML Special Attributes

id

Action

refid, Layout Constraints

initClass

Converters

SwiXML XML id Attribute

Initializes "client object" public field

```
<textfield id="tf" Columns="20" Text="Swixml"/>

public class HelloWorld {
  public JTextField tf;
  public void run() throws Exception {
    // this HelloWorld instance is the client object
    SwingEngine engine = new SwingEngine(this);
    JFrame frame =
      (JFrame)(engine.render("HelloWorld.xml"));
    tf.setText("I was rendered by SwiXML");
  }
}
```

SwiXML XML Action Attribute

Looks up corresponding Action valued public field

```
<button Text="Click Here" Action="doHelloWorld"/>

public class HelloWorld {
  public void run() throws Exception {
    SwingEngine engine = new SwingEngine(this);
    JFrame frame =
      (JFrame)(engine.render("HelloWorld.xml"));
      // engine.render() effectively does:
      // button.addActionListener(doHelloWorld);
  }
  public AbstractAction doHelloWorld = ... ;
}
```

SwiXML GridBagLayout Support

```
<panel layout="GridBagLayout">
  <button Text="Left">
    <gridbagconstraints
      gridx="1" gridy="0"
      insets="2,2,2,2"
      weightx="1" weighty="1"/>
  </button>
  <button Text="Right">
    <gridbagconstraints
      gridx="1" gridy="0"
      insets="2,2,2,2"
      weightx="1" weighty="1"/>
  </button>
</panel>
```

SwiXML refid Attribute

Copies attributes from specified element id

```
<panel layout="GridBagLayout">
  <button Text="Left">
    <gridbagconstraints id="defaultGBC"
      gridx="1" gridy="0"
      insets="2,2,2,2"
      weightx="1" weighty="1"/>
  </button>
  <button Text="Right">
    <gridbagconstraints refid="defaultGBC"
      gridx="1"/>
  </button>
</panel>
```

SwiXML initclass Attribute

Create constructor arguments, like Swing models

```
<table initclass="myPkg.MyTableModel"/>
```

Comparable to:

```
new JTable(new myPkg.MyTableModel());
```

- or -

```
new JTable(myPkg.MyTableModel.getInstance());
```

Other Special Attribute Conversions

Font	font="Georgia-BOLD-36"
KeyStroke	accelerator="control N"
Locale	locale="de"
Color	foreground="991144"
ImageIcon, Image	icon="myIconImage.png"
Insets	insets="1,2,3,4"
Dimension	size="100,200"
Point	origin="12,34"
Rectangle	bounds="10,10,250,200"
Border	border="TitledBorder(My Title)"

SwiXML Summary

SwiXml

- Strengths
  - Straightforward design
  - If you know the Swing API, you know SwiXML
  - Small (40K bytes!)
  - Integrates nicely with code
  - Some takeup by projects
- Weaknesses
  - Implicit schema is big
  - Documentation

Summing Up

Why We're Here (Reprise)

- Dynamic GUI generation
- Expressive
- Compact
- Convenience
- Increase leverage
- Perspective shift

Conclusions

- Two kinds of schema designs
  - Special purpose (SwingML)
  - Mirror Swing API (about 20)
- SwiXML is the strongest example of the latter
- Integration with scripting: inevitable
  - Less is more
  - Demands tools support
- Integration with an app framework: desirable
  - XML GUI can be part of the solution

Thanks You!

Many developers answered questions and provided feedback...

- Gaudenz Alder (http://www.jgraph.com)
- Richard Bair (http://sourceforge.net/projects/xom/)
- Gerald Bauer (http://xul.sourceforge.net)
- Shawn Curry (http://jfcml.sourceforge.net/)
- Arron Ferguson (http://geekkit.bcit.ca/xui)
- Pierre Morel (http://www.ultrid.com)
- Robert Morris (http://swingml.sourceforge.net)
- Wolf Paulus (http://www.swixml.org/)
- James Strachan (http://xulux.org/)
- Jakob Wenzel (http://xmlgui.tigris.org/)
- Jon Lipsky (http://www.xesoft.com/products/xmlface/)

Q&A

Submit Session Evaluationsfor Prizes!

Your opinions are important to Sun

- You can win a $75.00 gift certificate to the on-site Retail Store by telling Sun what you think!
- Turn in completed forms to enter the daily drawing
- Each evaluation must be turned in the same day as the session presentation
- Five winners will be chosen each day (Sun will send the winners e-mail)
- Drop-off locations: give to the room monitors or use any of the three drop-off stations in the North and South Halls

Note: Winners on Thursday, 6/30, will receive and can redeem certificates via e-mail.

Defining Swing GUIs Declaratively: A Survey

Hans Muller

Java™ Desktop Engineering
Sun Microsystems
http://www.sun.com

TS-7122

Java™