JavaOne℠

Touch Our Application! Building a Rich SVG UI for Java™ ME Platform

Karol Harezlak
Sun Microsystems
Software Developer

Session Goal

Touch Our Application! Building a Rich Touch-Enabled SVG UI for Java™ Platform, Micro Edition

Learn how to develop a slick, touch-enabled user interface with the Java ME platform using SVG Rich Components.

Agenda

- Introduction
- SVG UI Rich Components
  - User Interface - XML Snippets
  - Java ME Library
- Tools Support
  - NetBeans SVG Composer
  - NetBeans Visual Mobile Designer
- Summary and Q&A

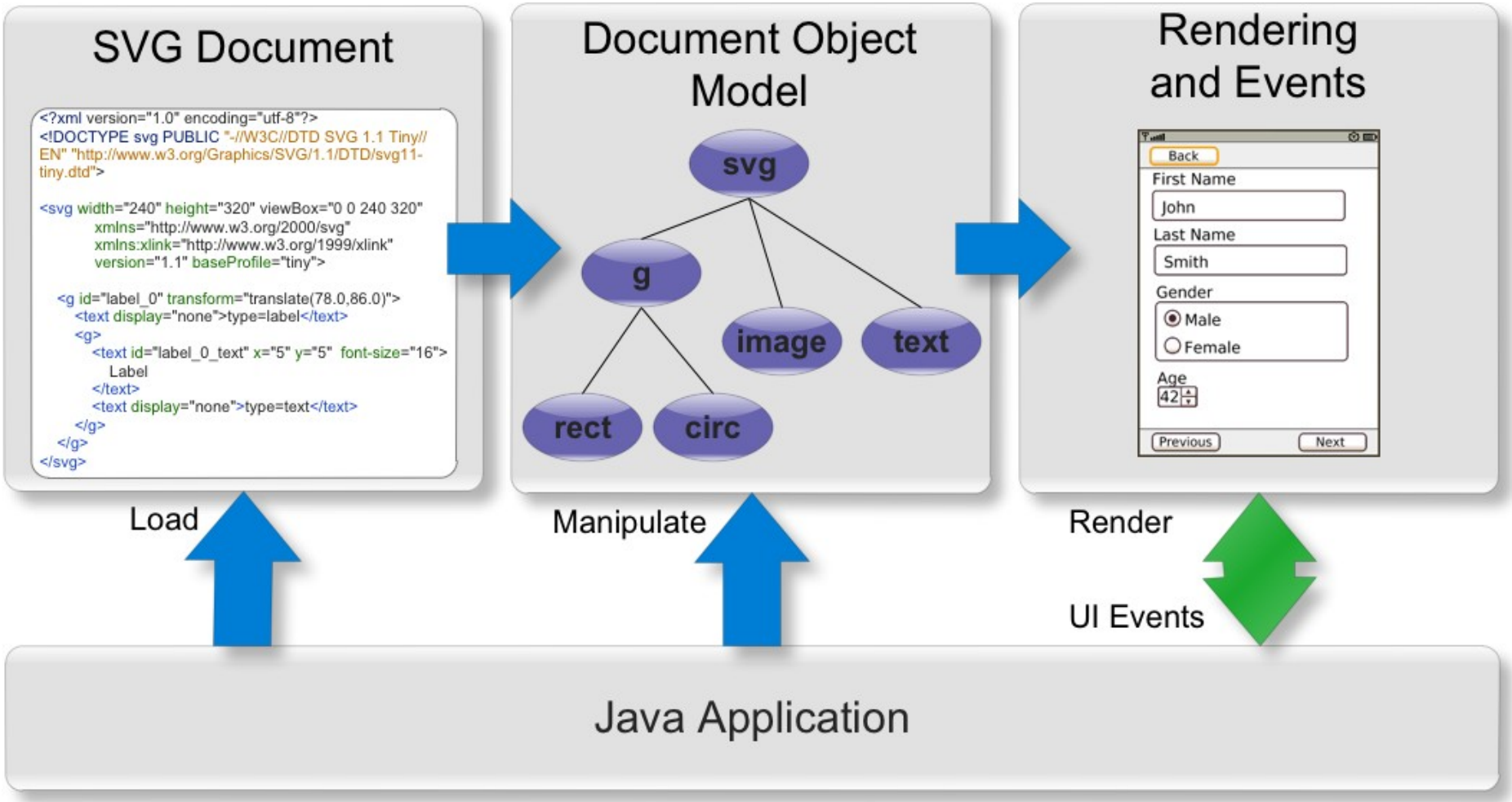
Agenda

- **Introduction**
- SVG UI Rich Components
  - User Interface - XML Snippets
  - Java ME Library
- Tools Support
  - NetBeans SVG Composer
  - NetBeans Visual Mobile Designer
- Summary and Q&A

Introduction
Scalable Vector Graphics

- Vector graphics
- Scalable without loss of quality
- Animations and events in XML
- Separates design from application logic
- W3C standards:
  - SVG Tiny 1.1, 1.1+ and SVG Tiny 1.2
- Designers produce SVG graphics with familiar tools

Introduction
Scalable Vector Graphics



Agenda

- Introduction
- **SVG UI Rich Components**
  - User Interface - XML Snippets
  - Java ME Library
- Tools Support
  - NetBeans SVG Composer
  - NetBeans Visual Mobile Designer
- Summary and Q&A

SVG UI Rich Components

- Architecture
  - SVG XML Snippets and Java ME Library
- Touch screen events support
- Clear separation between UI and application logic
- Customizable looks through XML Snippets
- Nine touch-enabled components available
  - Button, List, TextField, Spinner, RadioButton, ComboBox, Label, Slider, Checkbox

SVG UI Rich Components

Touch-Enabled Device in Landscape Mode

SVG UI Rich Components

Non Touch-Enabled Device

Agenda

- Introduction
- **SVG UI Rich Components**
  - **User Interface - XML Snippets**
  - Java ME Library
- Tools Support
  - NetBeans SVG Composer
  - NetBeans Visual Mobile Designer
- Summary and Q&A

SVG UI Rich Components

User Interface - XML Snippets

- UI components are defined in XML Snippets
- SVG Tiny 1.1 standard supported
- Customizable snippets
  - Appearance
  - Behavior
  - Animation
- UI stored as an SVG file

SVG UI Rich Components

User Interface - XML Snippets

Customized appearance of an SVG UI Component

Customized appearance of an SVG UI Component - XML Snippet

```
<!-- SVG Button Snippet -->
<g transform="translate(76.0,53.0)" id="button_1">
    <!-- Background Button image -->
    <image>
    <!-- Button shape -->
    <rect id="_102" x="0.0" y="0.0" rx="5" ry="5" width="80" height="30" fill="none" stroke="rgb(73,46,55)" stroke-width="1.5">
    </rect>
    <!-- Button animation on click event-->
    <g>
        <text display="none">type=body</text>
        <rect id="button_1_body" x="0.0" y="0.0" rx="5" ry="5"  width="80" height="30" fill="none" stroke="rgb(255,165,0)" stroke-width="0">
            <animate attributeName="stroke-width"  begin="button_1.focusin" dur="0.25s" fill="freeze" to="2"/>
            <animate attributeName="stroke-width"  begin="button_1.focusout" dur="0.25s" fill="freeze" to="0"/>
            <animate id="button_1_body_pressed" attributeName="stroke"  begin="indefinite" dur="0.25s" fill="freeze" to="red"/>
            <animate id="button_1_body_released" attributeName="stroke"  begin="indefinite" dur="0.25s"  fill="freeze" to="rgb(255,165,0)"/>
        </rect>
    </g>
    <!-- Button text -->
    <g>
        <text id="button_1_stext" x="27.999992" y="20.0" fill="black" font-size="15">  OK  </text>
        <text display="none">type=shadow_text</text>
    </g>
</g>
```

Agenda

- Introduction
- **SVG UI Rich Components**
  - User Interface - XML Snippets
  - **Java ME Library**
- Tools Support
  - NetBeans SVG Composer
  - NetBeans Visual Mobile Designer
- Summary and Q&A

SVG UI Rich Components
Java ME Library

- Handles touch screen events
- Controls interaction with SVG UI Components
- Provides an API to communicate with SVG UI components
- SVG Form – a container for SVG components
- Every SVG component has a unique ID
- Touch-enabled components and libraries come with NetBeans Mobility 6.7

SVG UI Rich Components
Java ME Library – Touch Screen Events

- Touch screen events are transparent for developers
- Touch screen events are forwarded to the SVGActionListener
- Internally SVG UI Components implement SVGEventListener to handle touch screen events
  - SVG UI Component pressed

```
public void pointerPressed(final int x, final int y) {...}
```

  - SVG UI Component released

```
public void pointerReleased(final int x, final int y) {...}
```

SVG UI Rich Components
Java ME Library

- Load an SVG UI Image

```
svgImageUI = (SVGImage) SVGImage.createImage(getClass()
    .getResourceAsStream("/package/svgUI.svg"), null);
```

- Create a new SVGForm

```
svgForm = new SVGForm(svgImageUI, getDisplay());
```

- Create a new SVG Component

```
svgButton = new SVGButton(svgForm, "button_0");
```

- Add a component to SVGForm

```
svgForm.add(svgButton);
```

SVG UI Rich Components
Java ME Library

- SVG TextField

```
svgTextField.setText(“First Name”);
```

- SVG CheckBox

```
svgCheckBox.setSelected(true);
```

- SVG Slider

```
svgSlider.setMax(100);
```

- All SVG Components

```
svgComponent.setFocusable(false);
```

Agenda

- Introduction
- SVG UI Rich Components
  - User Interface - XML Snippets
  - Java ME Library
- **Tools Support**
  - **NetBeans SVG Composer**
  - NetBeans Visual Mobile Designer
- Summary and Q&A

Tools Support
NetBeans SVG Composer

- Simple visual editor for UI
- XML source code editor
- SVG UI Preview
- UI Landscape mode
- Set of SVG components available from palette
- API to plug in customized SVG UI XML snippets into palette

Tools Support
Demo

- Building a touch-enabled UI with the NetBeans SVG Composer

Agenda

- Introduction
- SVG UI Rich Components
  - User Interface - XML Snippets
  - Java ME Library
- **Tools Support**
  - NetBeans SVG Composer
  - **NetBeans Visual Mobile Designer**
- Summary and Q&A

Tools Support
NetBeans Mobile Visual Designer

- All the benefits of NetBeans Mobility
  - Flow Designer
  - Screen Designer
- Code generation based on SVG UI image files
- Support for drag and drop of SVG UI image files
- New SVG palette component - SVG Form

Tools Support
Demo

- Creating a Rich SVG UI, touch-enabled application with Visual Mobile Designer

Summary

- SVG Rich UI Components simplify creation of touch-enabled SVG applications
- Nine touch-enabled SVG components available
- Appearance of the SVG components can be customized
- Tools Support
  - NetBeans SVG Composer
  - NetBeans Visual Mobile Designer

More Information

- Links
  - http://wiki.netbeans.org/MobilityDesignerRichComponents
  - http://www.netbeans.org/kb/trails/mobility.html
  - http://netbeans.org
- NetBeans Mobility User Mailing list
  - users@mobility.netbeans.org
- Other 2009 JavaOne Conference Sessions
  - LAB-5539 - Touch Your Application! Building Slick, Touch-Enabled UIs for Java™ Platform, Micro Edition

Questions & Answers

Karol Harezlak
karol.harezlak@sun.com

Sun Microsystems