



Graphical, Scripted, and Animated User Interfaces on the Java Platform

Nandini Ramani & Vincent Hardy

Senior Staff Engineers
Sun Microsystems, Inc.

TS-5743



Goal of This Session

Enabling Fluid Java™ Platform, Micro Edition
(Java ME platform) Graphical User Interfaces

Learn how to create graphically rich, animated and scripted applications for Java ME platform, today, tomorrow and in the future

Agenda

History of Java ME Platform User Interfaces

Demos

Application and Development Workflow

SVG Tiny 1.1 for the Java ME Platform,
Java Specification Request (JSR) 226

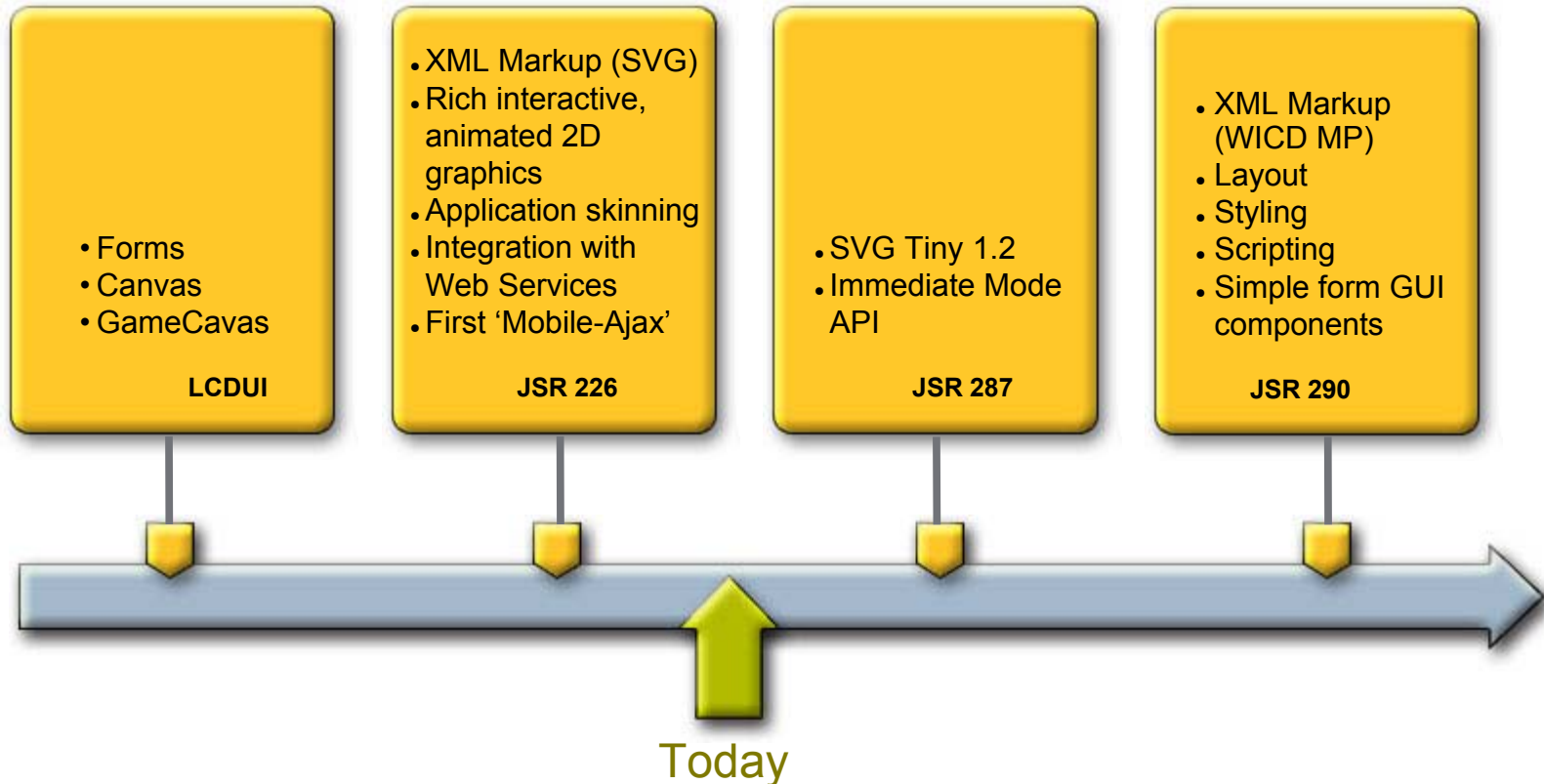
SVG Tiny 1.2 for the Java ME Platform, JSR 287

XML UI for the Java ME Platform, JSR 290

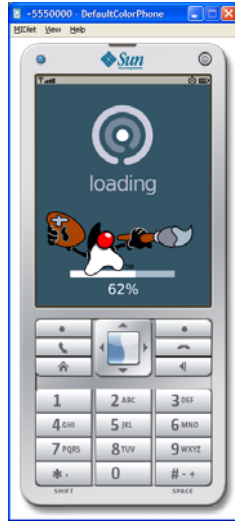
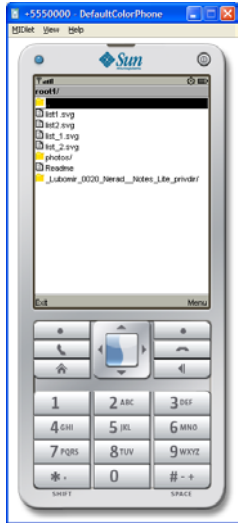
Q&A

Introduction

The history of Java ME platform graphics



Demos



LCDUI

Today
JSR-226

JSR 287

JSR 290

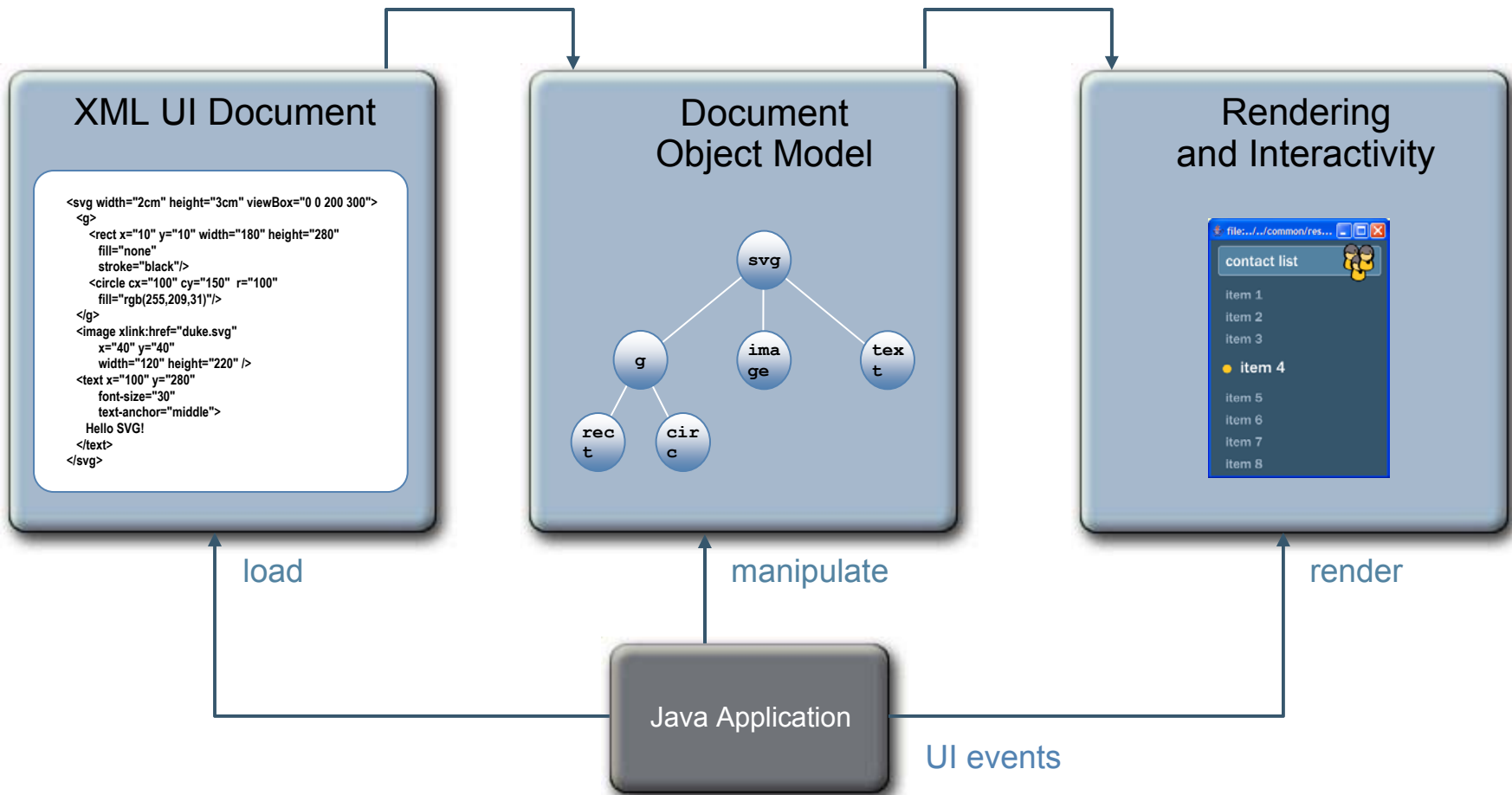


Demo

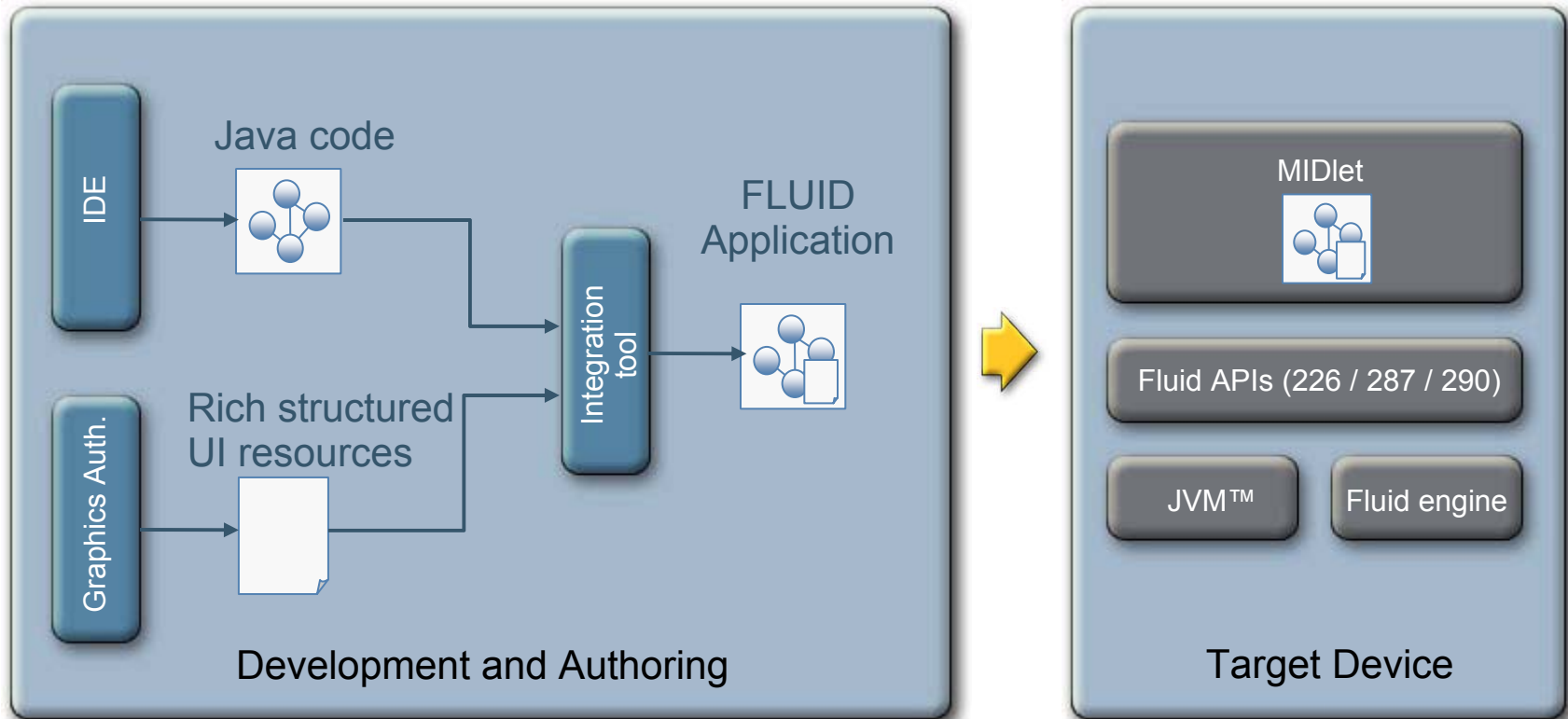
UI Time Travel



User Interface Application Model



User Interface Development Workflow



FLUID: FLeXible User Interface Development

The terms “Java Virtual Machine” and “JVM” mean a Virtual Machine for the Java™ platform

JSR 226—SVG Tiny 1.1

- Basic and complex shapes
- Rich text support
- Embedded fonts
- Animation
- Document Object Model (DOM) API
- Interactivity



SVG Tiny 1.1—Code Sample

JSR 226 sequence

```
SVGImage svgImage = (SVGImage)
    ScalableImage.createImage(svgURI, null);

// Build an SVGAnimator to play the SVG content
SVGAnimator animator =
    SVGAnimator.createAnimator(svgImage);

// Get the component where the image will play and
// display it.
Canvas cmp = (Canvas) animator.getTargetComponent();
Display.getDisplay().setCurrent(cmp);

// Play animation now and process user inputs
animator.play();
```

SVG Tiny 1.1—Code Sample

Graphics and animation

```

<svg ...>
  <g transform="translate(200,200)">
    <animateColor
      attributeName="fill"
      values="red;green;blue"
      begin="indefinite" dur="2s"/>
    <rect x="-30" y="-30" width="60" height="60"/>
    <circle id="myCircle" r="60" />
    <path d="M-20,-20 ... " />
  </g>
</svg>
  
```

SVG Tiny 1.1—Code Sample

Embedded font

```
<svg ...>
  <font ...>
    <font-face font-family="TestFont" .../>
    <missing-glyph d="M-0.5,... Z"/>
    <glyph unicode="A" d="M-0.5, ... Z"/>
  </font>
  <g transform="translate(200,200)">
    <text id="myText"
      font-family="TestFont">Hello</text>
  </g>
</svg>
```

SVG Tiny 1.1—Code Sample

JSR 226 DOM sequence

```
// Get the root of the document tree
Document doc = svgImage.getDocument();

// Access the circle element
SVGElement myCircle
    = (SVGElement) doc.getElementById("myCircle");

// Modify the circle's radius
myCircle.setFloatTrait("r", 30f);

// Trigger animation from code, with a 2s delay
SVGAnimationElement myAnim
    = (SVGAnimationElement) doc.getElementById("myAnim");
myAnim.beginElementAt(2);
```

JSR 287—SVG Tiny 1.2

- Gradients
- Opacity
- External `<use>`
- External fonts
- More powerful Document Object Model (DOM) API (e.g., dynamic animation creation)



SVG Tiny 1.2—Code Sample

JSR 287 sequence

```
public void onSMSReceived() {
    DocumentEvent docEvt = (DocumentEvent) doc;
    CustomEvent newSMS
        = docEvt.createEvent("CustomEvent");
    customEvent.initializeEvent(null,
                               "newSMS",
                               true,
                               true,
                               null);

    EventTarget svgRoot
        = (EventTarget) doc.getDocumentElement();

    svgRoot.dispatchEvent(newSMS);
}
```

SVG Tiny 1.2—Code Sample

Gradients and CustomEvent triggered animation

```

<svg id="root"...>
  <radialGradient id="grad" ...>
    <stop stop-offset="0" stop-color="white" />
    <stop stop-offset="1" stop-color="red" />
  </radialGradient>
  <rect fill="url(#grad)" x="0" ... >
    <animate attributeName="x" values="0;400"
      begin="root.newSMS" />
  </rect>
</svg>
  
```


JSR 290—WICD

- Java Programming Language and XML UI Markup Integration
- Leverage W3C CDF (WICD MP)
- Integration with Web technologies
 - XHTML Basic 1.1
 - ECMAScript CP
 - CSS 2.1 Mobile
 - SVG Tiny 1.2
- Binding between markup and Java code
- Connection between ECMAScript and Java platform



CDF—Code Sample

JSR 290 sequence

```
// Similar to JSR 226, but applies to multiple
// XML UI Formats (e.g., SVG, XHTML)
boolean disableScripts = false;
FluidImage fluidImage =
    ImageFactory.createImage(inputStream, null, null,
                            disableScripts);
FluidPlayer fluidPlayer =
    FluidPlayer.createPlayer(fluidImage);
FluidComponent component =
    fluidPlayer.getTargetComponent();
// Integrate the target component in the GUI toolkit.
...
// Now, start automatically playing animations
// and updating the display.
fluidPlayer.play();
```

CDF—Code Sample

JSR 290 sequence—text input

```
XMLImage xmlImage = ...;
Document doc = xmlImage.getDocument();
FluidPlayer fluidPlayer = ...;

// Implement the TextInputListener to perform any
// update needed on text entry
TextInputListener textInputListener = ...;

fluidPlayer.addTextInputListener(textInputListener);

// Start text input on the desired element
Element textElement =
    doc.getElementById("textInputElement");
xmlImage.setActiveTextInputElement(textElement);
fluidPlayer.startTextInput();
```

CDF—Code Sample

WICD MP example

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>CDF example</title>
  </head>
  <body style="background:blue" onclick="alert('hello')">
    <!-- Use SVG Image by Reference -->
    <object data="icon.svg"
            type="image/svg+xml" width="100%" />
  </body>
</html>
```

CDF—Code Sample

WICD MP example: simple menu layout

```

<html>
  <head>
    <style>
body {
  background: yellow;
  padding: 0;
  margin: 0;
  width: 100%;
  height: 100%;
}

object {
  width: 30%;
  height: 30%;
}
...

    </style>
  </head>
  <body>
    <object id="menuItem_0"
      type="image/svg+xml"
      data="item0.svg"
      <param name="focusable"
        value="flat" />
      ...
    </body>
</html>
.....

```



Demo

SVG Tiny 1.2—Multimedia Support



Combining Java Technology and Web Formats (1/2)

- Robust, secure, performance execution environment
- Rich API set
- High-end graphics, interactivity and animations
- Mobile Services Architecture (MSA)
Enables the development of services based on previously closed device capabilities



Combining Java Technology and Web Formats (2/2)

- Clear separation of application logic from application UI
 - Application UI in document Markup (SVG, XHTML, CSS)
 - Application logic in Java Platform
- Visual aspect and Interaction Model of application
 - Modified by replacing XML UI document
 - While retaining application logic

Examples of Synergies

- SVG animations can be triggered through JSR 226 APIs on messaging or Bluetooth events
- JSR 75 (PIM APIs) gives access to the contact list which can be richly animated/displayed in a Java application
- The look and feel for a game can be completely replaced in a JSR 226/287/290 based game

Conclusion

- Rich, scripted, animated, interactive and structured UI resources can be loaded, rendered, manipulated from the Java platform
- Graphic designers and Web developers can participate in the creation of elaborate applications
- This is an enabling platform
- You can start today with JSR 226 in MSA
- JSR 287 and JSR 290 will make this model even more compelling

For More Information

- **Session TS-5525:** Mobile AJAX for Java Technology
- **Session TS-5585:** Whiz Bang Graphics and Media Performance for Java™ ME Platform Applications
- **Session TS-5628:** Developing Flashy Mobile Applications Using SVG and JSR 226
- **Session TS-5626:** Data Binding and Java Platform ME
- **BOF 5677:** A Hands-On Introduction to SVG and JSR 226
- <http://www.svg.org>
- <http://www.w3.org>



Q&A

Nandini Ramani
Vincent Hardy





Graphical, Scripted, and Animated User Interfaces on the Java Platform

Nandini Ramani & Vincent Hardy

Senior Staff Engineers
Sun Microsystems, Inc.

TS-5743