Developing Flashy Mobile Applications Using SVG and JSR 226

Martin Brehovsky

Software Engineer
Sun Microsystems

Andrei Mihaila

Lead Engineer
Nokia Corporation

Marc Verstaen

Senior Director
Hyperion

TS-5628

DISCLAIMER BY

ORACLE® | Hyperion®

THE FOLLOWING IS INTENDED TO OUTLINE OUR GENERAL PRODUCT DIRECTION. IT IS INTENDED FOR INFORMATION PURPOSES ONLY, AND MAY NOT BE INCORPORATED INTO ANY CONTRACT. IT IS NOT A COMMITMENT TO DELIVER ANY MATERIAL, CODE, OR FUNCTIONALITY, AND SHOULD NOT BE RELIED UPON IN MAKING PURCHASING DECISION. THE DEVELOPMENT, RELEASE, AND TIMING OF ANY FEATURES OR FUNCTIONALITY DESCRIBED FOR ORACLE'S PRODUCTS REMAINS AT THE SOLE DISCRETION OF ORACLE.

Session Goal

Learn how to utilize SVG content developed by graphic artists to create good-looking and interactive Java™ Platform, Micro Edition (Java ME platform) applications

Agenda

Introduction to SVG

Java Specification Request (JSR) 226—Enabling SVG in Java ME applications

Tools for developing SVG/JSR 226 applications

What is coming

Summary

Q&A

Agenda

Introduction to SVG

Java Specification Request (JSR) 226—Enabling SVG in Java ME applications

Tools for developing SVG/JSR 226 applications

What is coming

Summary

Q&A

Introduction to SVG

- Vector graphics, animations and events in XML
- W3C standard
- SVG Tiny Viewer 1.1, 1.1+ and 1.2
- 150+ device types on the market, 225M+ devices
- Limitations:
  - No access to device features (camera, messaging…)
  - Accessing remote data might be difficult
  - No secure access to remote data

Source: http://www.svg.org

A Simple SVG File Example

```
<?xml version="1.0" encoding="utf-8"?>
<svg width="240" height="320" viewBox="-120 -160 240 320"
    xmlns="http://www.w3.org/2000/svg"
    version="1.1" baseProfile="tiny">

    <g id="Scene_1">
        <rect x="-120" y="-160" width="240" height="320"
              fill="#5382a1" stroke="#ff0000"/>
        <rect x="-110" y="-150" width="220" height="300"
              fill="#ED9B09">
              <animateTransform attributeName="transform"
                  type="translate" from="0,0" to="60,-40"
begin=”1s” dur="0.5s" fill="freeze"/>
        </rect>
    </g>
</svg>
```

DEMO

SVG Graphics in Action

SVG Tiny Viewer 1.1
7.6 kBytes

Agenda

Introduction to SVG

JSR 226—Enabling SVG in Java ME applications

Tools for developing SVG/JSR 226 applications

What is coming

Summary

Q&A

JSR 226—Enabling SVG in Java ME Applications

Where the two worlds meet

- JSR 226
  - Specification led by Nokia
  - Finalized in 2005
  - Supported by Ericsson, Motorola, Nokia, Sony Ericsson, Sun, Symbian…
  - Part of Mobile Service Architecture (JSR 248)
- SVG enables using declarative UI/graphics
- Java ME platform enables accessing device features
- Over 30 devices with JSR 226 support already available

Source: http://www.svg.org

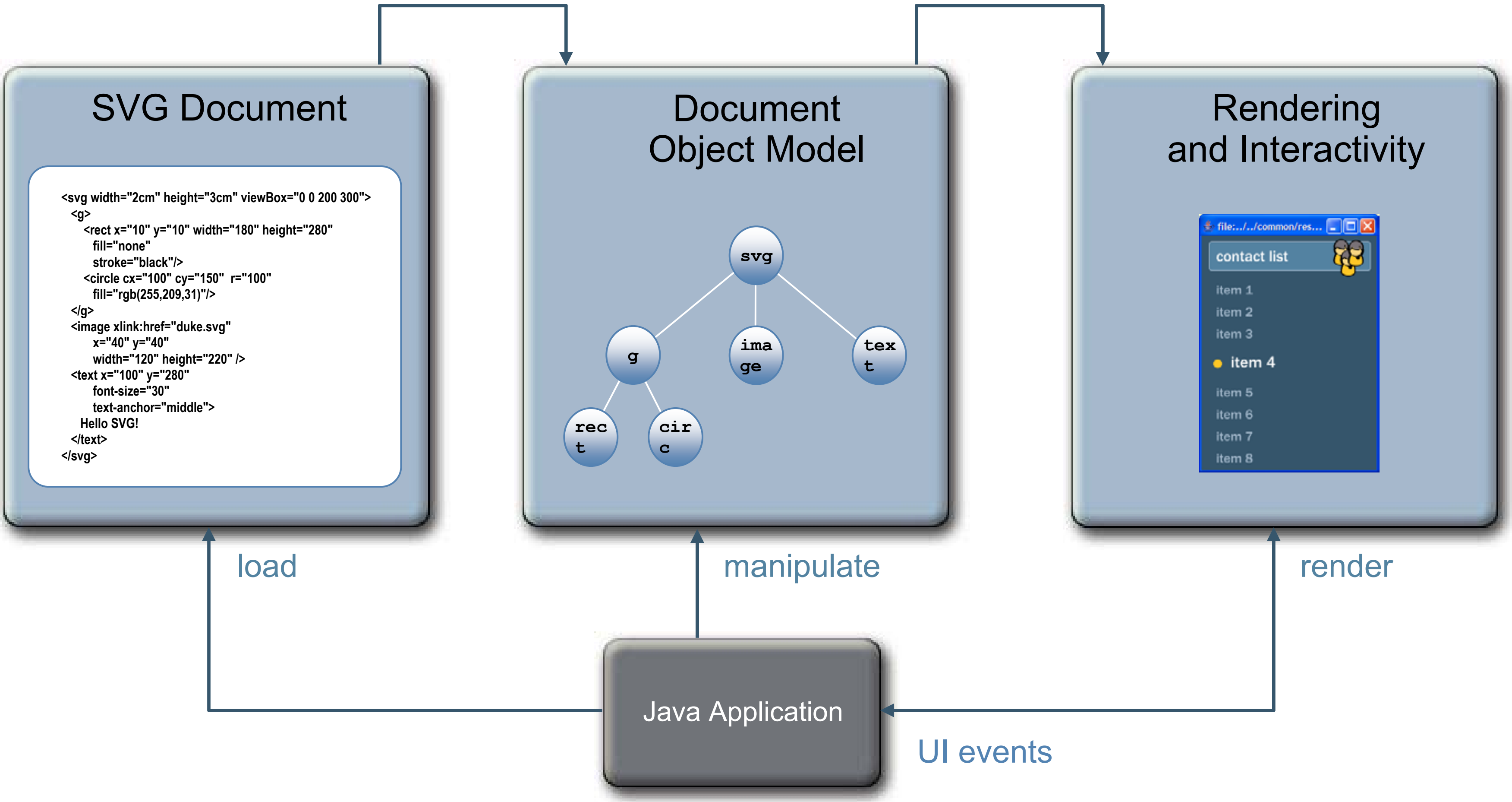
Showing SVG Graphics Using JSR 226

```
// load image
SVGImage svgImage = (SVGImage)ScalableImage.createImage(
       SVGDemo.class.getResourceAsStream("/simple.svg"),
             null);

// create animator
SVGAnimator animator = SVGAnimator.createAnimator(svgImage);

// get display component (Canvas)
Canvas canvas = (Canvas)animator.getTargetComponent();

// display the component
display.setCurrent(canvas);

// play the animation
animator.play();
```

DEMO

Java ME Applications Utilizing SVG

JSR 226 Application Model



```
<svg width="2cm" height="3cm" viewBox="0 0 200 300">
  <g>
     <rect x="10" y="10" width="180" height="280"
        fill="none"
        stroke="black"/>
     <circle cx="100" cy="150"  r="100"
        fill="rgb(255,209,31)"/>
  </g>
  <image xlink:href="duke.svg"
        x="40" y="40"
        width="120" height="220" />
  <text x="100" y="280"
        font-size="30"
        text-anchor="middle">
    Hello SVG!
  </text>
</svg>
```

JSR 226 Flexibility

It isn’t “All or Nothing”

- Using SVG is not “all or nothing”
- Several levels of SVG usage:
  - Use SVG rasterized to bitmaps during development time (JSR 226 not needed)
  - Pre-render SVG to bitmap on device and use bitmaps
  - Use “static” SVG graphics rendered to Canvas
  - Use fully dynamic SVG graphics together with events

DEMO

Various Levels of Using SVG
in Java ME Applications

Agenda

Introduction to SVG

JSR 226—Enabling SVG in Java ME applications

Tools for developing SVG/JSR 226 applications

What is coming

Summary

Q&A

Authoring Tools for SVG Content

The design domain

- Drawing tools producing SVG graphics:
  - Adobe Illustrator, Corel Draw, Inkscape, Sketsa, Hyperion Mobile Designer…
- Animation tools:
  - Ikivo Animator, Hyperion Mobile Designer…
- Interaction tools:
  - Adobe GoLive, Hyperion Mobile Designer…

Java ME Platform/JSR 226 Developer Tools

The code domain

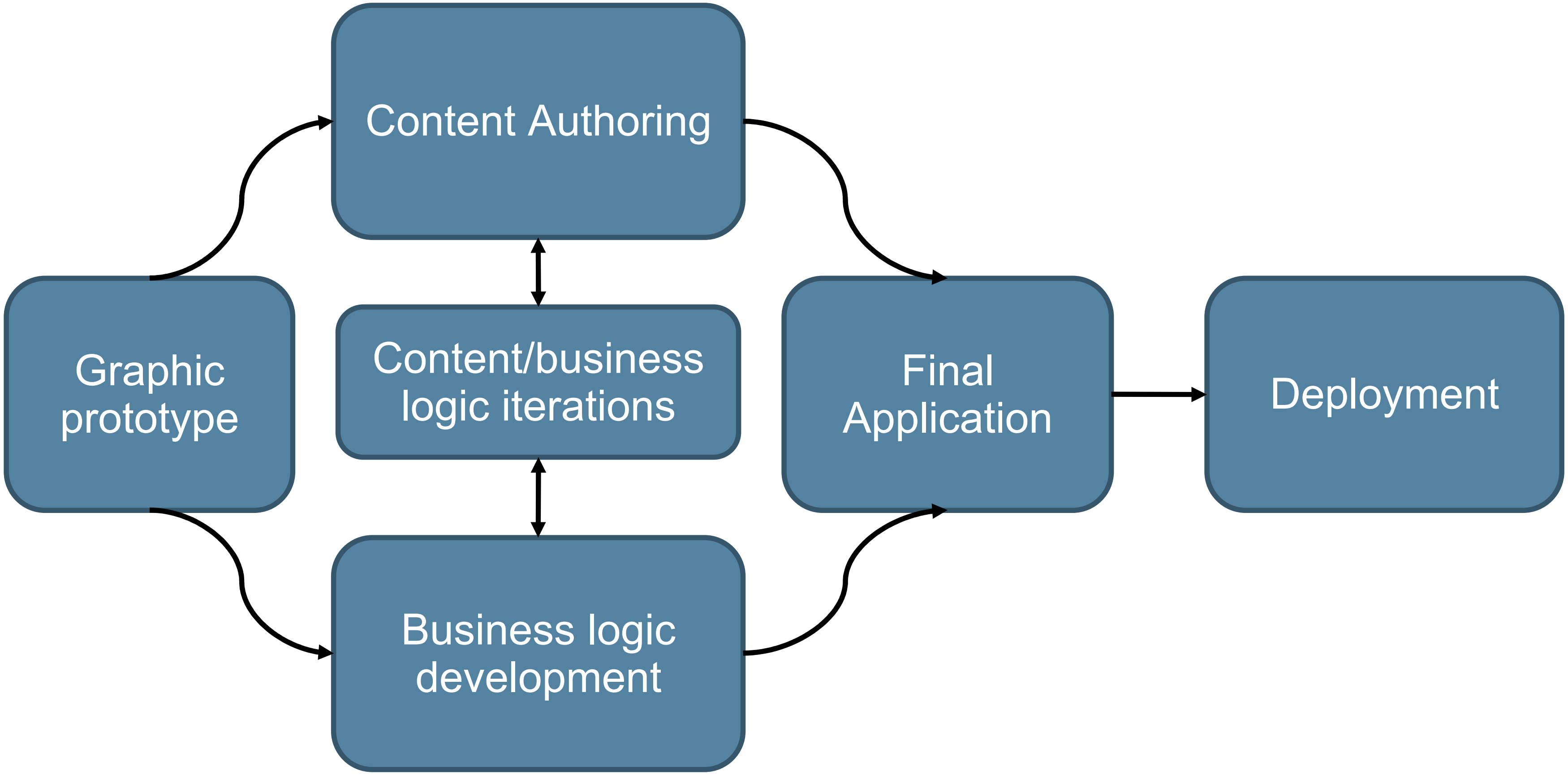
- Any IDE supporting Java ME platform is a good start
- SDK supporting JSR 226 is required:
  - Sun Java Wireless Toolkit 2.5
  - Nokia Series 40, 3rd edition, Feature Pack 2 SDK
  - Nokia S60, 3rd edition, Feature Pack 1 MIDP SDK
  - Others…
- A device supporting JSR 226:
  - Motorola, Nokia, Sony Ericsson…

Advanced Environment for Creating SVG/JSR 226 Applications

Combining the powers

- NetBeans™ Mobility Pack
  - Provides specific support handling SVG graphics and for developing JSR 226 applications
  - Integrates with SVG authoring tools
- Hyperion Mobile Designer
  - Provides support for creating interactive and animated SVG graphics
  - Based on NetBeans Platform
  - Integrates with NetBeans Mobility Pack

Typical Workflow for SVG/JSR 226 Application Development



DEMO

Developing SVG/JSR 226 Applications

Agenda

Introduction to SVG

JSR 226—Enabling SVG in Java ME applications

Tools for developing SVG/JSR 226 applications

What is coming

Summary

Q&A

What Is Coming

- More devices with JSR 226 support
- SVG Tiny Viewer 1.2
- JSR 287
- Advanced integration of NetBeans Mobility Pack and Hyperion Mobile Designer

Summary

- SVG is here!
- JSR 226 enables Java ME platform developers to easily use SVG content created by graphics artists
- SVG is already widely used for UIs on devices
- JSR 226 is part of MSA (JSR 248)—The next generation mobile platform
- First class tools for developing SVG/JSR 226 Java ME applications are available

For More Information

- Links:
  - SVG specification: http://www.w3.org/TR/SVG/
  - JSR 226 specification: http://www.jcp.org/en/jsr/detail?id=226
  - SVG.org homepage: http://www.svg.org
  - NetBeans IDE homepage: http://www.netbeans.org
- Other JavaOne Sessions:
  - TS-5743—Graphical, Scripted, and Animated User Interfaces on Java Platform, Micro Edition (Java ME)
  - TS-5626—Data Binding and Java ME

Q&A

Developing Flashy Mobile Applications Using SVG and JSR 226

Martin Brehovsky

Software Engineer
Sun Microsystems

Andrei Mihaila

Lead Engineer
Nokia Corporation

Marc Verstaen

Senior Director
Hyperion

TS-5628