Stylesheet Translations of SVG to VML

Student: Julie Nabong
Advisor: Dr. Chris Pollett
Committee Members:
  Dr. Agustin Araya
  Dr. Robert Chun

May 2004
Topics

- Introduction
  - Motivation
  - Objectives
  - Requirements
  - Approach
- Implementation Features
- Results
- Limitations
- Conclusion
- Demo
Introduction - Motivation

jpeg: 3001 b  gif: 3601 b  bmp: 150,742 b

svg: 274 b
Motivation (cont.)

- JPEG, GIF
  - longer download
  - zoom – lose quality
  - Paint, Macromedia,…

- SVG
  - smaller file size
  - zoom – same quality
  - text editor
Motivation (cont.)

SVG

- Scalable Vector Graphics
- W3C Recommendation
- Jan 2003
- two-dimensional graphics
- XML-based
Motivation (cont.)

- SVG
  - does not work in Netscape 6.01
  - can be viewed in I.E., but inconvenience
    - search viewer
    - download ~ 2MB viewer
    - install
Motivation (cont.)

- Solution – transform SVG to VML
  - works in I.E.
  - no extra software needed
  - text editor
  - XML
  - zoom – same quality
Motivation (cont.)

**VML**

- Vector Markup Language
- W3C Recommendation Candidate
- two-dimensional graphics
- XML-based
Project Objectives

- develop software
  - accepts SVG document input
  - outputs document with VML
- goal:
  - eliminate need for plug-in
  - display SVG transparently
Requirements

1. an SVG document – with supported tags
2. an XSLT stylesheet – *translator.xsl*
3. an HTML document – *display.html*
   a. loads SVG and stylesheet
   b. has transform method
   c. displays result
4. browser - I.E. 6.0
Approach

- World Wide Web Consortium (W3C)

- W3C Recommendations
XSLT

- eXtensible Stylesheet Language Transformation
- W3C Recommendation, Nov 1999
- XML-based
- Goal: transform XML documents
Approach (cont.)
Approach (cont.)

SVG

XSLT

XSLT Processor

browser

VML

web document
Approach (cont.)

SVG → XML Parser → browser

- svg
- circle
- line
- text
Approach (cont.)

XSLT File: Stylesheet

<..................JavaScript...............>
Approach (cont.)

XSLT

\{ instruction \}
\{ VML \}
\{ instruction \}
\{ VML \}
\{ instruction \}

output
Implementation Features

1. Direct mapping

2. No mapping

3. Scripting
   
   function getPoints(svgpoints) {... return points; }

4. Gradients

5. Events
Implementation Features – 1. Direct mapping
Implementation Features – 1. Direct mapping (cont.)

- Transformation of <rect> Element

SVG → Stylesheet → VML
Implementation Features – 2. No mapping

```xml
<g...>
  <rect../>
  <circle../>
  <polyline../>
</g>
```

SVG
Implementation Features – 2. No mapping (cont.)

1. check if parent is ‘g’
2. if yes, get info
3. if no, get rect’s info
4. render image
Implementation Features – 2. No mapping (cont.)

- Transformation of <g> Element

SVG → Stylesheet → VML
Implementation Features – 3. Scripting

<svg>
  <path points/>
  <v:curve/>
  <xsl:template>
    <......JavaScript......>
  </xsl:template>
</svg>

output
Implementation Features – 3. Scripting (cont.)

- Transformation of `<path>` Element
Implementation Features – 4. Gradients

- Gradient – smooth transition of one color to another

examples:

linear  radial
Implementation Features – 4. Gradients (cont.)

SVG radial gradient → Stylesheet → VML radial gradient

SVG circle to VML oval = OK, but…
Implementation Features – 4. Gradients (cont.)

SVG radial gradient → Stylesheet → VML radial gradient

SVG rectangle to VML rectangle = **not OK**
Implementation Features – 4. Gradients (cont.)

SVG → Stylesheet → VML

Solution
Implementation Features – 5. Events

- `<svg>` event
- `<script>` event
- XSLT output
- output
Results

- Stylesheet
  - 23 templates
  - JavaScript
  - 1,430 lines
Results (cont.)

- Experiment Data

<table>
<thead>
<tr>
<th>Num of Elements</th>
<th>Translation Time in Milliseconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>70</td>
</tr>
<tr>
<td>50</td>
<td>160</td>
</tr>
<tr>
<td>100</td>
<td>350</td>
</tr>
<tr>
<td>150</td>
<td>501</td>
</tr>
<tr>
<td>200</td>
<td>601</td>
</tr>
</tbody>
</table>
Results (cont.)

- Experiment Chart

![Chart showing translation time in milliseconds for different numbers of elements (circles, ellipses, rectangles). The x-axis represents the number of elements, and the y-axis represents the translation time in milliseconds. The chart shows a linear increase in translation time as the number of elements increases.]
Result (cont.)

- Transformation Snapshot
Results (cont.)

- project transforms SVG documents with the following elements:
  - geometric shapes
  - gradient fills
  - lines
  - scripts
  - events
  - etc.
Limitations

- no zooming on mouse click
- no dashed lines and arrows
- no shadows on images
- some SVG elements not supported
Conclusion

Points Learned:

1. need XML Path Language to use XSLT
2. XSLT
   a. unique language
   b. hard to debug
3. second transformation
4. need XML DOM
5. DOM2 Events not supported in I.E. 6.0
## References

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[DOM04]</td>
<td>W3C Document Object Model. <a href="http://www.w3.org/DOM/">http://www.w3.org/DOM/</a></td>
</tr>
</tbody>
</table>
## References (cont.)

<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[SVG03]</td>
<td>Scalable Vector Graphics (SVG) 1.1 Specification. <a href="http://www.w3.org/TR/SVG/">http://www.w3.org/TR/SVG/</a></td>
</tr>
<tr>
<td>[W04]</td>
<td>W3Schools. <a href="http://www.w3schools.com">http://www.w3schools.com</a></td>
</tr>
<tr>
<td>[XML03]</td>
<td>XML Path Language. <a href="http://www.w3.org/TR/xpath">http://www.w3.org/TR/xpath</a></td>
</tr>
<tr>
<td>[XSLT99]</td>
<td>XSL Transformations (XSLT). <a href="http://www.w3.org/TR/xslt">http://www.w3.org/TR/xslt</a></td>
</tr>
</tbody>
</table>
Demo
Thank You For Coming!

- Questions