SEDUX
Simple Event Display Using Xslt
A simple event display for the Atlas experiment at CERN
Sedux: Overview

• The Atlas experiment at CERN
• The Inner Detector
• Analysis and Event display
• Motivation
• Geometry
• Geometry Transform
• Event Files
• Trigonometrical Functions

• Transforming an Event
• Javascript for Pan and Zoom
• Extensions to the concept:
  • A web application
  • Additional geometries
  • Incorporating additional information
• Conclusion

Shaun Roe, SVG Open, August 2008
Detector:
25m Ø
46m length
7 000 t
~10^8 channels
3.2 Pbyte/yr

http://atlasexperiment.org/
2500 physicists
164 institutes

pictures: http://atlas.ch/

Shaun Roe, SVG Open, August 2008
One 12cm SCT module

The Inner Detector

Particles traversing the detector produce an electrical signal from which a ‘hit’ is deduced at a given position. 3D ‘spacepoints’ may be produced from hits.

<table>
<thead>
<tr>
<th></th>
<th>channels</th>
<th>resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>pixel</td>
<td>80 mi.</td>
<td>14 x 115 µm</td>
</tr>
<tr>
<td>SCT</td>
<td>6 mi.</td>
<td>22µm</td>
</tr>
<tr>
<td>TRT</td>
<td>400 k</td>
<td>170µm</td>
</tr>
</tbody>
</table>
Event analysis and display

Event data are analyzed by a C++ program, Athena. Analysis results are produced in the form of histograms in a binary format.

A ‘JiveXml’ option may be switched on to produce an XML representation of each event. This can be displayed by Atlantis, a Java program. (http://www.hep.ucl.ac.uk/atlas/atlantis/)
Sedux motivation

- During commissioning of the detector, users wanted to quickly see what event displays might be interesting.
- Events had simple topology
- The r-phi view is the most revealing
- Decided to use XSLT to transform the XML to SVG on-the-fly => direct display in a web browser
- Possibility to make modules into active links

Browser use:

<table>
<thead>
<tr>
<th>Browser</th>
<th>Platform</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firefox</td>
<td>Linux</td>
<td>76.8% (93)</td>
</tr>
<tr>
<td>Safari</td>
<td>Macintosh</td>
<td>15.7% (19)</td>
</tr>
<tr>
<td>IE7</td>
<td>Windows</td>
<td>7.4% (9)</td>
</tr>
</tbody>
</table>

- & Thought I could make a more appealing display :-(
Geometry description

These are ‘as designed’ dimensions
Transforming the geometry

- Prepare the geometry first, merge later
- Could use any language specific XML parser (e.g. libxml2)
- Decided to use XSLT2.0
  - total lines of code ~230
  - use extended XSLT functions for sin, cos

Repetitive graphic elements generated with XSLT 2.0 sequences.

```xml
<xsl:variable name="phiSequence" as="xs:integer*">
  <xsl:sequence select="for $i in 0 to $lastPhi return $i"/>
</xsl:variable>
```
Event ‘JiveXml’ files

• The transform of the event data has to be done in the browser => using XSLT 1.0

• Format problems:
  1. The points are presented as text lists of first x, then y, then z or even r then phi
  2. Can’t use extended XSLT trig functions to translate r, phi to x, y

```xml
<S3D count="15" storeGateKey=""
... identifier info omitted ...
</x>
-9.808979 -7.996168 24.583413 23.176761
22.379122 -13.450560 -12.680121 -12.613973
-12.551715 26.821852 26.190418
25.515184 -17.471010 32.997142 -20.881114
</x>
<y>
30.365820 -34.422605 -34.606149 -34.621908
-34.636740 35.695123 35.978239
36.280993 -41.291724 40.313217 -47.165986
</y>
<z>
36.022580 32.086557 23.600728 23.432253
22.893846 37.440609 41.358268 39.659175
38.060028 20.829506 21.015146
21.499875 38.969130 19.912280 40.171093
</z>
</S3D>
other detector specific info is also present
```
Trigonometry functions

- Use Taylor expansion of sin, and map cos onto same function using trig identities.

Want a 10cm diameter display to display a straight line ~ perfectly => use a 5 term expansion

(error on sin approximation calculated by xsltproc)
Transforming the event

• CSS used to determine colours, can be overwritten by using a local CSS

• user right-clicks on JiveXml file to ‘open with’ Firefox/Safari (~2s render); the JiveXml file is associated with a stylesheet which needs to be in the same directory, as does the svg for geometry

niggles: a) Firefox enforces the ‘same origin’ policy, Safari does not. If a local geometry file is not found by Safari, it can load it from a url. Firefox displays a blank screen...
b) Text variations between browsers

one of the first cosmic rays seen in the Atlas inner tracker
Pan and Zoom; Javascript

- The overall SVG is scaled so the detector fits into a 10cm window with origin at the display centre.

- Pan and Zoom alter these directly in the DOM. The Javascript is injected during the geometry transform.

```html
<svg viewBox="{$viewXMin} {$viewYMin} {$viewWidth} {$viewHeight}" height="20cm"
    width="20cm">
    <script type="text/ecmascript">
      <![CDATA[
        var scale0 = 0.8;
        var dx0 = 0;
        var dy0 = 0;
        function changeScale(multiple) {
          var picture = document.getElementById("detectorPicture");
          var newScale = scale0 * multiple;
          scale0 = newScale;
          newTransform = "scale(" + newScale + ", -" + newScale + ") translate(" + dx0 + "," + dy0 + ")";
          picture.setAttribute("transform",newTransform);
        }
      ]]>}
    </script>
    <g id="ZoomIn" class="buttons" onclick="changeScale(1.25)"
      >
      <circle r="{3 * $unit}" cx="0" cy="0"/>
      <path d="M 0,{2 * $unit} L 0,-{2 * $unit} M -{2 * $unit},0 L {2 * $unit},0"/>
    </g>
</svg>
```

closeup of a cosmic shower
Extending the concept

1. Pure web application
   no downloads; user simply goes to the web page and uploads his/her file, which is then displayed
   7 lines of code!

   ```php
   <?php header("Content-type: text/xml");
   @$ftmp = $_FILES['datafile']['tmp_name'];
   $fp = fopen($ftmp, 'r');
   $content = fread($fp, filesize($ftmp));
   fclose($fp);
   echo $content;
   ?>
   ```

2. Additional geometry views
   e.g. an endcap quadrant:

![ATLAS Sedux Run 7269, event 4 diagram]
Detector Control System provides additional information based on detector component position and time

RESTful database interface (based on CherryPy) provides status information in XML

Will integrate this with the event display
Conclusions

• SVG has been demonstrated to be a useful, lightweight event display technology

• With XSLT, never fails to produce a ‘wow!’ effect making it more appealing (despite the drawbacks inherent in XSLT i.e. lack of scientific functions)

• Could find much wider application to replace existing methods of producing web-immediate plots

• Needs a few SVG/XSLT missionaries to convert the masses.

http://sroe.home.cern.ch/sroe/display/example.xml.