Web Technologies

Presenting Information

Why Web Technologies?

- Presentation of findings in an interactive way
  - Animation with, e.g., Flash
  - Graphical user interface with, e.g., TclTk
  - Dynamic Web pages with, e.g., JavaScript
  - Interactivity with XML, e.g. KML for Google Earth and SVG in browser

Our choice of topics is not meant to imply endorsement of this approach over others

Alternative Mode of Presentation

- Growing demand for interactive presentations of data
- Example: GapMinder, where time moved from the x-axis to the time domain, i.e. animation
- Public expects data to be presented in a way that they can interact with
- Opportunity to bring research into the public forum

Examples

- Google Earth presentation of analysis of elephant seal journey
- SVG interactive plot in geo-location analysis
- Interactive Web page – Communication between SVG plot and Google Earth embedded in Web page for presenting earthquake data

XML Concepts

- Declarative language, rather than procedural
  - Describe *what* the program should accomplish, rather than *how* to go about accomplishing it.
- Separate content from form
  - Separate what it is, e.g. section, from how to display/format it
- Structure hierarchical
  - Basic unit – element, node, chunk
  - Needed for signifying nesting

XML

- XML is a meta-language facility for defining markup languages
- Framework for supplying meta-information to data
  - Elements are delimited by tags
  - Tags open and close the elements – e.g. <coordinates>-122.08,37.42,0</coordinates>
  - Attributes supply additional information about the element, e.g. <Cube currency = "CZK">

Examples of XML grammars

- KML – keyhole Markup Language: geographic annotation and visualization on Internet-based maps and 3D earth browsers
- DocBook – semantic markup language for technical publications
- docx, xlsx and pptx – XML-based formats for office suite
- SGML, CSML, SDMX, …

Well-formed XML

- Elements – start with an open tag <para> and end with a closing tag </para>
- Document must have one root element
- Tag names are case sensitive
- Attribute values must appear in quotes
- Restrictions on tag names

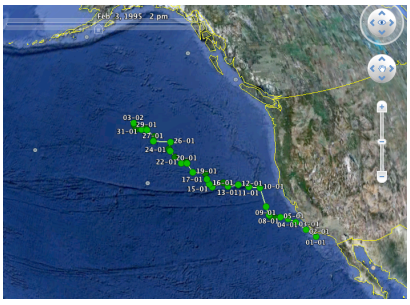
Pros

- Standard tools can be used to parse
- Highly extensible – add your own elements to a grammar
- Easily machine generated *and* plain text
- Separates content from form
- Supports complex data structures

Cons

- Tends to be very verbose
- Unforgiving format

Google Earth - elephant seal journey



Why KML?

- A grammar of XML in widespread use – e.g. GoogleEarth
- Progression from HTML presentation of earlier work (e.g. deconstruct-reconstruct HW)
- Build on understanding of a tree, e.g. file system, lists in R
- KML as an alternative language

Introduction to KML for students

- Basic concepts
- Short set of tags
- KML tutorial – leave it to the students to fill in the details
  - http://code.google.com/apis/kml/documentation/kml_tut.html
  - http://code.google.com/apis/kml/documentation/kmlreference.html

KML - Placemark

The Placemark element is one of the most commonly used features. It marks a position on the Earth's surface.

The simplest Placemark includes only a <Point> element, which specifies the location of the Placemark.

Placemark Example

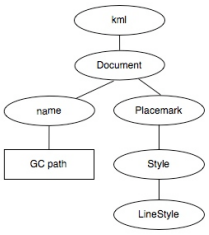
```
<Placemark>
  <name>Simple placemark</name>
  <description>Attached to the ground. Places itself at the height of the terrain.
  </description>
  <Point>
    <coordinates>-122.08,37.42,0</coordinates>
  </Point>
</Placemark>
```

Other KML notions

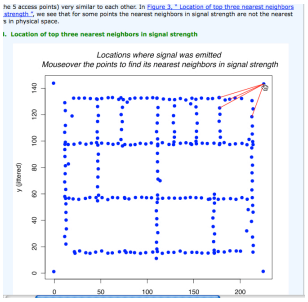
- Ground overlay: The <GroundOverlay> drapes an image onto the Earth's terrain
- Path: made with <LineString>, a child of <Placemark>
- Polygon: the <Polygon> tag is also a child of <Placemark>
- Styles: <LineStyle> for paths and <IconStyle> for points. All are children of <Style>

Create KML programmatically

```
kml = newXMLNode("kml")
doc = newXMLNode("Document", parent=kml)
newXMLNode("name", "GC path", parent=doc)
pm = newXMLNode("Placemark", parent=doc)
style = newXMLNode("Style", parent=pm)
lsty = newXMLNode("LineStyle", parent=style)
...
saveXML(kml,"~/Documents/greatCircle.kml")
```



Nearest-Neighbor in geo-location



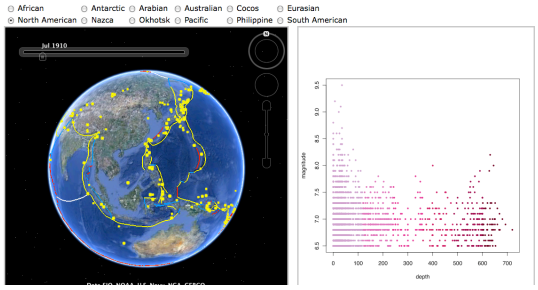
Why Scalable Vector Graphics (SVG)?

- XML format for describing 2-D graphical displays
- Supports interactivity, animation, and filters for special effects
- Vector-based system that describes an image as a series of geometric shapes. (not raster based that uses pixels)
- Shapes infinitely scalable because of vector descriptions
- Many commonly used Web browsers support SVG
- Can be included in HTML and PDF
- R has an SVG graphics device

Features of SVG

- Root tag: <svg>
- Basic shapes are provided via: <line>, <rect>, <circle>, <ellipse>, and <polygon>
- <path> tag provides the information needed to draw a curve
- Attributes support mouse events, e.g. onmouseover = ”Javascript call”
- Supports animation

Interactive Web page – 100 years of Earthquakes



Why JavaScript?

- Interactivity in the browser
- Optimized for use in the browser
- Widely used
- Exposure to a different language
  - Object oriented reference-based language
  - Asynchronous, event-driven programming
- Students will not confuse it as an alternative to R because it is for a different purpose
- Interpreted scripting language

Interactivity

- Used in HTML to respond to user actions on buttons, menus, checkboxes, etc. in HTML forms
  - in-line within <script> elements within HTML/SVG
  - value of attributes of HTML or SVG element
- Can dynamically and programmatically construct HTML and SVG

Computational model

- Similar to C++, Python and Java in its computational model.
- Much of its use focuses on classes and instances (objects) of these classes and their methods
- Not a vectorized language
- Requires variables to be declared within the scope in which they are used

Syntax

- Executable statements end with a semicolon
- Variables are declared via the var prefix

```
var global = 1;
```

- Curly braces group JavaScript statements together into executable blocks

```
if (x < 2) {
 ...
} else if( x > 2 && x < 10) {
 ...
} else ...
```

(Control flow and comparison operators are similar to R)

Variables

- Supports arrays, including multi-dimensional, ragged arrays
  - uses 0-based indexing for arrays
  - e.g. expression x[1][0] returns the first element in the second array
- equal sign (=) is the assignment operator.
  - x += y is equivalent to the assignment x = x + y
  - -=, *=, and /= are similarly defined.

JavaScript functions

- functions are defined using the keyword function as a prefix to the definition as in

```
function functionName(var1, var2, ..., varN)
{
body-code
}
```

Benefits to Student

- Opportunity for students to practice "learning how to learn"
  - Give them a brief introduction to the technical subject and then let them loose to learn about it
  - Must figure out how to combine modern methods, visualization, programming in a new, modern venue
- Learn practical tools that can be used in real settings

Benefits to Student

- Students can showcase their work, e.g. create portfolio of their projects, which can help them get jobs
- In the future, when in the workforce, our students will spend a lot of their time preparing data for presentation (as opposed to applying methods to data)

Plus…

- It is a lot of fun for the students
- Students get to be creative
- Capstone to students' work in the course

Hans Rosling's musings

- We should work on being as exportable and importable as possible
- Innovate the interface – e.g. use animation to communicate time
- Sharing information does not lead to understanding
- Each stat department should have a stable URL for best student work
- Need to put people together into teams: code, stats, design
- Need to put students in a position to innovate and communicate – tell a story
- Keep on-line tools explorative, don't make a weak analytic tool, use R instead
- R is great but if you want to teach the broad community, use Flash and put on the Web (unfortunately can't be imported into ppt)
- Create a competition for best video presentation of information
- 2-types of video watchers: lean-forward and dig deeper vs lean backward, push button once and sip a latte. Don't just develop for the first kind of watcher