Map.Harvard.Edu

The 21st Century Harvard Map

www.map.harvard.edu/mapserver/campusmap.htm

NEARC

May 11, 2010

Presented by Peter Siebert

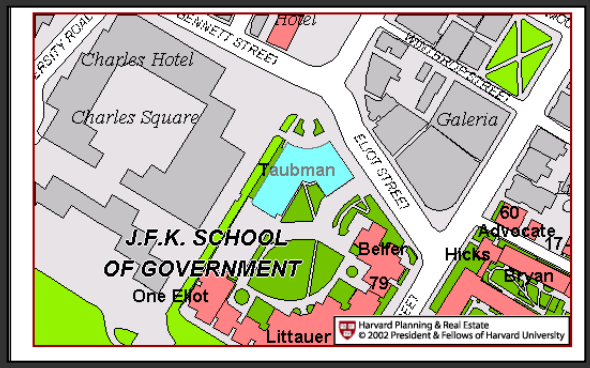
Harvard University Planning Office

peter_siebert@harvard.edu

The Harvard On-line Map: first 10 years

Initial Web Mapping - 1997

- AutoCAD + Autodesk MapGuide Server
  - Full functionality: navigation, zoom, roll-overs
  - Complicated map authoring
  - Required plug-in



Current Harvard Map – 2003 to present

- Cold Fusion back end for navigation and database
- Simulated interactivity
- Image tiles: ArcMap >PDF >Adobe Illustrator >GIF

Choosing a new platform

Why not just use Google?

- "Value added": Accuracy, Currency, Specificity
- Control the user experience – look and feel
- Beyond Google, beyond maps: data delivery to The Cloud

Map service architecture

- Internet feeds for the GIS desktop
- Packaged data for web developers

Wish list for the new architecture

Simplify authoring

- Use existing GIS data & composition tools (ArcMap)

Variety of delivery protocols

- ArcGIS Services, WMS, KML
- Consumer chooses best match

"Main Stream" technology

- Support, User community, Maintenance

Good performance

- Tile generation and Caching where appropriate

Make it "Googly"

✓Simplified authoring

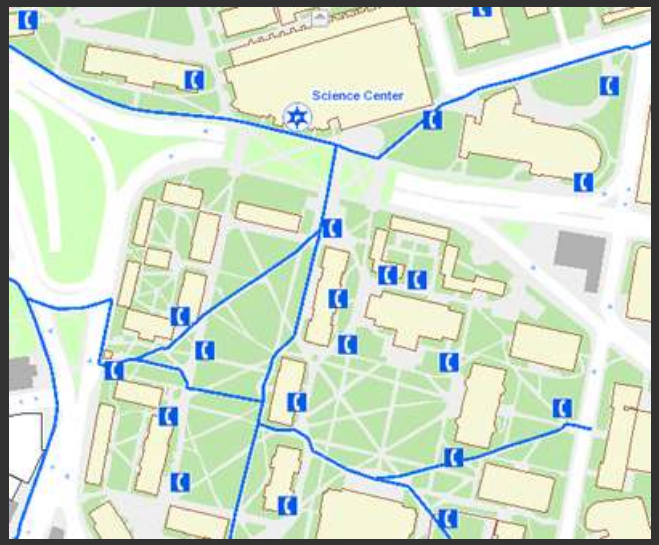
Maps generated directly from ArcMap

- Map service creation/updating is quick

No need for manually coding symbology or rendering (see ArcIMS)

- Programmatic control of rendering is optional

This:



Not this:

```
<LAYER type="featureclass" name="Accessible Paths" visible="true" id="PATHS">
    <DATASET name="Access_paths" type="line" workspace="shp_ws-0" />
    <GROUPRENDERER>
    <SCALEDEPENDENTRENDERER upper="1:10799" lower="1:7200">
    <VALUEMAPRENDERER lookupfield="Layer">
      <EXACT value="APATH" label="APATH">
        <SIMPLELINESYMBOL width="2" captype="round" color="0,0,255" antialiasing="true"/>
      </EXACT>
    </VALUEMAPRENDERER>
    </SCALEDEPENDENTRENDERER>
    <SCALEDEPENDENTRENDERER upper="1:7199">
    <VALUEMAPRENDERER lookupfield="Layer">
      <EXACT value="APATH" label="APATH">
        <SIMPLELINESYMBOL width="2" captype="round" color="0,0,255" antialiasing="true"/>
      </EXACT>
    </VALUEMAPRENDERER>
    </SCALEDEPENDENTRENDERER>
    </GROUPRENDERER>
</LAYER>

<LAYER type="featureclass" name="Construction" visible="true" id="CONSTRUCTION">
    <DATASET name="Construction" type="polygon" workspace="shp_ws-0" />
    <SCALEDEPENDENTRENDERER upper="1:10801" >
    <SIMPLERENDERER>
      <SIMPLEPOLYGONSYMBOL boundarywidth="10" boundarytransparency=".0" boundarycolor="25
fillcolor="255,255,255" boundarycaptype="round" antialiasing="true" />
    </SIMPLERENDERER>
    </SCALEDEPENDENTRENDERER>
</LAYER>

    <LAYER type="featureclass" name="Basetext" visible="true" id="TXT">
```

✓Full complement of geo services

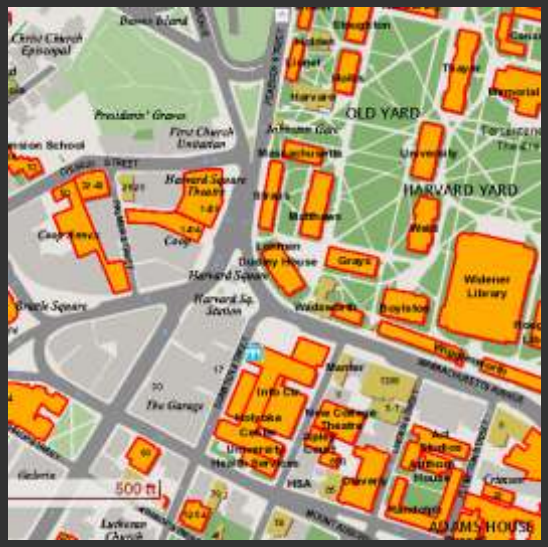
Variety of clients

- Desktop (ArcMap), Web, Mobile

Variety of services

- Maps, imagery, geoprocessing
- Native and Open standards (WMS/WFS)
- KML
- Geocoding, Geoprocessing, Routing
- 3D (ArcGlobe)

Map Service



KML Service

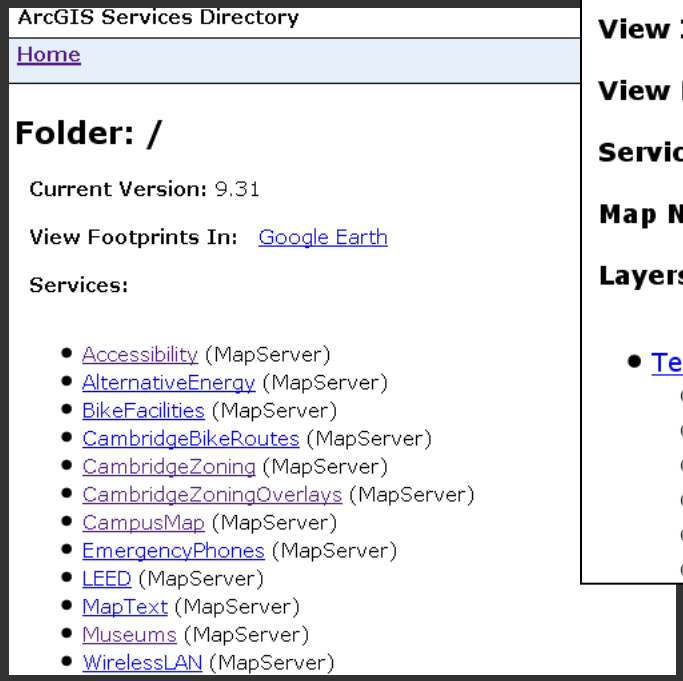
✓Good developer support

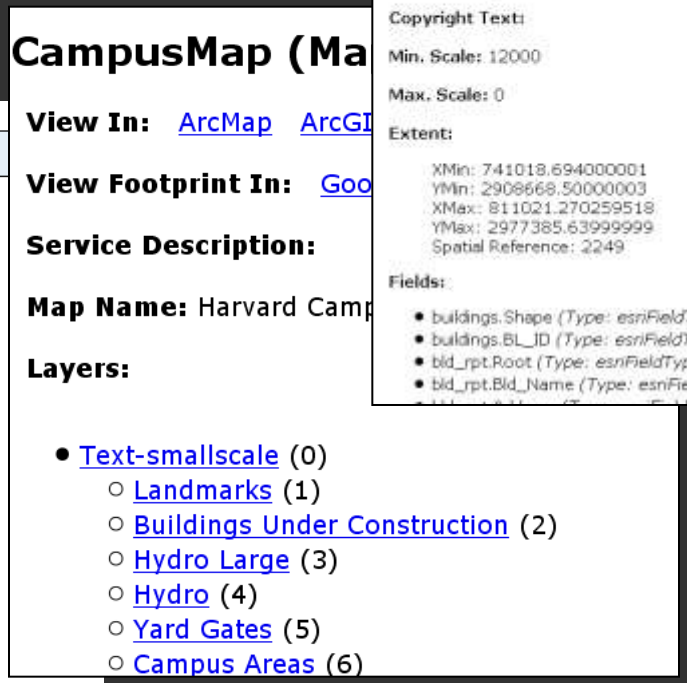
APIs reasonably well documented

Good selection of sample code

Large developer community

Easy to explore map service details





Which API?

.NET, Java

- Take advantage of server-side functionality
- Probably most feature rich

Flex, Silverlight

- Great for apps with non-mapping elements (charting, animation)
- Reliant on availability of client-side plug-ins

Google

- Wide adoption, familiar interface
- Simple UI - No need to program controls
- Global street or image base map

Javascript

- Essentially universally supported
- Dojo and other libraries offer many widgets and etc.
- Stateless, asynchronous, client-side processing presents challenges

Application design and functionality goals

Clean interface

- Minimal tool clutter
- Not a GIS

Maximize map area

- Automatically fill the screen

Multiple data layers

- Value added
- Requires interface for selection

Web map API for linking, embedding

Easily configurable

Demo

Current beta release: http://map.harvard.edu/mapserver/campusmap.htm

Release candidate: http://map.harvard.edu/mapserver/campusmapv5.htm

"Bare map": http://map.harvard.edu/mapserver/map.htm

http://map.harvard.edu/mapserver/renewable.htm

Technical design elements

User Interface: Standard HMTL, CSS, JS

- Adaptable for alternative apps, embeded maps, mobile
- Dojo adds some “syntactical sugar” and widgets

Map “Themes” as discreet services

- Easier maintenance
- Web developers can pick and choose

Configuration file connects the two

- Service parameters: URL, transparency, click tolerance
- Layer parameters
  - Order, visibility, legend info, fields

Current map status

Fine tuning

- Tinkering with page design
- Additional data layers: Libraries, Event venues, Trees, Parking, Dining facilities

Rounding out core functionality

- Printing/Export
- Metadata

Enhancements

- Queries
- Theme ordering
- Autocompletion for Search
- Mobile compatible version

Our experience so far

The Bad

- Specific functionality missing in JS API: Legends, Printing/Image export
- Occasional bugs (zoom slider, layer visible) and documentation omissions
- Appearance of special widgets (zoom, pan, info window) difficult to customize
- Combining multiple services presents challenges

The Good

- Fidelity of map service to MXD: WYSIWIG
- Overall good performance
- Dynamic labeling
- Good documentation with examples
- Strong vendor commitment to support and on-going development

The Ugly?