GeoMoose and Describing Maps on the Web

Dan “Ducky” Little / @theduckylittle

21 September 2020

What is GeoMoose

GeoMoose is a WebGIS that was originally developed at the City of Saint Paul, Minnesota, USA. The first release was in 2004 with an introduction made at the 2005 MapServer User Meeting.

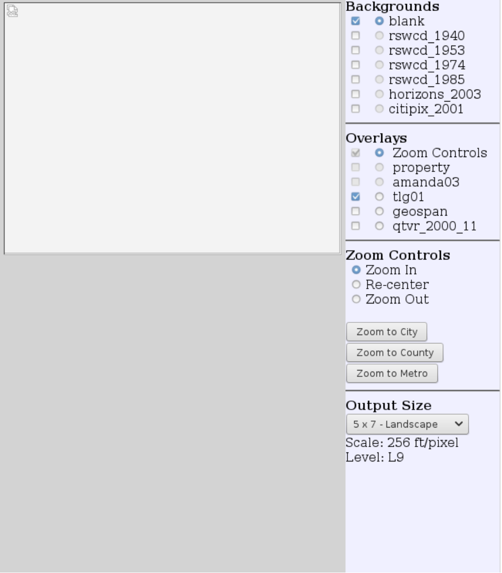
Features through the years

- Popups using Image-maps
- Detailed measure tools
- Connections to OGC services
- Support for an increasing number of "web friendly" formats: GML, GeoJSON, Mapbox Vector Tiles
- Two way communication (editing) via WFS-T

Initial goals

- Talk to MapServer!
- Use all the cool features of IE6!
- Create a user experience that enabled the organization, display, and selection of hundreds of layers.

Genesis, GISmo "v0"



Backgrounds

- ☑ ● blank
- □ ○ rswcd_1940
- □ ○ rswcd_1953
- □ ○ rswcd_1974
- □ ○ rswcd_1985
- □ ○ horizons_2003
- □ ○ citipix_2001

Overlays

- ☑ ● Zoom Controls
- □ ○ property
- □ ○ amanda03
- ☑ ○ tlg01
- □ ○ geospan
- □ ○ qtvr_2000_11

Zoom Controls

- ● Zoom In
- ○ Re-center
- ○ Zoom Out

Output Size

5 x 7 - Landscape

Scale: 256 ft/pixel
Level: L9

Figure 1: First version of GISmo

The early days (2003 - 2007)

- We started with data!
- MapServer is configured using Mapfiles.
- For independence of editors, we considered every Mapfile to be a “layer”.
- Just like the real world collections we put every “layer” into a “mapbook”
- XML provided us with a way to define the mapbook, creating the first “mapbook” XML format!
- GeoMoose itself was really a fancy XSLT with a big Javascript file next to it

What did this look like!

Nothing is more fun than a code snippet:

```
<mapbook>
  <catalog>
    <layer label="Parcels" mapfile="./parcels/parcels.map" status="on" />

    <group exclusive="true" label="Backgrounds">
      <layer
        label="2005 FSA"
        mapfile="./aerials/2005-fsa.map" status="off"
      />
      <layer
        label="2004 Ramsey County Hi-Rez"
        mapfile="./aerials/2004-hirez.map" status="off"
      />
      <layer
        label="1990 B&W Scans"
        mapfile="./aerials/1990-bw.map" status="off"
      />
    </group>
  </catalog>
</mapbook>
```

Eventually we got it pretty

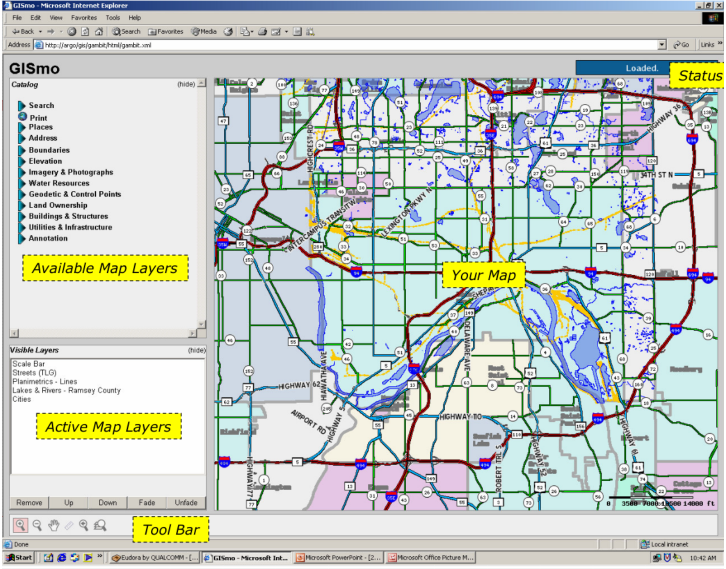
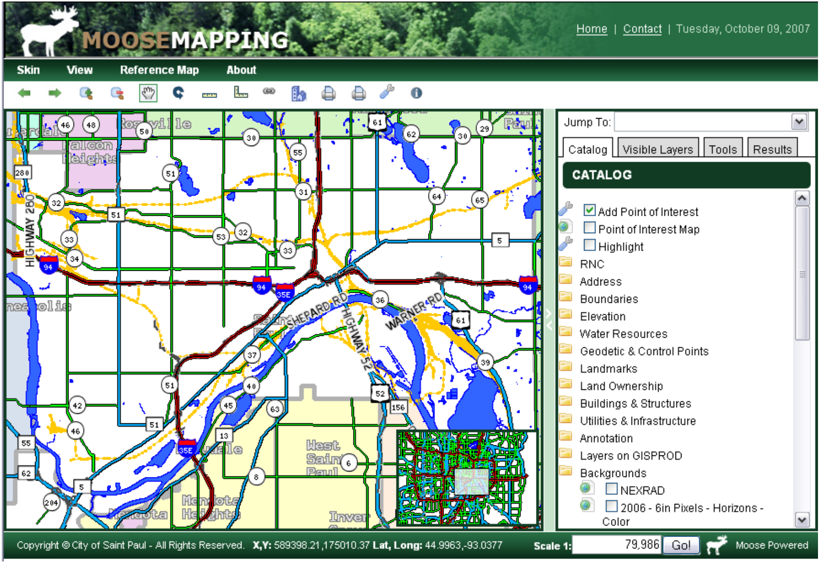


Figure 2: Look at that bottom toolbar! GISmo v1.0

We evolved it



☑ Add Point of Interest

□ Point of Interest Map

□ Highlight

RNC

Address

Boundaries

Elevation

Water Resources

Geodetic & Control Points

Landmarks

Land Ownership

Buildings & Structures

Utilities & Infrastructure

Annotation

Layers on GISPROD

Backgrounds

□ NEXRAD

□ 2006 - 6in Pixels - Horizons - Color

Copyright © City of Saint Paul - All Rights Reserved. **X,Y:** 589398.21,175010.37 **Lat, Long:** 44.9963,-93.0377 **Scale 1:** 79,986

Figure 3: GeoMoose 1.x

What ruled about this

- It was fast to render. Even on old hardware, in limited browsers, we could mapbooks of 200-ish layers.
- There was a great 1:1 between what was in the catalog in the XML and what the user would see in the interface.
- Data defined the user interface! It was structured!

But there were problems

- The order in the catalog was the order seen in the map. That means putting the "backgrounds" on the top would lay backgrounds over the map.
- There were not ways of defining different sources for maps.
- Every layer was an HTTP request. 10 layers on? 10 `<img>` tags were necessary to render it.

GeoMoose 2.x (2007-2017)

- GeoMoose 1.x was built using a custom mapping library.
- At this point, OpenLayers had caught up with our internal map library.
- Plus, it had much better *cross-browser* support. In 2007, it was still somewhat acceptable to target a SINGLE browser (usually IE). We targeted IE 6, late versions of ***Netscape*** and early versions of Firefox.
- Early versions of Javascript classes were available (AMD, Dojo)
- **We needed to fix the relation between a LAYER and a SOURCE.**

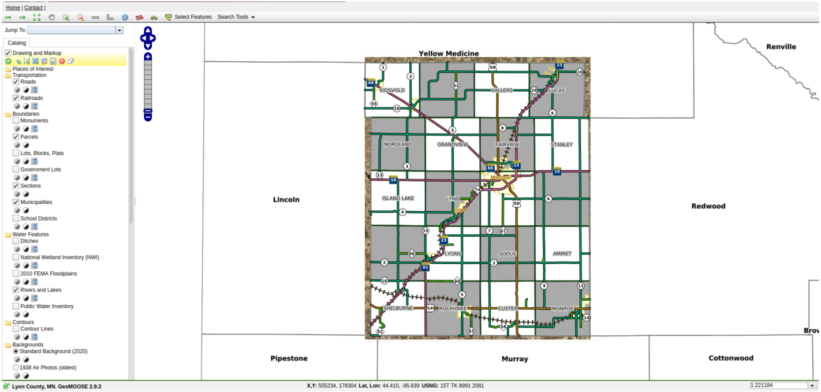
More XML!

The map source:

```
<mapbook version="2.0">
  <map-source name="firestations" type="mapserver">
    <file>./demo/firestations/firestations.map</file>
    <layer name="firestations" />
  </map-source>

  <catalog>
    <layer
      label="Firestations"
      path="firestations/firestations" status="off"
    />
  </catalog>
</mapbook>
```

And all was good



☑ Drawing and Markup
Places of Interest
Transportation
☑ Roads
☑ Railroads
Boundaries
☐ Monuments
☑ Parcels
☐ Lots, Blocks, Plats
☐ Government Lots
☑ Sections
☑ Municipalities
☐ School Districts
Water Features
☐ Ditches
☐ National Wetland Inventory (NWI)
☐ 2010 FEMA Floodplains
☑ Rivers and Lakes
☐ Public Water Inventory
Contours
☐ Contour Lines
Backgrounds
● Standard Background (2020)
○ 1938 Air Photos (oldest)

Lyon County, MN. GeoMOOSE 2.9.3

X,Y: 555234, 178304 **Lat, Lon:** 44.410, -95.639 **USNG:** 15T TK 8991 2081

1:221184

Figure 4: All the OpenLayers you can handle

Rough bits

- Custom Java compiler to stitch the classes together.
- Components would not render properly on modern browsers.
- Functional programming models were not available.
- Mapbooks got unwieldy as we attempted to put a full application configuration, including layouts, into the same file.

Some notable events

- Google tiled the whole world.
- Everyone put a super computer in their hands.
- OSM has become the reference map data in many areas of the world.
- Vector data has gone to the web!
  - GeoJSON
  - Mapbox vector tiles
- Cloud first GIS with Cloud Optimized GeoTIFF, Streamable JSON, Cheap “cloud compute” and “access cost” storage for terabytes of open data.

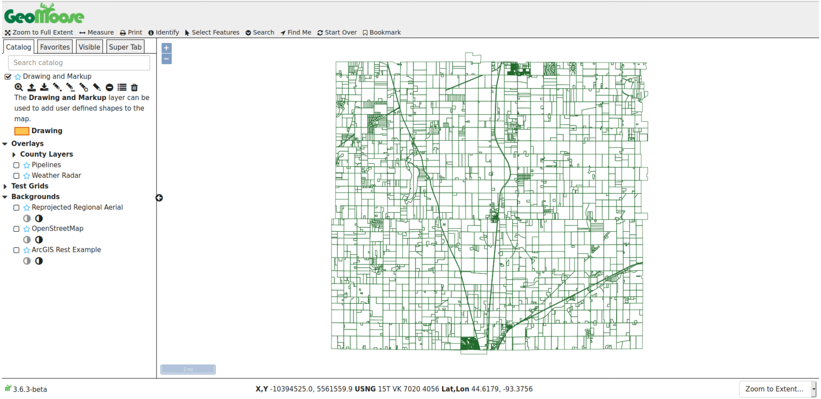
GeoMoose 3.x (2016-present)

- Attempts to address a number of new challenges.
- Sometimes better than others.
- Uses a “standard” stack: React, Webpack, OpenLayers
- Still defines the map and catalog as XML!

So fresh, so clean clean

Figure 5: So fresh, so clean clean

Sorry, GeoMoose 3. . . so fresh and clean



The **Drawing and Markup** layer can be used to add user defined shapes to the map.

X,Y -10394525.0, 5561559.9 **USNG** 15T VK 7020 4056 **Lat,Lon** 44.6179, -93.3756

Figure 6: Wow, such web app

The sometimes better

- We've created cleaner definitions for defining the sources.
- `query-as` is a concept we developed to allowing a different type of `map-source` for display versus query.
- Support native parser for GeoJSON and MVT!

The sometimes others

- Cartography is hard. MapCSS, SLD, MapBox GL Stylesheets, OpenLayers Styling, MapServer Mapfiles
- Performance is hard. The browser does not have a native way of taking in vector data and rendering it efficiently. (1)
- Complex definitions are hard. They `query-as` layers can get hard to explain.

1. *Yes this is still true when using WebGL.*

Future challenges are nearer than expected

- 3D. Especially in the form of Point Clouds are already a reality.
- Mobile. Scaling a map, styles, and interactivity changes.
- Dynamic rasters. More bit depth, more than RGB.

Big Take-Aways

- You can XML it!
- The `map-source` vs `layer` distinction is *very very* important.
- Styling is hard and needs to be considered.
- Someone will try to load multiple gigabytes of data and render it.

Fin.

Thank you!