SVG Web Mapping

Four-dimensional visualization of time- and geobased data

SVGOpen 2008 in Nuremberg
27th of August, 2008
Dipl.-Math. Dorothee Langfeld
Universitiy of Osnabrück
Content

- Web Mapping Application
  - Template
  - Client Server communication
  - Interaction and Animation with ECMAScript
  - Examples
- Java Program
  - Configuration
  - Different base data
  - Data processing
- Improvements
- Conclusion
Template Concept

- skeletal structure, filled with data at runtime
  - three basic parts
    - Structure
    - Data
    - Application logic
  - display for maps and georeferenced data
    - overview map
    - interaction elements for navigation and zooming
    - additional interaction elements
Template Concept

- skeletal structure, filled with data at runtime
- three basic parts
  - Structure
  - Data
  - Application logic
Server Client Communication

Server

Client
Server Client Communication

Server

- SVG template

Client
Server Client Communication

Server
- SVG template
- ECMAScript (creating navigation icons, filling the template etc.)

Client
# Server Client Communication

<table>
<thead>
<tr>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVG template</td>
</tr>
<tr>
<td>ECMAScript (creating navigation icons, filling the template etc.)</td>
</tr>
<tr>
<td>PHP Script (calculating the data which has to be loaded)</td>
</tr>
</tbody>
</table>

| Client |
# Server Client Communication

<table>
<thead>
<tr>
<th><strong>Server</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>SVG template</td>
</tr>
<tr>
<td>ECMAScript (creating navigation icons, filling the template etc.)</td>
</tr>
<tr>
<td>PHP Script (calculating the data which has to be loaded)</td>
</tr>
<tr>
<td>SVG Fragments (tiles)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Client</strong></th>
</tr>
</thead>
</table>
Server Client Communication

**Server**

- SVG template
- ECMAScript (creating navigation icons, filling the template etc.)
- PHP Script (calculating the data which has to be loaded)
- SVG Fragments (tiles)

**Client**

loading template
## Server Client Communication

### Server
- SVG template
- ECMAScript (creating navigation icons, filling the template etc.)
- PHP Script (calculating the data which has to be loaded)
- SVG Fragments (tiles)

### Client
- SVG template
Server Client Communication

<table>
<thead>
<tr>
<th>Server</th>
<th>Loading Script</th>
<th>Client</th>
</tr>
</thead>
<tbody>
<tr>
<td>• SVG template</td>
<td></td>
<td>• SVG template</td>
</tr>
<tr>
<td>• ECMAScript (creating navigation icons, filling the template etc.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PHP Script (calculating the data which has to be loaded)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• SVG Fragments (tiles)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Server Client Communication

**Server**
- SVG template
- ECMAScript (creating navigation icons, filling the template etc.)
- PHP Script (calculating the data which has to be loaded)
- SVG Fragments (tiles)

**Client**
- SVG template
- ECMAScript (navigation and loading data)
Server Client Communication

Server
- SVG template
- ECMAScript (creating navigation icons, filling the template etc.)
- PHP Script (calculating the data which has to be loaded)
- SVG Fragments (tiles)

Client
- SVG template
- ECMAScript (navigation and loading data)

sending query
## Server Client Communication

### Server

| PHP script calculates the data which has to be loaded by using viewBox-coordinates, generates XML as answer |

### Client

| SVG template |
| ECMAScript (navigation and loading data) |

---
Server Client Communication

**Server**
- SVG template
- ECMAScript (creating navigation icons, filling the template etc.)
- PHP Script (calculating the data which has to be loaded)
- SVG Fragments (tiles)

**Client**
- SVG template
- ECMAScript (navigation and loading data)

receiving data
## Server Client Communication

### Server
- SVG template
- ECMAScript (creating navigation icons, filling the template etc.)
- PHP Script (calculating the data which has to be loaded)
- SVG Fragments (tiles)

### Client
- SVG template
- ECMAScript (navigation and loading data)
- data
Interaction and Animation

- Realised with ECMAScript
- Possibilities of interaction
  - Zoom and Pan
  - Selecting a rectangular area
  - Layer
  - Choosing different time steps
  - Executing an automatic Animation
Interaction with ECMAScript

- DOM API
- Changing attributes like the `viewBox` for zoom and pan
- Loading data
  - `getURL` or `XMLHttpRequest` calling the PHP script
  - only the changed or new data will be the answer (AJAX concept)
Server side

- SVG Map divided into tiles, stored on the server
- PHP-Script with old and new viewBox coordinates as parameters
- Calculating the zoom step and the chosen region depending on the given viewBox
  - A bit more data than only the viewBox content to avoid waiting time while panning
Server side

- SVG Map divided into tiles, stored on the server
- PHP-Script with old and new viewBox coordinates as parameters
- Calculating the zoom step and the chosen region depending on the given viewBox
- A bit more data than only the viewBox content to avoid waiting time while panning

Answer as XML (only new data)
<data><delete/><append/></data>

IDs to delete
Fragments to append (own parent attribute)
Server side

- SVG Map divided into tiles, stored on the server
- PHP-Script with old and new viewBox coordinates as parameters
- Calculating the zoom step and the chosen region depending on the given viewBox
- A bit more data than only the viewBox content to avoid waiting time while panning

Answer as XML (only new data)

<data><delete/><append/></data>
Server side

- SVG Map divided into tiles, stored on the server
- PHP-Script with old and new viewBox coordinates as parameters
- Calculating the zoom step and the chosen region depending on the given viewBox
- A bit more data than only the viewBox content to avoid waiting time while panning

Answer as XML (only new data)

<data><delete/><append/></data>

IDs to delete
Fragments to append (own parent attribute)
Server side

- SVG Map devided into tiles, stored on the server
- PHP-Script with old and new viewBox coordinates as parameters
- Calculating the zoom step and the chosen region depending on the given viewBox
- A bit more data than only the viewBox content to avoid waiting time while panning

<data><delete/><append/></data>

IDs to delete

Fragments to append (own parent attribute)
Server side

- SVG Map divided into tiles, stored on the server
- PHP-Script with old and new viewBox coordinates as parameters
- Calculating the zoom step and the chosen region depending on the given viewBox
  - A bit more data than only the viewBox content to avoid waiting time while panning
- Answer as XML (only new data)
  - <data><delete/> <append/></data>
  - IDs to delete
  - Fragments to append (own parent attribute)
Interaction with ECMAScript

- Loading data after receiving the answer
  - `getElementById`, `parentNode`, `deleteChild`
  - `deleteAttribute`, `importNode`, `getElementById`, `appendChild`
- Animation realised with `setTimeout`
  - Existing time steps are stored in an array
  - Similar query as described before, additional parameter for adapted PHP-script
- Four dimensions:
  - the two-dimensional geography
  - the overlaying data
  - the time
Examples

- Map of Osnabrück
Examples
Examples

Map of Osnabrück

Weather data

Traffic data

- Grünflächen
- Gewässer
- Straßen
- Straßennamen
- Points of Interest

Status: Zoomfunktion aussuchen
Examples

- Map of Osnabrück
- Weather data
Examples

SVG Web Mapping: 3 Tage Wettervorhersage

- Temperatur
- Luftdruck
- Niederschlag
- Wind

Status: Drag and Drop in Karte

Institut für Informatik, 2006
Wetterdaten von Raymarine.com

27th of August, 2008
Dipl.-Math. Dorothee Langfeld
Examples

- Map of Osnabrück
- Weather data
- Traffic data (provided by DDG Gesellschaft für Verkehrsdaten mbH)
Examples

- Map of Osnabrück
- Weather data
- Traffic data (provided by DDG Gesellschaft für Verkehrsdaten mbH)

27th of August, 2008
Dipl.-Math. Dorothee Langfeld
Examples

- Map of Osnabrück
- Weather data
- Traffic data (provided by DDG Gesellschaft für Verkehrsdaten mbH)
Generation of the application

- Java program
- Configuration via XML-File, defining
  - Layout of the application
  - Layers, layout of the data
  - Timesteps
  - Zoomsteps
Base Data

- Different base data
  - ESRI Shapefiles (binary data)
  - GRIB Files (raster data)
  - Streets and traffic data (XML) provided by DDG
- Algorithms to convert the base data into Java objects which represent Points, Polylines and Polygons (possibly with several parts)
  - Projection
  - Vectorizing
- Objects have methods for
  - Clipping
  - Labeling
  - Appending into a DOM
  - Aggregation
Data flow

<table>
<thead>
<tr>
<th>Base data</th>
<th>Creation of graphics</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Shape files, GRIB files, XML-Files with streets and traffic data</td>
<td></td>
</tr>
<tr>
<td>- Configuration file</td>
<td>- Java objects represent shapes</td>
</tr>
<tr>
<td></td>
<td>- Methods for clipping, labelling, inserting into the DOM</td>
</tr>
<tr>
<td></td>
<td>- Data aggregation</td>
</tr>
<tr>
<td></td>
<td>- DOM-Builder and XML-Serializer</td>
</tr>
<tr>
<td>Data processing</td>
<td>ECMA/PHP scripts</td>
</tr>
<tr>
<td>- Reading data</td>
<td>SVG template</td>
</tr>
<tr>
<td>- Geographical projection</td>
<td></td>
</tr>
<tr>
<td>- Setting layout</td>
<td></td>
</tr>
</tbody>
</table>
Some special approaches

- Own clipping algorithms
  - Lines, the border and the filling of polygons must be devided into several paths

- Labeling (simple approach)
  - Coordinates for points
  - Barycenter for polygons
  - Text on path for polylines
    - Offset calculation for correct labeling when tiles are used
Improvements

- Java program / generation
  - Configuration of the application via a Graphical User Interface (additional Java program)
  - Implementing better algorithms for labeling
  - Using existing GIS to generate the SVG fragments

- Application
  - Serving the data on the fly instead of precalculating
  - Display of data values on mouseover in tooltips
  - The change of symbolization options, like size, color etc.
Conclusion

- Easy generation of a complete Web Mapping Application
- Only JRE and PHP enabled Server needed
- Different georeferenced data can be included
  - Up to now Shapefiles, GRIB Files, traffic data
  - Modular buildup allows integration of other data
- Works in all major Browsers (current versions)
  - With native support (FF, Opera, Safari)
  - With Adobe SVG Viewer (IE, FF, Safari)
Online examples and contact

- http://www.inf.uos.de/svgopen2008/
- dlangfel@uos.de