



FOSSGIS 2010, Osnabrück

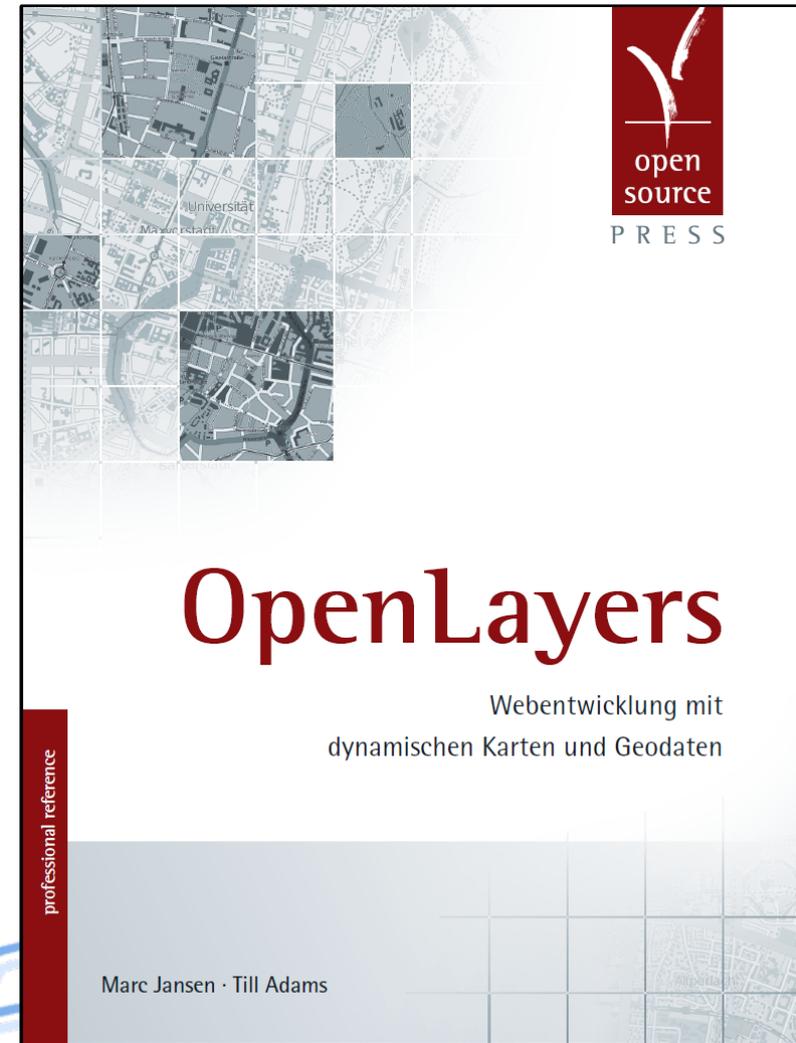
WYTIWYS (What You Type Is What You See)

**Webbasierter interaktiver Terminal
als Schnittstelle zwischen
OpenLayers und PostGIS**

Marc Jansen, terrestris GmbH & Co. KG

- Motivation
- Was ist WYTIWYS?
- Aktuelle Umsetzung
- Einschränkungen
- Demonstration
- Erweiterung und Potenzial
- Zusammenfassung / Fazit

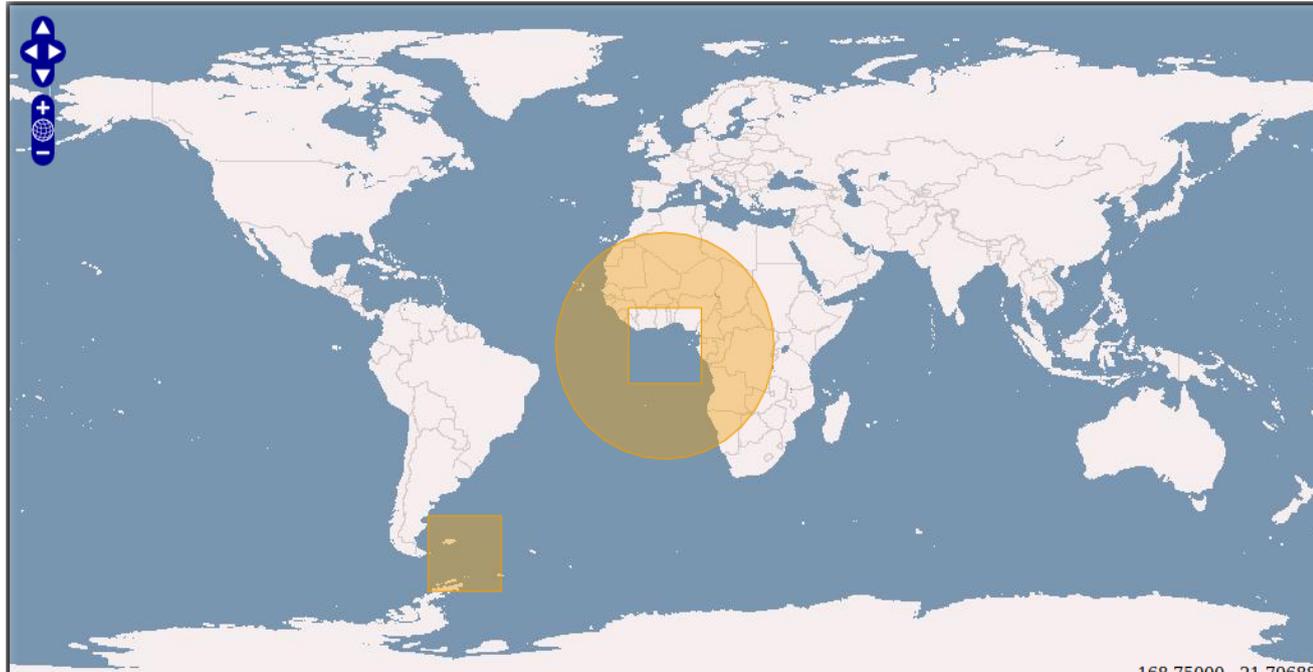
- OpenLayers-Buch
- OpenSource-GIS-Stack
- Visualisierung abstrakter Konzepte der Datenbank
 - Mit Bildern!
- In einem OpenLayers-Buch natürlich über einen WebClient: **WYTIWYS!**



Was ist WYTIWYS?

- Minimales WebGIS
- Eingabefeld für SQL-Queries
- Minimales Backend, welche Queries entgegennimmt und verarbeitet
- Ergebnisse des Queries werden
 - Visualisiert (für Geometrien)
 - Kommuniziert (für Informationen)
- AJAX mit Well-Known-Text als Austauschformat

Was ist WYTIWYS?



```
SELECT ST_AsText(  
  ST_Union(  
    ST_Difference(  
      ST_Buffer(ST_GeometryFromText('POINT(0 0)', 4326), 30),  
      ST_Envelope(  
        ST_Buffer(ST_GeometryFromText('POINT(0 0)', 4326), 10)  
      )  
    ),  
    ST_Envelope(  
      ST_Buffer(ST_GeometryFromText('POINT(-55 -55)', 4326), 10)  
    )  
  )  
);
```

168.75000, -21.79688

Daten absenden



Aktuelle Umsetzung

- PostGIS 1.4.1
- GEOS 3.1.0
- PROJ Rel. 4.6.1
- OpenLayers 2.8
- PHP 5

- WYTIWYS steht unter der BSD-Lizenz

OpenLayers-Code, Teil I

```
1  var map, vectorLayer;
2  -function init() {
3      map = new OpenLayers.Map( 'map' );
4      var wms = new OpenLayers.Layer.WMS(
5          "OpenLayers WMS",
6          "http://labs.metacarta.com/wms/vmap0",
7          { layers: 'basic' }
8      );
9      vectorLayer = new OpenLayers.Layer.Vector(
10         "Vektorlayer"
11     );
12     map.addLayers( [ wms, vectorLayer ] );
13     map.addControl( new OpenLayers.Control.MousePosition() );
14     map.zoomToMaxExtent();
15 }
```

OpenLayers-Code, Teil II

```
1 -function processSpatialQuery() {
2 -    var req = OpenLayers.Request.GET( {
3 -        url: './process-spatial-query.php',
4 -        params: {
5 -            command: OpenLayers.Util.getElement('command').value
6 -        },
7 -        callback: function(req) {
8 -            var response = req.responseText || '';
9 -            if ( response === '' || (/^FEHLER|INFO/).test( response ) ) {
10 -                alert( response );
11 -            } else {
12 -                var wkt_format = new OpenLayers.Format.WKT();
13 -                var vectorFeature = wkt_format.read( response );
14 -                if ( vectorFeature ) {
15 -                    vectorLayer.addFeatures( vectorFeature );
16 -                } else {
17 -                    alert('String konnte nicht als WKT interpretiert werden.');
```

Serverseitiger Code

- Rudimentäre Validierung des SQL:
 - Nur ein Query?
 - SELECT?
- Ausführen des SQLs als User mit minimalen Rechten
 - Nur eine Zeile/Feld?
 - Regulärer Ausdruck entscheidet ob...
 - ...Well-Known-Text-Format
 - ...oder unspezifische Information

Serverseitiger Code

- Vor der Ausführung werden Variablen ausgetauscht:
 - `SELECT ST_AsText(ST_Union(geom))
FROM (SELECT #{C} UNION SELECT #{H}) AS
foo(geom)`
 - `==> MULTIPOLYGON(((0 0,0 2,2 2,2 0,0 0)),((2.5
2.5,2.5 4.5,4.5 4.5,4.5 2.5,2.5 2.5)))`

Einschränkungen

- Projektion: EPSG:4326, WGS84, lon/lat
- Eine Zeile, ein Feld
- Nicht persistent
- Performance von Vektorgeometrien
- Keine Managementfunktionen von PostGIS
- Keine komfortable Konsole
- SQL-Injection?

Demonstration

- Osnabrück:
`SELECT 'POINT(8.043056 52.278889)';`
- Haarlem, NL (Städtepartnerschaft seit 1961)
`SELECT 'POINT(4.640556 52.380278)';`
- Luftlinie:
`SELECT 'LINESTRING(4.640556 52.380278,
8.043056 52.278889)';`

Demonstration

- Länge: Osnabrück - Haarlem, NL (Grad!)
`SELECT ST_Length('LINESTRING(4.640556 52.380278, 8.043056 52.278889)::geometry');`
- Länge auf einer Kugel:
`SELECT ST_Distance_Sphere('POINT(4.640556 52.380278)::geometry, 'POINT(8.043056 52.278889)::geometry');`
- Entfernung München-Hamburg

Demonstration

- Entfernung München-Hamburg
`SELECT ST_Distance_Sphere('POINT(11.566667
48.133333)::geometry, 'POINT(9.993333
53.550556)::geometry);`

Erweiterung und Potenzial

- Andere Projektionen
 - Reprojektion on-the-fly?
- SQL => Anweisung in menschlicher Sprache
- Anweisung in menschlicher Sprache => SQL
- WMS als Mittelschicht?



Erweiterung und Potenzial

- PostGIS 1.5
 - Neuer Datentyp „GEOGRAPHY“
- Zeilen / Feldbegrenzung aufheben
- Andere Austauschformate:
 - KML
 - GML
 - GeoJSON

Zusammenfassung / Fazit

- Projekt befindet sich in einer sehr frühen Phase
- Feedback / Wünsche lassen sich noch relativ leicht integrieren
- Soll man mehr Energie in WYTIWYS stecken?

Zusammenfassung / Fazit

- Ähnliches Ziel mit anderem Ansatz wurde auf der PostGIS Mailingliste von Nicklas Avén diskutiert:

<http://postgis.refractions.net/pipermail/postgis-devel/2009-December/007909.html>

<http://postgis.refractions.net/pipermail/postgis-devel/2009-December/007931.html>

- Tim Schaub (u.a. OpenLayers)
 - Ansatz über GeoScript <http://geoscript.org/>
 - Paul Ramsey (PostGIS) sieht großes Potenzial

<http://blog.cleverelephant.ca/2010/02/nyc-sprint-day-3.html>

Vielen Dank! Fragen?

- Vielen Dank für Ihre Aufmerksamkeit!
- Bitte diskutieren Sie mit mir über WYTIWYS

- **Autor:**

Marc Jansen

terrestris GmbH & Co. KG

Irmintrudisstr. 17

53111 Bonn

- **Kontakt:**

0228 - 96 28 99 -53

jansen@terrestris.de

<http://www.terrestris.de>

- **Lizenz:**

<http://creativecommons.org/licenses/by/3.0/deed.de>

