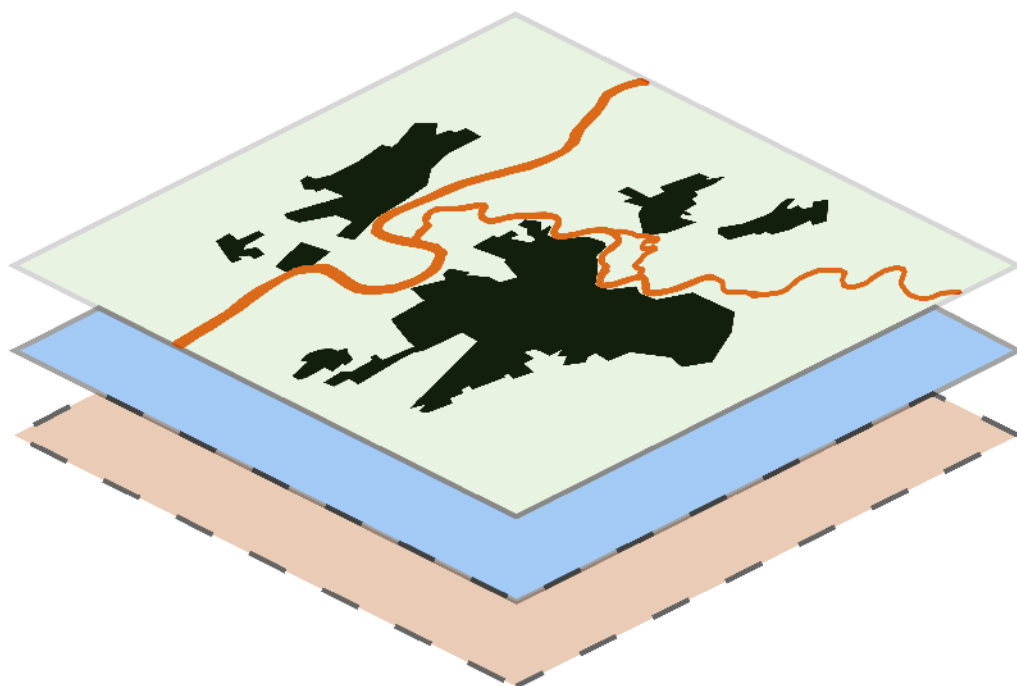




Anwenderkonferenz für Freie und Open Source Software für Geoinformationssysteme

Dessau 20. - 22. März 2012

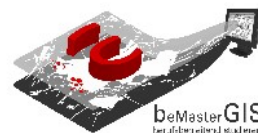


Mit freundlicher Unterstützung von:

Metaspatial • Omniscale GmbH • disy Informationssysteme GmbH • 52° north GmbH • geOps GeoInformatics • WhereGroup GmbH & Co. KG • in medias res GmbH • beMasterGIS • GDB Spatial Consult • Camptocamp SA • MapMedia GmbH • Intevation GmbH • CSGIS José Canalejo & Ruth Schönbuchner GbR • GEOFABRIK • FOSS Academy • terrestris GmbH & Co. KG • mapwebbing • Wichmann • Boinx Software Ltd



HOCHSCHULE ANHALT
Anhalt University of Applied Sciences





Anwenderkonferenz für Freie
und Open Source Software
für Geoinformationssysteme

Herausgeber: FOSSGIS e.V.
<http://www.fossgis.de>

Hinweise zur digitalen Version dieses Tagungsbandes

Die digitale Version des Tagungsbandes liegt in zwei Versionen vor:

FOSSGIS_12_Tagungsband_300dpi.pdf mit hoch aufgelösten Grafiken sowie
FOSSGIS_12_Tagungsband_150dpi.pdf mit geringer aufgelösten Grafiken.

Darüber hinaus enthält die digitale Version drei Artikel, die es terminlich leider nicht in den gedruckten Tagungsband geschafft haben:

Konzeption eines Spatial Business Intelligence System zur Analyse und Steuerung von Re-Produktionsketten auf Grundlage von OpenSource Komponenten.....	35
OSGeo Jahresbericht 2011.....	181
Fallbeispiel Katzenbuckel – Optimierung der TETRA Funknetzabdeckung als Gruppenentscheidung mit GRASS GIS und KML.....	186

Inhaltsverzeichnis

Hinweise zur digitalen Version dieses Tagungsbandes.....	4
Public-Private-Partnership und freie Software am Beispiel OpenLayers mobile.....	7
gvSIG Community Edition (CE): Entwicklungsgeschichte und aktueller Stand des Projektes.....	10
QGIS auf Android / Ein vollwertiges GIS auf Android-Geräten.....	11
QGIS und gvSIG – Desktop GIS Lösungsansätze.....	15
Ermittlung und Visualisierung von Einsatzgebieten der Polizei, Feuerwehr oder von Ersthelfern auf Basis von OpenStreetMap	18
Eigene OpenStreetMap Kartendienste.....	19
Mobiles ortsbezogenes soziales Netzwerk: Ein Geowiki auf OSM-Basis.....	22
Forschungsdatenmanagement mit Open-Source-Software.....	26
Das Anlagen- und Indirekteinleiterkataster der Stadt Bielefeld.....	31
Konzeption eines Spatial Business Intelligence System zur Analyse und Steuerung von Re-Produktionsketten auf Grundlage von OpenSource Komponenten.....	35
Ableitung von topographischen Webkarten aus amtlichen Geobasisdaten.....	43
Anwendung im Bereich Bauzonenstatistik in der Schweiz.....	56
DIVE4elements als Business Intelligence Werkzeug.....	61
NAS - BasisDLM Aufbereitung mit gdal/PostNAS.....	62
PGVS - Konkurrenzierendes Editieren von PostGIS Layern und offline Editieren.....	73
PostGIS 2.0 - Was bringt die neue Version?.....	74
Neues vom QGIS Server und -Webclient.....	77
TinyOWS - der schlanke WFS Server.....	80
MXD2map - ein freier Konverter von ArcGIS MXD-Dateien zu UMN MapServer.....	81
www.geoportal.de - deutschlandweit Geodaten finden und nutzen.....	84
Geoportal DACH+ - Relaunch mit Drupal, OpenLayers+GeoExt.....	91
Wie wird das Kartenportal sichtbar für die Suchmaschinen?.....	93
HTML5-Editor für OpenStreetMap.....	98
Video-Mapping / Des Mappens letzter Schluss?.....	99
Geocoding mit OpenStreetMap.....	101
Synchronisation zwischen OpenAddresses und OpenStreetMap.....	104
JOSM: Der Java OpenStreetMap Editor.....	109
Nutzung amtlicher Geodaten - aktueller Stand.....	110
Am Puls der Zeit: Minütliche Daten-Updates bei OpenStreetMap.....	111
History-Informationen in der OSM Datenbank, Reloaded.....	115
Das Osmium-Framework.....	119
Overpass API.....	120
OpenStreetMap und R.....	134
MapFish, ein OSGeo WebGIS 2.0 Framework.....	135
Mapbender3 – Next Generation PHP Mapping.....	136
SHOGun - Spring, Hibernate, OpenLayers, GeoExt und weitere	138
Ausschreibungen und Open Source Software.....	142

Neuigkeiten aus dem Open Geospatial Consortium – Rückblick 2011 und eine Zusammenfassung der letzten OGC Technical Committee Meetings.....	143
Open Source GIS Software in der UNIGIS Fernlehre.....	145
Freie Software und infektiöse Krankheiten: Von Rohdaten zu ökologischen Indikatoren.....	149
Verarbeitung sehr großer räumlicher Datensätze mit GRASS GIS.....	152
SEXTANTE in gvSIG CE.....	153
MapProxy - das "Schweizer-Taschenmesser" für Kartenserver.....	154
MapServer MapCache - der neue WMTS Tile Cache.....	157
WMS2GO – Der WMS zum Mitnehmen.....	158
Couchbase Mobile, GeoCouch und MapQuery.....	159
Mobiler Haltestellenclient zur Erfassung barrierespezifischer Mobilitätskriterien und Angebotsqualität des ÖV auf Basis von OSM-Daten.....	160
GeoExt Mobile (GXM).....	161
WPS-Appstore.....	165
Marble - ein Schweizer Taschenmesser für Karten.....	165
OSM 3D.....	166
OpenIndoorMap – Erfassung von Innenräumen mittels Smartphone.....	167
INSPIRE View Service in MapServer.....	175
Das OGC Coverage Konzept: Daten und Dienste.....	175
EOxServer, GDAL, MapServer - Zugang zu großen Archiven von Erdbeobachtungsdaten.....	176
djeo: eine einheitliche Open Source API für unterschiedliche geographische Web-Engines.....	177
SOS-Importer - Werkzeug zum Veröffentlichen von Daten im Sensor Web.....	178
OSGeo Jahresbericht 2011.....	181
Fallbeispiel Katzenbuckel – Optimierung der TETRA Funknetzabdeckung als Gruppenentscheidung mit GRASS GIS und KML.....	186

Public-Private-Partnership und freie Software am Beispiel OpenLayers mobile

Andreas Hocevar

Zusammenfassung

Die populäre Open Source JavaScript-Bibliothek OpenLayers bietet seit der Version 2.11 auch auf mobilen Endgeräten wie Smartphones (etwa iPhone oder Android-Geräte) und Tablet-Computern (z.B. iPad) eine hervorragende Grundlage zur Erstellung von webbasierten geographischen Informationssystemen. Dieser Beitrag zeigt die vielfältigen neuen Möglichkeiten auf und erklärt deren Verwendung in eigenen Projekten. Vor allem aber wird auf die Entstehungsgeschichte der Version 2.11 eingegangen: Dank einer Public-Private-Partnership, welche einen Codesprint mit Teilnehmern aus aller Welt möglich machte, bietet OpenLayers nun die Funktionen, die WebGIS auf mobilen Endgeräten ermöglichen.

Der Autor ist Kernentwickler von OpenLayers und war in der glücklichen Lage, auch am bereits erwähnten Entwicklertreffen teilzunehmen. In diesem Beitrag wird von den technischen und organisatorischen Hintergründen der aktuellen OpenLayers Version berichtet und sozusagen aus dem „OpenLayers-Nähkästchen“ geplaudert. Dabei wird deutlich, wie die Unterstützung von Open Source durch öffentliche Verwaltungen als Innovationsmotor wirken kann.

OpenLayers mobile

Auch wenn die Überschrift vermuten lässt, dass es sich bei „OpenLayers mobile“ um ein eigenständiges Produkt handelt, sind damit lediglich die im Februar 2011 entwickelten Funktionalitäten gemeint, die den Einsatz von OpenLayers auf mobilen Endgeräten (Smartphones, Tablets) ermöglichen. Seit der Version 2.11 sind diese fixer Bestandteil von OpenLayers. OpenLayers? Das ist der quasi-Industriestandard für web-basiertes Open Source GIS. Als Mitglied der OSGeo-Projektfamilie bietet es Zugriff auf Geodaten aus unterschiedlichsten Quellen, von OpenStreetMap über proprietäre Anbieter wie Google Maps bis hin zu eigenen Daten aus WMS, WFS, GeoRSS, GeoJSON und vielen anderen Formaten. Dazu bietet OpenLayers Interaktivität wie Info-Popups, Digitalisieren im Browser, Messfunktion, und mehr. Seit der Version 2.11 (aktuell: 2.12) steht all das auch auf mobilen Endgeräten mit Touch-Bedienoberfläche zur Verfügung. Dazu kommt noch die Möglichkeit, die Karte auf die aktuelle GPS-Position des Benutzers zu zentrieren.

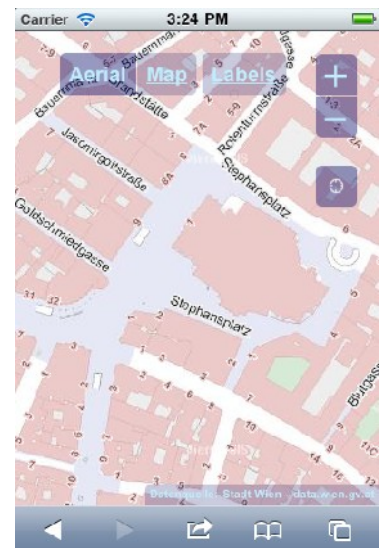


Abb. 1: Mobile Wien-Karte mit OpenLayers

Public-Private-Partnership – warum und wie?

Wie entsteht Open Source Software? Nur in Ausnahmefällen sind es jugendliche Kaffee-Junkies, die ihre gesamte Frei- und Schlafzeit vor dem PC verbringen und zwischen Türmen halbleerer Pizza-Kartons und Kaffeebecher zum Vergnügen an der Software basteln. Gerade im Umfeld der OSGeo-Projekte gibt es ein Netzwerk von Unternehmen, die ihre Geschäftsmodelle rund um Open Source Software aufgebaut haben. Der Grundgedanke ist immer der, dass neue Features in Open Source Projekten zu einem möglichst großen Teil aus Kundenanforderungen heraus entwickelt werden. Danach stehen sie allen Benutzern zur Verfügung.

Public-Private-Partnership und freie Software am Beispiel OpenLayers mobile

Dieses Geschäftsmodell führt, je nach konkreter Ausprägung, entweder zu hohen Preisen für neu zu entwickelnde Features, oder zu einem hohen Risiko für das Unternehmen, das daran arbeitet. Erstes ist der Fall, wenn das anbietende Unternehmen keine Folgeaufträge erwartet und nicht mit Interesse anderer Kunden für das gleiche Feature rechnet. Somit werden die tatsächlichen Entwicklungskosten plus der Overhead für das Community-Management im Open Source Projekt voll dem Kunden verrechnet. Ein hohes Risiko für das anbietende Unternehmen entsteht dann, wenn dem Kunden nur ein Teil dieser Kosten verrechnet wird, weil das neu zu entwickelnde Feature strategische Bedeutung hat. Wenn sich dann wider Erwarten doch keine Möglichkeit ergibt, Kostenanteile über andere Aufträge oder Supportleistungen zu decken, muss das Unternehmen den Mehraufwand selbst abdecken.

Ein gängiger Lösungsansatz für dieses Problem ist es, mehrere Interessenten an einem neuen Feature zu finden, sodass die Entwicklungskosten aufgeteilt werden können. Die Umsetzung muss nicht durch ein einzelnes Unternehmen erfolgen, sondern wird auch oft zwischen mehreren Organisationen aufgeteilt. Üblicherweise geht die Initiative dazu von Softwareanbietern aus, die Kunden sind aber häufig öffentliche Auftraggeber (z.B. mehrere Bundesländer, Gemeinden etc. mit ähnlichen Anforderungen).

OpenLayers mobile entstand aus einem ähnlichen Modell, aber mit folgenden wesentlichen Unterschieden:

- Die Initiative zur Vernetzung potenzieller Auftraggeber ging nicht von Softwareanbietern, sondern von öffentlichen Organisationen aus (geo.admin.ch und swisstopo).
- Es gab kein Lasten-/Pflichtenheft, sondern den offen formulierten Auftrag: „Macht OpenLayers fit für mobile Endgeräte mit Touch-Bedienoberfläche“.
-

Durch dieses Modell, das eher dem Sponsoring als einer klassischen Auftragsvergabe entspricht, bot sich für die Umsetzung das Format eines mehrtägigen Code Sprint an: 15 Entwickler kamen im Februar 2011 an der EPFL Lausanne zusammen, um an einem von Camptocamp organisierten 5-tägigen Sprint teilzunehmen, der von insgesamt 16 Organisationen mit einer 5-stelligen Euro-Summe gesponsert wurde. Mit Rahmenprogramm, aber auch mit langen Nächten des Programmierens. Das Ergebnis konnte sich sehen lassen: GPS-Unterstützung, Navigation und Digitalisieren über Touch-Gesten sowie zahlreiche Proof-of-Concept Implementierungen, wie z.B. zur Offline-Speicherung von digitalisierten Geometrien oder Kartenkacheln. Im Herbst letzten Jahres kam schließlich die OpenLayers Version 2.11 heraus, die all diese neuen Features beinhaltet. Und der Pool von Proof-of-Concept Implementierungen dient nach wie vor als Quelle neuer Features.

Lessons learned: Vorteile der Public-Private-Partnership

- **Für die Sponsoren:** Ohne Lastenheft gibt es mehr Funktionalität für viel weniger Geld. Obwohl das Ergebnis nicht genau den Anforderungen entspricht, kann ein eventuell geplantes Vorhaben mit wenig Aufwand bzw. einem kleinen Zusatzauftrag umgesetzt werden, und weitere Innovationen aus dem Sprint stehen für die Zukunft ebenfalls zur Verfügung. Beispiel: von terrestris umgesetztes mobiles Geodatenportal <http://mobile.map.geo.admin.ch> und das daraus hervorgegangene GXM (GeoExt Mobile) Framework.
- **Für die Open Source Unternehmen:** Möglichkeit der Umsetzung strategisch wichtiger Features durch die entsandten Entwickler im Rahmen des Code Sprints. Dazu kommt die Chance, über Folgeaufträge die Kosten der Weiterentwicklung begonnener neuer Features abzudecken.
- **Für die Entwickler und das Open Source Projekt:** 1-wöchiger direkter Kontakt in einem Umfeld, wo man üblicherweise nur über Skype-Konferenzen, Chat und Mailinglisten kommuniziert. Regelmäßige Code-Sprints sind ein wesentlicher Erfolgsfaktor für das Funktionieren von Open Source Communities.



Abb. 2: OpenLayers-Sprinter an der EPFL Lausanne

Kontakt zum Autor:

Dipl.-Ing. Andreas Hocevar
OpenGeo
148 Lafayette Street, Floor 12, New York, NY 10013, USA
+43-720-503462
ahocevar@opengeo.org

gvSIG Community Edition (CE): Entwicklungsgeschichte und aktueller Stand des Projektes

Benjamin Dücke

Die gvSIG Community Edition ist ein neues Open-Source-GIS-Projekt, das 2011 ins Leben gerufen wurde. Es handelt sich um eine, durchaus kontrovers diskutierte, Abzweigung des „offiziellen“ gvSIG-Projekts der Conselleria de Infraestructuras y Transporte (CIT) in Valencia. Hauptziel des Projektes ist es, die Zusammenarbeit mit anderen relevanten Projekten wie SEXTANTE, GRASS GIS und SAGA mit gvSIG sowie die Internationalisierung der Software und Anwendergemeinschaft voran zu treiben.

Technisch gesehen, basiert gvSIG CE auf gvSIG OADE (2010), den aktuellen Verbesserung aus dem CIT-Codezweig, sowie der aktuellsten Version von SEXTANTE GIS. Besonderes Merkmal von gvSIG CE ist die Einbindung von mehr als 700 Funktionen zur Geodatenprozessierung aus den genannten Partnerprojekten. Die Entwicklung von gvSIG CE geschieht auf Englisch, um die Kommunikation zwischen allen Entwicklern und Anwendern zu vereinfachen. Die Projektstruktur ist bewusst offen und weitestgehend frei von Hierarchien gestaltet.

Der Vortrag erläutert die technischen, ökonomischen und sonstigen Aspekte der Gründung von gvSIG CE. Er wägt dessen Entwicklungsmodell gegen das der CIT ab und gibt einen Einblick in den momentanen Stand der Entwicklung und die Dynamiken hinter dem Projekt. Eine kurze Live-Einführung der aktuellen Version sowie die Vorstellung der konkreten Web-Organisation des Projektes runden den Vortrag ab.

Weitere Informationen erhalten Sie auf der Internetseite des Projektes: <http://www.gvsigce.org>.

QGIS auf Android / Ein vollwertiges GIS auf Android-Geräten

Marco Bernasocchi

Die gute Verfügbarkeit von mobilen Geräten macht die Verwendung deren Verwendung heutzutage zur einer alltäglichen Angelegenheit. Bei der Datenerfassung oder Verwaltung im Feld ist die Verwendung von sperriger Hardware sehr unbequem und unproduktiv. Das Zusammenführen von erstklassiger Software wie QGIS und der letzten Generation von Android Tablett-Geräten kann die Produktivität vieler Berufs-, Firmen- und Privatbenutzer bedeutend verbessern. Diese Präsentation wird den aktuellen Status und die zukünftigen Pläne für den Android-Port von QGIS aufzeigen.

Dank seiner Multi-Plattform-Natur (Windows, Mac, Linux und Android) und seines breiten Feature-Sets (Desktop, Server, Web-Client), ist QGIS eine der am weitesten verbreiteten Open-Source GIS Software und wird bereits von vielen Institutionen, Firmen und Behörden als GIS-Software der Wahl benutzt.

Die Ergänzung der QGIS Suite mit einer Tablett-fähigen Version (und in naher Zukunft auch eine Version für kleine Bildschirme) bietet den Anwendern eine vollwertige Infrastruktur zur Verwaltung von Daten im Feld.

Videos

<http://www.bernawebdesign.ch/byteblog/2012/02/16/qgis-on-android-videos/>

Installation

- Installer von <http://android.qgis.org/download/qgis-installer.apk> herunterladen
- Version auswählen und installieren. Die nightly builds sind debug builds und werden jede Nacht aus dem neuesten Code automatisch erstellt.
- Nach der Installation auf Öffnen klicken. QGIS firstRun sollte starten und verschiedene Daten entpacken.
- QGIS öffnen, auf Anfrage Ministro II (Qt Library manager) installieren und anschließend zurück zu QGIS.
- Auf Anfrage Qt Bibliotheken installieren.
- QGIS Startet

Updates

- Momentan muss man manuell den QGIS installer starten und prüfen ob, es eine neue Version gibt.
- Die Qt -Bibliotheken werden bei jedem QGIS Start automatisch (via Ministro II) auf Updates geprüft. Verschiedene Versionen können über das Ministro Configuration Tool 2 eingestellt werden.

Benutzung

- Touch: QGIS auf Android funktioniert genauso wie QGIS Desktop. Der Touch auf den Bildschirm erzeugt einen Mouse Click, ein langer Touch (etwa eine Sekunde) erzeugt einen Rechtsklick.
- Daten: Jedes Android-kompatible Gerät unterstützt einen gemeinsamen "externen Speicher", auf dem Sie Dateien speichern können. Dies kann ein Wechseldatenträger (z. B. eine SD-Kar-

QGIS auf Android / Ein vollwertiges GIS auf Android-Geräten

te) oder ein interner (nicht entfernbarer) Speicher sein. Die GIS-Daten werden meistens auf dem "externen Speicher" gesichert. Um diese von den QGIS File Dialoge zu finden kann man den storage Link benutzen (wenn vorhanden – dies ist Gerät abhängig) oder die „nach oben“ Pfeile bis man auf /sdcard kommt.

Einschränkungen

- Momentan Kein Python Support (Sponsoren sind gesucht)
- Verschiedene Bugs, siehe: <http://hub.qgis.org/projects/android-qgis/issues>

Zur Behebung von Bugs und für die Entwicklung neuer Features sind Sponsoren, resp. Kunden gesucht.

Screenshots

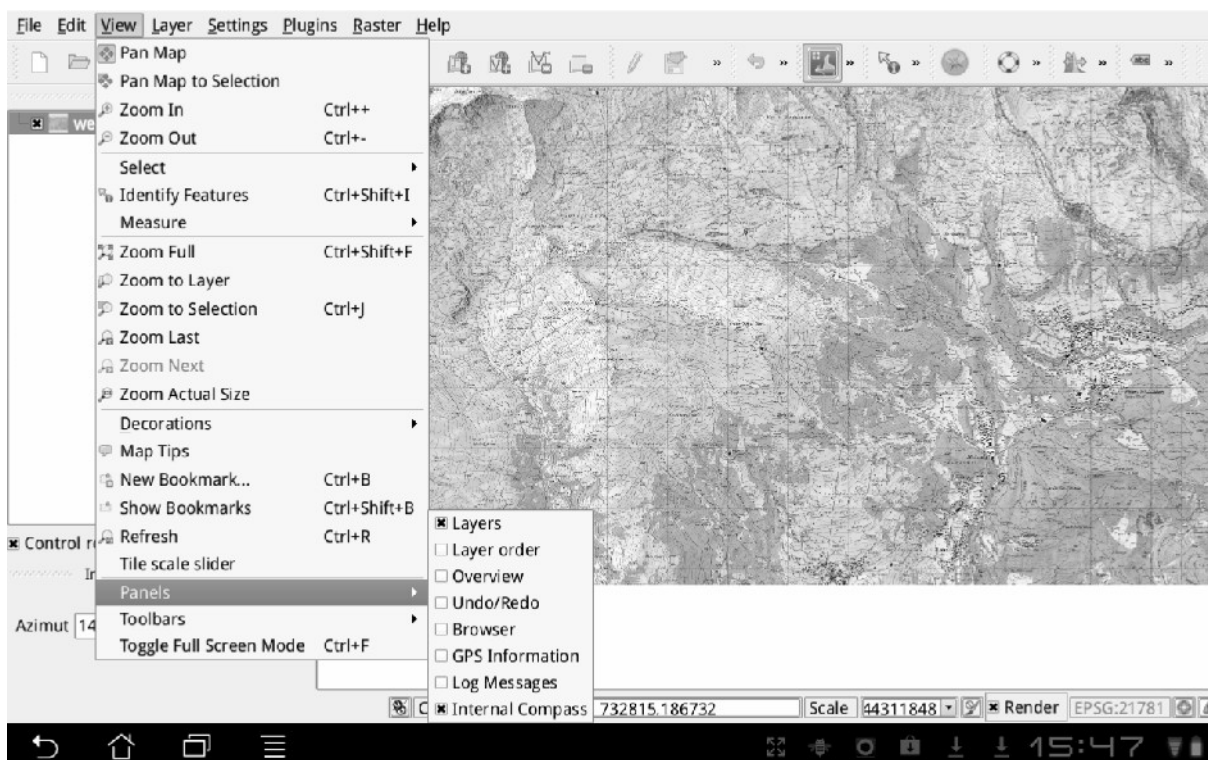


Illustration 1: Menüs und Submenüs

QGIS auf Android / Ein vollwertiges GIS auf Android-Geräten

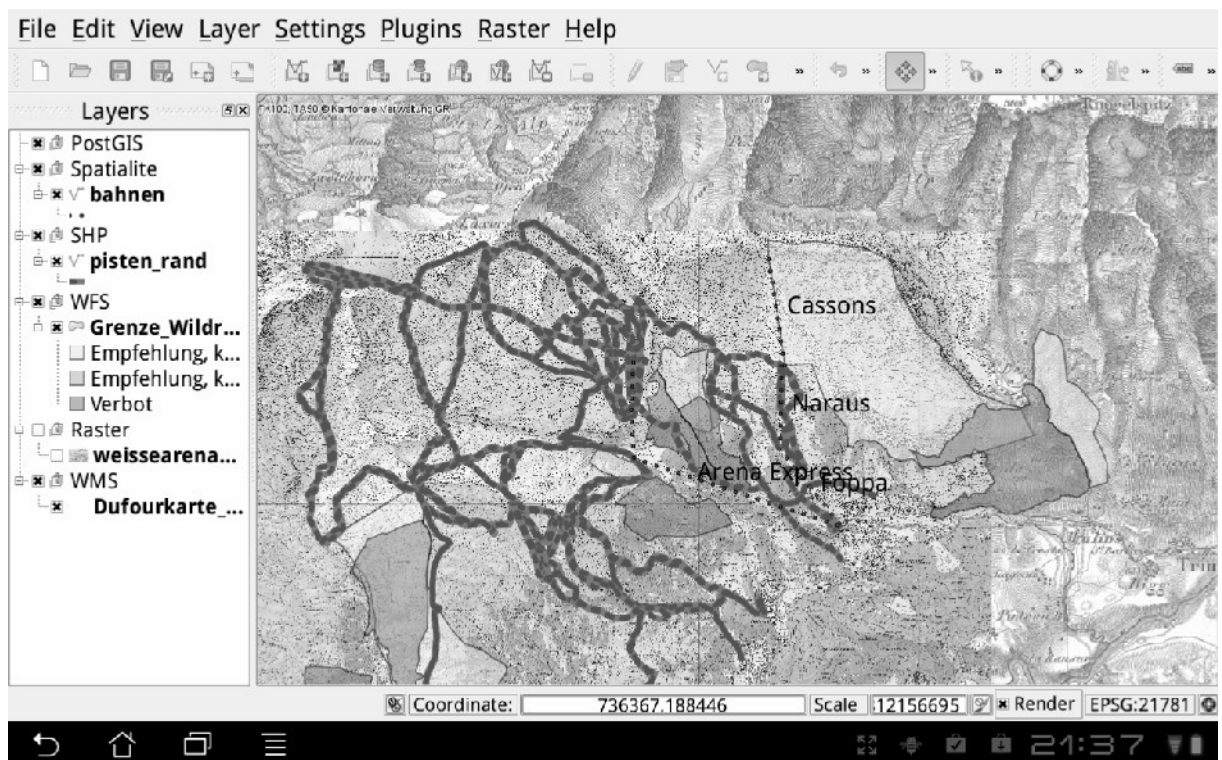


Illustration 2: Verschiedene Datenquellen

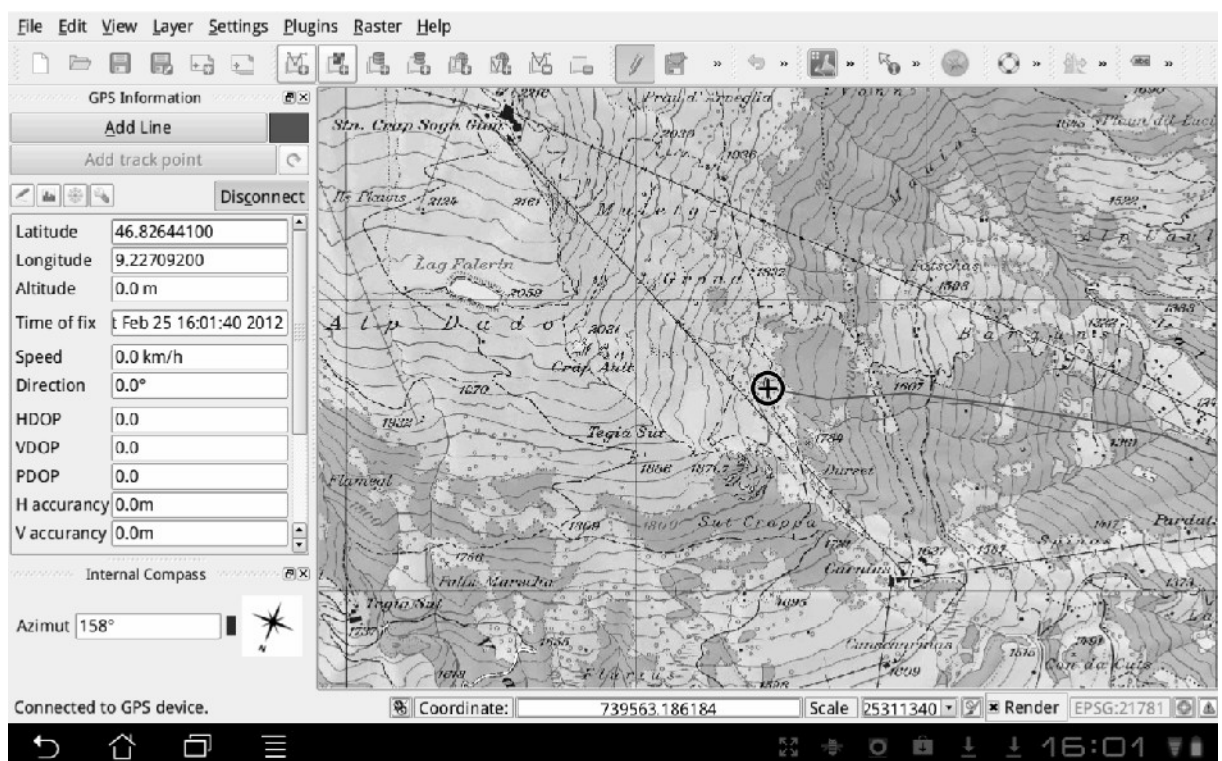


Illustration 3: GPS tracking und elektronischer Kompass (hier mit künstlichem GPS Signal)

Möglichkeiten der Unterstützung des QGIS auf Android Projekts

- Sponsoring (allgemein oder Bug/Feature-spezifisch)
- Testing und Erstellen von Bugreports auf <http://hub.qgis.org/projects/android-qgis/issues>
- Mitentwickeln <https://github.com/qgis/qgis-android>

Kontakt

Marco Bernasocchi
bernawebdesign.ch / opengis.ch
Via Cochetta 2
7152 Sagogn
Schweiz
+41794672470
marco@opengis.ch

QGIS und gvSIG – Desktop GIS Lösungsansätze

Till Adams



Zusammenfassung

Zunehmend rücken Open Source Desktop-GIS in den Fokus der öffentlichen Verwaltung und anderer GIS-Anwender. Dies liegt vor allem daran, dass auch im OpenSource Umfeld nun absolute Highend-Lösungen für Desktop-GIS verfügbar sind.

Wussten Sie, dass Sie in QGIS Google-Karten anzeigen oder dass Sie in gvSIG einfach 3D-Darstellungen erstellen können?

Dieser Beitrag versucht nicht nur solche Highlights darzustellen, sondern sich auch in einem Vergleich zwischen den beiden prominentesten Vertretern der Open Source Desktop GIS, dies jedoch ohne jeglichen Anspruch auf Vollständigkeit zu erheben.

Bei diesem Vergleich geht es auch nicht darum, Funktionslisten gegenüberzustellen und abzuarbeiten, sondern neben Funktionsgruppen auch die Philosophie der beiden Systeme zu vergleichen. An einigen Beispielen wird praktisch aufgezeigt, wie klassische GIS-Fragestellungen mit beiden Systemen gelöst werden können. Dabei wird jeweils eine Lösung derselben Aufgabe in beiden Systemen vorgestellt.

Desktop und GIS

Während bis vor einigen Jahren der klassische Aufbau einer Geodateninfrastruktur durch Kauf von entsprechenden Desktop-Lizenzen mit vorangegangenen Vorsingen der entsprechenden Anbieter vonstatten ging, sieht die Sache heute wieder einmal komplizierter aus:

Erstens können heutzutage viele GIS Arbeitsplätze durch die Einrichtung eines WebGIS und die Bereitstellung eines entsprechenden Zugang über einen Browser eingerichtet werden, zweitens kommen nun durch die extreme Dynamik in der Entwicklung auf Seite von Open Source Software auch noch neue Störenfriede hinzu und stoßen in den bis dahin nahezu hermetisch abgeschlossenen Markt der selbsternannten Weltmarktführer vor.

Aber auch wenn heute wegen der Weiterentwicklung im WebGIS Bereich theoretisch weniger Desktop GIS gebraucht werden als früher, so sind Sie dennoch für die Administration von WebGIS, die Geodatenerstellung und -pflege sowie für diverse andere Aufgaben unabdingbar.

Die Frage nach Open Source und Desktop GIS entlockt einem Open Source GIS Vertreter wie dem Autor nicht mehr einen undefinierbaren Laut und den anschließenden kleinlauten Verweis auf wirklich gute Software im Bereich WebGIS, sondern lässt einen mit stolzer Brust mindestens zwei Namen nennen:

QGIS und gvSIG

Man könnte nun einen einfachen Vergleich heranziehen und den ewig währenden Streit zwischen den beiden Völkern **Java** und **C** bemühen, um Ihnen eine einfache Wahl Ihres Werkzeugs zu überlassen.

Dies würde der oben genannten Software jedoch wenig gerecht werden.

Ebenso könnte man sagen, warum sollen hier Vor- und Nachteile aufgezählt werden, beides ist Open Source Software und damit die Einstiegshürde gering. Man kann also beides ausprobieren, auch wenn man sich eher dem ein- oder anderem Volke zugehörig fühlt. Tun Sie dies, dennoch werde ich versuchen einige Fakten zusammenzutragen.

QGIS und gvSIG – Desktop GIS Lösungsansätze

Und es sei an dieser Stelle noch der Verweis auf weitere Freie Desktop GIS wie OpenJump [1], UdlG [2] sowie SAGA und auch GRASS, auch wenn es über eine Erweiterung in QGIS geladen werden kann, erlaubt.

Der Vergleich

In mehreren Workshops mit dem Thema „*Morgens QGIS, Nachmittags gvSIG*“ habe ich versucht Meinungen und echte Fakten zum Beweis der Überlegenheit des Einen gegenüber dem Anderen System herauszubekommen, leider bisher ohne Erfolg. Die Kursteilnehmer waren dennoch sehr angetan, Nachmittags dieselben Aufgaben zu erledigen, wie Vormittags, gleiche Daten, nur anderer Software. Der Vorbereiter des Workshops übrigens auch, denn ich musste mir lediglich Aufgaben für einen halben Tag ausdenken.

Folgende Tabelle greift einige Aspekte auf und versucht diese – wenn auch nicht qualitativ - gegenüberzustellen

Tabelle 1: Gegenüberstellung von Funktionen (unvollständig)

Funktion	QGIS	gvSIG
Datenformate	alle GDAL/OGR	Übliche, Zugriff per GDAL/OGR in einer der nächsten Versionen geplant
Oberfläche	„klassisch“	„modern“
3-D	mit GRASS PlugIn	3-D Erweiterung
Erweiterungen	offizielles und freies Repisitory, Verwaltung über QGIS	manuelles Einbinden, Verwaltungstool vorgesehen
Rasterdatenverarbeitung	über GRASS	über Sextante
WebGIS	QGIS Server	--
Mobiles GIS	QGIS mobile nur Android	gvSIG Mobile

Jetzt mag sich ein Leser (berechtigterweise) Fragen, was nun mit „klassischer“ und „moderner“ Oberfläche gemeint ist. Für mich als ehemaligen ArcView 3 Benutzer erschließt sich QGIS sehr einfach, denn sowohl der Aufbau als auch der Workflow ist sehr ähnlich, wohingegen ich bei gvSIG meine Intuitivität etwas mehr strapazieren musste, das ist gemeint.

Neben diesen harten Fakten poche ich bei einem Vergleich auch immer darauf, sich das Umfeld der Open Source Projekte einmal genauer anzuschauen. Wer entwickelt die Software, wieviele Entwickler gibt es, wie dynamisch ist die Software usw.

Ein Blick auf die Seite der OSGeo verrät, QGIS ist inkubiertes OSGeo Projekt, gvSIG ist es auch. Damit sind schon einige der oben aufgeworfenen Fragen beantwortet, ohne der Notwendigkeit eine neue Tabelle zu erstellen. Auch dort befindet sich für jede Software ein guter Überblick über die angebotenen Funktionen. Es wird kein Projekt ohne entsprechenden Hintergrund oder funktionierende Community ein OSGeo-Projekt, dazu sind die Inkubationsregeln zu streng.

Vorstellung

Im Vortrag wird in 4 Schritten eine klassische GIS Aufgabe auf beiden Systemen vorgestellt. Dabei werden zunächst Daten geladen und ausgestaltet (1), danach wird ein Liniendatensatz neu erstellt (2), auf eine neu erzeugte Linie ein Buffer gerechnet (3) und anschließend ein rudimentäres Layout erstellt (4).

Fazit & Ziel des Beitrags

Mein Ziel ist es, die Einstiegshürde zur Benutzung Freier Software noch weiter abzusenken und Appetit zu machen, selber einmal eines der Freien GIS auszuprobieren. Denn die Verwendung von Freien Desktop GIS bietet neben vielen Vorteilen von Partizipation an einem Open Source Projekt, der hohen Flexibilität und Dynamik natürlich auch ein enormes Einsparpotential an Lizenz- und Folgekosten. Dieses Geld kann sinnvoll in benötigte Erweiterungen und Ausbildung der Mitarbeiter investiert werden. Die Migration kann dabei schrittweise erfolgen - so können zunächst einzelne Arbeitsplätze durch Freie Desktop GIS ersetzt werden, neu hinzukommende Mitarbeiterkünftig auf Freien Desktop GIS geschult werden, u. v. m.

Kontakt zum Autor:

Till Adams
terrestris GmbH & Co KG
Pützchens Chaussee 56, 53227 Bonn
+49 – (0)228 962 899 52
adams@terrestris.de

Literatur

- [1] <http://www.openjump.org/>
- [2] <http://udig.refractory.net/>
- [3] <http://www.saga-gis.org/>
- [4] <http://grass.fbk.eu/>

Ermittlung und Visualisierung von Einsatzgebieten der Polizei, Feuerwehr oder von Ersthelfern auf Basis von OpenStreetMap

Pascal Neis

Am Beispiel der Gemeinde Hünstetten im Taunus

Ein typisches Problem im Bereich der Standortanalyse im Bereich der Blaulichtorganisationen (BOS) ist: Welches Gebiet kann von einer Einsatzzentrale der Polizei, der Feuerwehr oder von Ersthelfern in einer vorgegebenen Zeit abgedeckt werden? In den meisten Fällen scheitert die Beantwortung einer solchen Frage entweder an dem Nichtvorhandensein der Daten oder am Fehlen entsprechender Software für die Analyse.

Als Datengrundlage kann in Deutschland inzwischen auf das bereits gute und kostenlose Datenmaterial des OpenStreetMap.org (OSM) Projektes zurückgegriffen werden. Im OSM Datenbestand werden neben Straßendaten auch die Standorte von Feuerwehren, Krankenhäusern oder Polizeistationen gelistet. Früher erfolgten einfache Darstellungen über einen Kreis auf einer Karte, der das zu erreichende Gebiet darstellt. Inzwischen kann auch die kostenlose Erreichbarkeitsanalyse [1] im Portal von OpenRouteService.org (ORS) verwendet werden. Der Dienst ermittelt dabei realistischere Isochronen als Polygone auf Basis des Straßennetzes. Diese Flächen repräsentieren dann das Gebiet, welches in einer vorgegebenen Zeit oder Entfernung erreicht werden kann.

Im Vortrag wird beispielhaft für die Gemeinde Hünstetten im Taunus gezeigt, welche Gebiete die gemeindeeigenen freiwilligen Feuerwehren in bestimmten Zeiten abdecken können. Dabei werden unterschiedliche Dienste, die Daten des OSM Projektes verwenden, genutzt. Angefangen bei der Overpass API [2], über die Erreichbarkeitsanalyse von ORS [3] zu der Darstellung der Ergebnisse mittels der freien Javascript Bibliothek OpenLayers [4]. Der Vortrag wird vor allem so aufgebaut sein, dass es anschließend für jeden möglich sein sollte, das Gezeigte für ein eigenes Gebiet wiederholen zu können. Weitere Informationen finden sich hier [5][6].

Referenzen:

[1] Neis, P., A. Zipf (2007): A Web Accessibility Analysis Service based on the OpenLS Route Service. AGILE 2007. International Conference on Geographic Information Science of the Association of Geographic Information Laboratories for Europe (AGILE). Aalborg, Denmark.

[2] http://wiki.openstreetmap.org/wiki/Overpass_API

[3] <http://wiki.openstreetmap.org/wiki/DE:OpenRouteService#WaskannstDuallesaufORSmachen.3F>

[4] OpenLayers: Free Maps for the Web - <http://openlayers.org/>

[5] <http://neis-one.org/2010/08/etwas-osm-fur-die-feuerwehr/> [6] <http://neis-one.org/2010/09/osm-fur-die-feuerwehr-2-0/>

Eigene OpenStreetMap Kartendienste

Oliver Tonnhofer

Mit freier Software eigene, individuelle Kartendienste auf OpenStreetMap-Basis aufsetzen.

Offizielle OpenStreetMap Karten

Die freien Geodaten des OpenStreetMap[1] Projekt haben mittlerweile einen Umfang erreicht, mit dem die Daten in vielen Anwendungsbereichen kommerzielle Daten ersetzen können. Dieses gilt insbesondere für Deutschland.

Das OpenStreetMap Projekt betreibt selbst Server die Karten zeichnen (rendern) und in eigene Webanwendungen eingebunden werden. Dieses ist für kleine Anwendungen auch erwünscht. Für größere und kommerzielle Anwendungen wird jedoch geraten eigene Kartenserver zu verwenden und nicht auf die knappen Ressourcen des Projektes zurückzugreifen.

Für das Aufsetzen eigener Server stellt das Projekt, neben den Kartendaten, auch alle benötigten Softwarekomponenten und Konfigurationsdateien unter freien Lizenzen bereit. Hiermit kann eine Kopie der offiziellen Karten erzeugt werden.

Das offizielle Kartenstyling kann nach Meinung des Autors jedoch nicht mit vielen kommerziellen Karten mithalten. Zum einen werden sehr viele Daten dargestellt, vom Briefkasten bis zur Hochspannungsleitung, was die Karten sehr komplex machen. Zum anderen werden in den Karten keine einheitlichen Farbschematas verwendet.

Die Ursache hierfür liegt zum Großteil daran, dass möglichst alles was auch von dem Projekt aufgezeichnet wird auch auf den Karten dargestellt wird (Umfang) und das Styling historisch gewachsen ist und neue Objekte nach und nach hinzugefügt wurden, ohne dass jedes mal auf ein harmonisches Kartenbild geachtet wurde (Optisch).

Für einen professionellen, kommerziellen Einsatz von OpenStreetMap Daten sollten daher angepasste Kartendesigns verwendet werden, die inhaltlich und farblich für den Einsatzzweck angepasst sind.

Solch ein Kartenstyling zu erstellen ist jedoch eine sehr komplexe Aufgabe und auch das Anpassen der bestehenden OpenStreetMap XML Styling-Dateien ist mit einem hohen Aufwand verbunden.

In den letzten ein, zwei Jahren hat sich, neben den offiziellen Tools des OpenStreetMap Projektes, weitere freie Software entwickelt, die bei der Erstellung eigener OpenStreetMap Kartendienste unterstützen kann.

Datenimport mit Imposm

Bevor eigene Karten mit Kartenservern wie Mapnik[2] oder MapServer[3] gezeichnet werden können, müssen diese in einer Geo-Datenbank wie zum Beispiel PostGIS importiert werden.

Hier wurde im letzten Jahr das Tool Imposm[4] von der Firma Omniscale entwickelt und unter Open Source gestellt. Ein Vorteil von Imposm ist der geringere Speicherverbrauch beim importieren der Daten. Außerdem lassen sich mit Imposm sehr einfach abweichende Datenbankschematas für die OSM Daten umsetzen. Hierfür kann in einem Mapping definiert werden, welche Objekte – abhängig von den Tags – in welche Tabelle importiert werden sollen.

Mit Imposm kann z.B. für eine Parkplatzkarte einfach eine Tabelle erstellt werden, in der ausschließlich Parkplätze gespeichert werden. In diesem Beispiel können die Werte vom Tag `capacity:disabled` automatisch in einer boolean Spalte (Ja/Nein) und einer numerischen Spalte eingetragen werden. Im-

Eigene OpenStreetMap Kartendienste

posm kann Werte automatisch vereinheitlichen (no/nein/none/0 werden so als Datenbankwert Falsch gespeichert). Mit solchen angepassten Datenbankschemas kann das Styling und das Zeichnen individueller Karten deutlich vereinfacht werden.

Für das Zeichnen von regulären Straßenkarten liefert Imposm bereits ein Standardmapping mit, dass von Straßen, Wäldern bis zu Gebäuden alles beinhaltet was auch auf den meisten Stadtplänen zu sehen ist.

Kartenstyling mit TileMill

Aufbauend auf diesem Datenbankschema gibt es bereits existierende Kartenstylings für MapServer[5] und Mapnik[6]. Ein manuelles bearbeiten dieser Stylings ist jedoch noch immer mit Aufwand verbunden.

Um dieses zu vereinfachen wurde im letzten Jahr die Software TileMill[7] entwickelt. TileMill wird von der Firma Development Seed entwickelt und steht ebenfalls als OpenSource zur Verfügung. TileMill setzt auf den Kartenrenderer Mapnik auf und ermöglicht es einfach eigene Kartendesigns zu erstellen.

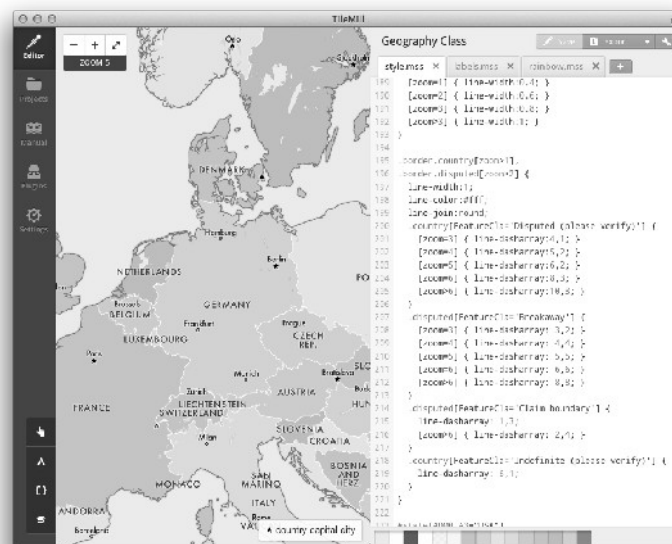


Abbildung 4: TileMill Beispielprojekt

TileMill verwendet eine an CSS angelehnte Styledefinition namens Carto[8]. Carto ermöglicht das Aussehen von Objekten Schritt für Schritt, vom Allgemeinen hin ins Detail zu definieren – von "alle Straßen sind Weiß" bis "Nebenstraßen in Level 14 sind 4 Pixel breit" – ohne dass Definitionen wiederholt werden müssen. Außerdem unterstützt Carto die Verwendung von Variablen, Rechenoperationen auf Schriftgrößen und Linienbreiten, sowie relative Farben (z.B. heller, satter).

TileMill integriert den Mapnik Kartenserver und liefert beim Designen der Karten umgehendend Feedback.

Kartenrendering mit Mapnik und MapProxy

Für das Zeichnen der Karten kann die Exportfunktion von TileMill verwendet werden. Zum einen können fertige Karten im Kachelformat exportiert werden und zum anderen kann das Carto Styling in Mapnik XML Styling überführt werden.

Eigene OpenStreetMap Kartendienste

Das Exportieren fertiger Karten erfolgt über das MBTiles Format. Hierbei werden alle Kacheln in einer SQLite Datenbank als PNG Bilder abgelegt. Mit dem Tool mb-utils[9] können diese Bilder aus dem Export extrahiert werden und so z.B. direkt über ein Webserver angeboten und in OpenLayers integriert werden. Bei dieser Variante müssen sämtliche Kacheln der Karte exportiert werden, ein dynamisches nacherzeugen von fehlenden Kacheln ist nicht möglich. Für kleinere Karten, z.B. für eine Stadt, kann dieses jedoch eine unkomplizierte alternative darstellen.

Für größere Karten ist jedoch schnell ein dynamisches Zeichnen notwendig, da ein Vorgenerieren aller möglichen Kacheln zu zeitaufwändig wird. Hierfür kann MapProxy[10] verwendet werden, mit dem die exportierten Mapnik Kartenstylings direkt eingebunden werden können. Mit MapProxy können polygongenaue Kartenbereiche vorgeneriert werden. Fehlende Kacheln werden von MapProxy bei Bedarf mit der Hilfe von Mapnik nachgezeichnet. Im behördlichen Umfeld können die gekachelten Daten sogar als WMS-Dienst bereitgestellt werden.

Fazit

In den letzten Jahren haben sich nicht nur die OpenStreetMap Daten selbst rasend gewachsen, auch die Software für OpenStreetMap Daten – und Software für moderne Kartendienste im Allgemeinen – hat sich weiterentwickelt. Mit den aufgeführten Softwarekomponenten lassen sich mit vertretbarem Aufwand eigene, individuelle und moderne Kartenanwendungen entwickeln.

Kontakt zum Autor:

Dipl.-Ing. (FH) Oliver Tonnhofer
Omniscale GmbH & Co. KG
Nadorster Straße 60
26123 Oldenburg
0441/93927742
tonnhofer@omniscale.de

Literatur

- [1] <http://www.openstreetmap.org/>
- [2] <http://mapnik.org/>
- [3] <http://mapserver.org/>
- [4] <http://imposm.org/>
- [5] <http://code.google.com/p/mapserver-utils/source/browse/trunk/README>
- [6] <https://github.com/mapbox/osm-bright>
- [7] <http://mapbox.com/tilemill/>
- [8] <https://github.com/mapbox/carto>
- [9] <https://github.com/mapbox/mbutil>
- [10] <http://mapproxy.org/>

Mobiles ortsbezogenes soziales Netzwerk: Ein Geowiki auf OSM-Basis

Klaus Stein, Christoph Schlieder

Einleitung

Ein wesentlicher Aspekt selbstbestimmten Wohnens und Lebens ist die Erhaltung der Mobilität im Alter und bei körperlichen Behinderungen. Hierzu zählt der Gang zum Bäcker genauso wie die Nutzung von Mitfahrgelegenheiten. Im BMBF-Projekt EMN-MOVES haben sich Universitäten, Wohnungsunternehmen und Pflegedienstleister zusammengeschlossen, um soziale und technische Dienste zu entwickeln, die Bewohner von Wohnquartieren befähigen, ihre individuelle Mobilität durch gemeinschaftliches Handeln zu bewahren. Durch die Verbindung von Crowdmapping und Wiki-Technologie entsteht eine Mobilitätsplattform zur kollaborativen Erfassung von Wegen und Routen sowie auch von Mobilitätshindernissen im Nahbereich, die den Gedanken der sozialen Mobilität umsetzt.

Soziale Mobilität

Klassische mobilitätsunterstützende Informationssysteme, allen voran die weit verbreiteten GPS-Navigationssysteme für den Individualverkehr, lösen Routingprobleme, d.h. Wegplanungen vom Startort zum Zielort. Die aktuelle Forschung im Bereich der Volunteered Geographic Information fasst dagegen Navigation als ein in der Gemeinschaft zu lösendes Problem auf, welches auf der Mitarbeit vieler beruht und die Mobilität der gesamten Gemeinschaft erhöht (Goodchild, 2007). Vorwiegend werden Ansätze verfolgt, die mobile soziale Netzwerke nutzen, deren Mitglieder Mobilitätshindernisse melden oder Strategien zu deren Beseitigung entwickeln (z.B. Beiträge in Zhou & Zie, 2009). Im Zentrum dieser Forschungsanstrengungen stehen informatische Ortsmodelle, die nicht nur die geographische Umwelt, sondern auch die Interaktion von Nutzer und Umwelt wiedergeben können (Schlieder & Henrich, 2011).

Eine soziale Mobilitätsplattform befähigt die Bewohner von Wohnquartieren, ihre persönliche Selbstständigkeit und individuelle Mobilität durch gemeinschaftliches Handeln zu bewahren. Im BMBF¹-Projekt EMN-MOVES (<http://www.emn-moves.de>) entwickeln Informatiker und Psychologen der Universitäten Bamberg und Erlangen zusammen mit Pflegedienstleistern und Wohnungsunternehmen in Nürnberg, Erlangen und Bamberg ein Set technischer und organisatorischer Dienste, die es möglich machen, eine nahtlose Mobilitätskette (von der eigenen Wohnung zum gewünschten Ziel und wieder zurück) planerisch zu erfassen und praktisch umzusetzen. Während der klassische Routenplaner die Frage beantwortet, wie *man* von A nach B kommt, beantwortet eine soziale Mobilitätsplattform die Frage, wie im Sinne einer Art Nachbarschaftshilfe eine Person X eine Person Y darin unterstützen kann, von A nach B zu gelangen. Im Rahmen des Projektes wird daher zusätzlich ein Matchmakingdienst für Fahrgemeinschaften und andere persönliche Hilfestellungen entwickelt und in die Plattform integriert, auf den wir im Folgenden nicht näher eingehen.

Primäre Zielgruppe des Projektes sind Senioren und betreuende Einrichtungen, allerdings legen wir Wert auf integrative und nicht ausgrenzende Maßnahmen, gerade die technischen Angebote richten sich daher an alle Bewohner eines Stadtviertels. Die im Rahmen des Projektes entwickelte Software bleibt nicht auf den Einsatz bei den Projektbeteiligten beschränkt, sondern wird als Open Source veröffentlicht.

1 Bundesministerium für Bildung und Forschung. Gefördert im Rahmen der Ausschreibung „Mobil bis ins hohe Alter – nahtlose Mobilitätsketten zur Beseitigung, Umgehung und Überwindung von Mobilitätsbarrieren“

Die Mobilitätsplattform



Abbildung 5: Prototyp Mobilitätsplattform

Die Karten von Openstreetmap bieten aufgrund ihres hohen Detaillierungsgrades mit Einbeziehung von kleinen Fußwegen wie auch interessanter POIs von der kleinen Bäckerei über die Bushaltestelle bis zum Briefkasten eine sehr gute Basis für die beschriebene Mobilitätsplattform. Zudem können sie bei Bedarf von jedem Nutzer erweitert und ergänzt werden. Doch selbst wenn jede Treppenstufe, die Breite jedes Fußwegs und die Höhe jedes Bordsteins in der Karte erfasst sind, muß die Plattform darüber hinaus die folgenden weiteren Aspekte abbilden.

Temporäre Mobilitätsbarrieren wie eine vereiste Fahrbahn (Illustration 2) stellen unter anderem für ältere Menschen ein besonderes Problem dar. Informationen über zur Zeit gesperrte oder nur eingeschränkt nutzbare Wege (Baustelle, Schneefall, etc.) sowie mögliche Alternativrouten können zwar gut auf einer Karte dargestellt werden, sollten aber aufgrund ihrer Kurzlebigkeit nicht in eine globale Kartendatenbank aufgenommen werden. Vielmehr muß die Plattform die Verarbeitung von schnell veraltenden Informationen beispielsweise durch die Angabe eines Verfallsdatums, Plausibilitätschecks oder anderer Werkzeuge zur technischen wie kollaborativen Qualitätssicherung unterstützen. Ebenso müssen ungenaue oder unsichere Angaben möglich sein, die gegebenenfalls dann durch andere Nutzer präzisiert werden können.



Abbildung 6: Vereiste Straße – eine temporäre Mobilitätsbarriere für manchen Fußgänger

Eine reine Kartendarstellung unterstützt weitere Beschreibungsformen von Text bis Photo nur eingeschränkt, für viele Informationen zur Mobilität wie Busfahrpläne oder Informationen zu Fahrgemeinschaften oder auch Parkregelungen bietet sich eher ein wikiartiges System mit Möglichkeiten zur Georeferenzierung an, wie dies beispielsweise die Einbettung von OSM-Karten in die Wikipedia schon in

Mobiles ortsbezogenes soziales Netzwerk: Ein Geowiki auf OSM-Basis

Ansätzen zeigt. Dies erlaubt auch die Verarbeitung zusätzlicher Informationen zum jeweiligen Wohnquartier, wie sie beispielsweise im etwas größeren Maßstab auch in einem Stadtwiki² zu finden wären.

Die Mobilitätsplattform folgt wie OSM selbst, aber auch darauf aufbauende Projekte wie wheelmap.org einem Crowdmappingansatz, fokussiert aber stärker auf die Erfassung temporärer Mobilitätsbarrieren in Text, Bild und Karte. Im Gegensatz zu Projekten wie Mobidat³, die stark strukturierte Daten zu Mobilitätsbarrieren mit speziell geschulten Mitarbeitern erfassen, liegt der Schwerpunkt hier auf der technischen Unterstützung kollaborativer Qualitätssicherung auf semistrukturierten Daten. Der wesentliche Aspekt ist die soziale Komponente. Zur Zeit existiert eine prototypische Testumgebung mit Einbindung von OSM-Karten in ein Semantic Mediawiki, die allerdings nur eine lose Kopplung bietet.

Zielgruppe

Die Mobilitätsplattform richtet sich an alle Bewohner und Besucher eines Quartiers oder Stadtviertels sowie an Dienstleister, die Hilfestellungen zur Erhöhung der Mobilität anbieten. Grundsätzlich sind die Beteiligten dabei nicht nur Informationsempfänger, also passive Rezipienten, sondern stellen im Sinne einer kollaborativen Plattform auch Informationen bereit.

Allerdings unterscheiden sich die Nutzer in mehrfacher Hinsicht. Zum einen sind sie unterschiedlich mobil, der Rollstuhlfahrer kämpft mit anderen Barrieren als der gebrechliche Fußgänger, der Vater mit Kinderwagen hat andere Anforderungen als die Joggerin.

Zum anderen sind die Beteiligten unterschiedlich technikaffin. Während Mrs. Smartphone direkt vor Ort ein Photo von der vereisten Gefahrenstelle macht und dieses gleich georeferenziert ins System hochlädt, gibt es gerade unter den älteren Bewohnern noch viele, die gar keinen Computer haben und nutzen, und die entsprechende Informationen über die Kinder, das Seniorenbüro, ehrenamtliche Helfer oder die Hausverwaltung bekommen oder auch telefonisch nachfragen.

Und schließlich treten die Nutzer in verschiedenen Rollen auf. Die Interessen und Möglichkeiten des einzelnen Bewohners können sich dabei deutlich von denen eines Dienstleisters oder der Hausverwaltung unterscheiden, die gegebenenfalls direkt auf solche Gefahrenstellen reagieren muß.

Forschungsfragen

Neben den technischen Implementierungsaspekten stellen sich daher die folgenden Forschungsfragen, die im Rahmen der zweijährigen Projektlaufzeit untersucht werden:

- Ortsinformationen und Wegbeschreibungen können in sehr unterschiedlicher Weise angeboten werden. Dies reicht von umgangssprachlichen Ortsbezeichnungen „am großen Brunnen“ über Adressen „Feldkirchenstraße 21, 96045 Bamberg“ bis hin zu Geokoordinaten „10.905653, 49.907141“. Welche Formen der semantischen Verarbeitung und Darstellung in Text und Karte bieten sich hier an?
- Gerade für Nutzer mobiler Endgeräte ist eine vom aktuellen Standpunkt abhängige Sicht auf die Plattform, gegebenenfalls auch für spezialisierte mobile Clients, hilfreich. Wie kann das System den für den Nutzer passende Kontext bestimmen?
- Welche Aspekte eines „Sozialen Netzwerks“ muß eine Mobilitätsplattform abbilden, um unterschiedliche personalisierte Sichten und die Zusammenarbeit der Nutzer der unterschiedlichen Rollen wie ehrenamtlicher Helfer, Bewohner usw. zu verbessern? Wie wird eine wiki-typische Zusammenarbeit an georeferenzierten Inhalten, insbesondere die Qualitätssicherung bei der Verfeinerung oder Korrektur von Georeferenzen unterstützt?

2 <http://allmende.stadtwiki.info/wiki/Node:Portal>

3 <http://www.mobidat.net/>

Mobiles ortsbezogenes soziales Netzwerk: Ein Geowiki auf OSM-Basis

- Wie können die unterschiedlichen Anforderungen der Nutzer mit unterschiedlichen Mobilitäts-einschränkungen am besten abgebildet werden? Welche Klassifizierungen eignen sich hier?
- Wie sieht ein sinnvoller Umgang mit kurzfristigen Mobilitätsbarrieren aus? Sollten Hindernisse wie Baustellen oder zugeschneite Wege ein Verfallsdatum haben?
- Wie kann eine Kopplung verschiedener lokaler Mobilitätsplattformen zu einem größere Gebiete umfassenden System aussehen?

Kontakt zu den Autoren:

Dr. Klaus Stein, Prof. Dr. Christoph Schlieder
Lehrstuhl für Angewandte Informatik
in den Kultur-, Geschichts-, und Geowissenschaften
Otto-Friedrich-Universität Bamberg
96045 Bamberg
+49 951 863 2845
klaus.stein@uni-bamberg.de
christoph.schlieder@uni-bamberg.de

Literatur

- [1] *Goodchild, M. F.*: Citizens as Sensors: the World of Volunteered Geography. *GeoJournal* 69(4), 211–221. 2007.
- [2] *Schlieder, C.; Henrich, A.*: Spatial Grounding with Vague Place Models. In: *SIGSPATIAL Special Issue on Geographic Information Retrieval.*, 2011.
- [3] *Zhou, X., Xie, X.*: Proceedings of the First International Workshop on Location Based Social Networks. *ACM Digital Library.*, 2009

Forschungsdatenmanagement mit Open-Source-Software

C. Willmes, S. Brocks, C. Hütt, D. Kürner, K. Volland und G. Bareth

Einführung

Im Rahmen des von der Deutschen Forschungsgemeinschaft (DFG, <http://www.dfg.de>) geförderten Sonderforschungsbereich 806 (SFB806, <http://www.sfb806.de>) wird eine webbasierte Forschungsdatenbank (<http://crc806db.uni-koeln.de>) entwickelt. Der SFB806 ist ein Forschungsprojekt mit über 100 beteiligten Wissenschaftlern, das sich mit der Ausbreitung des modernen Menschen (*Homo sapiens*) von Afrika nach Europa befasst.

Die SFB806-Datenbank erfüllt zwei Hauptaspekte: Zum ersten stellt sie eine Möglichkeit zur Publikation und zur Langzeitarchivierung von Daten die im Rahmen des SFB806 produziert werden zur Verfügung. Zum zweiten stellt sie den Forschern des SFB806 eine umfangreiche Forschungsdatenbasis zur Verfügung.

Durch die Forschungsdatenbank des SFB806 sollen die beteiligten Forscher einen intuitiven und integrierten Zugang zu relevanten Daten aus den Bereichen Paläoumwelt, pleistozäner und holozäner Archäologie sowie der Geoarchäologie erhalten. Dazu wurde und wird ein einheitliches Datenmodell [1,2] zur Beschreibung der drei genannten Domains auf der Basis von *Semantic Web Technologie* [3] fortlaufend entwickelt.

In diesem Beitrag wird im Folgenden die Bereitstellung der webbasierten Infrastruktur der SFB806-Datenbank unter Einsatz von standardisierten Datenformaten/Interfaces und Open-Source-Software beschreiben.

Datenmanagement

Die DFG fordert von allen Sonderforschungsbereichen (SFB) die Implementation eines Datenmanagements [4], das die Langzeitarchivierung und Verfügbarkeit der Forschungsergebnisse des SFB für eine Zeitspanne von mindestens 10 Jahren nach Ablauf des Forschungsprojektes gewährleistet. Das bedeutet bei einer maximalen Laufzeit eines SFB von 12 Jahren eine Datengewährleistung von bis zu 22 Jahren. Daten 22 Jahre zugreifbar vorzuhalten ist alles andere als trivial, wenn man betrachtet in welchen Datenformaten und auf welchen Datenträgern vor 22 Jahren (1990 – das Web 1.0 wurde gerade erst von Tim Berners-Lee erfunden) Forschungsdaten archiviert und zur Verfügung gestellt wurden. Diese Erkenntnis legt nahe, Daten ausschließlich auf der Basis von gut dokumentierten bzw. standardisierten Formaten und Interfaces bereitzustellen bzw. zu archivieren.

Aus diesem Grund wird das Datenmanagement des SFB806 zum einen auf der Basis von Open-Source-Software implementiert um die Nachvollziehbarkeit der Systemarchitektur zu gewährleisten und eine höhere Kontrolle über die Implementation des Systems zu haben und ggf. individuelle Anpassungen oder Erweiterung der Software vornehmen zu können. Zum zweiten wird das System auf der Basis von standardisierten Interfaces und Datenformaten implementiert, um die langfristige Nutzbarkeit der Datenbasis zu gewährleisten. Zum dritten werden die Daten in RDF modelliert um die Semantik der Daten klar und eindeutig zu definieren und um die Nutzbarkeit bzw. Integration der Forschungsergebnisse für zukünftige (externe) Forschungen und Anwendungen zu ermöglichen.

Implementation des Datenmanagement

Die von den Wissenschaftlern des SFB806 produzierten Daten werden zunächst im Format in dem der Wissenschaftler die Daten selbst nutzt und verwaltet in einen mehrfach gesicherten Speicherbereich gespeichert.

Der Speicherbereich basiert auf dem verteilten Dateisystems AFS [5], und wird durch das Rechenzentrum der Universität zu Köln (seit vielen Jahren ausfallsicher und datenverlustfrei) betrieben. Die Forscher haben beim Hochladen der Daten die Möglichkeit, Metadaten zur Beschreibung ihrer Daten anzugeben. Hierzu kann der Forscher aus vorgegebenen Vokabularen (Ontologien) zum Beschreiben seiner Daten auswählen. Momentan stehen die internen Ontologien für Archäologie, Palaeoumwelt und Geoarchäologie, die im Rahmen der Doktorarbeit des Erstautors formalisiert werden, sowie die weit verbreiteten Dublin Core [6] und ISO19115 [7] Vokabulare, zur Verfügung.

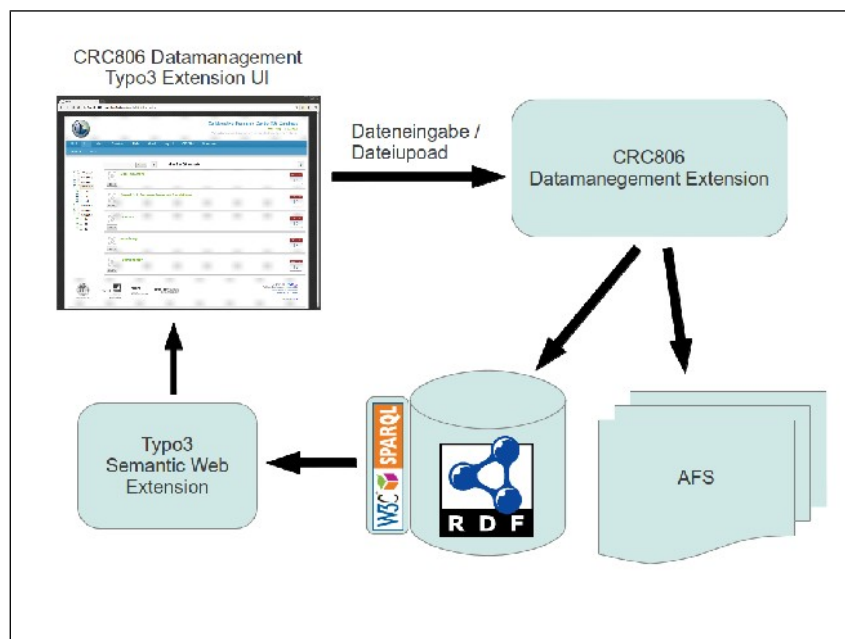


Abbildung 1: CRC806 Datamanagement Typo3 Extension.

In Abbildung 1 ist der Datenupload der SFB806-Datenbank schematisch dargestellt. Der Anwender lädt über das Interface der von uns entwickelten *CRC806 Datenmanagement Typo3-Extension* eine Datei hoch und stellt dazu ggf. Metadaten zur Verfügung. Aufgrund dieser Informationen wird einerseits die Datei im Dateisystem abgelegt und werden andererseits die Metadaten in RDF modelliert. Die RDF-modellierten Daten können über ein weiteres Interface, das die *Typo3 Semantic Web Extension* einsetzt, innerhalb der SFB806-Datenbank Webanwendung über den SPARQL-Endpoint durchsucht und angezeigt werden.

Das Frontend der SFB806-Datenbank ist eine Typo3 [8] basierte Webanwendung, für die eine Extension entwickelt wurde und wird, die das SFB806-Datenamangement (Dateiupload, Dateneingabe, Editieren und Anzeigen/Browsen der Daten) implementiert.

Forschungsdatenbank

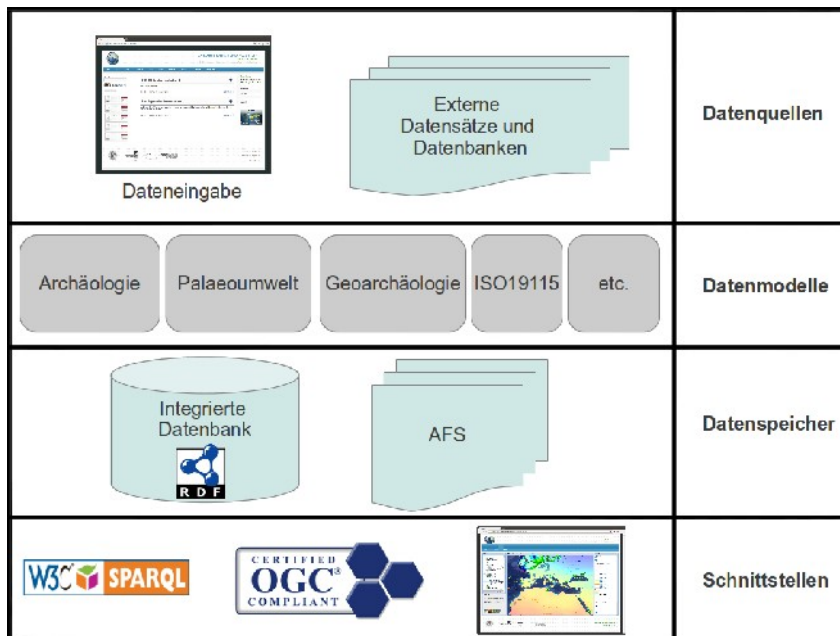


Abbildung 2: Systemarchitektur der Forschungsdatenbank.

Die Forschungsdatenbank dient als zentraler Datenpool für die Forscher des SFB806. Sie umfasst, wie erwähnt, vor allem Daten aus den Domains Archäologie, Geoarchäologie und Paläoumwelt. Die Datenbasis wird ständig erweitert, zum einen durch Integration von Publizierten Datensätzen aus den relevanten Fachdisziplinen und zum zweiten durch Eingabe von Daten durch die Forscher des SFB806.

Zusätzlich werden die Daten der Forschungsdatenbank fortlaufend mit externen Datenmodellen bzw. Ontologien/Vokabularen verlinkt. Zum einen um die semantische Aussagekraft der Daten zu erhöhen und zum zweiten um die Interoperabilität mit anderen Datenmodellen und Anwendungen zu unterstützen. Diese Technik ist auch unter dem Begriff *Linked Data* [9] bekannt.

Implementation der Forschungsdatenbank

Technisch werden die Daten in einer Graphdatenbank (Sesame [10]) gespeichert. Dazu werden die zu integrierenden Daten in den entwickelten Datenmodellen [1,2] in RDF [11] formuliert um sie dann im Sesame-Triplestore zu speichern.

Zu räumlichen Daten werden i.d.R. nur die Metadaten im RDF-Format gespeichert, der Datensatz selbst wird in einem herkömmlichen GIS-Datenformat (PostgreSQL/PostGIS Datenbank, als ESRI Shapefiles und/oder im GeoTiff-Format) gespeichert um diese über OGC WMS-, WFS- und WCS-Interfaces, und als Dateidownload zur Verfügung zu stellen.

Zum Zugriff auf die Daten der Forschungsdatenbank werden bis dato die folgenden Interfaces bereitgestellt:

- SPARQL-Endpoint
- OGC WMS, WFS, WCS
- GeoExt basiertes WebGIS

Weitere Interfaces sind in Planung (z.B. eine Exhibit [12] Timelinevisualisierung).

Ergebnisse

Bis jetzt sind ca. 20.000 archäologische, ca. 10.000 paläoumwelt und ca. 500 geoarchäologische Datensätze in die Forschungsdatenbank integriert. Diese Daten stehen somit den Forschern zur integrierten Analyse über die genannten Interfaces zur Verfügung. Zusätzlich gibt es eine Datenbank für GIS-Daten, die nicht in die drei genannten Modelle einzuordnen sind (v.a. Höhenmodelle/Bathymetrie und rezente/aktuelle Klimadaten und Umweltdaten), eine Literaturdatenbank und eine Mediendatenbank zur Speicherung aller Datensätze die in keinem der genannten Modelle modelliert werden (z.B. Projektberichte oder Videoaufnahmen von Vorträgen etc.).

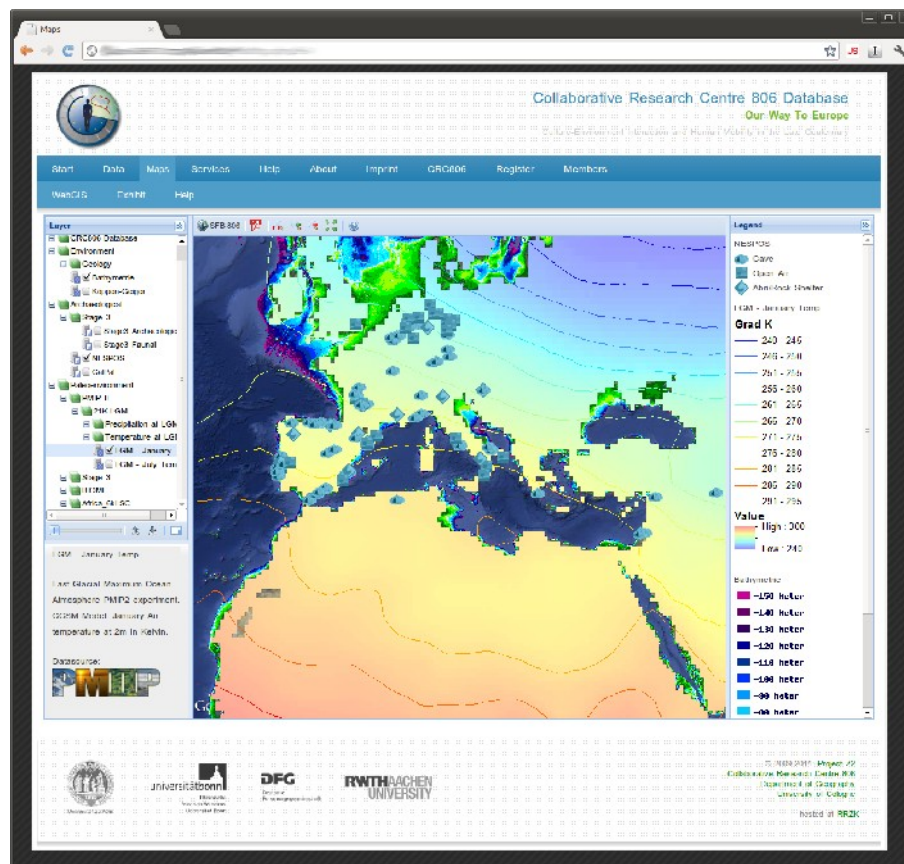


Abbildung 3: Screenshot des WebGIS-Interface.

Des weiteren steht der interessierten Öffentlichkeit ein Großteil der integrierten Datenbasis als OGC Webservices zur Verfügung. Der Zugriff auf den SPARQL-Endpoint ist vorerst nur für Mitglieder des SFB806 vorgesehen, da der Informationsvorsprung, der durch die integrierte Datenbank gegeben ist, zunächst intern für Forschungen genutzt werden soll. Nach Ablauf der aktuellen Projektphase (Sommer 2013) soll der SPARQL-Endpoint auch öffentlich zugänglich gemacht werden.

Fazit & Ausblick

Durch den Einsatz von Semantic Web Technologien ist es möglich, heterogene Datenbanken konsistent und mit überschaubarem Aufwand in einheitliche und eindeutige Modelle zu integrieren. Die Anwendung von Linked Data Methoden ermöglicht bzw. verbessert die Integration der entwickelten Modelle (und auf ihnen basierende Daten) in externe Anwendungen und umgekehrt die Einbindung externer Daten in SFB806-Anwendungen. Ein weiterer Vorteil von Graphdatenmodellen wie RDF ist, dass sie prinzipiell Schemaunabhängig sind. Dies ermöglicht die Weiterentwicklung von Datenmodellen unabhängig von der Anwendungsschicht [13].

Forschungsdatenmanagement mit Open-Source-Software

Das System befindet sich zur Zeit in der internen alpha-Testphase, der öffentliche Start der SFB806-Datenbank Webseite (<http://crc806db.uni-koeln.de>) ist für Sommer 2012 geplant.

Kontakt zum Autor:

(Dipl.-Geogr.) Christian Willmes
Geographisches Institut, Universität zu Köln
Albertus-Magnus-Platz, 50923 Köln
+49 (0)221 470 6234
c.willmes@uni-koeln.de

Literatur

- [1] *Willmes, C. und Bareth, G.*: A dataintegration concept for an interdisciplinary research database. In: Proceedings of the GI_Zeitgeist Young Researchers Forum, Universität Münster, 2012.
- [2] *Willmes, C., Brocks, S., Hoffmeister, D., Hütt, C., Kürner, D., Volland, K. und Bareth, G.*: Facilitating spatio-temporal visualization and analysis of heterogeneous archaeological and palaeoenvironmental research data. In: Proceedings of the XXII IRSPRS Congress, Melbourne, 2012.
- [3] Allemang, D. and Hendler, J.: Semantic Web for the working Ontologist – Effective Modeling in RDFS and OWL. Morgan Kaufman Publishers/Elsevier, Amsterdam, Second Edition, 2011.
- [4] Effertz, E.: The funders perspective: Data management in coordinated programmes of the German research Foundation (DFG). In: Curdt, C. und Bareth G. (Hrsg.), Proceedings of the Data Management Workshop 29.-30.10.2010, Kölner Geographische Arbeiten, Heft 90, Universität zu Köln, pp. 35-38.
- [5] AFS – Andrew File System: http://de.wikipedia.org/wiki/Andrew_File_System.
- [6] DublinCore Metadata Initiative: <http://dublincore.org/>.
- [7] Geographic Information – Metadata ISO19115: http://de.wikipedia.org/wiki/ISO_19115.
- [8] Typo3 CMS: <http://typo3.org/>.
- [9] Linked Data: http://en.wikipedia.org/wiki/Linked_data.
- [10] SESAME: <http://www.openrdf.org/>.
- [11] Resource Description Framework - RDF: <http://www.w3.org/RDF/>.
- [12] Exhibit - Publishing Framework for Data-Rich Interactive Web Pages: <http://www.simile-widgets.org/exhibit/>.
- [13] Segaran, T., Evans, C. and Taylor, J.: Programming the Semantic Web. O'Reilly, Sebastopol, CA, USA. 2009.

Das Anlagen- und Indirekteinleiterkataster der Stadt Bielefeld

Gerhard Genuit

1. Entstehung

Die Ursprünge des Katasters gehen in die frühen 90er Jahre zurück. Mit der Einführung von Einzelplatz-PCs sollte die bis dahin im Einsatz befindliche Großrechneranwendung Uschi (Umweltschutzinformationssystem) abgelöst werden. Angestrebt war eine Lösung mit Informix als Datenbank und Frontends, die in Smalltalk programmiert waren. Aufgrund verschiedenster Unzulänglichkeiten konnte sich diese Programmierung jedoch bei den Anwendern und Anwenderinnen nicht durchsetzen.

Da die PCs aber mittlerweile mit MS-Access ausgestattet waren, begannen zahlreiche Sachbearbeiter und Sachbearbeiterinnen für ihre Probleme individuelle Lösungen zu schaffen. Diese wurden Mitte der 90er Jahre auf Basis von Access zu einer Datenbank zusammengeführt. Um eine ausreichende Sicherheit der Daten zu gewährleisten, wurden die Tabellen bald auf ein mehrbenutzerfähiges Backend übertragen. Hierfür wurde zunächst eine DB2 und später eine Informix Datenbank eingesetzt.

Im Jahre 2005 wurden die Frontends auf Java portiert und seit letztem Jahr liegen unsere Daten in einer PostgreSQL[1] Datenbank.

2. Aufgaben

Das Kataster soll schnell einen möglichst umfassenden Überblick über alle umweltrelevanten Anlagen an einem Standort bieten. Zudem soll es auch einen fach- und organisationsübergreifenden Informationsaustausch ermöglichen. Details aus der Sachbearbeitung werden sicher noch eine ganze Zeit in Hausakten abgelegt, aber die grundlegenden Daten der Bearbeitung werden auch über Abteilungsgrenzen hinweg bereitgestellt.

Für die Wartung und Funktionskontrolle von Lager- oder Produktionsanlagen müssen oft Fristen von mehreren Jahren nachgehalten werden. Dies ist bei einer großen Anzahl von Anlagen nur noch über ein EDV gestütztes System möglich.

Die Vergangenheit hat gezeigt, dass oft erst nach Jahren eine Belastung auf einem Standort festgestellt wird. In diesem Fall kann der Verursacher oft nur schwer oder gar nicht mehr ermittelt werden. Daher ist es wünschenswert, die Entwicklung eines Standortes möglichst vollständig dokumentiert zu haben. Nur auf diese Weise kann man Verunreinigungen einem Verursacher zuordnen, auch wenn dieser schon lange den Standort verlassen hat.

Stadt Bielefeld Umweltamt		Flaschen		Probenahmeauftrag E005/12	
Probenahmeprotokoll für Abwasser E-Satzung					
Anlagenbetreiber/in Einleiter/in					
Name/Firma: Duminy GmbH & Co KG					
Straße, Hausnr.: Bielefelderhof 14					
Ausschleisschleuse		Ausschleisschleuse		Ausschleisschleuse	
* Bei mehreren Versuchen bitte den zu versuchenden Versuchsgegenstand					
Probeentnahmezeitpunkt					
Revisionsnummer an der Grundstücksgrenze		Baujahr des Gebäudes		Datum	
Eigenschaft des Grundstückes		Baujahr des Gebäudes		Datum	
Abwasser		Umweltlabor		Umweltlabor	
Beobachtungen im Abwasserstrom					
Turbid.		Schwebstoffe		Schwebstoffe	
Turbid.		Schwebstoffe		Schwebstoffe	
Örtliche Verhältnisse:					
am Entnahmesteg		am Vorlag		am Vorlag	
ja/nein		ja/nein		ja/nein	
ja/nein		ja/nein		ja/nein	
Messungen vor Ort					
Temperatur		pH-Wert		Leitfähigkeit	
Temperatur		pH-Wert		Leitfähigkeit	
Bemerkungen:					
Die oben genannten Arbeiten wurden ausgeführt von:					
Name: Genuit, Gerhard					
Stellen, Stellung: 380 35, 2521/51 807					
Unterschrift des Probennehmers/der Probennehmerin:					
Stempel des Umweltamtes/der Umweltamtin:					

Abb 1: Probenahmeauftrag

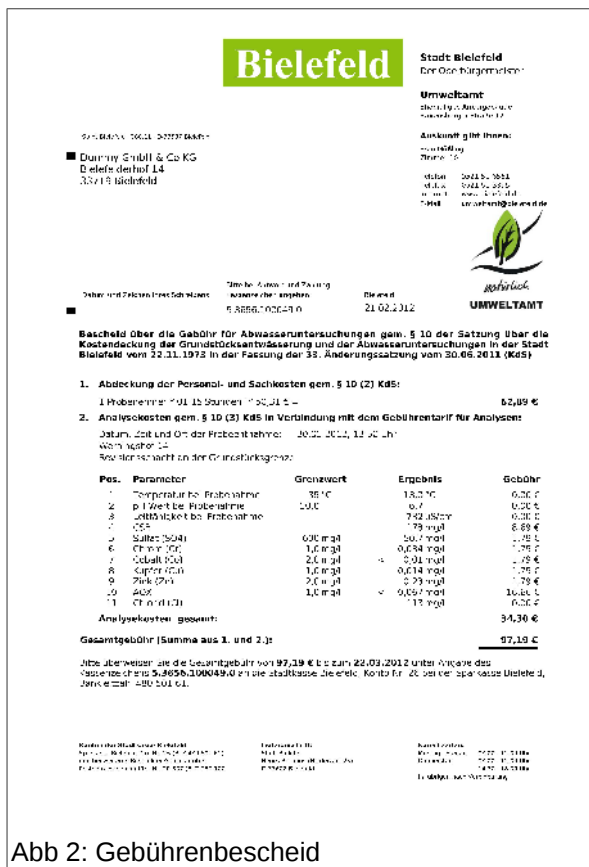


Abb 2: Gebührenbescheid

Eine weitere wichtige Aufgabe des Katasters besteht in der Erstellung von statistischen Auswertungen. Diese werden immer wieder und immer öfter von stadtinternen Stellen, aber auch von vorgeetzten Dienststellen angefordert.

In der derzeitigen Form verwenden wir das Programm für die Überwachung von Indirekteinleitern, also Firmen, die Abwasser mit gefährlichen Stoffen in den Kanal einleiten, und Anlagen, in denen mit wassergefährdenden Stoffen umgegangen wird. Ein Ausbau auf andere Bereiche ist nicht konkret geplant, durch die modulare Struktur aber jederzeit möglich. In dem Kataster können Informationen zu Objekten gespeichert werden, die über einen Standort, einen Betreiber und eine Objektart definiert sind. In der einfachsten Form muss lediglich eine neue Objektart angelegt werden, und die neuen Objekte können mit einem Minimum an Informationen erfasst werden. Sollen weitere Details gespeichert werden, geht dies über eine zusätzliche Fachdatentabelle und weitere Bearbeitungsmasken.

Um einen möglichst großen Nutzen zu erreichen, ist es wichtig, dass die Daten, wenn es möglich ist, automatisch importiert werden. Dies haben wir beispielsweise im Bereich der Probenahmen realisiert,

wo ein vollständiger Workflow in dem System entstanden ist. Dieser beginnt beim Erstellen eines Probenahmeauftrags über den Import der Ergebnisse und die Freigabe der Daten und geht bis hin zum Erstellen eines Gebührenbescheides und der Übergabe einer Textdatei an die Stadtkasse.

3. Programmierung

Wie bereits erwähnt, sind die Frontends in Java programmiert. Für eine ansprechende Optik sorgen die Jgoodies[1] und als Persistenzschicht übernimmt Hibernate[2] die Kommunikation zwischen dem Frontend und den Datenbanktabellen. Hierdurch wird die Programmierung vereinfacht und ein Wechsel des Datenbank Backends ist recht einfach möglich. Grafische Auswertungen von Messdaten machen wir mit JFreeChart[3] und für die Ausgabe von PDF-Dokumenten bis hin zur automatischen Erstellung von Gebührenbescheiden für Abwasserproben nach Entwässerungssatzung kommt Jasper[4] zum Einsatz.

Von allen eingesetzten Komponenten wurden nur die als OpenSource verfügbaren Varianten verwendet, und auch unsere eigene Programmierung haben wir unter einer GPL-Lizenz freigegeben.

4. Bedienung

Beim Start des Programms öffnet sich nach der Authentifizierung ein Fenster zum Suchen eines Standortes, da dies der häufigste Einstieg in die Bearbeitung ist.

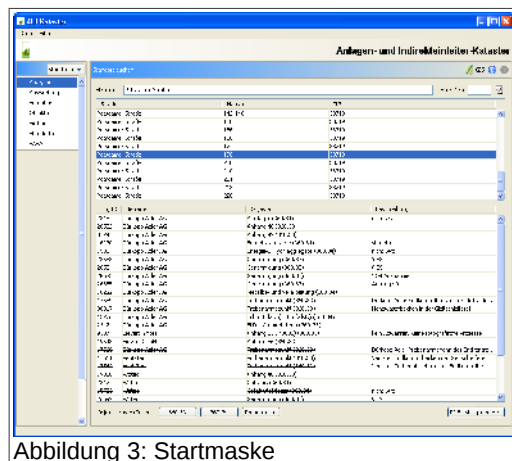


Abbildung 3: Startmaske

Das Anlagen- und Indirekteinleiterkataster der Stadt Bielefeld

Wenn hier ein Standort ausgewählt wird, bekommt man im unteren Bereich einen Gesamtüberblick über alle erfassten Objekte. Durch Doppelklick auf ein Objekt können die weiteren Details aufgerufen werden.

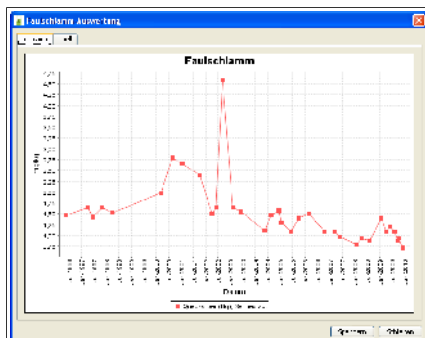


Abbildung 4: Messwertdiagramm

Das Programm besteht aus verschiedenen Modulen, die zu Kategorien zusammengefasst sind. Das o. g. Modul „Standort suchen“ befindet sich z. B. in der Kategorie „Betriebe“. Weitere Kategorien sind: Auswertung, Klärschlamm, Sielhaut, Labor und VAWS (Anlagen zum Umgang mit wassergefährdenden Stoffen).

Messwerte sind in dem System zu Abwasser-, Klärschlamm- und Sielhautuntersuchungen (dies sind Untersuchungen im Kanalnetz, die Informationen über die Qualität des Abwassers liefern) enthalten. Diese können, wie bereits oben erwähnt, automatisiert über CSV-Dateien importiert und, grafisch aufbereitet, dargestellt werden.

5. GIS Schnittstelle

Als Geoinformationssystem kommt bei der Stadt Bielefeld u. a. Quantum GIS[5] zum Einsatz. Das Programm deckt alle Anforderungen der Sachbearbeitung, bei denen es nur sehr selten um die Erfassung von Flächen geht, vollständig ab. Darüber hinaus kann es als OpenSource Produkt auf beliebig vielen Rechnern installiert werden. Um dieses Programm nun möglichst komfortabel direkt aus dem Kataster heraus nutzen zu können, wurde eine lockere Anbindung programmiert.

Diese erlaubt es, das Programm mit einem lokal gespeicherten Projekt, ausgehend von einem Standort, zu starten und den Rechts- und Hochwert eben dieses Standortes über zwei Umgebungsvariablen einem Plugin zu übergeben, mit dem man dann dort hin navigieren kann. Umgekehrt können Rechts- und Hochwerte aus der Karte abgegriffen und über die Zwischenablage in einen Datensatz übernommen werden.

Das Screenshot zeigt das Fenster 'Anlagen- und Indirekteinleiter-Kataster' mit einer Tabelle der Standorte. Die Tabelle enthält Spalten für Adresse, Hausnummer, PLZ, Koordinaten und weitere Details. Ein Kontextmenü ist über einer der Zeilen geöffnet.

Adresse	Hausnummer	PLZ	Koordinaten	Weitere Details
Werninghof	4	33745	4.75	Werninghof 4
Werninghof	6	33745	4.75	Werninghof 6
Werninghof	8	33745	4.75	Werninghof 8
Werninghof	10	33745	4.75	Werninghof 10
Werninghof	12	33745	4.75	Werninghof 12
Werninghof	14	33745	4.75	Werninghof 14
Werninghof	16	33745	4.75	Werninghof 16
Werninghof	18	33745	4.75	Werninghof 18
Werninghof	20	33745	4.75	Werninghof 20
Werninghof	22	33745	4.75	Werninghof 22

Abbildung 4: GIS Aufruf

Eine weitere Möglichkeit, zu einem Standort zu navigieren, besteht über eine Programmierung, die ebenfalls von dem „zoom-to-point“ Plugin abgeleitet wurde. Diese greift auf eine Tabelle zu, in der zu jedem Standort mit Straße und Hausnummer ein Rechts- und Hochwert hinterlegt ist. Wird hier ein Standort ausgewählt, wird der Bildausschnitt aufgrund dieser Werte dort hin verschoben.

Seit dem Umstieg auf die PostgreSQL/PostGIS Datenbank Anfang letzten Jahres, wurde diese in "fiumwelt", was für Fach-Informationen-System-Umwelt steht, umbenannt. Neben dem Schema "auik", in dem die Sachdatenhaltung des Katasters stattfindet, gibt es in dieser Datenbank ein Schema "gis" für Geoinformationen. Hier sind z. B. der gesamte Kanalbestand, die Wasserschutzgebietsgrenzen, Altlastenverdachtsflächen, Zuständigkeitsbereiche usw. gespeichert. Da unser Katasteramt eine Vielfalt von Karten als WMS zur Verfügung stellt, ist es nun auch wenig geübten Benutzern und Benutzerinnen möglich, ein QGIS-Projekt mit Hintergrundkarten und Fachdatenebenen zusammen zu stellen.

Das Anlagen- und Indirekteinleiterkataster der Stadt Bielefeld

Ein weiteres Schema mit dem Namen "oberflgw" ist mittlerweile hinzugekommen. Hier werden die Daten zur Bearbeitung von Einleitungen in Oberflächengewässern gespeichert. Sobald das Programm "AqualInfo", das bei uns zur Erfassung der Grundwassermessstellen, Trinkwasserbrunnen und Sondierungen eingesetzt wird, die PostgreSQL Datenbank unterstützt, werden auch diese Informationen hier abgelegt und damit über die gleiche Schnittstelle im QGIS verfügbar sein.

Durch diese Struktur sind nun auch Abfragen und Veränderungen in den Datensätzen entsprechend ihrer örtlichen Lage einfach möglich. Beispielhaft sollen hier ein select und ein update Befehl für Standorte, die in dem Sachdatenschema „auik“ gespeichert sind, nach ihrer Lage in einem Entwässerungsgebiet bzw. einem VAWS-Bezirk, die in dem „gis“ Schema liegen, genannt werden:

```
SELECT st.id, st.strasse, st.hausnr, st.entgebid, ent.entw_geb
FROM auik.basis_standort AS st, gis.abk_entwaesserungsgebiete AS ent
WHERE Contains ( ent.the_geom, st.the_geom ) = TRUE
      AND ent.entw_geb = '1.01';
```

```
UPDATE auik.basis_standort SET wassereinzgebid = vbez.ezgb_id
FROM   gis.vaws_bezirke AS vbez
WHERE Contains ( vbez.the_geom, auik.basis_standort.the_geom ) = TRUE;
```

Nähere Informationen erhalten Sie bei Gerd Genuit (gerd.genuit@bielefeld.de). Der Quellcode kann unter <http://wald.intevation.org/projects/auik/> heruntergeladen werden. Hier wird auch in Kürze eine Webseite zu dem Projekt entstehen.

Kontakt zum Autor:

Gerhard Genuit
Stadt Bielefeld
Ravensberger Straße 12
33602 Bielefeld
0521/51-2832
gerd.genuit@bielefeld.de

Quellen

- [1] <http://www.postgresql.de/>
- [2] <http://www.jgoodies.com/>
- [3] <http://www.hibernate.org/>
- [4] <http://www.jfree.org/>
- [5] <http://jasperforge.org/projects/jasperreports>
- [6] <http://www.qgis.org/>
- [7] <http://www.pgadmin.org/>

Konzeption eines Spatial Business Intelligence System zur Analyse und Steuerung von Re-Produktionsketten auf Grundlage von OpenSource Komponenten

Martin Becker^a, Christian Wolff^b

^a Institut für Architektur, Facility Management und Geoinformation, Hochschule Anhalt, Dessau, M.Becker@afg.hs-anhalt.de

^b Institut für Architektur, Facility Management und Geoinformation, Hochschule Anhalt, Dessau, C.Wolff@afg.hs-anhalt.de

Zusammenfassung - Das vorliegende Dokument stellt die Konzeption einer Anwendung zur Analyse und Steuerung von regionalen Wertschöpfungsketten in den Mittelpunkt. Die Intention ist die innovative Kombination von Methoden der Business Intelligence mit denen der Geoinformationstechnologie in den Bereichen des nachhaltigen Landmanagements und Erneuerbare Energien um Entscheidungsunterstützungs- und Planungsprozesse zu fördern. Durch die gewonnene Akzeptanz und Bedeutung von Geoinformationen ergeben sich bei der Konzeption eines räumlichen Entscheidungsunterstützungssystems für eine breite Anwenderschicht neue Herausforderungen, um räumliche Informationen zu kommunizieren. In diesem Paper werden auf die Anforderungen, Methodik und Organisationsstruktur des zu konzipierenden Systems eingegangen. Es konzentriert sich dabei auf die Einbindung und Nutzung von OpenSource Komponenten in einem komplexen und interdisziplinären Arbeitsumfeld.

1 Einführung

Re-Produktionsketten beschreiben Kombinationen technischer Prozesse des Wasser- und Energiesektors im ländlichen Raum, die in stofflicher, energetischer oder wirtschaftlicher Hinsicht miteinander in Beziehung gesetzt sind und als Ziel die lokale Wertschöpfung und Ressourceneinsparung verfolgen. Reststoffe aus der Land- und Forstwirtschaft, geklärtes Abwasser, Klärschlamm und Faulgas, Abwärme aus Kläranlagen und Biogasanlagen – was bislang Abfall war, soll in Zukunft der Rohstoff für regionale Wertschöpfungsketten werden. Unternehmen, Kommunen und Bürger – alle, die bislang diese Abfälle produzieren, werden so zu Lieferanten dieser Sekundärressourcen. Angesichts steigender Preise für Energie und Rohstoffe interessieren sich Kommunen und Unternehmen sehr für den Aufbau solcher regionalen Wertschöpfungsketten. In diesem interdisziplinären Arbeitsfeld aus diversen Praxis- und Wissenschaftspartnern⁴ setzt sich das Forschungsprojekt „Re-Produktionsketten in der Wasser- und Energieinfrastruktur in schrumpfenden Regionen“⁵ mit der Identifizierung, dem Aufbau und der Steuerung derartiger regionaler Stoffkreisläufe auseinander.

Im Teilprojekt „Räumliche Abbildung und Steuerung“ wird für die räumlich-analytische Untersuchung der starken räumlichen Beziehungen zwischen den Akteuren, den technischen und natürlichen Elementen der Re-Produktionsketten ein räumliches Entscheidungsunterstützungssystem (Spatial Business Intelligence System, kurz SBIS) auf Grundlage von OpenSource Komponenten aufgebaut, welches auf kommunaler Ebene zum Einsatz kommen soll. Die tragende Herausforderung ist dabei die kombinierte räumliche Betrachtung von GIS-Analysen und Business Intelligence in einer Web-basierenden Anwendung. Ein solches System besteht aus verschiedensten Technologien um den umfangreichen Prozess aus Datenrecherche, -beschaffung, -bereitstellung, -integration, -modellierung und -visualisierung zu verarbeiten und zu steuern.

4 Die Partner aus Verwaltung und Wissenschaft in dem Verbundprojekt sind die Klimaschutzregion Elbe-Elster, die Bioenergieregion Wittenberg, inter3 als Verbundkoordinator, die BTU Cottbus, die TU Berlin und die Hochschule Anhalt.

5 Siehe <http://www.reproketten.de/>

Als interdisziplinäres Fachgebiet zeigt die Geoinformatik in der Verknüpfung von Informatik und den raumbezogenen Wissenschaften ihre besondere Bedeutung. Die verstärkte Nutzung erneuerbarer Energien, aufgrund der Energiewende⁶, mit dem von Politik und Gesellschaft beschlossenen Ausstieg aus der Kernenergie und der verstärkten Entwicklung der dezentralen Energieversorgung im ländlichen Raum mit der Bildung von energieautarken Kommunen, wie z.B. Feldheim in Brandenburg⁷, eröffnen der Geoinformatik ein wachsendes Anwendungsspektrum als technologischer Begleiter der Energiewende (Eder 2011, S.41[2]). Auch die Trendanalyse des Jahres 2011 vom Runden Tisch GIS e.V. zeigt die aktuelle Bedeutung des Einsatzes von Geoinformationssystemen (GIS) im Zusammenhang mit den erneuerbaren Energien (Kutzner et al. 2011[3]).

2 Problemstellung und Methodik

2.1 Problemstellung

Für die Entwicklung des SBIS in diesem Projekt ergaben vor allem durch den wissenschaftlichen Hintergrund einige Herausforderungen, die es zu lösen galt. Eine große Herausforderung war, dass in dem SBIS Lösungen integrieren werden müssen, die von den Projektpartnern erst während der Projektlaufzeit entwickelt werden. Außerdem werden während der Projektarbeit ständig neue Probleme erkannt, was einen sehr dynamischen Softwareentwicklungsprozess nötig macht. Zu Beginn des Forschungsprojekts waren nur Grundanforderungen an das zu entwickelnde SBIS für die Umsetzung des Requirement Engineerings festlegbar. Problematisch ist dabei auch die breite Definition der Zielgruppe, die sowohl engagierte Bürger, Investoren, fachlich versierte Fachplaner, Verwaltungsangestellte sowie regionale Akteure einschließt. Dadurch können nur wenig technische und fachliche Anforderungen vorausgesetzt werden.

Weiter Herausforderungen ergeben sich durch den fachlichen Hintergrund des Projekts. So wird zur Planung von Re-Produktionsketten eine Vielzahl an Daten aus unterschiedlichsten Quellen benötigt. Dies reicht von verschiedensten amtlichen Daten bis zu Daten, die mittels Crowd Sourcing erhoben wurden. Neben den Fragestellungen zur Integration und Homogenisierung der Daten gestaltet sich auch die Nutzung und Bereitstellung öffentlicher Daten oft schwierig. Bei der Planung und Umsetzung von Re-Produktionsketten muss außerdem Rückhalt und Einbeziehung aller Beteiligten, also Akteure, Investoren, Verwaltung und Bürger, beachtet werden. Wichtig ist auch bei der Entwicklung des SBIS Möglichkeiten zur Verstetigung / Verbreitung und Aktualisierung der Werkzeuge und des integrierten Wissens einzuplanen.

2.2 Methodik

Um den organisatorischen Herausforderungen Rechnung zu tragen, wird als Softwareentwicklungsmethode „Prototyping“ eingesetzt. Dies ermöglicht das relativ schnelle Prüfen und Anpassen des jeweiligen Prototyps an die aktuellen Projekterkenntnisse, sowie das Testen neuer Ideen. Lasten- und Pflichtenheft dienen zur formalen Definition eines Prototyps, sowie zur Festlegung von Vision und Zielen. Zu Beginn des Teilprojekts wurden die Grundaufgabenbereiche des zu entwickelnden Systems abgesteckt. Diese sind:

- Datenerfassung
- Datenintegration

⁶ Siehe <http://www.spiegel.de/thema/energiewende>

⁷ Siehe <http://www.brandenburg.de/sixcms/media.php/4055/feldheim.pdf>

Konzeption eines Spatial Business Intelligence System zur Analyse und Steuerung von Re-Produktionsketten auf Grundlage von OpenSource Komponenten

- Datenhaltung
- Datenanalyse
- Datenvisualisierung
- Grundsätze

Diese Aufgabenbereiche müssen durch Softwarekomponenten abgedeckt werden. Dabei kann die Auswahl der Komponenten von Prototyp zu Prototyp variieren. Die ausgewählten Komponenten werden dann zu einem Prototyp zusammengefügt und getestet. Um den Arbeitsaufwand dabei gering zu halten, muss möglichst wenig Arbeitsaufwand, beispielsweise in Form von Programmierung, in die Komponenten selbst fließen. Dadurch bleiben sie austauschbar. Deshalb besteht bei der Entwicklung des SBIS der Grundsatz „**Erweitern vor Ändern**“, also besser eine nötige Funktion durch eine zusätzliche Softwarekomponente integrieren als die Funktion in bestehende Komponenten selbst einzuprogrammieren. Dies erhält auch die Möglichkeit zum Updaten einer Komponente und trägt so zur Langlebigkeit bzw. Nachhaltigkeit des Systems bei. Um den speziellen Herausforderungen gerecht werden zu können, wurden weitere Grundsätze bei der Systementwicklung festgelegt. Da zu Beginn des Teilprojekts noch keine Spezifikationen erhoben werden konnten, sollen diese Grundsätze dazu beitragen das System an die Bedürfnisse und Qualitätsansprüche anzupassen. Nachfolgend werden diese näher vorgestellt:

Nachhaltigkeit

Nachhaltigkeit soll hier auch für die Entwicklung des Systems berücksichtigt werden. Dies bedeutet, dass die Infrastruktur anpassbar an neue fachliche Aufgaben sein soll. Außerdem sollen integrierte Informationen und Werkzeuge anpassbar und aktualisierbar sein. Weiterhin soll bei der Auswahl von Softwarekomponenten ebenfalls auf Langlebigkeit geachtet werden. Durch den reinen Einsatz von OpenSource Komponenten kann die Weiterentwicklung des Systems durch eine Community erfolgen.

Serviceorientierung

Da hier keine Software im eigentlichen Sinne neu entwickelt wird, sollen hierbei nur Konzepte der SOA beachtet werden. Folglich sollen die Softwarekomponenten durch Services Daten austauschen und steuerbar sein können. Außerdem sollen durch Komposition, also durch das Zusammenspiel verschiedenster Softwarekomponenten neue Anwendungen entwickelt werden. Dadurch können einzelne Softwarekomponenten leicht ersetzt, neue Komponenten integriert oder einzelne Komponenten, beispielsweise in die Cloud, ausgelagert werden.

Metadaten zur Datenverwaltung

Seit einigen Jahren wird die Entwicklung des Web 2.0 zum Semantic Web, also Web 3.0, vorausgesagt. Durch den Einsatz von Metadaten sollen auch hier die großen Datenmengen strukturiert und besser verwaltbar gemacht werden.

Motivation schaffen und Aufwand senken

Um ein neues System zu etablieren und Personen zur Mitarbeit am System zu bewegen ist es wichtig die Personen entsprechend zu motivieren und den Aufwand zur Einarbeitung in das System zu senken. Konkret soll beispielsweise das Gesamtsystem auf Linuxbasis als virtueller Server bereitgestellt werden. Außerdem soll der Erstaufwand möglichst gering gehalten werden. Positiv dazu trägt die Nutzung von OpenSource Komponenten bei. Dadurch fallen keine Lizenzkosten an. Durch die lose Struktur kann das System auch relativ einfach in bestehende Systeme integriert werden. Deshalb wird bei der Komponentenauswahl auch auf die Verbreitung der potenziellen Softwarekomponenten geachtet. Gleiches gilt für die Voraussetzungen für den Nutzer des Systems. Hier soll keine spezielle Software vorausgesetzt werden sondern nur Standardsoftware, für die auch freie Versionen verfügbar sind.

Konzeption eines Spatial Business Intelligence System zur Analyse und Steuerung von Re-Produktionsketten auf Grundlage von OpenSource Komponenten

Mit der genannten Zielsetzung steigen die Anforderungen an das System. Zur Steigerung der Akzeptanz des Entscheidungsunterstützungssystems werden dabei folgende Aspekte berücksichtigt:

- Einfache Oberfläche und hohe Benutzerfreundlichkeit mittels Profil- und Nutzerorientierung
- Ansprechendes Look & Feel
- Hohe Performance auf Echtzeitniveau

Mit der Einhaltung der aufgezeigten Bedingungen werden hohe Zugriffszahlen und Informationsübermittlungen geliefert. Das System darf in diesem Zusammenhang nicht nur als Werkzeug für Fachanwender, sondern ebenfalls für den normalen, fachfremden Nutzer konzipiert werden. Es wird nur ein Mehrwert für den Anwender erzeugt, wenn die Bedürfnisse der jeweiligen Nutzergruppen erfüllt werden. Dabei gilt es nicht nur den Grundbedürfnissen nachzukommen, sondern auch Zusatznutzen und Anreize für den Nutzer zu generieren, um die Weiterentwicklung des Systems zu gewährleisten.

Offenheit und Transparenz

Da durch das zu entwickelnde SBIS auch hoheitliche Aufgaben durchgeführt werden sollen, muss das System offen und transparent gestaltet werden. Folglich soll es niemanden benachteiligen oder ausschließen. Daraus folgt auch der Anspruch auf Transparenz bzw. Verständlichkeit / Nachvollziehbarkeit des Systems. Dies kann ohnehin nur bei einem quelloffenem System gewährleistet werden.

Das Gelingen eines kommunalen Projektes hängt von seiner Akzeptanz, auch in der regionalen Bevölkerung, ab. Die offene Kommunikation und Transparenz des Systems dient dabei als vertrauens- und akzeptanzbildender Ansatz, um Vertrauen zu Bürger, Entscheidungsträgern und Investoren aufzubauen. Die Methoden der Geoinformationstechnologie können die Kommunikation durch einfache räumliche Visualisierung wesentlich unterstützen und Informationen für jedermann direkt und einfach liefern. Mit dieser Art der Bürgerbeteiligung soll das SBIS nicht nur als politisches, sondern auch als gesellschaftliches Instrument dienen.

Diese **Grundsätze** werden auch bei der Auswahl der Softwarekomponenten beachtet. Dadurch begrenzt sich die Menge der zu prüfenden, verfügbaren Softwarekomponenten für eine Aufgabe bereits auf eine einstellige Anzahl. In einigen Fällen, beispielsweise bei Contentmanagementsystemen (CMS), gibt es allerdings auch eine Vielzahl an OpenSource Softwaresysteme. Bei der Auswahl der Komponenten wird hier zunächst mithilfe einer Nutzwertanalyse die Anzahl der potenziellen, verfügbaren Softwarekomponenten auf nicht mehr als drei eingeschränkt, die dann einer genaueren Prüfung unterzogen werden. Für die Auswahl von OpenSource Softwarekomponenten gibt es viele Ansätze, beispielsweise die Evaluationskriterien von Cruz et al.[1]. Für die hier durchgeführte Nutzwertanalyse wurden, abhängig von der gewünschten Funktionalität, eigene Kriterien mit eigenen Gewichten ausgewählt.

Dabei wurde ausgehend von der wichtigsten Komponente, also der mit den wenigsten Alternativen und der für das System wichtigsten Funktion eine Beispielauswahl aufgebaut. Als Ausgangspunkt für die Komponentenauswahl wurde hier das DBMS ausgewählt, da die Zahl der Alternativen überschaubar und die übernommene Funktion, hier vor allem Datenhaltung georeferenzierter Daten, in diesem System ein Muss ist. Die verfügbaren OpenSource Alternativen waren SQLite mit Spatialite, PostgreSQL 9 mit PostGIS, Ingres mit Geo Community und MySQL 5. Über die Kompatibilität zum hier ausgewählten DBMS PostgreSQL und dem Anwenden der Nutzwertanalyse wurden dann weitere Komponenten ausgewählt. Grund für den Einsatz der Nutzwertanalyse war die einfache Anpassbarkeit an eigene Anforderungen. Nachteilig dabei ist, dass die Bewertungen insbesondere bei vielen qualitativen Kriterien zur Subjektivität neigen. Aus diesem Grund wurden möglichst quantitative Kriterien integriert, beispielsweise die Communitygröße eines OpenSource Projekts da diese sich positiv auf die Softwarequalität auswirkt. Außerdem fördert die diese Umfang und Bestand einer OpenSource Software. Um Hinweise auf die Communitygröße eines OpenSource Projekts zu erhalten kann eine Vielzahl an Indi-

Konzeption eines Spatial Business Intelligence System zur Analyse und Steuerung von Re-Produktionsketten auf Grundlage von OpenSource Komponenten

katoren untersucht werden, beispielsweise Downloadzahlen, Anzahl der Erweiterungen, Anzahl Suchtreffer bei Suchmaschinen, Anzahl gemeldeter Installationen, etc..

3 Systemkonzeption

Im folgenden Abschnitt wird die Konzeption des Systems, mit den Schwerpunkten Organisationsstruktur, das Konzept des derzeitigen Prototypen und die Datenintegration mittels ETL-Prozess.

3.1 Organisationsstruktur

Der Systemaufbau in Abbildung 1 zeigt die Plattform als vernetzte Struktur mit eingebetteten technischen Elementen. In dieser Organisationsstruktur werden die Benutzerbereiche nach Funktionen gegliedert. Das allgemeine Grundkonzept sieht eine dezentrale Struktur vor, welches die Einbindung mehrerer Verarbeitungsstellen ermöglicht. So werden mehrere Grundkomponenten integriert, die auf einem oder mehreren Servern installiert sein können und miteinander kommunizieren bzw. Daten austauschen. Von Nutzerseite aus soll der Zugriff einzig und allein über den Browser möglich sein. In einem ersten Schritt wird eine zentrale, stets erweiterbare Informationsplattform aufgebaut, damit die verschiedenen Nutzer von verschiedenen dezentralen Orten aus, Informationen abrufen und aktiv eingeben können. Der Einsatz des CMS wird hier als Mittel gesehen, die Geodatendarstellung und -verwaltung schneller, flexibler und bürgerfreundlicher zu gestalten. Das CMS ist für die zentrale Freigabe der Beiträge und Qualitätskontrolle zuständig. Die offene Organisationsstruktur erlaubt ein individuelles Arbeiten und zum anderen die mögliche Integration von weiteren Geowerkzeugen und -server in die IT-Infrastruktur.

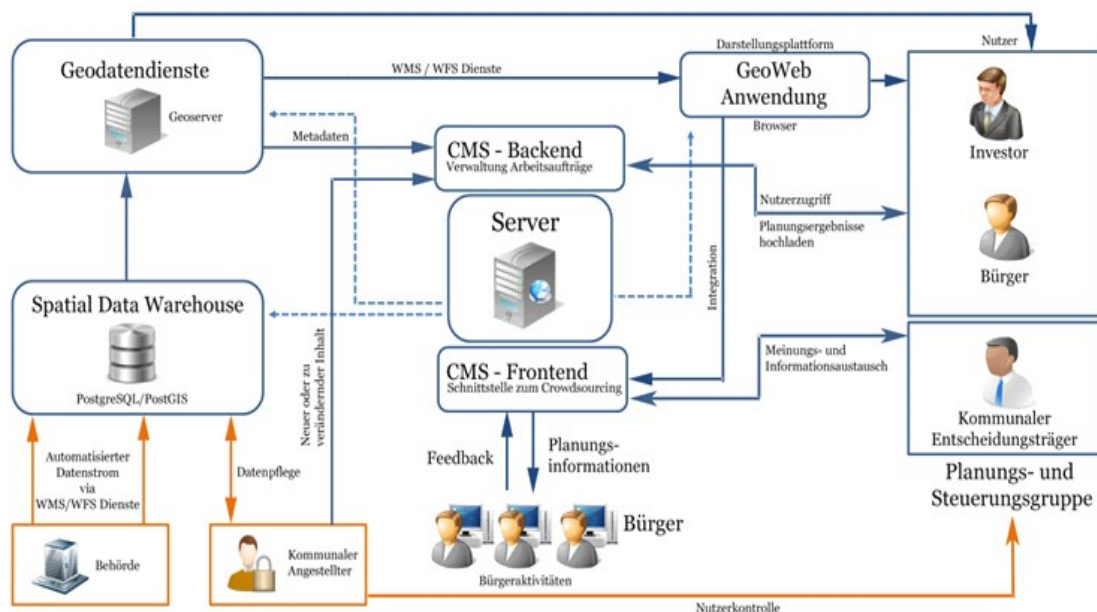


Abbildung 1: Darstellung der Organisationsstruktur des SBIS (Eigene Darstellung)

Im Folgenden werden die Bedienungsabläufe für die Organisationsstruktur näher erläutert. Es gliedert sich in drei größere Organisationsbereiche. Zum einen die Planungs- und Steuerungsgruppe, welche aus Investoren, engagierten Bürgern und kommunalen Entscheidungsträgern besteht, welche je nach Nutzerrechte zu den unterschiedlichen Bereichen des Systems des WebGIS Zugriff besitzen. Die behördliche Seite wird in der zweiten Gruppe dargestellt, die zum Einen zukünftig über Web-Service Daten automatisiert in das System zur Verfügung stellen sollen, sowie ein kommunaler Angestellter, der als Administrator die Datenpflege betreibt, wobei auch Personas aus einer anderen Gruppe diese Aufgabe erfüllen könnten und ein größerer Pool von engagierten Personen aus verschiedensten Fachbereichen zur Transparenz, Aktualität und Akzeptanz beitragen würden. Bürgeraktivitäten können direkt

Konzeption eines Spatial Business Intelligence System zur Analyse und Steuerung von Re-Produktionsketten auf Grundlage von OpenSource Komponenten

über das CMS gesteuert werden, wobei das CMS auch als zentraler Informationsmittelpunkt fungiert. Grundlegende technische Bestandteile sind das Spatial Data Warehouse als Verwaltungsbereich des gesamten Datenbestandes und die Bereitstellung von Geodatendiensten über einen Mapserver. Nähere technische Umstände werden in den folgenden Abschnitten erläutert.

3.2 Konzept des aktuellen Prototyps

Die Entwicklung des Prototypen erfolgte auf Grundlage der in diesem Dokument vorgestellten Methodik. Die Abbildung 2 stellt die Grundaufgabenbereiche mit den eingesetzten OpenSource Komponenten dar, die im folgenden näher erläutert werden.

Beim aktuellen Prototypen wurde als DBMS PostgreSQL 9 in Verbindung PostGIS eingesetzt. Dieses übernimmt die Datenhaltung und im späteren Verlauf auch einige Verarbeitungsfunktionen. Zur Datenintegration wurde das ETL Tool Talend Open Studio mit dem Sextante Plugin verwendet. Dabei werden automatisch Metadaten abgefragt und an Geonetwork gesendet und in einer Metadatenbank gespeichert. Da die Metadaten auch zur Datenverwaltung, also auch zur Zugriffsrechteverwaltung genutzt werden sollen, reicht die Speicherung der Metadaten in Geonetwork allein nicht aus (siehe Abbildung 3).

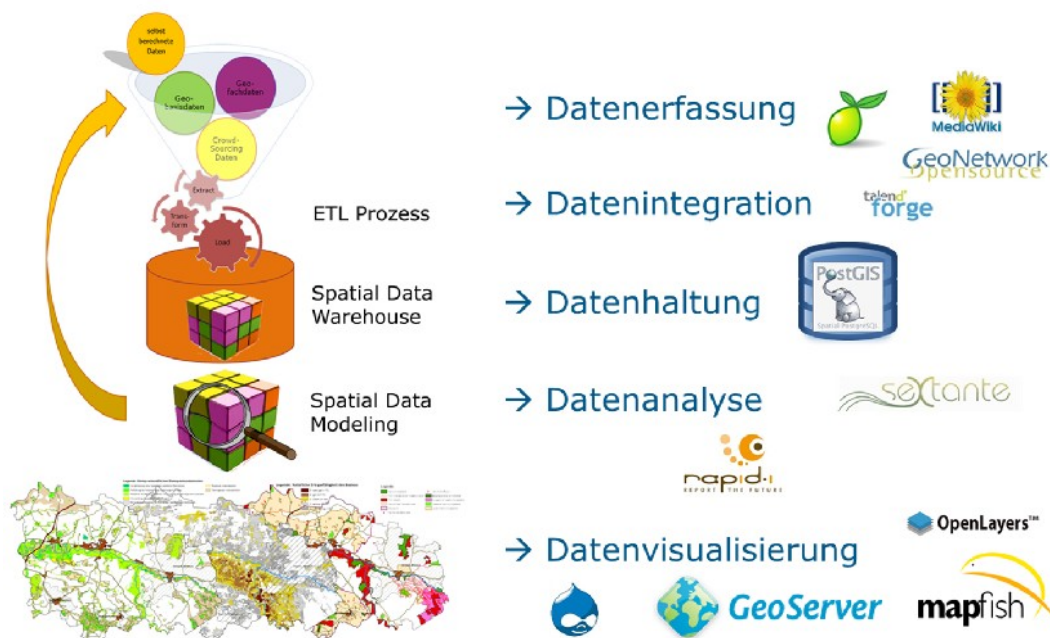


Abbildung 2: Darstellung der Systemgrundaufgaben und den eingesetzten Komponenten (Eigene Darstellung)

Detailliertere Informationen zum ETL-Prozess werden im Abschnitt 3.3 ETL ausgeführt. Geoserver fungiert als Mapserver zur Übertragung der Geodaten an das WebGIS, womit die Einbindung der Daten auch in externen Geoinformationssystem möglich sein wird. Das WebGIS wurde mit der OpenLayers und der GeoExt Bibliothek als Modul für das CMS Drupal konfiguriert.

Da die Zusammenarbeit und Kommunikation zwischen den Beteiligten sehr wichtig ist, wurde Drupal als zentrale Schnittstelle ausgewählt, das es unter anderem sehr gute Communityfeatures⁸ bietet. Es wird hier neben der Vernetzung von Akteuren auch für die Projektarbeit der Beteiligten, der Öffentlichkeitsarbeit sowie als allgemeines Werkzeug zur Planung und Steuerung der Re-Produktionsketten genutzt. Weiterhin sind im aktuellen Prototyp MediaWiki und Limesurvey integriert. MediaWiki soll als

8 Siehe <http://drupal.org/features>

Konzeption eines Spatial Business Intelligence System zur Analyse und Steuerung von Re-Produktionsketten auf Grundlage von OpenSource Komponenten

Werkzeug zum Wissensmanagement für Wissen dienen, welches keinen direkten regionalen Bezug sondern sich nur über die Zeit verändert. Es soll außerdem genutzt werden um durch CrowdSourcing möglichst viel relevantes Wissen für die Beteiligten zusammenzutragen. Es übernimmt somit auch Aufgaben der Öffentlichkeitsarbeit und verbessert die Transparenz. Limesurvey bietet die Möglichkeiten Befragungen an Personen(-gruppen) zu richten, um diese für die Projektarbeit automatisch auszuwerten und wiederum mit räumlichen Analyseergebnissen zu verknüpfen. Zusätzlich wird aktuell die Integration von *RapidMiner* und *Rapid Analytics* als Werkzeug für Datenanalysen und *Mapfish* als WebGIS getestet.

3.3 Datenintegration mittels ETL – Extract Transform Load

Der folgende Abschnitt verdeutlicht die Datenintegration in das SBIS mittels dem ETL Prozess. Amtliche Geofachdaten wurden ebenso wie Daten aus dem OpenStreetMap Datenbestand und mittels Crowd Sourcing gewonnene Daten in einen homogenen Datenbestand extrahiert. Abbildung 3 zeigt die Datenintegration im aktuellen Prototyp. Dabei werden mithilfe von Talend Open Studio Daten aus verschiedenen Quellen in die PostgreSQL-Datenbank geladen. Beim Einladen werden Metadaten abgefragt und in Geonetwork und eine Metadaten-Datenbank integriert. Über die Daten aus der Metadaten-Datenbank können dann die Zugriffsberechtigungen in GeoServer und Drupal festgelegt werden.

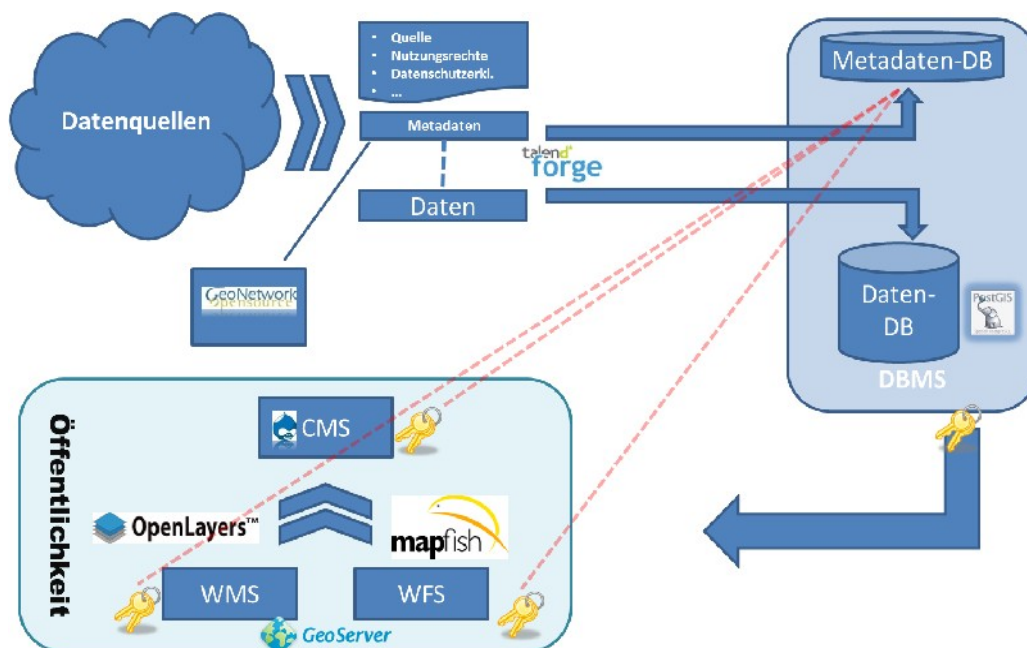


Abbildung 3: Darstellung des Datenintegrationsmanagements (Eigene Darstellung)

Mittelfristig soll die Integration der amtlichen Geofachdaten automatisch über Webservices erfolgen und via WMS oder WFS direkt von öffentlicher Seite aus integriert werden. Im aktuell erstellten Integrationsprozess wurden diese Daten aus Shapefiles gelesen und die Metadaten vom Benutzer abgefragt.

5 Fazit und Ausblick

Die Bedeutung von Geoinformationen hat durch die Verbreitung von sowohl kommerziellen, als auch freien Diensten und der deren Zugänglichkeit in den vergangenen Jahren einen ganz neuen Stellen-

Konzeption eines Spatial Business Intelligence System zur Analyse und Steuerung von Re-Produktionsketten auf Grundlage von OpenSource Komponenten

wert erlangt. Die hier vorgestellte Konzeption eines räumlichen Entscheidungsunterstützungssystems, welches im Umfeld des regionalen energetischen Wertschöpfung eingesetzt werden soll, zeigt das Potenzial, mittels räumlichen Visualisierungen von Sachverhalten, Informationen für jedermann nutzbar und abrufbar zu gestalten. Als Zwischenfazit aus der bisherigen Entwicklungszeit steht fest, dass die grundlegenden Bausteine des Systems lauffähig und einsetzbar sind, welche durch projektinterne Analysen bereits aufgezeigt wurden. Die Aufgaben bis zum Projektende im November 2012 fokussieren sich daher in der Verbesserung mit der Interaktion des Systems, da es nicht nur als analytisches sondern auch als gesellschaftliches Instrument zum Einsatz kommen soll. Auch die Integration von Medien des Web 2.0, beispielsweise eines Wikis mit technischen, finanziellen und ökologischen Hintergründen, sind wichtiger Bestandteil des Systems und kombinieren die Geoinformationstechnologie mit dem Informationsmanagement auf eine neue Ebene und benötigen noch weitere Arbeit in den nächsten Monaten. Weitere Informationen über den Entwicklungsstand, sowie erste öffentliche Einblicke in das System werden ab Mitte des Jahres über die Internetseite <http://www.reproketten.de> einsehbar sein.

Kontakt zu den Autoren:

Martin Becker
Christian Wolff
Hochschule Anhalt - Fachbereich AFG
Bauhausstraße 8
(0340) 5197-1558 bzw. 1543
M.Becker@afg.hs-anhalt.de, C.Wolff@afg.hs-anhalt.de

Literatur

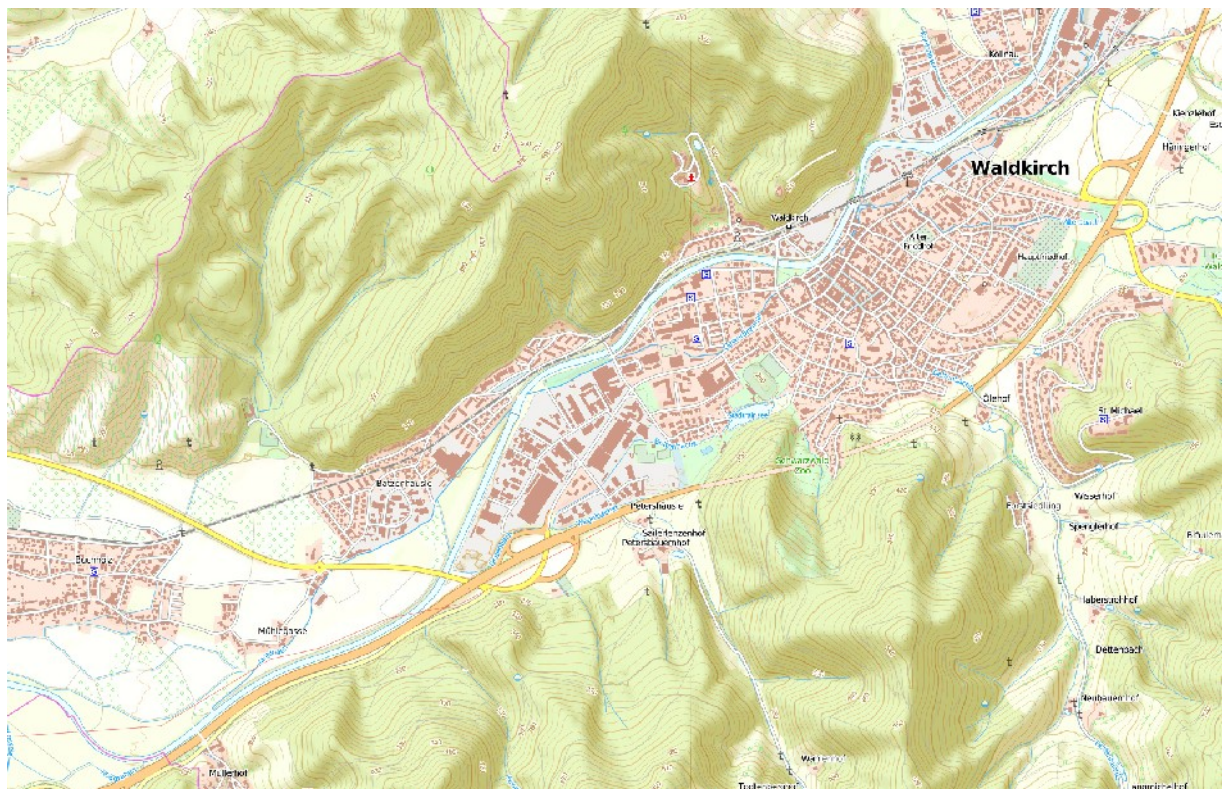
- [1] Cruz, David; Wieland, Thomas; Ziegler Alexander: Evaluation Criteria for Free/Open Source Software Products Based on Project Analysis, SOFTWARE PROCESS IMPROVEMENT AND PRACTICE 2006;11: 107–122
- [2] Eder, T. Energiestrategien im Fokus. In: *GIS.TRENDS+MARKETS* (6) 2011, S. 40 bis 44.
- [3] Kutzner, T.; Brück, A.; Fischl, G.; Kronen, M.; Machl, T.; Obermeier, C. et al. (2011): Die INTERGEO 2011 – Auf der Suche nach dem GIS-Trend der Zukunft. Hg. v. Runder Tisch GIS e.V. München. Online verfügbar unter http://www.geobranchen.de/images/produkte/trendanalyse_intergeo_2011_rtg.pdf, zuletzt geprüft am 28.02.2012.

Ableitung von topographischen Webkarten aus amtlichen Geobasisdaten

Peter Kunz, Bundesamt für Kartographie und Geodäsie

Für die Bereitstellung aktueller Geoinformationen wurde beim Bundesamt für Kartographie und Geodäsie [1] ein Verfahren entwickelt, mit welchem aus verschiedenen amtlichen Geobasisdaten topographische Webkarten ähnlich denen von Google Maps abgeleitet werden können. Diese Karten, die den amtlichen topographischen Karten der Bundesrepublik Deutschland ähnlich sind, können in verschiedenen Zoomstufen und Darstellungsvarianten präsentiert werden. Das Ziel war der Aufbau eines Kachelarchivs, welches als Datenquelle für einen performanten, internetbasierten Kartendienst bereitgestellt werden kann. Solche Kartendienste greifen auf vorprozessierte Rasterkacheln zurück und werden in Form von sog. WMTS-Diensten (Web Map Tile Service) angeboten.

Neben den Daten des Digitalen Basis-Landschaftsmodells Deutschlands (Basis-DLM) wurde auf weitere Geobasisdaten zurückgegriffen. Für kleinere Zoomstufen wurde das Digitale Landschaftsmodell 1 : 250 000 (DLM250) eingesetzt. Die Darstellung der Gebäude in größeren Zoomstufen erfolgte mithilfe der Daten der amtlichen Hausumringe. In der höchsten Zoomstufe wurden auch Hausnummern der Gebäude mithilfe des Datensatzes „Georeferenzierte Adressdaten Bund“ platziert, welcher u.a. auf den amtlichen Hauskoordinaten der Länder beruht. Höhenlinien und eine Schummerung wurden aus dem Digitalen Geländemodell mit einer Gitterweite von 10m (DGM10) gerechnet. Auskunft über die Daten erteilt das Geodatenzentrum (GDZ) des Bundesamtes für Kartographie und Geodäsie [2] und die AdV [3].



Darstellung des Reliefs mit Höhenlinien und Schummerung

Das Verfahren erlaubt zur Zeit die Ableitung von Karten in 12 verschiedenen Maßstäben bzw. Zoomstufen, von der deutschlandweiten Übersichtskarte bis hin zur detaillierten Stadtkarte. Die höchste Zoomstufe zeigt hierbei eine Auflösung von 0,4m pro Pixel. Die Kartengrafik selbst kann in vier verschiedenen Darstellungsvarianten erzeugt werden. Neben der sog. Volltonvariante können auch Kar-

Ableitung von topographischen Webkarten aus amtlichen Geobasisdaten

tenbilder als Halbtonvariante (Pastellfarben) oder Graustufenvariante angeboten werden. Zusätzlich kann eine Volltonvariante einschließlich einer schattenplastischen Geländedarstellung (Schumme- rung) erzeugt werden.

Verwendete Software

Aufgrund der Vorgaben wurde ein Verfahren benötigt, welches die sehr große und rechenintensive Datenmenge in einem vertretbaren Zeitraum erzeugen konnte. Im Rahmen des OpenStreetMap-Projektes [4] wird die Software Mapnik [5] erfolgreich für das Rendern von Karten aus diesen Daten eingesetzt. Insbesondere werden die Kartenkacheln der Hauptdarstellung der OpenStreetMap-Karte, die sog. Slippy Map, mithilfe von Mapnik erstellt. Besonders überzeugend ist dabei die sehr gute grafische Qualität der entstehenden Kartenbilder. Diese gut lesbare Kartengrafik ist insbesondere auf die Anwendung eines Antialiasing beim Rendern zurückzuführen. Mapnik verwendet hierzu die Bibliotheken des AGG-Projektes (Anti-Grain Geometry) [6].

Die guten Ergebnisse im Falle der OSM-Karten waren Anlass, diese Software auch für das hier vorgestellte Projekt einzusetzen. Bei Mapnik handelt es sich um eine freie Software, welche eine saubere Trennung der Programmierlogik von der Datenhaltung und der Kartengestaltung erlaubt.

Bei dem Verfahren wurde neben Mapnik noch weitere freie Software eingesetzt:

- Die Datenhaltung erfolgte in einer PostGIS-Datenbank [7]
- Die PostGIS-Daten wurden mit Quantum GIS visualisiert und überprüft [8]
- Als PostgreSQL Data Provider für .NET wurde NPGSQL eingesetzt [9]
- Mithilfe der Skriptsprache Python wurde Mapnik aufgerufen [10]
- Mit der Programmbibliothek GDAL (Geospatial Data Abstraction Library) wurden aus dem DGM10 Höhenlinien und Schummerungen abgeleitet [11]
- Die Bibliothek OGR diente zum Clippen der Höhenlinien und zum Export der Höhenlinien in die PostGIS-Datenbank [11]
- Die Bibliothek AForge.Net wurde für die Nachbearbeitung der Rasterdaten der Schummerung eingesetzt [12].

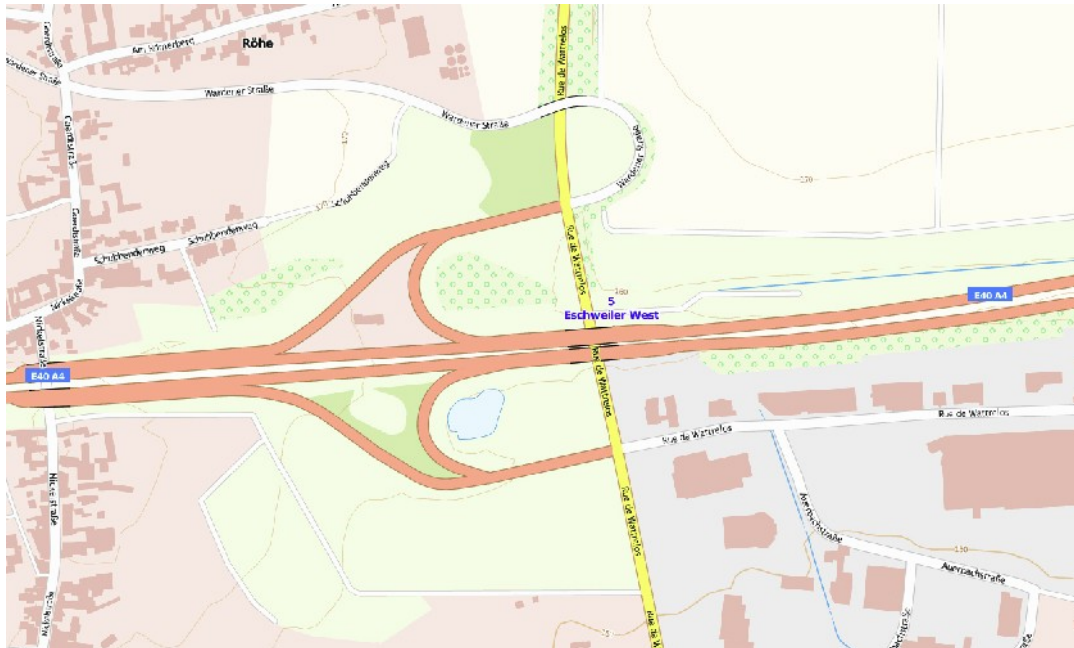
Datenaufbereitung und Import der GIS-Daten in eine PostGIS-Datenbank

Alle vektoriellen Geobasisdaten wurden zunächst in PostGIS-Datenbanken übertragen. Für die Aufbereitung der GIS-Daten und für den Import in die PostGIS-Datenbanken wurde die Software Safe FME [13] eingesetzt. Die Original-GIS-Daten wurden hierbei verschiedenen Bearbeitungsschritten unterzogen, um insbesondere das Rendern der gewünschten Kartengrafik zu ermöglichen und um die spätere Schriftplatzierung durch Mapnik zu verbessern. Beispiele:

- Umbenennung und Vereinheitlichung von Attributbezeichnungen und Zuweisung von Attributwerten entsprechend den Angaben in den Mapnik-Styledateien
- Prüfen der Daten auf invalide Polygone und Korrektur vor dem Import nach PostGIS
- Klassifizieren des Gewässernetzes anhand der Gewässerkennziffer und Gewässerbreiten
- Ermitteln der Lage von Objekten auf Brücken oder Tunnels. Die Darstellungspriorität dieser Objekte wurde aufgrund eines Attributes und einer Zusatztable berechnete, mit welcher im Basis-DLM Unterführungsbeziehungen gekennzeichnet werden. Das „unter“ einem Objekt liegende Objekt kann demnach eine Brücke oder ein Tunnel sein. Hat ein Objekt keine Unterführungsbeziehung, wird es höhengleich mit anderen Objekten dargestellt.
- Im Falle von Straßenkreuzungen oder -einemündungen hat normalerweise die Straßenfüllung einer Autobahn eine höhere Darstellungspriorität als die Straßenfüllung einer untergeordneten Straße. Mit FME wurden die Stellen bestimmt, an denen eine Straße höherer Priorität in eine Straße niederer Priorität einmündet (z.B. Autobahnabfahrt mündet in Landesstraße). Für diese Einmündungen wurde die Darstellungspriorität so geändert, dass die niederrangige Straße die

Ableitung von topographischen Webkarten aus amtlichen Geobasisdaten

höhere Darstellungspriorität hat. Auf diese Weise entstehen später in Mapnik gute grafische Lösungen für diese Einmündungen.



Anpassung der Darstellungspriorität von Straßenfüllungen:

Im oben gezeigten Beispiel hat die Landesstraße (gelb) Vorrang vor der Autobahn (rot)

Vorbereiten der Daten für eine optimierte Schriftplatzierung:

- Dissolveren oder Aggregieren von Flächen und Linien (teilweise auch nur in der Nachbarschaft)
- Berechnung von Flächen und Längen und Einteilung der Objekte in Klassen (z.B. Darstellung von Seen/Wäldern usw. nur ab einer best. Größe, Beschriftung entsprechend der Klasseneinteilung nach EWZ oder Flächengröße)
- Feststellen der Zeichenanzahl von Straßennummern. Je nach Zeichenanzahl wird auf unterschiedlich große Symboldateien (Kästchen für die Straßennummern) zurückgegriffen
- Feststellen der Länge von Straßen und der Zeichenanzahl von Straßennamen (Unterschiedliche Schriftgrößen je nach Straßenlänge bzw. Zeichenanzahl)
- Ermitteln der erforderlichen Textbestandteile aus einem Attribut für die Beschriftung von Verkehrsknoten (Anschlussstellennummer)
- Konkatenieren von Objektbeschreibung und Name (z.B. „Raststätte XYZ“ oder „Segelfluggelände XYZ“)
- Versetzen der Beschriftung von Siedlungspolygonen in das rechte obere Viertel der Bounding Box einer Siedlungsfläche.

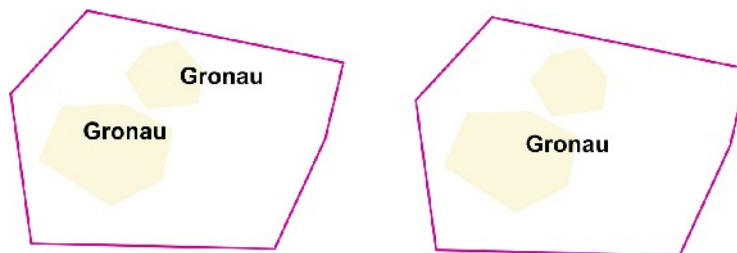
Ableitung von topographischen Webkarten aus amtlichen Geobasisdaten



- Ermitteln von möglichen Konfliktsituationen und Versetzen von Beschriftungspunkten in entgegengesetzter Richtung, falls Schriften sich überlagern würden



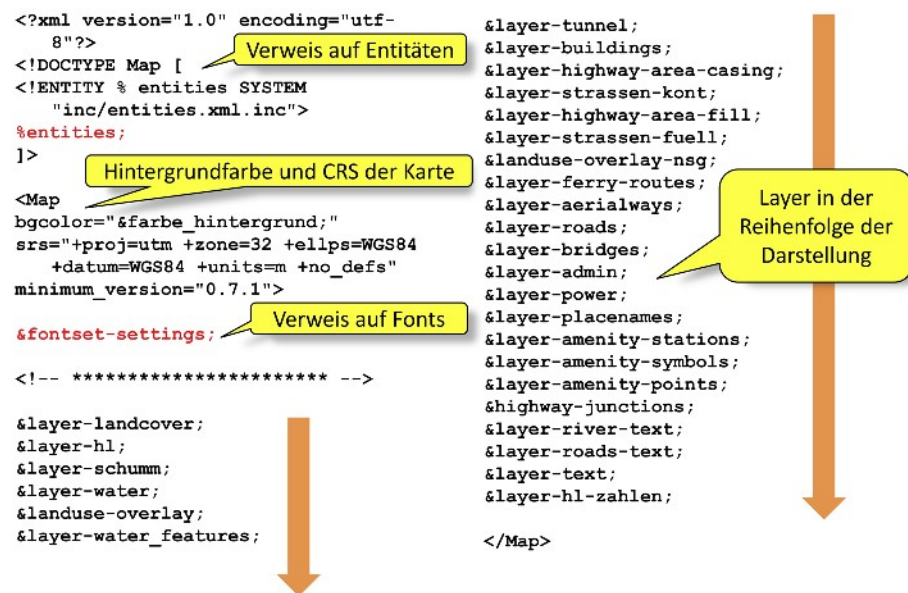
- Verhindern der doppelten Beschriftung von Wohnplatznamen und Beschriften der Ortslagen an der größten Fläche



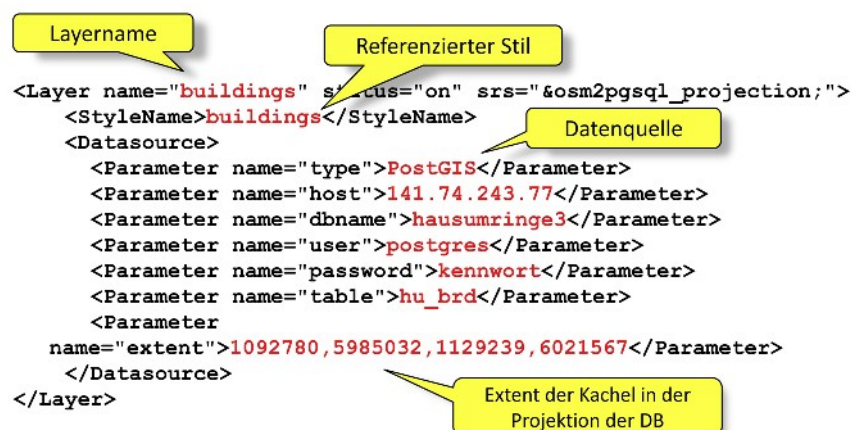
Sofern entsprechende PostGIS-Kenntnisse vorhanden sind, können viele dieser Arbeitsschritte aber auch in PostGIS selbst durchgeführt werden oder als Bestandteil einer SQL-Abfrage in einer Layerdefinition in Mapnik enthalten sein. Allerdings erleichtert FME hier den Umgang mit den Daten enorm.

Mapnik – eine Werkzeugsammlung für das Rendern von Karten

Die Software Mapnik steht für die Plattformen Windows, Mac und Linux zur Verfügung. Die Software kann für das Rendern auf dem Desktop oder für ein serverseitiges Rendern auf dem Webserver eingesetzt werden. Der Zugriff auf Mapnik erfolgt über Programmierschnittstellen für Python und C++. Damit können Karten auch rein programmatisch erzeugt werden. Eine weitere Möglichkeit ist aber die Beschreibung einer Karte in Form eines Stylesheets oder Mapfiles. Diese XML-Datei beschreibt neben dem Koordinatenreferenzsystem der Karte den Kartenaufbau und insbesondere die Datenquellen für die Kartenobjekte und die Art und Weise, wie diese darzustellen sind. So besteht eine Karte aus mehreren Layern. Jeder Layer verweist auf eine bestimmte Datenquelle und kann im Falle von Datenbanken auch SQL-Abfragen enthalten, um die Objektauswahl einzuschränken.



Stylesheet oder Mapfile mit der Beschreibung des Kartenaufbaus und der darzustellenden Layer



Ein Layer beschreibt was dargestellt wird und woher die Daten kommen

Jeder Layer ist wiederum verknüpft mit einer Beschreibung, wie die Objekte darzustellen sind. Dieser sog. Style enthält eine oder mehrere Regeln (Rules) in denen beschrieben wird, in welchen Zoom-

Ableitung von topographischen Webkarten aus amtlichen Geobasisdaten

stufen ein Objekt wie grafisch dargestellt wird. Mit sog. Filtern kann auch hier die Objektauswahl eingeschränkt werden. Jede Regel beschreibt, mit welchen der sog. Symbolizern ein Objekt wie gezeichnet werden soll. So stehen Symbolizer für die Darstellung von Punkten, Polygone, Linien, Texten, Symbolen, Rasterdaten u.a. zur Verfügung. Hier wird dann definiert, welche Farben, Strichstärken, Symbolgrafiken, Schriftarten und -größen usw. für das Rendern eines Kartenelements verwendet werden sollen. Mit sog. Scaledenominatoren wird festgelegt, in welchem Maßstabsbereich ein Objekt gezeichnet werden soll oder nicht.

```
<Style name="landuse_overlay">

  <Rule>
    &maxscale_zoom12;
    &minscale_zoom12;
    <PolygonSymbolizer>
      <CssParameter name="fill">&farbe_haus12;</CssParameter>
      <CssParameter name="gamma">0.7</CssParameter>
    </PolygonSymbolizer>
    <LineSymbolizer>
      <CssParameter name="stroke">&farbe_haus12;</CssParameter>
      <CssParameter name="stroke-width">0.2</CssParameter>
    </LineSymbolizer>
  </Rule>

</Style>
```

Darstellung nur in Zoomstufe 12

Polygonfüllung

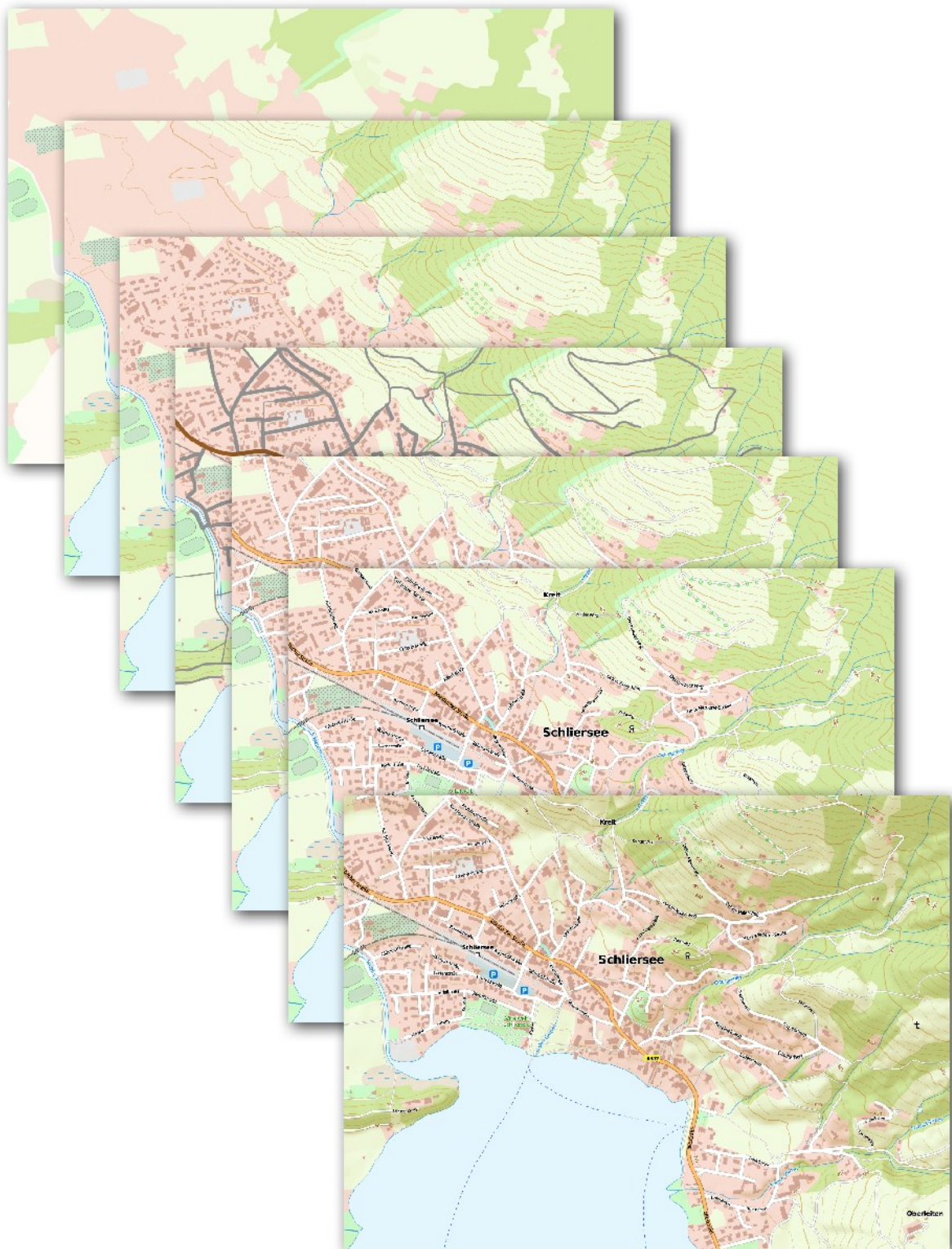
Kontur des Polygons

Ein Style mit einer oder mehreren Regel beschreibt wie etwas dargestellt wird

Die Reihenfolge, in der die Layer gezeichnet werden, und die Reihenfolge der Objekte in einem Layer bestimmen, wie der grafische Kartenaufbau erfolgt. So kann bei der Verwendung von PostGIS-Datenbanken in der Layerdefinition neben der Objektauswahl auch die Sortierung der Daten mit einer SQL-Abfrage festgelegt werden.

Im Falle der Textplatzierung kann Mapnik auf bereits platzierte Schrift Rücksicht nehmen und Schriftüberlagerungen verhindern. Dabei werden die Objekte in der Reihenfolge beschriftet, die durch eine SQL-Abfrage vorgegeben wurde. In der SQL-Abfrage können hier auch weitere Hilfsattribute vor Beginn des Renderprozesses berechnet werden. Die Objekte können dann nach Flächengröße, Länge oder Zahl der Zeichen in einem Feld sortiert werden. Wenn z.B. in einer Layerdefinition mithilfe einer SQL-Abfrage die Straßenlänge berechnet wird und die Straßen absteigend nach der Straßenlänge sortiert werden, erfolgt anschließend zuerst die Beschriftung der langen Straßen, erst danach die der kürzeren Straßen, sofern für deren Beschriftung noch Raum zur Verfügung steht.

Da eine solches Mapfile sehr umfangreich sein kann, kann mithilfe sog. Entitäten (entities) auf Inhalte in externen Dateien verwiesen werden. Dabei wird im Mapfile vor allem die Reihenfolge der darzustellenden Layer aufgeführt, deren Beschreibung und die zugehörigen Styledefinitionen erfolgen aber jeweils in weiteren Dateien. So können auch die Beschreibungen bzw. Definitionen der Hauptdatenquelle (datasource-settings.xml.inc), der Grundeinstellungen (settings.xml.inc), der einzubindenden Layer (layers.xml.inc), der verwendeten Schriftarten (fontset-settings.xml.inc) oder der Maßstäbe der Zoomstufen (hier entities.xml.inc) in externe Dateien ausgelagert werden.



Kartenaufbau durch das Überlagern verschiedener Layer (von oben nach unten):
Landbedeckung, Gewässer und Höhenlinien, Gebäude, Straßenkonturen, Straßenfüllungen, Schrift,
Schummerung

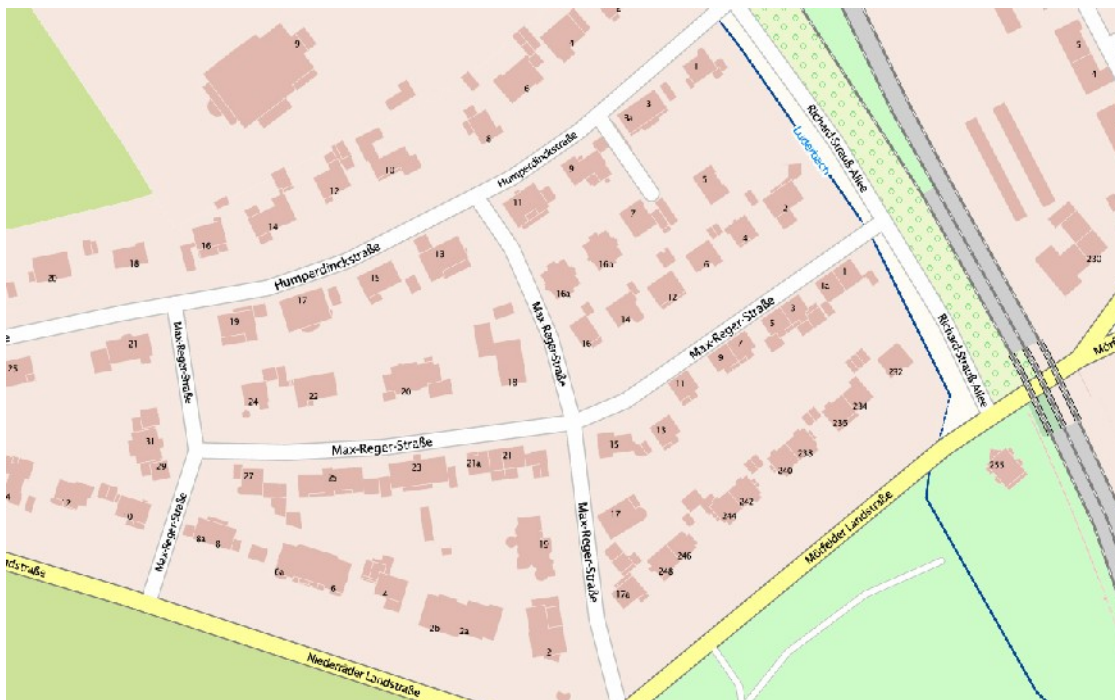
Ableitung von topographischen Webkarten aus amtlichen Geobasisdaten

Bei dem hier vorgestellten Verfahren wurden auch die Farbdefinitionen der verschiedenen Darstellungsvarianten (Vollton, Halbton, Graustufen) in solche externen Dateien ausgelagert. Es genügt daher die Änderung einer Farbdefinition an einer Stelle und alle Symbolizer greifen auf diese Farbdefinition zurück.

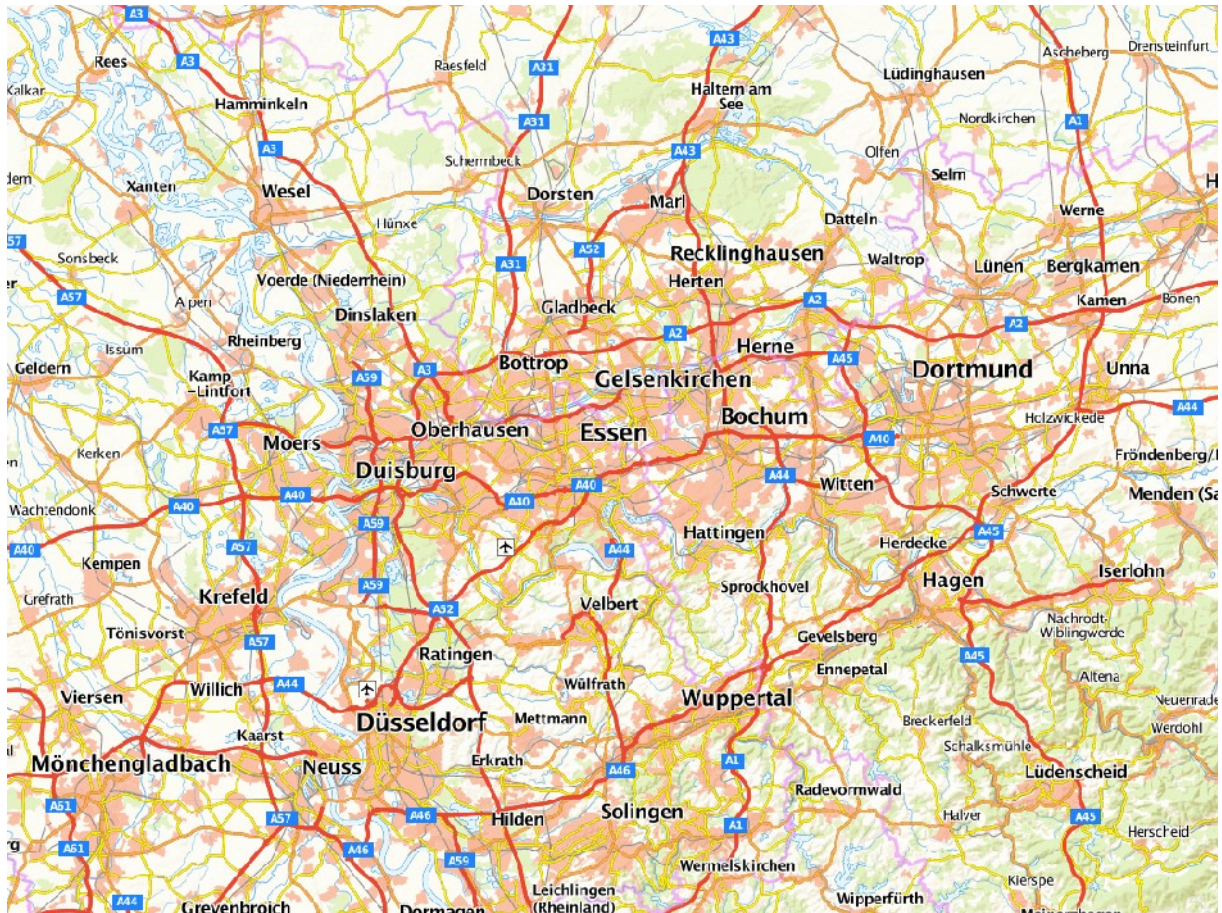
Bei Mapnik handelt es sich um eine Bibliothek, nicht um eine selbstständige Anwendung. Um Karten zu rendern muss diese Bibliothek in eine eigene Anwendung eingebunden werden (C++ oder Python). Beim vorgestellten Verfahren wurde Mapnik mit Python aufgerufen und der Rendervorgang gestartet. In diesem Fall wird in einem kleinen Pythonskript auf die Kartenbeschreibung (das mapfile) verwiesen. Weiterhin wird die Größe des gewünschten Bildes in Pixel angegeben, das gewünschte Dateiformat der Bilddatei (z.B. PNG) und die Bounding Box des zu rendernden geographischen Bereichs. Zusammen mit Mapnik werden zwei Pythonskripte für das Rendern zur Verfügung gestellt:

- `generate_image.py` erzeugt ein einzelnes Bild entsprechend der Kartendefinition (mapfile) und des gewünschten räumlichen Ausschnitts (Bounding Box)
- `generate_tiles.py` ist vergleichbar, erzeugt aber ein Kachelarchiv für den Webserver in den gewünschten Zoomstufen und für einen angegebenen räumlichen Ausschnitt

Als Datenquellen kommen für Mapnik sowohl Vektor- als auch Rasterdaten in Frage. Über die eingebundenen Bibliotheken GDAL und OGR kann auf eine Vielzahl von Datenformaten zurückgegriffen werden. Vor allem jedoch kommen für Mapnik PostGIS-Datenbanken und Shapefiles als Datenquelle in Frage.



Darstellung von amtlichen Hausumringen und Hauskoordinaten (höchste Zoomstufe)



Visualisierung des Digitalen Landschaftsmodells 1 : 250 000 in den kleineren Zoomstufen

Rendern der Kartendaten

Bei dem hier vorgestellten Projekt wurden große Kartenbilder mit 2560 x 2560 Pixel in dem Koordinatenreferenzsystem UTM32 berechnet. Man spricht in diesem Zusammenhang vom Rendern von Metakacheln. Diese Bilder wurden anschließend in kleine Kacheln von 256 x 256 Pixel unterteilt und in die Verzeichnisstruktur des WMS-Proxies MapProxy verteilt. Das Rendern großer Bilder hat den Vorteil, dass Konflikte bei der Beschriftung über Kachelgrenzen hinweg minimiert werden.

Für die Kartenproduktion wurde ein eigenes Kachelsystem (ein eigener Blattschnitt) im Koordinatenreferenzsystem UTM32 entwickelt, welches mit FME berechnet wurde. Die Naturstrecke der Kachelbreite wurde so gewählt, dass bei einer Bildbreite von 2560 Pixel eine Auflösung von 0,4m pro Pixel in der höchsten Zoomstufe entstand. Gleichzeitig wurden für jede Kachel die Koordinaten der Bounding Box in UTM-Koordinaten und in Koordinaten der Sphärischen Mercatorabbildung berechnet. Auf diese Weise wurden Blattschnitte für 12 Zoomstufen berechnet, wobei mit jeder nachfolgenden Zoomstufe die Auflösung halbiert wurde. Die Ergebnisse wurden ebenso in einer PostGIS-Datenbank gespeichert.

Mit einem Visual Basic-Programm wurden mithilfe von NPGSQL die Koordinaten für eine bestimmte Kachel aus der PostGIS-DB gelesen und in das Pythonskript zum Starten des Renderns eingesetzt. Auf diese Weise konnte bei der Verfahrensentwicklung jederzeit eine ganz bestimmte Kachel in einer bestimmten Zoomstufe und in einer gewünschten Darstellungsvariante berechnet werden. Da die GIS-Daten der PostGIS-Datenbank in der Sphärischen Mercatorabbildung vorlagen, wurden die Koordinaten dieser Abbildung verwendet, um den gewünschten Ausschnitt (den Extent) jedes Layers in den

Ableitung von topographischen Webkarten aus amtlichen Geobasisdaten

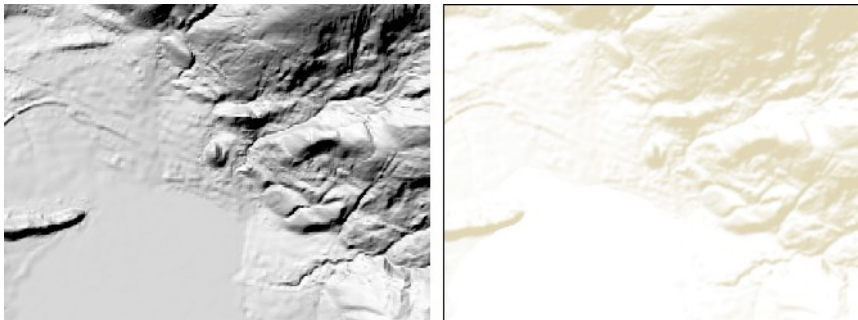
Mapnik-Stylesheets angeben zu können. Die Angabe des Extents bei der Verwendung von PostGIS-Daten beschleunigt deutlich den Zugriff auf diese Daten. Die Koordinatenangaben für diesen Extent einer Kachel wurden programmatisch mit Visual Basic in alle Layerdefinitionen eingefügt. Für das Rendern wurden zudem Extents angegeben, die größer waren als der Kachelausschnitt selbst. Nur so wurde gewährleistet, dass beim Rendern, und hier insbesondere bei der Schriftplatzierung, auch die Daten der Nachbarkacheln mit berücksichtigt werden konnten.

Die Automation der Kartenherstellung erfolgte mithilfe von Tools, die in VB.Net bzw. in C#.Net geschrieben wurden. Das Rendern selbst erfolgte sowohl unter Windows als auch unter Linux mit der Mapnikversion 0.7.1. Das Rendern aller Kartendaten für eine Darstellungsvariante ohne Schummerung dauert ca. 14 Tage für das Gebiet der BRD. Die dabei entstehende Gesamtdatenmenge beträgt ca. 300GB für eine Darstellungsvariante.

Darstellung des Reliefs mit Höhenlinien und Schummerung

In den Karten werden Höhenlinien und in einer Darstellungsvariante auch eine Schummerung für die Darstellung des Reliefs verwendet. Für die Berechnung der Höhenlinien und der Schummerung aus dem DGM10 wurde auf die freie Bibliothek GDAL zurückgegriffen. Das hochaufgelöste DGM10 lag als binäres ESRI-Grid vor und hat ein Datenvolumen von 20GB. Mit GDAL wurde aus dem DGM der gepufferte Bereich einer Kachel geclippt und anschließend die Höhenlinien berechnet. Mit OGR wurden diese Höhenlinien auf den Bereich der Kachel geclippt und in eine PostGIS-Datenbank geschrieben. Diese Höhenlinien konnten dann für das Rendern der Kartenbilder verschiedener Zoomstufen verwendet werden.

Im Falle der Schummerung wurde individuell pro Kachel eine temporäre Schummerung berechnet. Dieses Rohbild wurde anschließend unter Zuhilfenahme des Open-Source-Frameworks AForge.NET so aufgehellt, dass es problemlos mit Mapnik in ein Kartenbild zu integrieren war. Obwohl das DGM mit 10m Gitterweite eine hohe Auflösung zeigt, ist diese Auflösung doch bei weitem zu gering, um in den höchsten Zoomstufen eine artefaktfreie Schummerung berechnen zu können. Daher wurde mit GDAL vor dem Berechnen der Schummerung die Auflösung des DGM-Ausschnitts hochgerechnet. Die daraus erzeugten Schummerungen zeigten ein grafisch ansprechenderes Bild. Im Falle der höchsten Zoomstufe wurde auf die Berechnung einer Schummerung verzichtet.



Links: Originalschummerung, Rechts: Schummerung nach der Bildbearbeitung durch AForge

Ableitung von topographischen Webkarten aus amtlichen Geobasisdaten

```
<Style name="hillshadestyle2">  
  <Rule>  
    <RasterSymbolizer>  
      <CssParameter name="scaling">fast</CssParameter>  
      <CssParameter name="opacity">0.8</CssParameter>  
      <CssParameter name="mode">multiply</CssParameter>  
    </RasterSymbolizer>  
  </Rule>  
</Style>
```

Rastersymbolizer mit Parametern

```
<Layer name="schummerung" status="on">  
  <StyleName>hillshadestyle2</StyleName>  
  <Datasource>  
    <Parameter name="type">raster</Parameter>  
    <Parameter name="file">&rasterdaten;/schummerung.tif</Parameter>  
    <Parameter name="format">tiff</Parameter>  
    <Parameter name="lox">582192</Parameter>  
    <Parameter name="loy">5235730</Parameter>  
    <Parameter name="hix">598576</Parameter>  
    <Parameter name="hiy">5238976</Parameter>  
  </Datasource>  
</Layer>
```

Referenzierter Stil

Datenquelle

Layerdefinition und Style für eine Schummerung

Fazit und Ausblick

Mit dem Verfahren wurde gezeigt, dass bei weitgehender Nutzung freier Software erfolgreich sehr große Datenmengen mit amtlichen Geobasisdaten in Kachelarchive für einen Web Map Tile Service überführt werden können. Besonders überzeugend waren hierbei die gute Verfügbarkeit der PostGIS-Datenbanken und die sehr guten Renderergebnisse von Mapnik. Mapnik erzeugt eine schöne Kartengrafik. Die Karten sind auch bei kleinen Schriftgrößen bzw. Linienbreiten unter einem Pixel gut lesbar. Dies wird durch ein Subpixel-Rendering mit Antialiasing erreicht. Mapnik erzielt ebenfalls sehr gute Ergebnisse bei der Platzierung von Schriften.

Zukünftig wird das Verfahren für die Darstellung weiterer europäischer Geobasisdaten für Gebiete außerhalb der BRD erweitert. Diese Daten stehen für mittlere Maßstäbe in einheitlicher Form zur Verfügung. Für die Darstellung der höchsten Zoomstufen kommen aber auch Daten aus dem OpenStreet-Map-Projekt in Frage.

Kontakt zum Autor:

Peter Kunz
Bundesamt für Kartographie und Geodäsie
Richard-Strauss-Allee 11
60598 Frankfurt am Main
Telefon: +49 (0)69 6333-332
peter.kunz@bkg.bund.de

Ableitung von topographischen Webkarten aus amtlichen Geobasisdaten

Literatur:

- [1] Bundesamt für Kartographie und Geodäsie, Webseite: <http://www.bkg.bund.de>
- [2] GeoDatenZentrum, Webseite: <http://www.geodatenzentrum.de>
- [3] AdV, Webseite: <http://www.adv-online.de>
- [4] OpenStreetMap, Webseite: <http://www.openstreetmap.org>
- [5] Mapnik, Webseite: <http://www.mapnik.org>
- [6] Anti-Grain Geometry, Webseite: <http://www.antigrain.com>
- [7] PostGIS, Webseite: <http://postgis.refrations.net/>
- [8] Quantum GIS, Webseite: <http://www.qgis.org/>
- [9] NPGSQL (PostgreSQL Data Provider für .NET), Webseite: <http://npgsql.projects.postgresql.org/>
- [10] Python, Webseite: <http://python.org>
- [11] GDAL und OGR, Webseite: <http://www.gdal.org/>
- [12] AForge, Webseite: <http://www.aforogenet.com/>
- [13] Safe FME, Webseite: <http://www.safe.com/>

Anwendung im Bereich Bauzonenstatistik in der Schweiz

Christian KARRIÉ, Hans-Jörg STARK

Zusammenfassung

Durch die fortschreitende Zersiedlung der Flächen in der Schweiz und die daraus resultierenden Verlust des Gesamtüberblicks über den Bebauungszustand einzelner Flächen wurde im Jahr 2007 erstmals vom Bundesamt für Statistik (BFS), Schweiz die «Bauzonenstatistik der Schweiz» [1] veröffentlicht. Der Bericht ermöglicht erstmals „einen gesamtschweizerischen Überblick über die Grösse und die Lage der Bauzonen in der Schweiz“, sowie einen Überblick über unüberbaute Bauzonen. Als Ergebnis beschreibt der Bericht weitere Merkmale von Bauzonen, wie beispielsweise deren Verkehrsanbindung, Erschliessung oder Nutzung.

Da die Erfassung der Daten bei den Kantonen liegt, wurde für den Kanton Baselland eine Webanwendung entwickelt, die es – zunächst in einem Testlauf über drei Gemeinden – ermöglicht, Flächen über eine webbasierte Anwendung sowohl hinsichtlich ihrer Geometrie als auch der entsprechenden Merkmale zu verwalten. Durch eine dynamische Attributierung der Flächen wird sichergestellt, dass neben vordefinierten Eigenschaften (Parzelle überbaut, nicht überbaut, erschlossen, nicht erschlossen usw.) auch freie Attribute vergeben werden können. Neben der Beurteilung der Überbauung bzw. der Baureife ermöglicht die Anwendung sogenannte „Innenentwicklungspotenziale“ aus bestehenden Flächen zu generieren.

Einführung

Seit dem 1. Juli 2008 ist das Geoinformationsgesetz (GeoIG) [2] in der Schweiz in Kraft. Der Einfluss des GeoIG auf die Nutzungsplanung bedeutet für das Bundesamt für Raumentwicklung (ARE), dass unter Mitwirkung der Kantone „ein minimales Geodatenmodell sowie ein Darstellungsmodell für die Nutzungspläne erarbeiten muss. Damit wird erstmals ein gesamtschweizerisch verbindliches minimales Geodatenmodell für die Nutzungspläne vorliegen.“ ([1]).

Die vorgestellte Anwendung wurde im Auftrag der Bau- und Umweltschutzdirektion des Kantons Baselland vom Institut Vermessung und Geoinformation der Fachhochschule Nordwestschweiz FHNW erstellt. Sie gibt den Anwendern die Möglichkeit, über eine web-basierte Anwendung sowohl die Geometrie als auch die Sachdaten von Parzellen einzusehen und ggf. anzupassen.

Die Anwendung kann in zwei Modi eingeteilt werden: Modus 1 zur Erfassung der Überbauung und Baureife von Flächen, Modus 2 für die Verwaltung von Innenentwicklungspotenzialen von Flächen. Für die Erhebung der Baureife (Status der Überbauung) wird ein Formular verwendet, das den Stand der Erschliessung, Massnahmen und Bebauung erfragt.

Innenentwicklungspotenziale (Potenziale für die Entwicklung der Flächennutzung nach „innen“, siehe Wikipedia „Innentwicklung“) können aus einer oder mehreren Flächen bestehen. Die in Modus 2 erstellten Potenziale erhalten eine von ihrer Basisfläche unabhängige Attributierung.

Zielgruppe der Erhebung von Statistiken und Potenzialen – also die Nutzer dieser Webanwendung – sind Fachpersonen, die im Bereich Kantonale Raumplanung angeordnet sind.

Gesamtkonzept und Softwarekomponenten der Anwendung Bauzonens-tistik

Nicht nur aus wirtschaftlichen Gründen, sondern auch weil von der Auftraggeberin verlangt, wird für die Anwendung OpenSource Software (OSS) eingesetzt. Im Umfeld der Verwaltung lässt sich generell beobachten, dass OSS zunehmenden Einsatz findet und teilweise geschlossene Software ablöst oder diese ergänzt oder zumindest als Alternative zu geschlossener Software geprüft wird. Des Weiteren lässt sich ein Trend von Gesamtsystemen zu modularen Systemen erkennen. Das Konzept Open-Source erlaubt eben durch die Spezialisierung einer Komponente auf ein Themengebiet einen modularen Aufbau einer vollwertigen Anwendung. Aus diesen Gründen wurde die Entwicklung für die Anwendung Bauzonenstatistik vollständig mit OSS Komponenten durchgeführt. Abbildung 1 skizziert die einzelnen Komponenten client- und serverseitig.

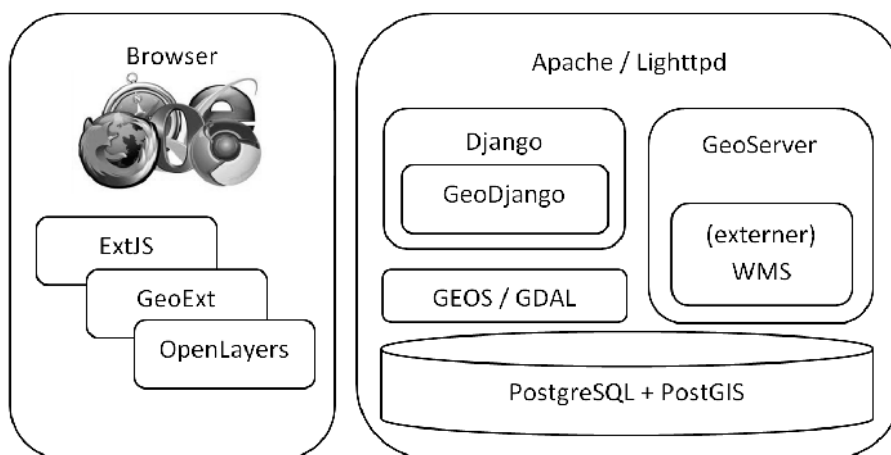


Abb. 1: Client- und Serverseitig eingesetzte Softwarekomponenten

Das Webframework Django bietet dank seinen zahlreichen Komponenten (Benutzerverwaltung, WSGI-Unterstützung, Unterstützung zahlreicher DBMS, Geo-Modul, Templatesystem, Verwaltung von URLs, ORM, Administrationsoberfläche, Sprachunterstützung, u.v.m) eine vollständige Grundlage zur Entwicklung eigener Webanwendungen in Python. Dabei spielt das Geo-Modul (GeoDjango) eine entscheidende Rolle in der Verwaltung der Geometrieobjekte (in diesem Fall vorwiegend Multipolygone für die Verwaltung der Flächen).

Auf Datenbankseite wird PostgreSQL mit PostGIS 2.0 eingesetzt. Dank PostGIS 2.0 wird vor allem die neue Split-Funktion `ST_SPLIT()` verwendet, die das Auftrennen von Flächen in einer Funktion ermöglicht (vgl. Abbildung 2). Die Split-Funktion löst damit umständliche und fehleranfällige Workarounds ab.



Abb. 2: Zeichnen einer Trennlinie über eine bestehende Parzelle

Die Generierung der Karte (Flächenkarte) übernimmt der GeoServer. Dank der Unterstützung von PostgreSQL Views können auf Basis komplexer Abfragen die Parzellen thematisch unterschiedlich eingefärbt werden.

In Bezug auf die Historisierung von Flächenänderungen – wie sie seit Jahren von der amtlichen Vermessung gefordert wird, aber bisher in der Schweiz technisch nicht umgesetzt worden ist – wurde ein Mechanismus entwickelt um „Teilungs-Historien“ darzustellen und gegebenenfalls rückgängig zu machen. Auf eine Historisierung der Attribute wurde aus Gründen der Komplexität verzichtet. Allerdings werden Metadaten zur letzten Änderung gespeichert (Benutzername, Zeitpunkt) (vgl. Abbildung 3).

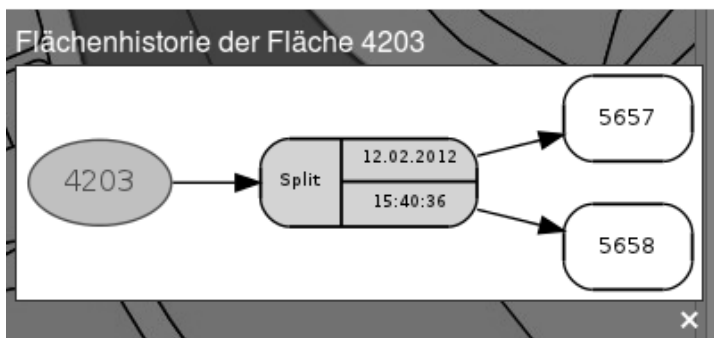
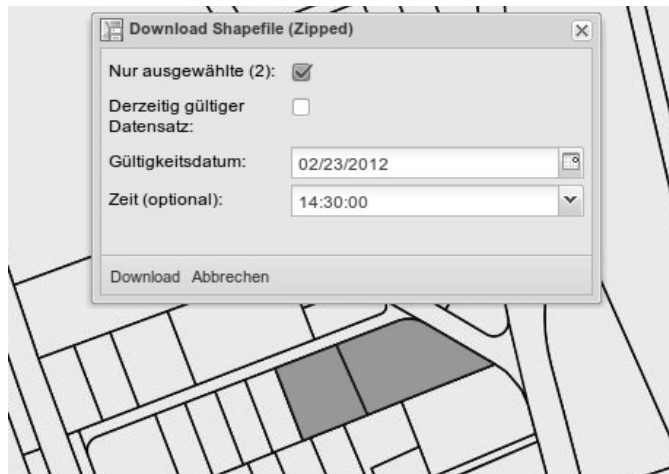


Abb. 3: Flächenhistorie einer gesplitteten Fläche

Eine weitere Funktion ist der Export der Flächen in eine Esri Shape-Datei (vgl. Abbildung 4). Hier kann der Zeitpunkt bestimmt werden, in dem sich die ausgewählten Flächen befunden haben (Beispielweise für einen Vergleich vor/nach mehrmaligem Split).

Anwendung im Bereich Bauzonenstatistik in der Schweiz



1.

Abb. 4: Export von Flächen

Ein weiteres Merkmal der Anwendung ist die dynamische Verwaltung von Attributen. Dabei existieren die Attribute als Schlüssel/Werte-Paare in einer separaten Tabelle und sind nur mittels einer Zwischentabelle mit der Fläche verknüpft. Diese Methode hat den Zweck, basierend auf dem Hauptattribut „Überbaut“ = „ja/nein“ weitere Attribute zuzulassen oder nicht. Des Weiteren besteht die Möglichkeit, den Datentyp eines Attributes festzulegen (Datum, vordefinierte Einzelauswahlliste, vordefinierte Mehrfachauswahlliste oder Freitext).

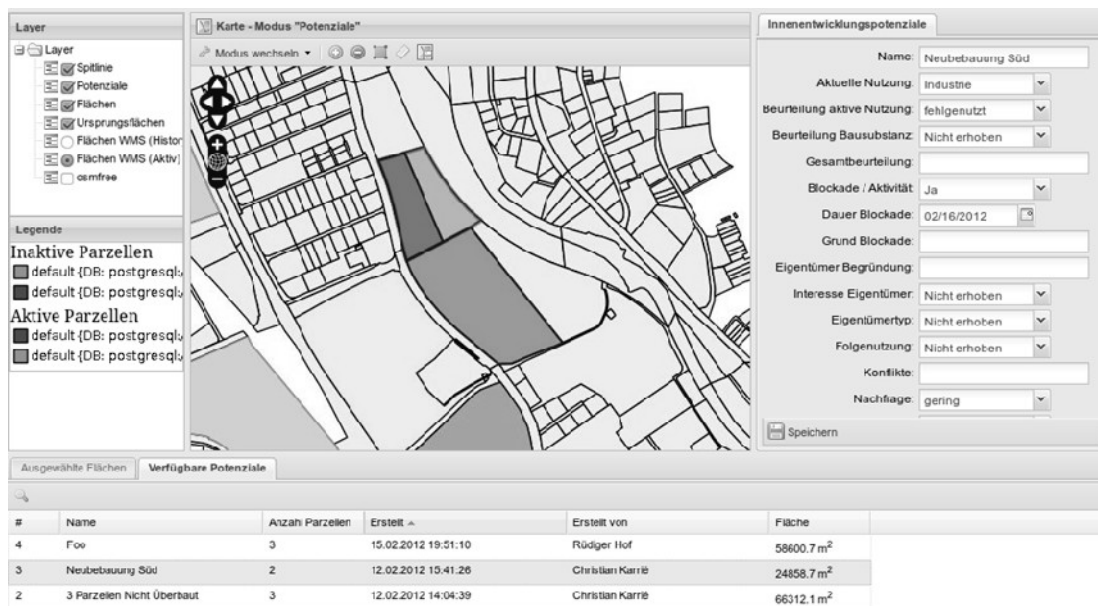


Abb. 5: Ansicht gesamte Webanwendung

Für das Deployment der Anwendung stehen viele Möglichkeiten frei. Für Entwicklungszwecke bietet Django einen integrierten Webserver an. Stabilere Möglichkeiten wären der Einsatz eines Apache Servers mit WSGI-Modul oder eines Lighttpd-Servers.

Zur Verwaltung des Webseitenlayouts wird die JavaScript-Bibliothek ExtJS eingesetzt. ExtJS bietet die Unterstützung von sogenannten Widgets an, die die Funktionalität und das Aussehen von Komponenten einer graphischen Benutzeroberfläche innerhalb eines Webbrowsers simulieren (vgl. Abbildung 5). Neben den Widgets liegt die Stärke von ExtJS in der Verwaltung von Daten und in der umfangreichen Unterstützung von AJAX.

Anwendung im Bereich Bauzonenstatistik in der Schweiz

Zur Darstellung der verschiedenen Attribute wird serverseitig dynamisch ein JSON im ExtJS-Stil erzeugt, das das Eingabeformular abhängig von den Attributen erzeugt.

Fazit

Die vorgestellte Anwendung ist zur Zeit der Verfassung dieses Artikels noch in der internen Einführungsphase, so dass noch keine Aussagen über den Praxiseinsatz gemacht werden können. Jedoch kann bereits zu diesem Zeitpunkt festgestellt werden, dass sich OSS als gute Alternative zu geschlossener Software erwiesen hat. Die Auftraggeberin ist mit den Ergebnissen der Entwicklung sehr zufrieden und zuversichtlich, dass die Nutzer eine gute und praxistaugliche Anwendung zur Verfügung gestellt bekommen werden.

Kontakt zu den Autoren:

Prof. Hans-Jörg Stark
Christian Karrié
Fachhochschule Nordwestschweiz
Hochschule für Architektur, Bau und Geomatik
Institut Vermessung und Geoinformation
Gründenstrasse 40
4132 Muttenz
+41 61 467 4605
hansjoerg.stark@fhnw.ch
christian.karrie@fhnw.ch

Literatur

- [1] Bundesamt für Raumentwicklung ARE (2007): Bauzonenstatistik Schweiz 2007. [<http://bit.ly/wN-Cuon>, Zugriff: 24.2.2012]
- [2] Schweizerische Eidgenossenschaft (2008): Bundesgesetz vom 5. Oktober 2007 über Geoinformation (Geoinformationsgesetz, GeolG). [http://www.admin.ch/ch/d/sr/c510_62.html, Zugriff: 24.2.2012]

DIVE4elements als Business Intelligence Werkzeug

Wasserwirtschaft und Ozeanographie: Maßgeschneiderte Arbeitsprozesse Analysen und Produkte

Sascha L. Teichmann

DIVE4elements ist ein BI-Werkzeug zur Analyse, Modellierung und Visualisierung von Arbeitsprozessen auf großen, (geo-)bezogenen Datenmengen und wird z.Z. in der Ozeanographie und Wasserwirtschaft eingesetzt.

Herzstück von DIVE4elements ist die flexible Konfigurierbarkeit von Workflows, die auf Basis von komplexen Abfragen auf angeschlossenen Datawarehouses (Data Mining) zu Produkten in Form von Diagrammen, Karten (OGC konform WMS/WFS) und Datenexporten führen. Das System wurde auf leichte Verständlichkeit hin entworfen, um gut auf individuelle Anforderungen reagieren zu können.

Durch den modularen, REST-basierten Aufbau des Systems ist eine Nutzung verschiedener Arten von Klienten möglich: Kommandozeile, Web-Interfaces (klassisch/AJAX) und Desktop.

Durch die Mehrschichtigkeit ist serverseitig eine effektive Lastverteilung möglich. Dies ist gerade bei intensivem Geo-Processing von Vorteil. Skalierbarkeit wird zusätzlich durch integriertes Caching gewährleistet. Datenbankabstraktion ermöglicht die Anbindungen verschiedener Datenquellen.

Der Vortrag zeigt exemplarisch zwei Anwendungsfälle der Software im Produktivbetrieb. Die erzeugten Karten werden hier mit Hilfe von UMN MapServer und OpenLayers erstellt.

DIVE4elements ist in Java implementiert und als Freie Software unter der LGPL veröffentlicht.

Weitere Informationen sind und unter <http://dive4elements.org/> zu finden.

NAS - BasisDLM Aufbereitung mit gdal/PostNAS

Tobias Dick / Armin Retterath

Einleitung

Seit der Integration einer lesenden Schnittstelle für die NAS Daten in die gdal Bibliothek (1.8+), kann beinahe jede Software, die gdal verwendet, auch NAS XML Dateien lesen und bei Bedarf darstellen. Die Schnittstelle ermöglicht eine Überführung der kompliziert modellierten NAS Objekte in eine relativ flache Datenstruktur. Die Beziehungen der Objekte untereinander werden dabei in einer eigenen Tabelle abgelegt. Auf Grundlage dieses Konverters, der aus dem PostNAS Projekt (<http://www.post-nas.org>) stammt, wurde schon ein Auskunftssystem für Katasterdaten aufgesetzt (ALKIS Auskunft des Kommunalen Rechenzentrums Minden-Ravensberg/Lippe). Das System basiert auf den FOSS Komponenten Mapserver, Mapbender und PostGIS. Da die Datenstruktur von ATKIS mit der von ALKIS identisch ist, läßt sich der Konverter auch für die Daten des Digitalen Landschaftsmodells (DLM) einsetzen.

In einem **Proof of Concept** wurden die Daten des BasisDLM's (Vektordaten 1:25.000) von Rheinland-Pfalz über die PostNAS Schnittstelle in eine PostGIS-DB überführt und dort mit PostGIS Funktionen in bis zu sechs Stufen generalisiert. Das Rendering übernimmt hier, wie auch bei der o.g. ALKIS Auskunft, der Mapserver. Beim Aufsetzen der Logik wurde besonderes auf ein *performantes Rendering* der Vektordaten geachtet. Der einzelne Renderingprozess braucht in keiner Stufe mehr als 0.8s. Da der Importprozess für das Land Rheinland-Pfalz mehr als 5h benötigt, werden die Daten über das s.g **NBA** Verfahren (**N**utzerbezogene **B**estandsdaten**a**ktualisierung) fortgeführt. Hier werden täglich nur die Änderungen übertragen. Durch diese Vorgehensweise erhält man einen WMS mit einer tagesaktuellen Präsentation ausgewählter Objektarten des BasisDLM. Das Verfahren eignet sich sehr gut für Nutzer, die über wenig bzw. keine finanziellen Mittel verfügen um sich teure, proprietäre AAA-Konverter anzuschaffen.

Bei der Generalisierung wurde darauf geachtet, dass die Geometrien valide sind, es also nicht zu topologischen Fehlern wie Überschneidungen innerhalb von Polygonen kommt. Dies ist wegen des sequentiellen Ablaufs der Generalisierung – Abstraktionsebene n leitet sich von $n-1$ ab – sehr wichtig, da es sonst zu Folgefehlern kommt.

Die Prozessierungskette besteht im Wesentlichen aus der Vereinfachung der Objektgeometrien, dem Ausschließen zu kleiner Objekte und ggf. dem Korrigieren der Objektgeometrie. Dem schließt sich eine Verschneidung mit der Landesfläche an, um überstehende Objekte grenzscharf zu trimmen und überdies die Ladezeit für große Polygone zu verbessern.

Umsetzung von ATKIS Daten mittels PostNAS

Der PostNAS Konverter funktioniert so, dass er aus einer gegebenen NAS XML Datei die Objekte automatisch ermittelt und hieraus ein PostGIS Tabellenschema erstellt. Der Konverter arbeitet die einzelnen XML Dateien dabei

[illegible]

Abbildung 1: Ausschnitt NBA Erstausrüstung XML Datei

ter arbeitet die einzelnen XML Dateien dabei nacheinander ab. Für das Land Rheinland-Pfalz ent-

NAS - BasisDLM Aufbereitung mit gdal/PostNAS

spricht die Standardabgabegröße einer NAS XML Datei einem Gebiet von 2x2 km² im Blattschnitt der TK25. Für den Erstimport mussten also mehr als 230 XML Dateien mit einem Datenvolumen von 12GB (entpackt – gepackt ca. 500MB) prozessiert werden. Ein Ausschnitt aus einer NBA Erstaussat-tung Datei ist in Abbildung Fehler: Referenz nicht gefunden zu sehen. Da die genutzte Version des Konverters Probleme mit der automatischen Erweiterung von Datentypen einzelner Tabellenspalten hatte, wurde der Importprozess zunächst iterativ durchgeführt. Mit dieser Vorgehensweise konnte ein zentrales Datenbankschema ermittelt werden, in das sich die NAS Dateien fehlerfrei importieren las-sen. Der initiale Importprozess (Erstaussattung) selbst dauert auf einer VM mit einer 2GHz CPU ca. 5 Stunden. Die Abbildung Fehler: Referenz nicht gefunden zeigt einen Ausschnitt der neuen Datenstruk-tur in der PostGIS DB.

Abbildung 2: Datenstruktur PostNAS

Man erkennt, dass die objektorientierten Strukturen der NAS Daten in ein flaches Tabellenschema ab-bildet werden. Die Objekte mit Raumbezug (s.g. REO⁹s – z.B. *ax_strassenachse*) erhalten, in der Spalte **wkb_geometry**, ihre Geometrie als Simple Feature im OGC WKB (**Well-KnownBinary**) Format. Die Indizes (auch die räumlichen) sind Bestandteil des Da-tenschemas und werden schon während des Imports aktualisiert. Alle Relationen werden in einer Tabelle **alkis_beziehungen** abgelegt. Diese Tabelle verknüpft die Objektidentifikatoren über eine Spalte **beziehungsart**. Die Verknüpfung kann, wie in Abbildung 3 ersichtlich, un-terschiedlichster Art sein (z.B. *istTeilVon*, *hatDirektUnten*, ...).

Auf die generierte Datenstruktur lässt sich mit allen möglichen GIS Program-men zugreifen und es können Auswer-tungen aller Art durchgeführt werden. Ein Rendern der Objekte, bspw. mit QGIS, wird so überhaupt erst performant möglich¹⁰.

Abbildung 3: Relationstabelle

9 Raumbezogene Elementarobjekte

10 In der originalen Datenhaltungskomponenten (DHK) liegen sie, aufgrund des MDA (model-driven architecture) Ansatzes, über mehrere hundert Tabellen verteilt vor! :-)

Um mit den Daten arbeiten zu können, ist es notwendig sich vorher mit ihrer (komplexen) Modellierung auseinandergesetzt zu haben. Viele Objekte, die man für spezielle Auswertungen benötigt, müssen zunächst über **joins** gebildet werden. Die nötigen Informationen zur Datenmodellierung findet man im **ATKIS-Objektartenkatalog** (ATKIS - OK Basis DLM) auf <http://www.adv-online/> → AAA-Modell → Dokumente der GeoInfoDok.

Differenzupdate per NBA Verfahren

Die grundlegende Datenstruktur für ein performantes Rendering der Daten liegt nach dem Import vor. Wie aber schon erwähnt, dauert der Import des gesamten Datenbestandes für ein mittelgroßes Bundesland ca 5 Stunden. Um die Datenmenge und Rechenzeiten in Grenzen zu halten, gibt es das **NBA** Verfahren, das nur die Änderungsinformationen übermittelt. Diese Datenmenge liegt täglich nur zwischen 0,1% und 0,2% des Gesamtdatenbestandes. Es bietet sich also an dieses Verfahren zur Aktualisierung eines Sekundärdatenbestandes einzusetzen.

Das Update erfolgt, wie auch zuvor, über den Import von XML Dateien mittels **ogr2ogr** auf der Kommandozeile. Es gibt in den NBA XML Dateien **wfs:Delete** und **wfsex:Replace** Tags¹¹.

Die Datenbank wird in einem dreistufigen Verfahren aktualisiert. Im ersten Schritt werden die nur die wfs:Delete und wfs:DeleteObjects Objekte über ogr2ogr in eine temporäre Tabelle importiert. Mit einer Postgres Funktion werden dann alle dort vorhandenen Datensätze über ihre **gml_id** aus den Objekttabellen sowie der Relationstabelle gelöscht. Die dritte und letzte Stufe entspricht einem normalen Import aller Objekte über ogr2ogr. Das tägliche Aktualisieren dauert i.d.R. nur wenige Minuten.

The screenshot shows an XML document with the following structure:

```

<?xml version='1.0' encoding='UTF-8'>
<wfs:Delete xmlns:wfs='http://www.opengis.net/wfs'>
  <wfs:DeleteObjects>
    <wfs:Object gml:id='1' />
  </wfs:DeleteObjects>
</wfs:Delete>

```

Abbildung 4: NBA XML Update Datensatz

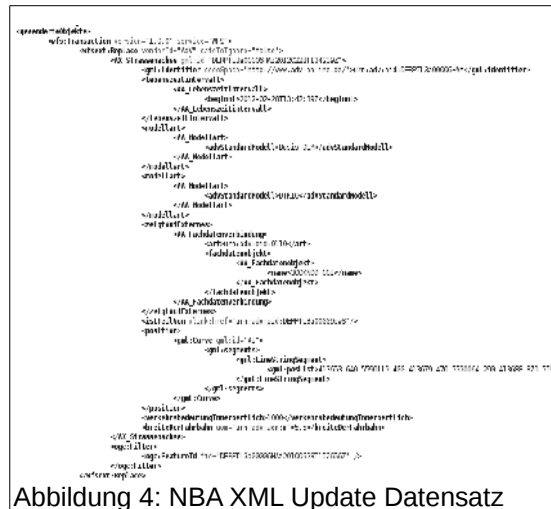


Abbildung 4: NBA XML Update Datensatz

Generalisierung mit generischen PostGIS Funktionen

Einleitung

Ziel des Projektes war die Erstellung einer hochperformanten **und** tagesaktuellen Präsentation von amtlichen, topographischen Basisdaten zur Nutzung als WMS Dienst für die Geodateninfrastruktur Rheinland-Pfalz. Der Bedarf für einen solchen Dienst bestand schon seit 2006 und er wurde in der ersten Realisierungsstufe (2006-2007) durch eine maßstabsabhängige Präsentation von Vektordaten des DLM1000, DLM250¹² und des BasisDLM gedeckt. Damals standen noch keine AAA Daten zur Verfügung und der Datenaustausch erfolgte über den Import von Shape Dateien. Es war klar, dass eine solche Lösung nicht optimal sein konnte. Das Problem besteht in der unterschiedlichen Aktualität der Datenbestände. Es konnte vorkommen, dass in einem Maßstabsbereich von z.B. 1:100.000 (DLM1000) Autobahnabschnitte nicht vorhanden waren, aber dann beim Hineinzoomen in einen größeren Maßstab plötzlich sichtbar wurden. Es musste also eine Möglichkeit geschaffen werden, die Präsentationsgraphiken in allen Maßstabsbereichen auf Basis der gleichen Datengrundlage zu generieren. Im Jahr

11 Man erkennt hier die enge Beziehung zur WFS Spezifikation des OGC.

12 Das DLM1000 sowie das DLM250 werden vom Bundesamt für Kartographie und Geodäsie in Frankfurt gepflegt. Weitere Informationen erhält man unter http://www.bkg.bund.de/nn_170890/DE/Bundesamt/Produkte/Geodaten/Landschaftsmodelle/DLM-Deutschland/DLMdeutschland_node.html nnn=true

2007 konnte ein Verfahren implementiert werden, dass mit Hilfe sehr einfacher PostGIS Funktionen, die Daten des BasisDLM (1:25.000) innerhalb von 10 Minuten in 5 verschiedenen Stufen generalisierte. Das Verfahren basierte noch auf dem Import von Shape Dateien, die zuvor per FME aus den originalen ATKIS Datenstrukturen (EDBS) extrahiert wurden. Es war rein auf die Ableitung einer einfachen Navigationsgraphik im Stil von Google Maps ausgerichtet und es wurden nur Objekte und Attribute exportiert, die für die Darstellung und Signaturierung unbedingt benötigt wurden. Mit der Umsetzung der neuen AAA Objektstruktur im ATKIS Umfeld und der parallel verlaufenden Entwicklung des PostNAS Konverters war im Sommer 2011 die Zeit reif für eine Umstrukturierung des Workflows.

Umsetzung der Generalisierung

Die Generalisierung der BasisDLM Daten in der PostNAS Datenbank erfolgt grundsätzlich über PostGIS Funktionen. Es wird hier zwischen der Generalisierung flächenhafter und linienförmiger Objekte unterschieden. Da sich die Generalisierung linienhafter Objekte mit PostGIS auf Grund ihrer Geometrie als recht einfach erweist, wird im Folgenden die Flächengeneralisierung am Beispiel der Waldflächen erläutert.

Das Ziel der Generalisierung digitaler Kartenmedien ist es, die Komplexität der Objektgeometrien zu verringern, sodass diese Objekte schneller geladen werden können (geringe Datenmenge zur Beschreibung der Geometrie notwendig) und sie auch in unterschiedlichen Maßstabsebenen lagerichtig dargestellt werden können. Um das zu erreichen kann man sich zum Teil der generischen PostGIS Funktionen bedienen, die etwa das Ausdünnen der Stützpunkte über die Funktion `ST_Simplify()` anbieten. Diese Methode ist jedoch nur für einfache Geometrien uneingeschränkt zu empfehlen. Waldflächen des BasisDLM (1:25.000) stellen aber in der Regel komplexe Objekte dar, da sich in ihrem Inneren Freiflächen, wie etwa Lichtungen befinden, die nicht der Nutzungsart Wald zuzuordnen sind. PostGIS verwendet zur Definition der OGC Simple Features unter anderem das menschenlesbare WKT-Format (WellKnownText). Ein klassischer WKT-String für ein Quadrat mit quadratischer Freifläche würde beispielsweise so geformt:

```
POLYGON((0 0,10 0,10 10,0 10,0 0),(5 5,7 5,7 7,5 7, 5 5))
```

In der Realität sind solche Objekte deutlich komplexer aufgebaut, wie die Abbildung 5 belegt: Sie zeigt einen 25 Quadratkilometer großen Ausschnitt des Pfälzer Walds bei Landau. Man kann sehr gut die großen Freiflächen und Schneisen erkennen; da diese das Polygon jedoch nicht teilen, handelt es sich um ein einzelnes – und im Hinblick auf die Geometrie valides – Objekt. Arbeitet man direkt mit diesem komplexen Ringpolygon weiter und reduziert die Anzahl der Stützpunkte mittels `ST_Simplify()`, sieht man sich unweigerlich mit fehlerhaften Geometrien konfrontiert, da sich ein Polygon beispielsweise niemals selbst überlappen darf – die Objektgeometrie wird nicht valide. Leitet man von solchen fehlerhaften Geometrien neue Objekte ab, ist das Chaos vorprogrammiert. Um das zu umgehen, zerteilt man die zusammengefassten Objekte mit `ST_DumpRings()` in einzelne Polygone, die sich dann leicht individuell bearbeiten lassen (Abbildung 6). Dies hat außerdem den Vorteil, dass man für Außen- und Innenpolygone unterschiedliche Schwellenwerte für die Stützpunktvereinfachung angeben oder zu kleine innere Elemente einfach löschen kann.

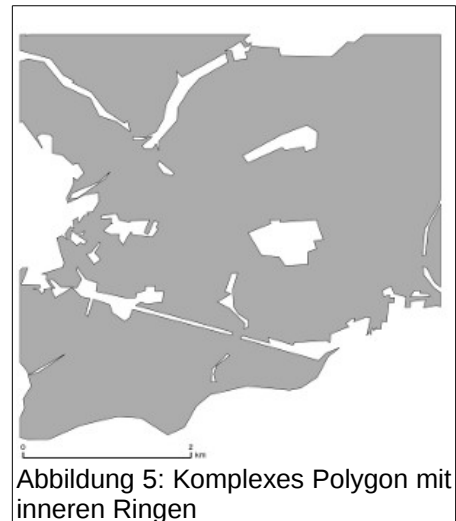


Abbildung 5: Komplexes Polygon mit inneren Ringen

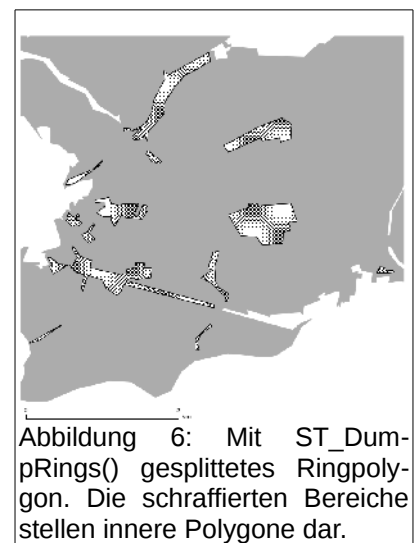


Abbildung 6: Mit `ST_DumpRings()` gesplittetes Ringpolygon. Die schraffierten Bereiche stellen innere Polygone dar.

NAS - BasisDLM Aufbereitung mit gdal/PostNAS

Hat man die Geometrieabstraktion auf den inneren und äußeren Ringen abgeschlossen, führt man alle Teilobjekte wieder zu einem zusammen. Dazu verwendet man zum Beispiel folgendes Konstrukt:

```
ST_MakePolygon(  
  ST_ExteriorRing(outer.wkb_geometry),  
  ST_Accum(ST_ExteriorRing(inner.wkb_geometry)  
) AS wkb_geometry
```

Einen simplen aber nichtsdestotrotz sehr wirksamen Ansatz zum Abstrahieren von Objektgeometrien findet sich auf der Website von Paul Ramsey [3] unter dem Stichwort Removing Complexities: Dazu wird ein Polygon zunächst mittels `ST_Buffer(geom, n)` vergrößert, anschließend zieht man denselben Wert durch `ST_Buffer(geom, -n)` wieder vom Objekt ab. Man erzielt damit eine Vereinfachung der Form, da man mit dem ersten Buffer Einstülpungen oder Löcher im Polygon einfach auffüllen kann (Abbildung 7). Durch den zweiten Buffer wird sichergestellt, dass die Lage und Ausdehnung der äußeren Form wieder auf den Ausgangswert zurückgesetzt wird. Abhängig von der Größe des Buffers kann man so auch größere Lücken in Ringpolygonen füllen, wobei man damit aber auch den äußeren Ring zum Teil sehr stark glättet, was unter Umständen nicht gewünscht ist.

Da im Zuge der Vereinfachung Objekte entstehen können, die über die Landesgrenze hinausragen, müssen diese nach erfolgter Generalisierung mit einem Polygon der Landesfläche verschnitten werden. Nur so ist es möglich, die Grenzscharfe zu gewährleisten – wichtig im Hinblick auf die Bündelung der WMS-Dienste mehrerer Bundesländer.

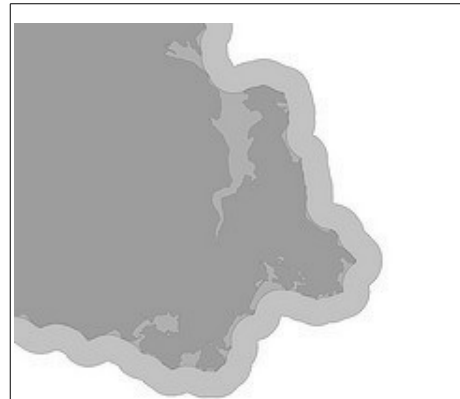


Abbildung 7: Die inneren Strukturen konnten durch `ST_Buffer()` gefüllt werden; die äußere Form wurde geglättet. (Quelle: siehe [3])

Verbesserung der Performance

Die Bereitstellung über WMS übernimmt ein UMN Mapserver, der die Vektordaten aus PostGIS abrufen und on-the-fly rendert. Es gibt also keine Kacheln, die wie etwa bei Google Maps oder OpenStreetMap serverseitig vorgehalten werden – alles geschieht in Echtzeit.

Daher ist es extrem wichtig, die Zeitspanne zwischen der Anfrage an den Mapserver und der Antwort durch den PostgreSQL-Server möglichst kurz zu halten. Um das zu gewährleisten, sind zunächst Indizes essentiell, allerdings ist es auch wichtig zu verstehen, wie PostGIS von diesen profitieren kann und wie man seine Queries entsprechend anpasst, um im Bezug auf die Laufzeit teure Zugriffe zu minimieren.

Die zweite große Stellschraube heißt Datenreduktion. Hier kann vor allem durch das Ausdünnen von Stützpunkten viel erreicht werden, wobei es hier einen klaren Trade-Off zwischen der Darstellungsqualität und dem Laufzeitgewinn gibt.

Datenreduktion kann aber auch dadurch erzielt werden, dass man – vor allem größere Objekte – in mehrere kleinere aufsplittet, da man aufgrund der Größe des Kartenviewers oft ohnehin nur einen Teil dieser großen Objekte darstellen kann, aber intern die gesamte Geometrie verarbeitet werden muss.

Anwendungsbeispiel „Kachelung“

Da im ATKIS-Datenbestand kein Polygon der Gesamtlandesfläche Rheinland-Pfalz existiert, muss diese einmalig über Aggregation der Gemeinden aus `ax_kommunalesgebiet` generiert werden (Abbil-

NAS - BasisDLM Aufbereitung mit gdal/PostNAS

dung Fehler: Referenz nicht gefunden). Danach wird das umschließende Rechteck (engl. *bounding box*) des Objekts mit einer selbst entwickelten PostGIS-Funktion *grid()* in quadratische Kacheln umgewandelt und mit der Landesfläche verschnitten. Die Kantenlänge einer Kachel variiert dabei zwischen 5 und 50 Kilometer je nach Maßstabsebene (siehe Abbildungen Fehler: Referenz nicht gefunden und Fehler: Referenz nicht gefunden).

Mit Hilfe dieser Kacheln kann man zum einen die überstehenden Geometrien in Grenznähe sauber abtrennen. Zum anderen ist es aber auch möglich, durch das Auftrennen eines großen Polygons in mehrere kleinere die transferierte Datenmenge zu reduzieren.

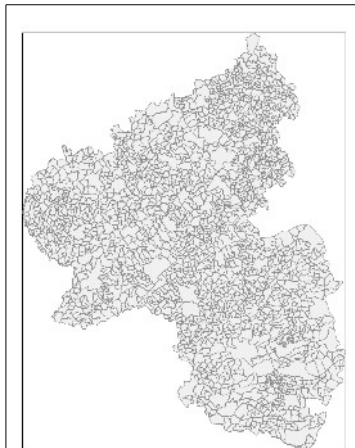


Abbildung 8: Die Ortsgemeinden werden zu einem Polygon zusammengefasst, der Rahmen entspricht der bounding box.

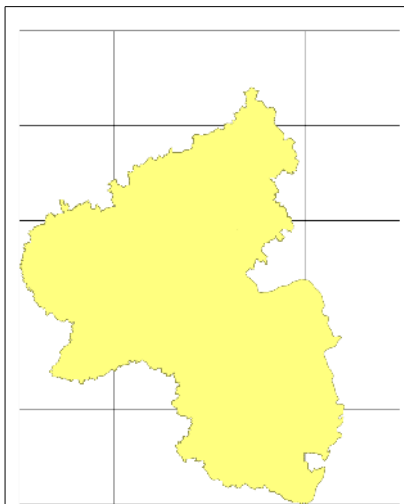


Abbildung 9: Anschließend werden mit *grid()* 50 x 50 km² große Kacheln über die bounding box gelegt.

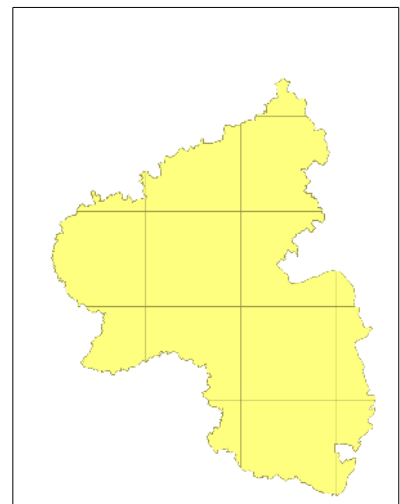


Abbildung 10: Durch die Verschneidung beider Datensätze erhält man eine gekachelte Landesfläche, die Grundlage aller weiteren Verschneidungsprozesse ist.

Beispiel Waldpolygone

Im Pfälzer Wald gibt es sehr große zusammenhängende Waldflächen, die in PostGIS als einzelne Objekte vorliegen (Abbildung 11). Bei einem Maßstab von 1:100.000 kann man in einem WebGIS nur Teile dieses Polygons darstellen, es wird allerdings intern zunächst die gesamte Geometrie, die sich hinter der *bounding box* verbirgt, geladen. Eine Abhilfe bietet hier die Verschneidung solcher Polygone mit den oben genannten Kacheln: Man erhält statt einem Objekt viele kleinere, die ihrerseits auch eine *bounding box* besitzen.



Abbildung 11: Zusammenhängendes Polygon im Pfälzer Wald (bounding box als Rahmen erkennbar)

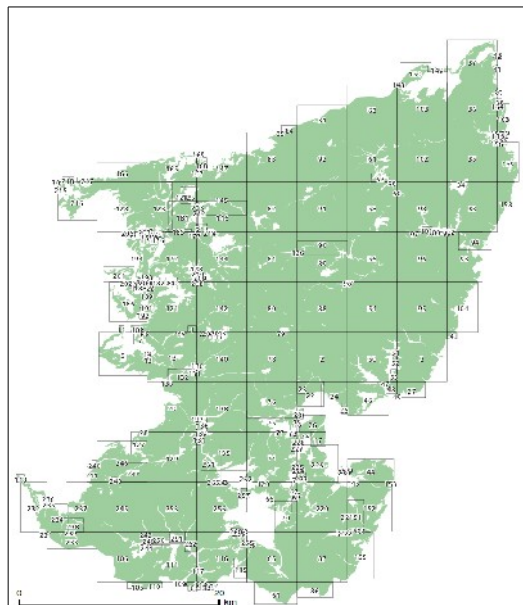


Abbildung 12: Identische Waldfläche nach der Kachelung mit 5 x 5 Kilometer

Bei einer Anfrage an den Mapserver werden über die *bounding box* des gezeigten Ausschnitts alle Geometrien aus PostGIS abgerufen, deren eigene *bounding box* innerhalb dieses Bereichs liegen. Liegt ein kleiner Teil des Waldpolygons aus Abbildung 11 in diesem Bereich, wird das gesamte Objekt mit seinen etwa 100.000 Stützpunkten aus der Tabelle geladen.

Nach der Kachelung ist es möglich, differenziert einzelne Stücke dieses Waldgebietes anzusprechen. Im Durchschnitt umfasst eine dieser Kacheln nur noch 7000 Stützpunkte.



Abbildung 13: 5x5 km²

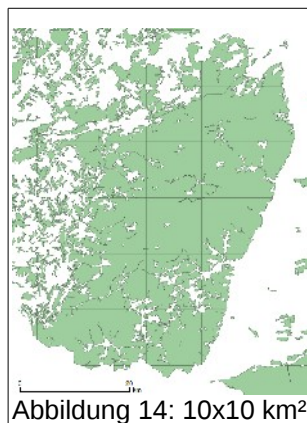


Abbildung 14: 10x10 km²

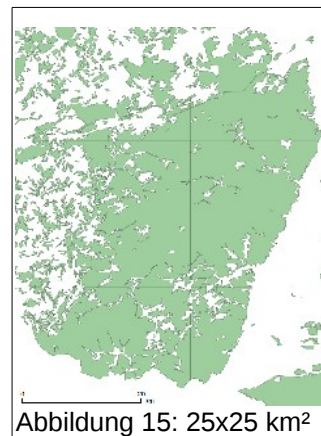


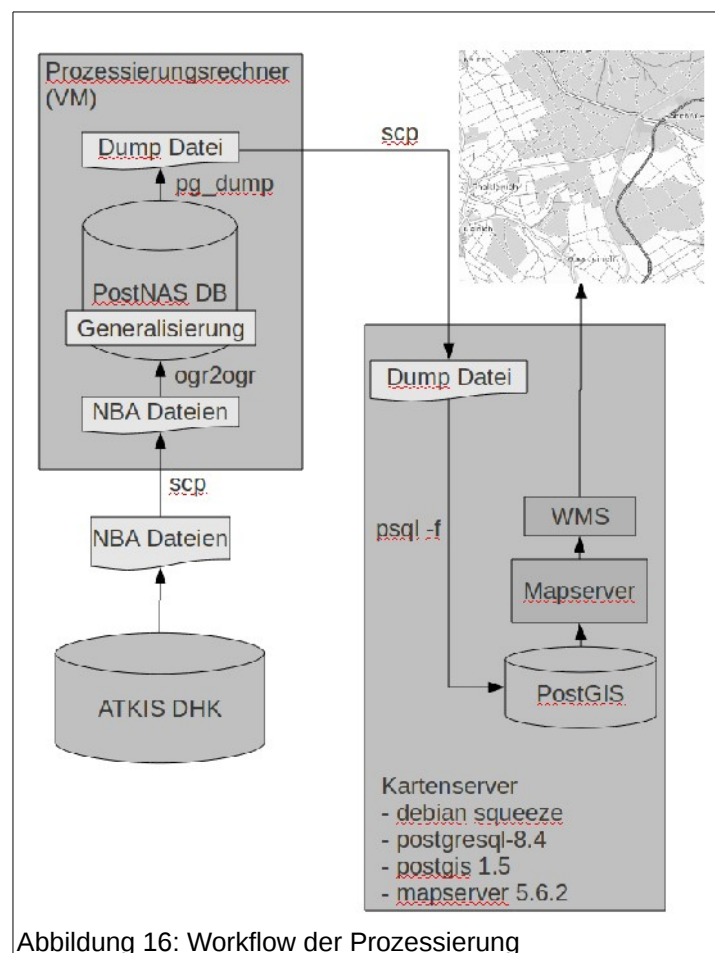
Abbildung 15: 25x25 km²

Allerdings enthalten alle Objekte aus Abbildung 12 zusammengekommen etwa 180.000 Stützpunkte, also deutlich mehr als das zusammenhängende Polygon, weshalb es wichtig ist, die Größe der Kacheln maßstabsabhängig zu variieren, da es ansonsten zu einer Verschlechterung der Performance kommt (siehe Abbildungen 13, 14 u. 15).

Ergebnisse

Implementierung des Workflows

Der im GeoPortal.rlp zu Navigationszwecken genutzte Kartendienst konnte im November 2011 auf das im Vorfeld beschriebene Verfahren umgestellt werden. Die Daten werden über Nacht in eine Sekundärdatenbank importiert und dort mit den beschriebenen Verfahren in verschiedenen Stufen generalisiert. Der Import der NBA Daten dauert nur wenige Minuten, die Generalisierung ca. 2.5h. Nach Ablauf der Prozessierungsschritte werden die benötigten Tabellen gedumpt und auf den für das Rendering verwendeten Webserver kopiert. Für Rheinland-Pfalz kommt so ungefähr 400MB an Datenvolumen in Form eines Postgres Dump Files zusammen. Der Import des Dumps in die Rendering Datenbank dauert ebenfalls nur wenige Minuten. Um keine Ausfallzeiten zu erhalten, wird die Datenbankverbindung des Kartenserver für die Dauer des Imports auf eine andere Datenbank umgestellt. Der gesamte Workflow (Abbildung 16) ist über UNIX-Shellscripte voll automatisiert und läuft jetzt schon mehr als 4 Monate reibungslos.



Leistungsfähigkeit des WMS

Die Dauer des Renderingprozesses für eine Kartenanfrage liegt, je nach Darstellungsmaßstab, zwischen 0.2s und 0.8s. Die Leistungsfähigkeit des auf generalisierten Vektordaten basierenden WMS Dienstes übersteigt - in einigen Maßstabsbereichen - selbst Dienste, die auf Rasterdaten aufgesetzt wurden. Die Tabelle 1 enthält einige Beispielanfragen. Es wurden immer 400x400 Pixel im Bildformat **image/png** abgefragt. Im Mapserver wird das Rendering über die AGG Bibliothek durchgeführt. Stellt

NAS - BasisDLM Aufbereitung mit gdal/PostNAS

man die Ausgabe auf GD Rendering ohne *antialiasing* um, so reduzieren sich die Bilddateigrößen auf ca. ein Viertel, was dann aber stark zu Lasten der Graphikqualität geht.

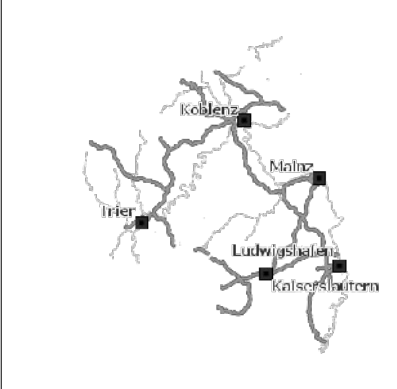
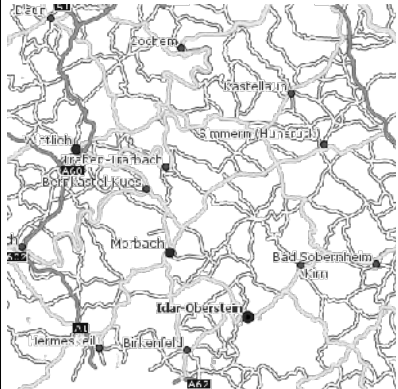
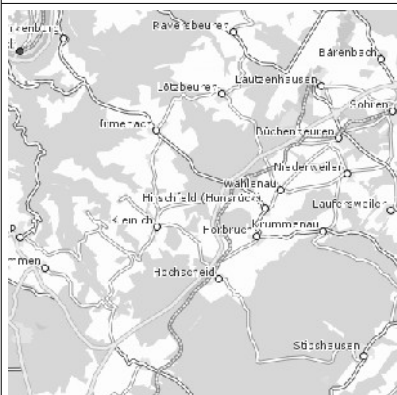
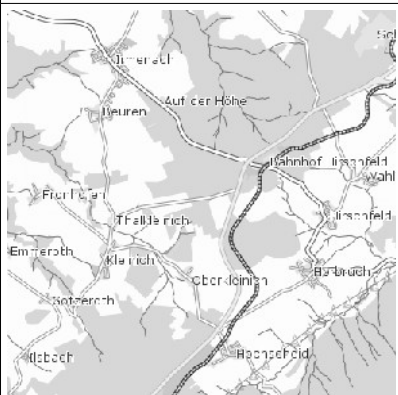
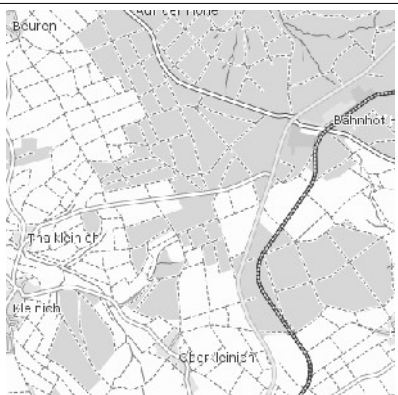
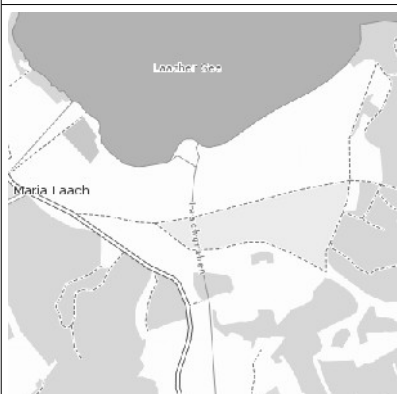
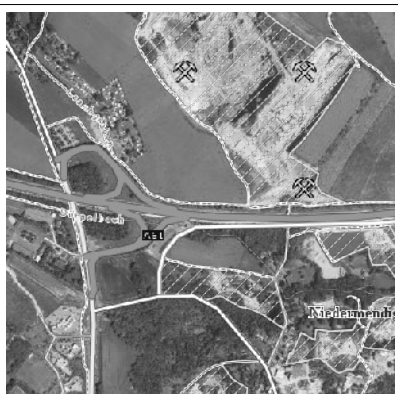
		
1:1.800.000/49kB/230ms	1:500.000/240kB/330ms	1:300.000/240kB/330ms
		
1:100.000/150kB/265ms	1:50.000/145kB/222ms	1:25.000/193kB/180ms
		
1:12.500/100kB/111ms	1:11.000/101kB/111ms	1:7.500/87kb(50kB)/112ms(82ms)

Tabelle 1: Beispielgraphiken

Im letzten Beispiel (1:7.500) ist zusätzlich das Luftbild eingeblendet. Dieses wird über einen optimierten, auf Rasterdaten basierenden Kartendienst (ebenfalls Mapserver) bereitgestellt. Zum Vergleich sind die Performanceangaben für diesen Kartenaufbau in Klammern angegeben. Im Laufe der letzten 4 Jahre hat sich gezeigt, dass Nutzer im Maßstabsbereich größer als 1:15.000 eigentlich immer eine Kombination von Luftbild und Vektorgeometrie bevorzugen. Eine flächenhafte Darstellung des Landschaftsmodells (z.B. mit dem Inhalt der TK25) wird als irritierend empfunden, zumal nicht alle dort erfassten Objektarten für den jeweiligen Anwendungsfall benötigt werden. Das menschliche Gehirn erfasst die jeweils relevanten Inhalt eines Luftbildes extrem schnell.

NAS - BasisDLM Aufbereitung mit gdal/PostNAS

Der aufgezeigte Ansatz, WebMapServices direkt auf den vektoriellen Datenbeständen aufzubauen, hat gegenüber dem aktuell weit verbreiteten Hype, für die Verbreitung von Kartenbildern im Netz auf Tile Cache Verfahren zu setzen, einige **entscheidende** Vorteile:

- Freie Wahl beliebiger Koordinatensysteme
- Freie Wahl eines Ausschnittes und Maßstabsbereiches ohne Qualitätsverlust bei der Kartographie
- Sehr schnelle Aktualisierungsmöglichkeit (Tagesaktualität problemlos realisierbar)
- Generierung dynamischer Legenden
- Dynamische Schriftplatzierung
- Der Nutzer kann einzelne Objektbereiche selbst an- und abwählen
- Abfragbarkeit von Sachdaten einzelner Objekte (z.B. Flurstücksinformationen oder auch Eigentümerdaten)
- Erzeugung beliebiger Produkte durch verschiedene Rendering Vorgaben (z.B. mapfiles), ohne zusätzlichen Bedarf an Speicherplatz
- Nutzung des gleichen Datenbestandes über WFS

Es zeigt sich, dass bei einer guten Aufbereitung der Vektordaten, ähnliche gute Performance Werte erreichbar sind wie auch bei vorgerenderten Bildkacheln. Man wird zwar die Performance von Tile Cache Services nie ganz erreichen, jedoch überwiegen die o.g. Argumente. Gerade im Hinblick auf die Bereitstellung von Daten innerhalb von Geodateninfrastrukturen fehlt es den Tile Cache Services an der notwendigen Flexibilität und Interoperabilität.

Offene Punkte / Pitfalls / TODOs / Perspektiven

PostNAS

- In der Version 1.9 der gdal Bibliothek gibt es noch einen Fehler, der dazu führt, dass bei den replace Datensätzen alle Relationen gelöscht werden. Es dürfen aber nur die Relationen gelöscht werden, die über den wfsex:Replace Datensatz neu erzeugt werden. An der Umsetzung wird zur Zeit gearbeitet (ggf. bis zur FOSSGIS 2012 gefixed).

Generalisierung

- Die Skripte zur Generalisierung wurden unter PostGIS 1.5 entwickelt. Test unter der neuen Version 2.0 – alpha ergaben Probleme.
- Im Rahmen des Einsatzes von PostGIS 2.0 sollte darüber nachgedacht werden, Generalisierungsverfahren im Rasterbereich durchzuführen. Das neue hybride Datenmodell ist hierzu prädestiniert (siehe auch [7]).

Renderingprozess

- Momentan fehlt noch die vollständige Unterstützung des Prinzips der *hidden* WMS Layer beim Mapserver – sie ist erst für Version 6.2 geplant. In der aktuellen Version muss man noch zu lokalen Kaskaden greifen um die Layer der Generalisierungsstufen im Capabilities Dokument des WMS zu unterdrücken (siehe auch [8], [9], [10]).
- Die Signaturierung nutzt keine Über- bzw. Unterführungsreferenzen. Eine saubere topologische Darstellung bezgl. der 3. Dimension ist derzeit nicht möglich.

NAS - BasisDLM Aufbereitung mit gdal/PostNAS

Kontakt zu den Autoren:

Tobias Dick
Universität Trier
Fachbereich VI Geographie/Geowissenschaften
Abteilung Fernerkundung
Behringstr. 21
54286 Trier
0651/201-4594
s6todick@uni-trier.de

Armin Retterath
Zentrale Stelle Geodateninfrastruktur Rheinland-Pfalz
Landesamt für Vermessung und Geobasisinformation RP
Ferdinand-Sauerbruch-Straße 15
56073 Koblenz
0261/492-466
armin.retterath@lvermgeo.rlp.de

Weitere Informationen im Netz

- [1] <http://trac.wherogroup.com/PostNAS/wiki/ATKIS>
- [2] <http://map.krz.de/mapwww/?Themen:ALKIS>
- [3] Ramsey, Paul; <http://blog.cleverelephant.ca/2010/11/removing-complexities.html>
- [4] <http://www.adv-online.de>
- [5] <http://www.geoportal.rlp.de>
- [6] [http://www.gdi-rp-dienste2.rlp.de/cgi-bin/mapserv.fcgi?
map=/data/umn/geoportal/karte_rp/atkis3.map&REQUEST=GetCapabilities&VERSION=1.1.1&SER-
VICE=WMS](http://www.gdi-rp-dienste2.rlp.de/cgi-bin/mapserv.fcgi?map=/data/umn/geoportal/karte_rp/atkis3.map&REQUEST=GetCapabilities&VERSION=1.1.1&SERVICE=WMS)
- [7] <http://trac.osgeo.org/postgis/wiki/WKTRaster>
- [8] <http://mapserver.org/es/development/rfc/ms-rfc-67.html>
- [9] <http://trac.osgeo.org/mapserver/wiki/HidingLayersInOGCWebServices>
- [10] <http://trac.osgeo.org/mapserver/ticket/1632>

PGVS - Konkurrenzierendes Editieren von PostGIS Layern und offline Editieren

Horst Düster

Das versionierte Bearbeiten von PostGIS-Layern wird bedeutend, wenn mehr als eine Person gleichzeitig an einem Layer arbeitet, oder spezifische Projektstände der zu bearbeitenden Daten erhalten werden sollen. Das Ziel des PostGIS Versionierungs Systems pgvs ist das Management verschiedener konkurrenzierender Versionen eines einzelnen PostGIS Layers. Dabei ist das System so angelegt, dass es sich ähnlich wie Quellcode Versionierungssysteme wie z.B. CVS oder Subversion verhält. Mit pgvs werden Layer individuell bearbeitet und die Änderungen in die produktive Umgebung zurück gespielt. Allfällige Konflikte können bereinigt werden. Im Vortrag wird ausserdem das Zusammenspiel der PostGIS Versionierung mit der Möglichkeit PostGIS-Daten offline zu editieren präsentiert.

Der erste Teil des Vortrages führt in das Konzept und die Funktionsweise von pgvs als Erweiterung von PostGIS ein. Im zweiten Teil der Präsentation wird ein QGIS Plugin vorgestellt, das als Frontend zu pgvs dient und auf diese Weise einen einfachen Einsatz von pgvs ermöglicht.

PostGIS 2.0 - Was bringt die neue Version?

Astrid Emde

Die PostGIS Version 2.0.0 wurde für das erste Quartal 2012 angekündigt. Nur noch wenige offene Tickets [1] verhindern die Veröffentlichung. Ende Februar 2012 erschien die Version PostGIS 2.0.0beta1, die zum Test bereit steht [2]. Die Entwickler haben über 900 Tickets für die Version 2.0.0 bearbeitet. Zahlreiche Neuerungen erwarten uns mit diesem Release. Diese Version löst die Serie 1.5 ab, die im März 2010 startete. Schon ein Weilchen warten die Anwender gespannt auf die neue Version.



PostGIS 2.0.0 - Die Neuerungen

Die Liste der Neuerungen in der Version 2.0.0 ist sehr lang, und zu Recht wurde hier der Hauptversionssprung gewählt.

Nachfolgend werden einige Neuerungen aufgeführt, die im Vortrag vorgestellt werden sollen. Natürlich kann in der kurzen Zeit nur eine kleine Auswahl präsentiert werden. Wer sich ausführlich informieren möchte, findet anschauliche Beschreibungen der Neuerungen in der PostGIS Dokumentation. Es gibt jeweils ein eigenes Kapitel, in dem die Neuerungen der Version auflistet werden [3]. Außerdem beschreiben die Release Notes [4], was beim Wechsel nach PostGIS 2.0.0 zu beachten ist.

Hier nun eine Auswahl der Neuerungen:

- geometry_columns ist nun eine Sicht (wie auch geography_columns), die aus dem Systemkatalog aufgebaut wird, und keine Tabelle mehr
- Geometriespalten können über typmod angelegt werden
- Datenbankerweiterung um PostGIS über CREATE EXTENSION (ab PostgreSQL 9.1)
- neue Unterstützung für Raster [5]
 - Typ Raster und Rasterfunktionen wurden integriert
 - nahtlose Vektorintegration
 - GDAL Export Vektor/Raster in jedes von GDAL unterstützte Format
 - GDAL Import von Rasterdaten
 - Raster Transformation und Resampling
 - räumliche Verschneidung von Raster- und Vektordaten
 - Raster Analyse – Histogramme, Pixelwerte, Map Algebra
- Topologie-Unterstützung
 - setzt die SQL/MM Vorgaben um und stellt weitere Funktionen bereit [6]
 - über Typumwandlung kann eine topogeometry in eine Geometrie umgewandelt werden (SELECT topo::geometry FROM topotable;)
- verbesserte 3D Unterstützung
 - viele Funktionen unterstützen nun auch 3D (z.B. ST_Distance, ST_Transform, ST_3DDistance, &&&)
 - N-dimensionaler räumlicher Index
 - neue 3D Typen TIN und PolyHedralSurface
- viele tolle neue Funktionen
 - ST_Split
 - ST_Snap
 - Datenbereinigung: ST_IsValidDetail, ST_MakeValid, ST_RemoveRepeatedPoints
 - ST_OffsetCurve
 - ST_GeomFromGeoJSON
 - ST_FlipCoordinates

PostGIS 2.0 - Was bringt die neue Version?

- KNN Gist Index (K Nearest Neighbor Index) für schnellere Distanzabfragen mit Limit
- der Shapefile Loader kann nun mehrere Shape-Dateien laden

Quantum GIS verfügt bereits über ein PostGIS Topology Plugin (TopoViewer). Auch ein QGIS Raster Python Plugin liegt für QGIS schon vor. MapServer unterstützt ebenfalls PostGIS Raster. gvSIG verfügt über ein Raster-Plugin. Über Raster SQL-Funktionen wie ST_AsPNG können PostGIS Rasterobjekte beispielsweise als PNG ausgegeben werden und so leicht visualisiert werden. Hierzu kann beispielsweise der auf PHP basierende PostGIS Webviewer [14] verwendet werden.

Was ist beim Datenbankumzug auf PostGIS 2.0.0 zu beachten?

In den Release Notes [4] wird beschrieben, was beim Wechsel nach PostGIS 2.0.0 zu beachten ist. Beim Umzug von Datenbanken auf die neue Version ist ein Hard Upgrade [7] notwendig, d.h. es muss ein Dump erstellt werden und mit Hilfe des Skriptes `postgis_restore.pl` in die neue Datenbank eingeplayed werden.

Einige Funktionen wurden schon ab der Version 1.2 nicht mehr unterstützt und entfallen nun (> 250). So entfallen beispielsweise `Area`, `Length`, `Buffer`, `Intersects` und die jeweiligen Funktionen mit dem Präfix `ST_` (gemäß SQL/MM Standard) werden nun verwendet. Eine Liste der entfallenden Funktionen findet sich unter <http://trac.osgeo.org/postgis/ticket/722,7302>. Prüfen Sie, ob in Ihren Sichten, Funktionen oder Skripten noch veraltete Funktionen verwendet werden und passen Sie diese ggf. an. Schauen Sie sich die Liste der entfallenden Funktionen in der Datei `legacy.sql.in.c` [8] an. Die 3D Funktionen verwenden nun den Präfix `ST_3D` statt nur `ST_`.

Download

Die neueste Version kann über die Download-Seite [9] oder die tag-Liste [2] heruntergeladen werden. Auf der Download-Seite finden Sie auch einen aktuellen Snapshot aus dem SVN.

Vielen Dank

Vielen Dank an das PostGIS Team und alle, die das Projekt und die neue Version unterstützt haben. Uns allen viel Spaß mit der neuen PostGIS Version 2.0.0.

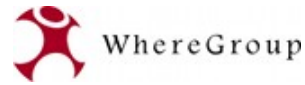
Weitergehende Informationen

Es finden sich schon viele Dokumentationen, Präsentationen und Videos rund um PostGIS 2.0.0 im Netz. Einen guten Einstieg bietet sicherlich die Trac-Wikiseite [10]. Sehr empfehlenswert ist auch das Buch *PostGIS in Action* [11], das im April 2011 erschienen ist und auf PostGIS 2.0, die Rasterunterstützung sowie PostgreSQL 9.1 eingeht und viele weitere Informationen rund um PostgreSQL/PostGIS beinhaltet. Auf der FOSS4G 2011 in Denver wurde PostGIS 2.0 ebenfalls in Präsentationen vorgestellt [13].

PostGIS 2.0 - Was bringt die neue Version?

Kontakt zur Autorin:

Astrid Emde
WhereGroup GmbH & Co KG
Eifelstraße 7, 53119 Bonn
+49 (0)228 / 90 90 38 19
astrid.emde@wheregroup.com
<http://wheregroup.com>
<http://foss-academy.eu>



Literatur

- [1] Trac Milestone PostGIS 2.0.0 <http://trac.osgeo.org/postgis/query?status=assigned&status=new&status=reopened&group=status&milestone=PostGIS+2.0.0>
- [2] Download tags: <http://trac.osgeo.org/postgis/browser/tags/>
- [3] PostGIS 2.0 Dokumentation aus dem SVN http://postgis.refrations.net/documentation/manual-svn/PostGIS_Special_Functions_Index.html#NewFunctions_2_0
- [4] http://postgis.refrations.net/documentation/manual-svn/release_notes.html#id3327378
- [5] http://postgis.refrations.net/documentation/manual-svn/RT_reference.html
- [6] <http://postgis.refrations.net/documentation/manual-svn/Topology.html>
- [7] http://postgis.refrations.net/documentation/manual-svn/postgis_installation.html#hard_upgrade
- [8] <http://trac.osgeo.org/postgis/browser/trunk/postgis/legacy.sql.in.c>
- [9] <http://postgis.org/download/>
- [10] Trac-Wiki mit Links zu Vorträgen und Videos: <http://trac.osgeo.org/postgis/wiki>
- [11] Obe, Regina; Hsu, Leo : PostGIS in Action, <http://www.manning.com/obe/>, 2011.
- [12] Obe, Regina; Hsu, Leo : Today and Tomorrow - PostGIS in Action, Geoinformatics, 12/2011.
- [13] Vortrag Obe, Regina; Hsu, Leo: PostGIS 2.0 the new stuff, <http://www.postgis.us/downloads/FOSS4G2011PostGIS20NewStuff.pdf> , Denver, FOSS4G 2011.
- [14] http://postgis.us/downloads/postgis_webviewer_php.zip

Neues vom QGIS Server und -Webclient

Dr. Marco Hugentobler

QGIS Server ist ein WMS Server, der auf den QGIS Bibliotheken aufbaut. Ein grosser Vorteil davon ist, dass ein existierendes QGIS Projekt sowohl für das DesktopGIS als auch für das WebGIS verwendet werden kann.

In QGIS Server gibt es einige wichtige Funktionen, die über den WMS Standard hinausgehen, wie zum Beispiel das webbasierte Drucken, Suche und Selektion. Um diese Funktionen zu verwenden, gibt es den QGIS Webclient, der auf GeoExt und OpenLayers aufbaut und die zusätzlichen Funktionen out of the box unterstützt.

QGIS Server und -Webclient werden in der Schweiz unter anderem vom Kanton Glarus und der Stadt Uster produktiv eingesetzt und in Zusammenarbeit mit der QGIS Community weiterentwickelt.

Suche und Selektion

Suche von Objekten und deren Selektion mit einer speziellen Farbe ist eine Funktion, die in Webkarten häufig gewünscht wird. In der WMS Spezifikation wird weder Suche noch Selektion behandelt. Trotzdem ist es praktisch, wenn man beides gleich mit dem WMS Server machen kann, anstatt nochmals einen Serverdienst dafür aufzusetzen (und zu konfigurieren).

QGIS Server führt für die Selektierung die Möglichkeit ein, in der GetFeatureInfo Anfrage die Position wegzulassen und dafür FILTER Parameter auf den Layers mitzugeben. Der Filter-String ist abhängig von der Datenquelle, in der Regel (PostgreSQL) ist es ein SQL-String. Um SQL-Injections zu verhindern, gibt es eine Reihe von Einschränkungen, die für einen Filterstring beachtet werden müssen.

Hier ein einfaches Beispiel, um in mylayer 1 alle Objekte mit Wert 3 in der Attributspalte OBJECTID und in mylayer2 alle mit dem wert 'blabla' in der Attributspalte text zu suchen:

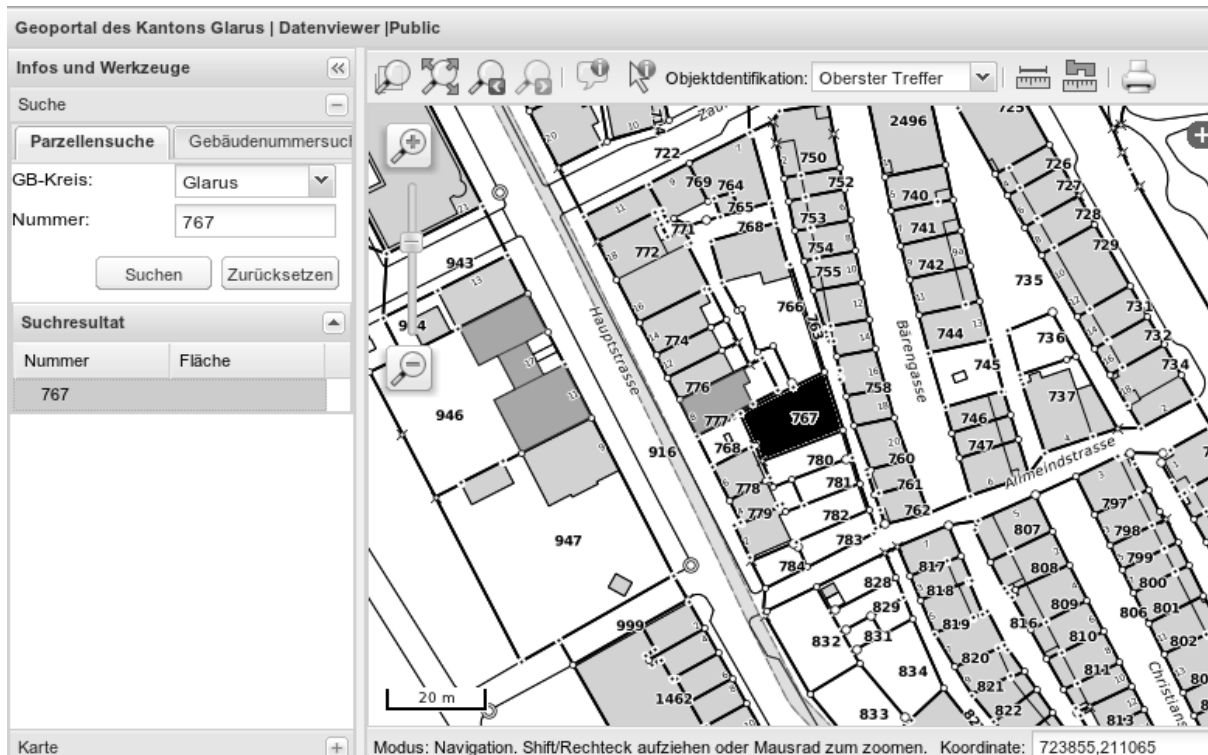
`http://myserver.com/cgi/qgis_mapserv.fcgi?REQUEST=GetMap&LAYERS=mylayer1,mylayer2&FILTER=mylayer1:"OBJECTID" = 3;mylayer2:'text' = 'blabla'&....`

In der GetCapabilities Antwort sind dann nicht nur die Attribute enthalten, sondern auch die Feature-IDs und die bounding boxes der gefundenen Objekte. Dadurch kann der Webclient auf die Auswahl zoomen und vom Server eine Karte mit selektierten Objekten anfragen. Im SELECTION Parameter werden dann die Ids der gewünschten Objekte mitgegeben:

`http://myserver.com/cgi/qgis_mapserv.fcgi?REQUEST=GetMap&LAYERS=mylayer1,mylayer2&SELECTION=mylayer1:3,6,9;mylayer2:1,5,6`

Neues vom QGIS Server und -Webclient

Eine graphische Benutzerschnittstelle für die Suche im QGIS Webclient sieht dann folgendermassen aus:



GetLegendGraphics

QGIS Server unterstützt auch den WMS Aufruf GetLegendGraphics. Es werden ausserdem noch eine Reihe zusätzlicher Parameter unterstützt, mit denen das Aussehen des Legendenbildes beeinflusst werden kann:

- BOXSPACE :Abstand zwischen Legendenrahmen und Inhalt (in mm)
- LAYERSPACE: Vertikaler Abstand zwischen Layertitel und den Layeritems
- SYMBOLSPACE: Vertikaler Abstand zwischen den Symboleneinträgen
- ICONLABELSPACE: Horizontaler Abstand zwischen Icon und Labeltext
- SYMBOLWIDTH: Breite der Vorschausymbole
- SYMBOLHEIGHT: Höhe der Vorschausymbole
- LAYERFONTFAMILY / ITEMFONTFAMILY: Schriftart für Layertitel und Text
- LAYERFONTBOLD / ITEMFONTBOLD : 'TRUE' für Fett
- LAYERFONTSIZE / ITEMFONTSIZE: Fontgrösse in Punkt
- LAYERFONTITALIC / ITEMFONTITALIC: 'TRUE' für kursive Schrift
- LAYERFONTCOLOR / ITEMFONTCOLOR: Schriftfarbe mit Hexadezimalcode (z.B. #FF0000 für rot)

Unterstützung für WMS 1.1.1

Als im Jahre 2006 mit der Programmierung des QGIS Servers begonnen wurde, war die WMS Spezifikation 1.3 bereits draussen und der Server wurde mit Fokus auf Version 1.3 entwickelt. In der Praxis

Neues vom QGIS Server und -Webclient

ist es aber so, dass die Version 1.1.1 viel weiter verbreitet ist und einige Clients mit 1.3 nichts anfangen können. Neuerdings wird die Version 1.1.1 auch von QGIS Server unterstützt.

Zukünftige Entwicklungen

Für die zukünftige Entwicklung ist geplant, noch mehr Parameter aus dem QGIS Projekt an den Webclient zu übergeben und in einem WebContext XML-Dokument zu Verfügung zu stellen (z.B. Startausdehnung oder welche Layer sichtbar sind).

Zudem gibt es Bestrebungen, mit QGIS Server in Zukunft auch WFS-Dienste anzubieten.

Kontakt zum Autor:

Dr. Marco Hugentobler
Sourcepole AG
Churerstrasse 22, CH-8038 Pfäffikon SZ
+41 44 440 77 11
marco.hugentobler@sourcepole.ch

Literatur

[1] QGIS 2012, http://hub.qgis.org/projects/quantum-gis/wiki/QGIS_Server_Tutorial

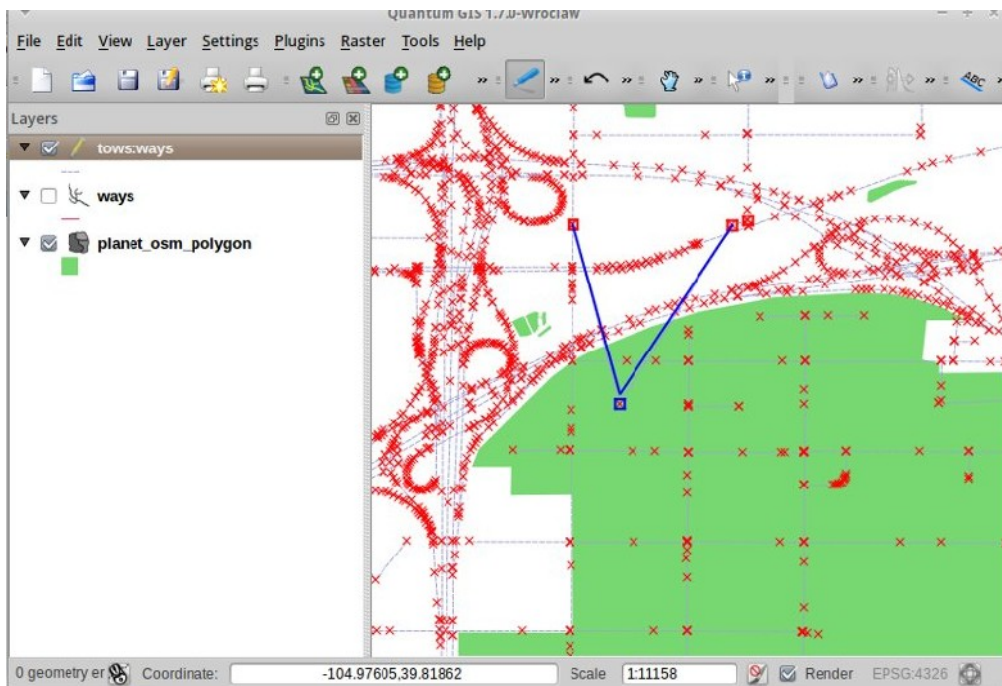
[2] Hugentobler, Marco; Neumann Andreas: WebGIS mit QGIS und GeoExt. FOSSGIS 2011

TinyOWS - der schlanke WFS Server

Pirmin Kalberer

TinyOWS ist ein hochperformanter, leichtgewichtiger und einfach einzurichtender Transaktionaler Web Feature Service (WFS-T). Betrieben wird er als CGI- oder FastCGI-Modul in einem Standard-Webserver mit PostGIS als Datenspeicher. WFS 1.0 und WFS 1.1 sind standardkonform implementiert und alle rund 1000 OGC CITE Tests wurden erfolgreich durchlaufen.

TinyOWS ergänzt einen WMS-Server (z.B. UMN MapServer oder QGIS Server) um die Fähigkeit, editierbare Daten über WFS-T für Desktop-Clients wie QGIS oder Web-Clients wie OpenLayers bereitzustellen. Die Konfiguration erfolgt über eine XML-Datei oder über ein Mapfile, welches gleichzeitig für den MapServer verwendet werden kann.



WFS-T Layer in QGIS im Editier-Modus

Kontakt zum Autor:

Pirmin Kalberer
Sourcepole AG
Churerstrasse 22
CH-8808 Pfäffikon SZ
+41 44 440 77 11
pka@sourcepole.ch

Links:

- <http://mapserver.org/trunk/tinyows/>
- http://live.osgeo.org/de/overview/tinyows_overview.html

MXD2map - ein freier Konverter von ArcGIS MXD-Dateien zu UMN Map-Server

OGC-konforme Dienste auf Basis von UMN MapServer mit ArcGIS erstellen

Stephan Holl <stephan.holl@intevation.de>, Intevation GmbH

Einleitung

ESRI ArcGIS ist in vielen Firmen und Institutionen vertreten und in den GIS-Arbeitsprozessen fester Bestandteil. GIS-Bearbeitung geschieht vielerorts mit diesem Desktop-GIS-Werkzeug. Um Arbeitsstände auszutauschen, besteht nur die Möglichkeit, Daten und Gestaltungsvorschriften z.B. per Mail zu verschicken. Dadurch entstehen redundante Datensammlungen, die zunehmend schwerer zu verwalten sind.

Im Zeitalter der vernetzten Informationswelt stehen unterschiedliche Möglichkeiten zur Verfügung, seine (Geo-)Daten zu teilen. MXD2map ist ein Werkzeug, welches auf Basis von lokal vorgehaltenen Geodaten und Gestaltungsvorschriften einen webbasierten (Geo-)Datendienst gemäß der OGC WMS-Spezifikation erstellen kann – und das ohne zusätzliche Lizenzierung von ESRI ArcGIS-Server-Komponenten.

Was ist MXD2map?

MXD2map erlaubt ein direktes Publizieren über eine OGC-konforme WMS- und WFS-Schnittstelle, direkt aus dem ArcGIS heraus. Durch den eingesetzten UMN MapServer werden die Daten direkt aus den primären Datenquellen als Dienst serviert. Es müssen nunmehr nur noch Links auf die Dienste der entsprechenden Daten verschickt werden. Die eigentlichen Rohdaten verbleiben beim Datenanbieter. Ein Teilen von Informationen ist nunmehr einfach und effektiv möglich.

Gestaltungsvorschriften wie Klassifikationen, Label, Layeranordnungen etc. werden über diesen Konverter in die Mapfile-Syntax umgewandelt und direkt über einen eigenen Webserver in einer Vorschauansicht dargestellt. Somit besteht die Möglichkeit, mit einem Knopfdruck einen OGC-konformen Webservice aus einem MXD-Projekt zu erstellen. Abbildung 1 zeigt schematisch die Funktionsweise von MXD2map sowie die involvierten Techniken und Technologien.

MXD2map ist ein freier Konverter zur Erzeugung von UMN MapServer Mapfiles aus ESRI ArcGIS MXD-Dateien. Das Kartenlayouten und -zusammenstellen der Layerinformationen geschieht weiterhin in ArcGIS, eine Darstellung als WMS- und WFS-Service wird über den UMN MapServer realisiert. MXD2map bietet diese Funktionalität als Toolbox-Eintrag direkt aus ArcGIS heraus.

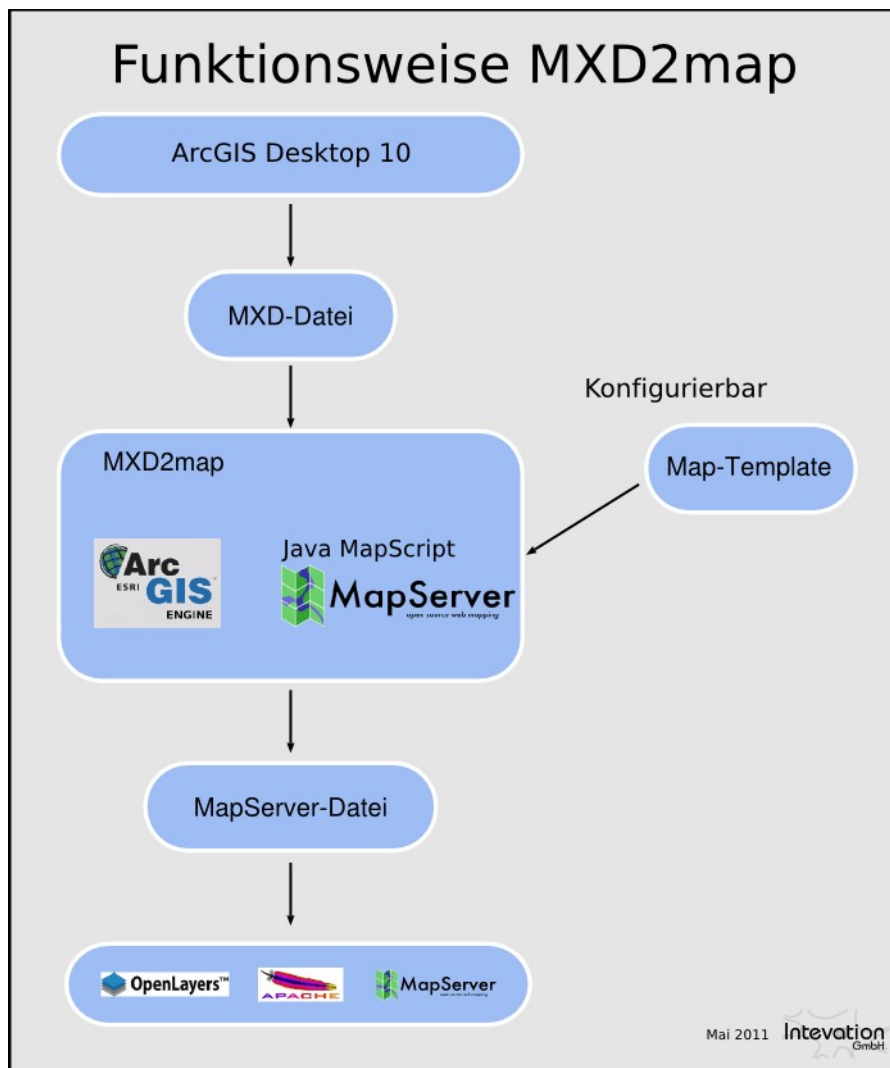


Abbildung 1: Funktionsschema von MXD2map

Durch die Nutzung der ArcGIS API wird ein Metaformat innerhalb des Konverters erstellt. Die Software MXD2map ist so generisch gehalten, dass über eine Plugin-Architektur weitere beliebige Ausgabe-Formate (wie z.B. QGIS, mapnik etc.) angeschlossen werden könnten. In weiteren Ausbaustufen in 2012 werden insbesondere die Labelplatzierungen sowie weitere Detailverbesserungen angegangen.

Es existieren einfache Installationspakete für Microsoft Windows, sodass der Konverter direkt installiert und ausprobiert werden kann. Mitgelieferte Beispieldaten zeigen die Funktionalität direkt nach der Installation.

MXD2map - ein freier Konverter von ArcGIS MXD-Dateien zu UMN MapServer

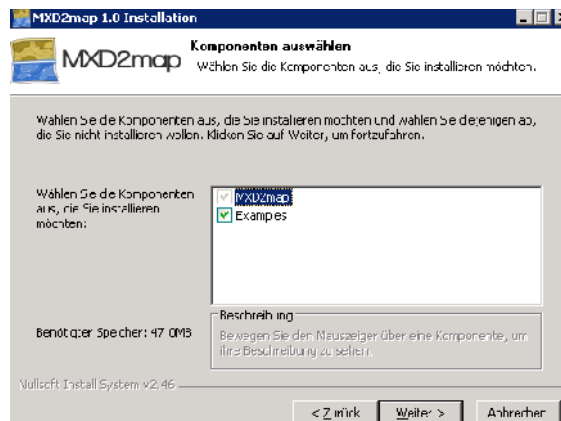


Abbildung 2: Installationsdialog von MXD2map

MXD2map ist Freie Software und unter der GNU Lesser Public License (LGPL) in der aktuellen Version 1.0 von der Webseite <http://www.mxd2map.org> verfügbar. Eine ausgiebige Dokumentation zur Einrichtung und Nutzung ist ebenso unter <http://www.mxd2map.org/documentation> verfügbar.

Der Vortrag gibt Einblicke in die Funktionalitäten, zeigt Usecase-Szenarien sowie eine Roadmap der weiteren geplanten Funktionen und lädt Interessierte zum Ausprobieren und mitmachen ein.

Kontakt zum Autor:

Stephan Holl <stephan.holl@intevation.de>
Intevation GmbH
Neuer Graben 17
49074 Osnabrück
+49-541-335083 663

Literatur

- MXD2map: <http://www.mxd2map.org>
- UMN MapServer: <http://www.mapserver.org>
- ESRI ArcGIS: <http://www.esri.com>

www.geoportal.de - deutschlandweit Geodaten finden und nutzen

Olaf Knopp, WhereGroup GmbH & Co. KG

Einführung

Auf der FOSSGIS 2011 wurde bereits über die Implementierung zweier zentraler Komponenten der Geodateninfrastruktur Deutschland (GDI-DE) – den „Geodatenkatalog-DE“ sowie die Migration der Mapserver-Komponenten des GeoPortal.Bund – berichtet [1].

Mit dem Aufbau des Geoportal.DE wurde nun auch die dritte Komponente durch die WhereGroup implementiert. Die Umsetzung erfolgte in Zusammenarbeit mit der Materna GmbH, die die Verantwortung für die CMS-Komponenten übernahm.

Der vorliegende Artikel erläutert Aspekte der technischen Umsetzung, wobei der Schwerpunkt auf der Webkomponente des Geoportals liegt. Geodatenkatalog-DE und MapServer werden der Vollständigkeit halber nur kurz angesprochen.

Ziel des Geodatenkatalog-DE ist der Aufbau eines zentralen Dienstes, der Katalogdienste bzw. deren Daten zusammenführt, an andere Strukturen abgibt und Recherchen erlaubt. Er umfasst Software zur Zusammenführung, Konsolidierung und Abgabe von Metadaten, ein entsprechendes Recherchewerkzeug sowie einen INSPIRE-konformen Metadateneditor zur Erfassung eigener Datensätze. Die Zusammenführung der Katalogdienste erfolgt über vorhandene Standard-Schnittstellen (OGC CSW).

Die Mapserver-Komponente stellt einen OGC-kompatiblen Internet Kartenserver dar, der den Anforderungen von INSPIRE und GDI-DE genügt. Die Hauptfunktionen dieses Moduls umfassen im Wesentlichen die Integration von Geodatendiensten und deren weitere Bereitstellung für Visualisierungskomponenten oder als INSPIRE-/GDI-DE-konforme Geodatendienste.

Die Freischaltung des Geoportal.DE stellte einen wichtigen Meilenstein beim Aufbau der GDI-DE dar. Seine frei verfügbare Webseite [2] bietet den zentralen Zugriff auf die Geodaten der öffentlichen Hand. Über den angebundenen Geodatenkatalog-DE lassen sich standardkonform vorgehaltene Datenbestände durchsuchen sowie weiterführende Informationen und Metadaten anzeigen. Über den integrierten Kartenviewer lassen sich Daten visualisieren, mit anderen Beständen kombinieren und in verschiedene Formate exportieren.

Die Umsetzung der Module erfolgt größtenteils mit freier Software. Eine Ausnahme stellt lediglich die Software Government Site Builder (GSB) dar, die als Content Management System zum Einsatz kommt [3]. Verwendet wurden folgende Komponenten:

- GeoNetwork opensource
- PostgreSQL/PostGIS
- MapProxy
- Mapbender3 inkl. OpenLayers
- Government Site Builder
- Apache Lucene und Solr

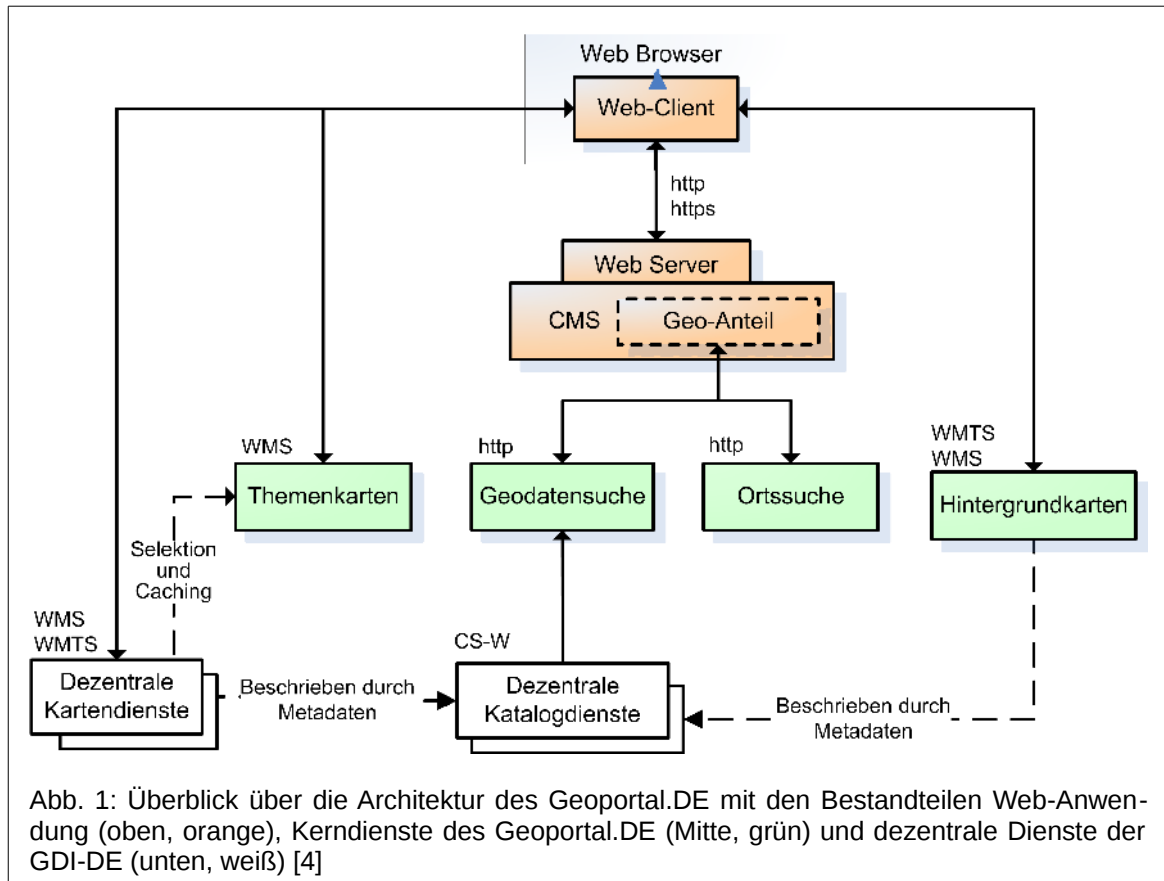
Die offizielle Freischaltung des Geoportal.DE erfolgte am 06. März 2012 im Rahmen der Cebit .

Technische Umsetzung

Die Umsetzung des Geoportal.DE und der angebundenen Komponenten erfolgte vollständig als dienstbasierte Infrastruktur. Grundlage bildet das Architekturkonzept der GDI-DE V2.0 auf Basis der standardisierten Schnittstellen OGC Web Map Service (WMS), OGC Web Feature Service (WFS) und

OGC Web Catalogue Service (CSW). Die Suche zusätzlich eine performanten Schnittstelle auf Basis von Lucene und Solr (s.u.).

Die Architektur ist in Abbildung 1 schematisch dargestellt.



Den Kern des Systems bildet der zentrale Katalogdienst, der mit Geonetwork umgesetzt wurde. Zur Zeit harvestet er knapp 30 Kataloge von Länderverwaltungen, Bundesämtern und -behörden sowie anderen öffentlichen Einrichtungen. Später sollen gut 50 Dienste angebunden werden.

Der als Mapbender3-Anwendung programmierte Geodatenkatalog-DE macht die Datenbestände des Katalogdienstes durchsuch- und anzeigbar. Zur räumlichen Suche bindet er eine Ortssuche an, die als JSON-Schnittstelle vom Geodatenzentrum angeboten wird.

Das Geoportal.DE integriert alle Komponenten und macht sie für den Anwender auf einer Webseite verfügbar. Der Zugang zu den Geodaten erfolgt über eine Suche im Geodatenkatalog.DE oder über angebotene Themenkarten in Form von vorkonfigurierten WMC-Dokumenten.

Bei wichtigen Diensten wie beispielsweise Hintergrundkarten, werden Performance und Verfügbarkeit durch die Mapserver-Komponenten sichergestellt.

Einzelne Aspekte und Module des Systems werden nachfolgend detaillierter betrachtet.

Die Suche

Der Suchclient des Geodatenkatalogs ermöglicht die Recherche in den Datenbeständen des Katalogdienstes. Wie auch beim Kartenclient wurde größter Wert auf eine einfache und intuitive Bedienung gelegt. Hierzu gehören z.B. Autocomplete- und LiveSuggest-Funktionen in Formularfeldern, „Meinten-Sie-Funktionen“ bei erfolglosen Suchen oder Vorschlagfunktionen durch Auswertung der beliebtesten Karten und häufigsten Suchanfragen.

Während der Katalogdienst nach Außen als CSW zum Harvesting angeboten wird, erfolgt die Suche innerhalb des Portals über eine eigene hochperformante Suchschnittstelle. Dazu wird der in PostgreSQL gespeicherte Datenbestand mittels Lucene indiziert. Gemäß den Vorgaben des AK Metadaten der GDI-DE, erfolgte dabei die Berücksichtigung bestimmter Rankingkriterien wie die Gewichtung von Metadatenelementen im Suchergebnis und eine Bewertung von Metadatenätzen. Als Suchschnittstelle kommt Apache Solr zum Einsatz. Das Suchergebnis wird gemäß den Vorgaben durch verschiedene konfigurierbare Kriterien beeinflusst:

- Gewichtungsklassen: Für jede Klasse können sogenannte Boostfaktoren bestimmt werden. Sie geben an, die wie stark ein Metadatenelement in die Berechnung der Relevanz einfließt.
 - Klasse 1: (sehr wichtig): z.B. Suchbegriff kommt im Titel vor.
 - Klasse 2: (wichtig): z.B. Suchbegriff kommt in den Schlüsselwörtern oder im Abstract vor.
 - Klasse 3: (weniger wichtig): z.B. Suchbegriff kommt im alternativen Titel/Anbieter vor
- Trunkierte Treffer: für jede Klasse kann ein Boostfaktor für einen direkten Treffer oder einen trunkierten Treffer bestimmt werden. So ist z.B. der Begriff "Wasser" ein direkter Treffer während der Begriff "Wasserschutzgebiet" einen trunkierten Treffer darstellt.
- Sortierung von Karten und Daten nach verschiedenen Kriterien wie Relevanz, Größe der Fläche oder Datum.

Die Ortssuche verwendet einen JSON-Suchdienst des Geodatenzentrums, der – ebenfalls mit Autocomplete- und „Meinten-Sie-Funktion“ ausgestattet – an verschiedenen Stellen des Portals eingebunden wurde.

Ergebnis der Suche ist eine übersichtliche Darstellung der gefundenen Darstellungsdienste, Datensätze und Portalinhalte. Sie ermöglicht die sofortige Anzeige von Geodaten im Kartenviewer des Geoportals sowie deren Metadaten, die Bewertung und Kommentierung von Datensätzen (nur für angemeldete Benutzer) oder den Download der Metadaten.

The screenshot displays the Geoportal.de search interface. At the top, there is a navigation bar with 'Karten', 'Suche', and 'Service' tabs. Below this, the 'Suche' (Search) section is active. On the left, a sidebar titled 'Themengebiete' (Thematic Areas) lists various categories like 'Wetter / Klima', 'Geologie / Bodenkunde', 'Wasser', etc. The main search area has two input fields: 'Nach was suchen Sie?' (What are you searching for?) with the text 'boden' and 'Wo suchen Sie?' (Where are you searching?). Below these are example suggestions. A 'Suchen' (Search) button is present. To the right of the search area, there are options to 'Suche speichern' (Save search) and 'Neue Suche' (New search). Below the search area, there is a section for 'Raumbezug über Karte einschränken' (Restrict search by map). The results are displayed in a table with three columns: 'Karten (727)', 'Geodaten (24209)', and 'Inhalte (31)'. The 'Geodaten' column is selected. Below the table, there is a pagination bar showing 'Treffer 71 - 80 von 727' and a list of page numbers. The first result is 'LGB: HGW Chrom - Chrom, Mittlerer Gesamtgehalt (Oberboden)' with a date of '2011-11-08' and a rating of 5 stars. It includes a description of soil formation and a button 'In Karte anzeigen' (Show in map). The second result is 'LGB: HGW Nickel - Oberboden' with a date of '2011-11-08' and a rating of 5 stars, also with a description and a 'In Karte anzeigen' button.

Abb. 2: Darstellung der Suchergebnisse

Eine Besonderheit stellt die Daten-Service-Kopplung dar. Alle Metadatenätze für Daten werden beim Harvesting vorprozessiert, damit eine performante Suche auf die gekoppelten Dienste erfolgen kann. Hierzu werden über den „resourceIdentifier“ die Dienste identifiziert, in denen die Daten angezeigt werden, und deren Capabilities-Aufruf gespeichert. Die URLs werden in der Datenbank abgelegt und dem Metadatenatz der Daten angehängt. Der umgekehrte Fall, also die Kopplung der Datensätze in den Dienstemetadaten, erfolgt aus dem Element „operatesOn“. Dort wird der „resourceIdentifier“ des Datensatzes kodiert.

In der Ergebnisanzeige der Suche werden Metadatenätze für Daten, für die ein passender oder mehrere passende Dienste gefunden wurde, mit einem Button "Anzeigen" versehen. Gibt es nur einen Dienst, wird der Button mit dem Capabilities-Link des Dienstes versehen. Kommen mehrere Dienste in Frage, hat der Anwender die Möglichkeit, einen Dienst aus einer Auswahlliste zur Anzeige auszuwählen.

Das Geoportal.DE

Bei der Konzeption des Geoportals stand eine intuitive und einfache Bedienung im Vordergrund. Das Portal soll es als zentrale Schnittstelle der GDI-DE allen Bürgern ermöglichen, sich im komplexen Themengebiet von Geodateninfrastrukturen, INSPIRE und Kartendiensten zurechtzufinden und so einen einfachen Zugang zu Geodaten in Deutschland ermöglichen.

Grundlage für die Umsetzung war ein Designentwurf, der vom BKG bereitgestellt wurde. Die Implementierung erfolgte mit Government Site Builder für die Webseite und Mapbender3 für die GIS-Komponenten.

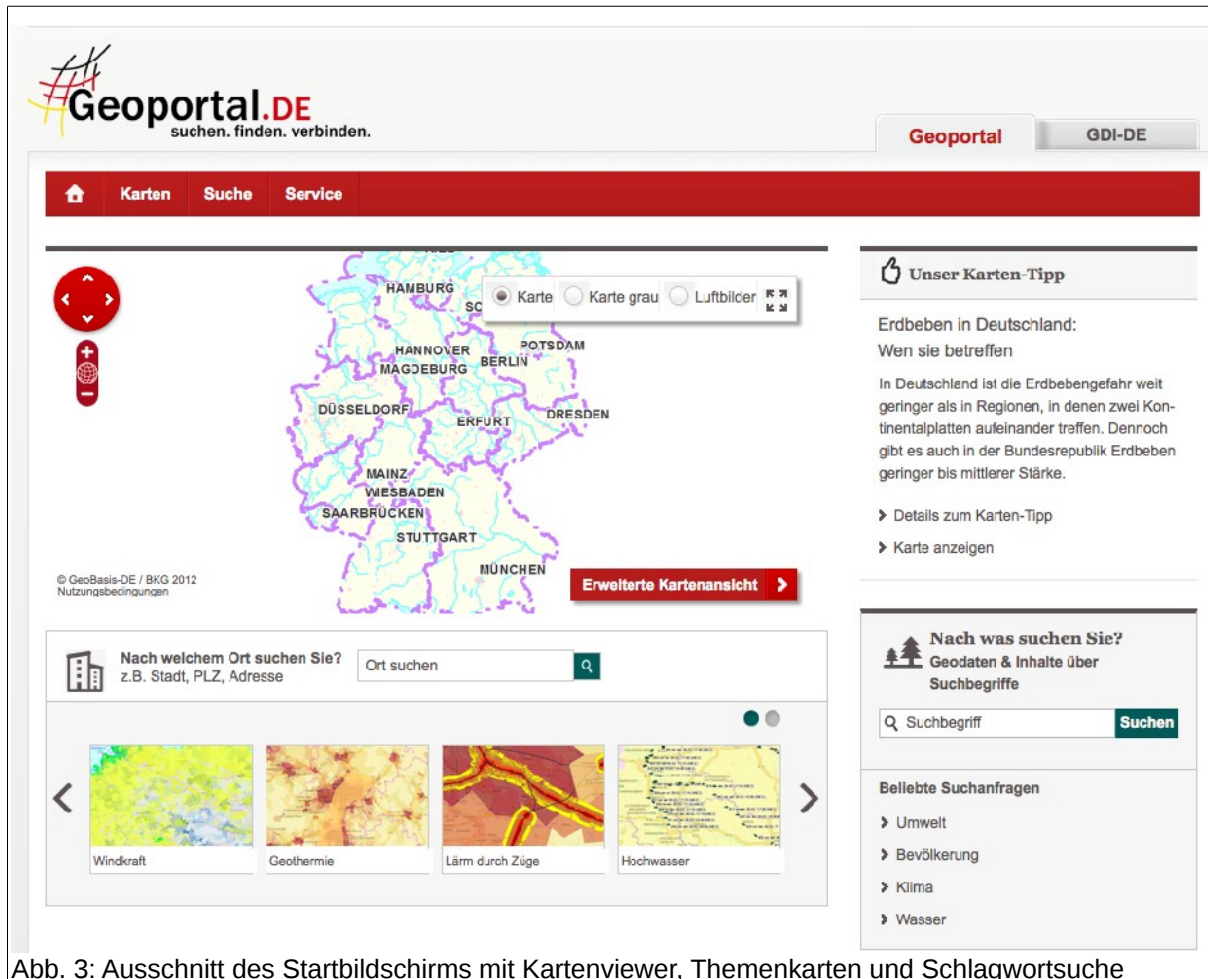


Abb. 3: Ausschnitt des Startbildschirms mit Kartenviewer, Themenkarten und Schlagwortsuche

Grundsätzlich werden zwei Wege angeboten, um Geodaten zu durchsuchen und anzuzeigen. Einen einfachen und schnellen Zugang bieten sogenannte Themenkarten. Diese konfigurierbaren Kartensammlungen in Form von WMC fassen interessante und aktuelle Inhalte zusammen und bieten sie mittels Ein-Klick-Funktion zur Anzeige an. Der schnellste Einstieg dafür wird bereits auf der Startseite angeboten (siehe Abb. 3). Unterhalb eines kleinen Kartenviewers werden verfügbare Themenkarten als Vorschaubilder angeboten. Ein Klick auf das Bild stellt die Daten im Kartenviewer dar.

Der Kartenviewer der Startseite kann zum einen in den Vollbildmodus umgeschaltet werden, zum anderen kann die Kartenanzeige inklusive der geladenen Themenkarte in den zentralen WebGIS-Client, der in Abbildung 4 dargestellt wird, übernommen werden.

Als zweiten, leistungsfähigeren Einstieg dient die oben beschriebene Freitextsuche über den Geodatenkatalog. Suchfelder und -formulare dafür sind an verschiedenen Stellen der Anwendung zu finden, beispielsweise auf der Startseite oder im Kopf jeder Seite.

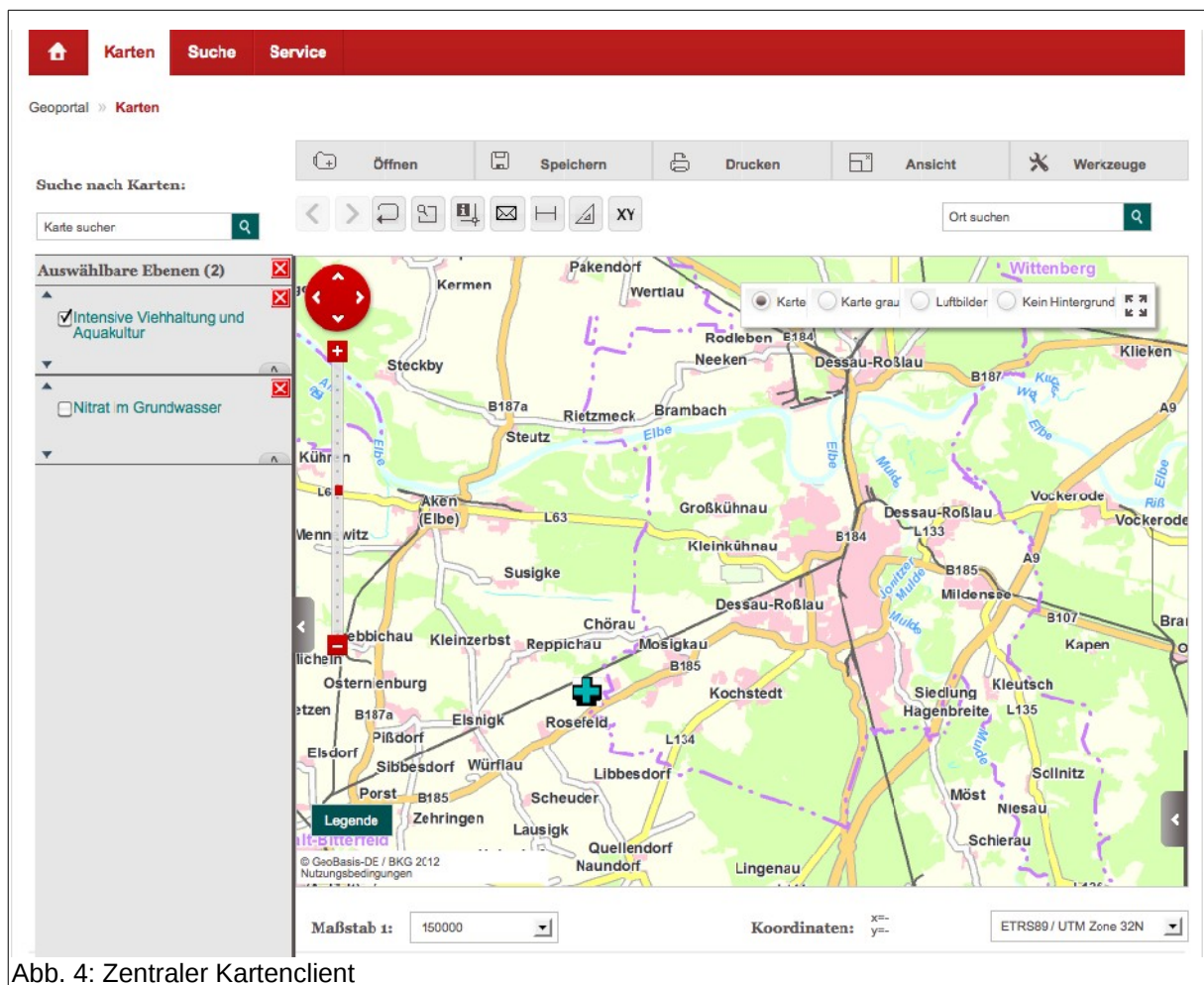


Abb. 4: Zentraler Kartenclient

Der zentrale Kartenclient bildet das Kernstück des Portals. Als Mapbender3-Anwendung dient, wie auch auf dem Startbildschirm, OpenLayers als Kartenclient. Neben den üblichen WebGIS-Funktionen wie Messen, Koordinatenausgabe oder das Speichern der Karte als WMC-Dokument, runden weitere Features den Funktionsumfang ab. Dazu gehören die Ausgabe der Karte als KML zur Anzeige in Google Earth, das Umschalten in den Vollbildmodus oder eine Funktion zum Weiterempfehlen der Karte als Email.

Der Layerbaum stellt ausschließlich Layer dar und verzichtet auf eine Verschachtelung von WMS-Diensten und Gruppen- oder Sublayer. Die Darstellungsreihenfolge der Layer und ihre Transparenz lassen sich per Drag-and-Drop bzw. Schieberegler verändern.

Zur einfachen Navigation in der Karte ist auch hier die Ortssuche eingebunden. Außerdem kann auch aus dem Kartenclient heraus auf den Geodatenkatalog zur Recherche in den Metadaten zugegriffen werden.

Mapbender3

Die Umsetzung der Geokomponenten des Geoportal.DE erfolgte in Absprache mit dem Auftraggeber mit Mapbender3 [5]. Diese Software ist die aktuellste Version der bewährten Mapbender Clientsuite und stellt ein vollständiges Redesign dar.

Mapbender3 basiert auf einer Vielzahl bewährter Frameworks wie Symfony2 (PHP), JQuery/Jquery UI (JavaScript) und OpenLayers (Kartenausgang). Für weitere Details zu Mapbender3 sei an dieser Stelle auf andere Quellen verwiesen (z.B. [6] und [7]).

Fazit

Mit dem Geoportal.DE wurde eine zentrale Zugangsplattform zur GDI-DE geschaffen, die es auch „GIS-Laien“ ermöglichen soll, einen schnellen Zugang zu aktuellen Geodaten zu finden. Viele Ideen der Konzeption und Umsetzung weisen einen neuen, untechnokratischen Weg in der Präsentation raumbezogener Informationen und Daten. Dazu gehören beispielsweise die Konzentration auf Layer ohne den Benutzer mit dem WMS-Konzept zu überfordern, die strikte Umsetzung der Daten-Service-Kopplung mit einer Ein-Klick-Funktionalität für die Visualisierung von Daten oder die komplexe Indizierung mit aktuellen Schlagworten und einer Vorschlagsfunktion.

Wie bereits bei den Vorgängerprojekten Geodatenkatalog-DE und Mapserver ist auch das Geoportal-DE ein gutes Beispiel für eine enge Kooperation zwischen Open Source Projekten und dem Auftraggeber. Ohne das Geoportal.DE hätte Mapbender3 noch nicht den Stand, den er heute hat.

Kontakt zum Autor:

Olaf Knopp
WhereGroup GmbH & Co. KG
Eifelstraße 8
53119 Bonn
0228/909038-27
olaf.knopp@wheregroup.com

Literatur

- [1] http://www.fossgis.de/konferenz/2011/programm/attachments/290_Abstract_gdi-de_Knopp_FOSSGIS_2011.pdf
- [2] www.geoportal.de
- [3] www.government-site-builder.de/
- [4] Kutterer, H., Lenk, M., Luckhardt, T.: Geoportal.DE – ein gemeinsames Geodatenportal von Bund, Ländern und Kommunen, Kartographische Nachrichten, 2012 (noch nicht veröffentlicht)
- [5] www.mapbender3.org
- [6] http://www.fossgis.de/wiki/Intergeo_2011/Vortragsprogramm#Mapbender_3_-_Aktuelle_Entwicklungen_im_Mapbender-Projekt
- [7] Wygoda, C.: Mapbender3 – Next Generation PHP Mapping, FOSSGIS Tagungsband 2012

Geoportal DACH+ - Relaunch mit Drupal, OpenLayers+GeoExt

Dipl.-Geogr. David Arndt

Im DACH+ Grenzraum wird ein Geoportal für die grenzüberschreitende Raumbeobachtung und Raumentwicklung aufgebaut. Die beteiligten Projektpartner 3 Regionalverbände (Deutschland), 9 Kantone (Schweiz), dem Land Vorarlberg (Österreich) und dem Fürstentum Liechtenstein verfolgen dabei folgende Zielsetzungen:

- Integration der vorhandener WMS zur Raumentwicklung und Raumbeobachtung der Projektbeteiligten und anderer Institutionen
- Aufbau eines grenzüberschreitenden Monitorings der Raumbeobachtung und Präsentation der Arbeitsergebnisse im Geoportal
- Integration weiterer Dienste ergänzender Themenbereiche z.B. Schutzgebiete Natur und Landschaft.

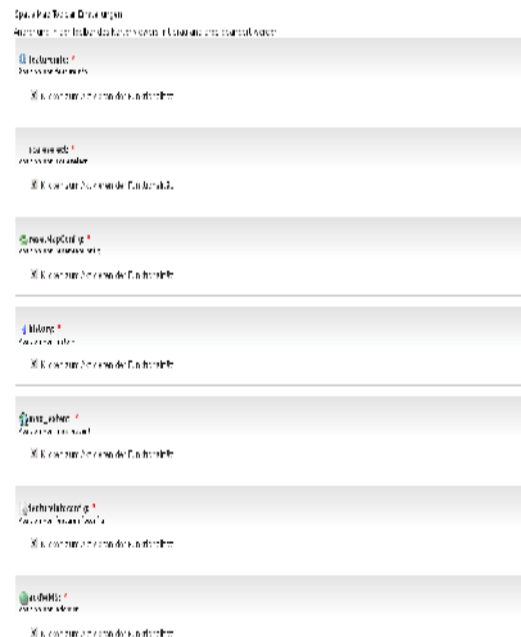


Abbildung 1: Admin-GUI zur Konfiguration der Toolbar

In einer ersten Phase (2006 – 2008) wurde die grundsätzliche technische Umsetzbarkeit eines grenzüberschreitenden Geoportals (unterschiedliche Projektionssysteme, eingesetzte Mapserver, Konfigurationen der WMS) bewerkstelligt. Der 2011 durchgeführte Relaunch hatte die Usability des Geoportals zum Ziel. Neben allgemeinen funktionalen Verbesserungen, war eine der zentralen Anforderungen an den Relaunch die Möglichkeit, bestehende Dienste bzw. einzelne Layer komplexer Dienste unterschiedlicher Herkunft, clientseitig möglichst einfach und dynamisch zu Themen zusammenzuführen und in das Themenangebot im Kartenviewer aufnehmen zu können.

Allein schon die wenig instutionalisierte Trägerstruktur legt den Einsatz von OpenSource-Produkten

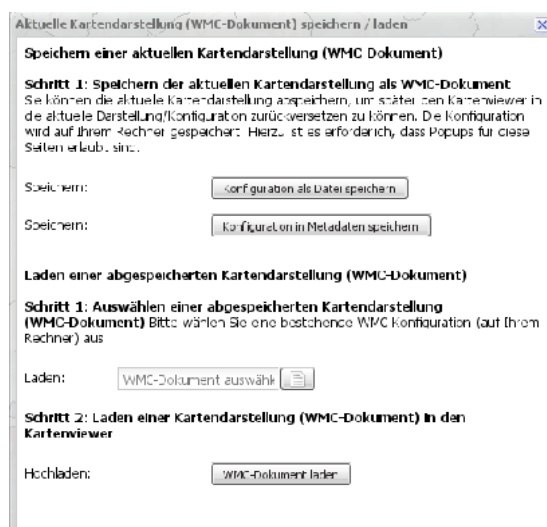


Abbildung 2: Speichern einer aktuellen Ansicht im Metadatenkatalog

nahe. Als integrierende Software der verschiedenen Module wird das CMS Drupal verwendet, welches sich im Geoportal Raumordnung BW in Zusammenspiel mit Openlayers/Mapfish bewährt hat. Allerdings gibt es einige Neuerungen. So wurde die Mapfish-API durch GeoExt ersetzt, und die Kartenkomponente ist jetzt vollständig in Drupal integriert. Diese Integration bietet weitreichende Möglichkeiten in der Ausgestaltung der Kartenvieweroberfläche. So lassen sich die einzelnen Module des Kartenviewers dem Rechtsmanagement unterwerfen. Dies ermöglicht eine zentrale Konfiguration des Gesamtportals, auch des Kartenviewers als Teil des Geoportals mit einem einfachen Client für den „Normalbenutzer“ und einen umfangreicheren Client für Fachanwender mit unterschiedlichem Funktionsumfang und thematischen Angebot.

Geoportal DACH+ - Relaunch mit Drupal, OpenLayers+GeoExt

Im Mittelpunkt des Kartenviewers steht die Funktionalität vorgefertigte Karten (sogenannte Themen) in den Kartenviewer einzuladen, die es auch ungeübten Nutzern ermöglichen soll einen Einblick in die Daten des DACH+-Projekts zu bekommen. Diese thematischen Zusammenstellungen können client-seitig zusammengestellt, als WMC gespeichert und im Metadatenkatalog GeoNetwork als ISO-konforme Metadaten gehalten werden. Der Einsatz des CMS Drupal als Metadateneditor ermöglicht die Erfassung weiterer Elemente zur Beschreibung der zusammengeführten Dienste/Layer und zur Konfiguration der Themen im Kartenviewer.

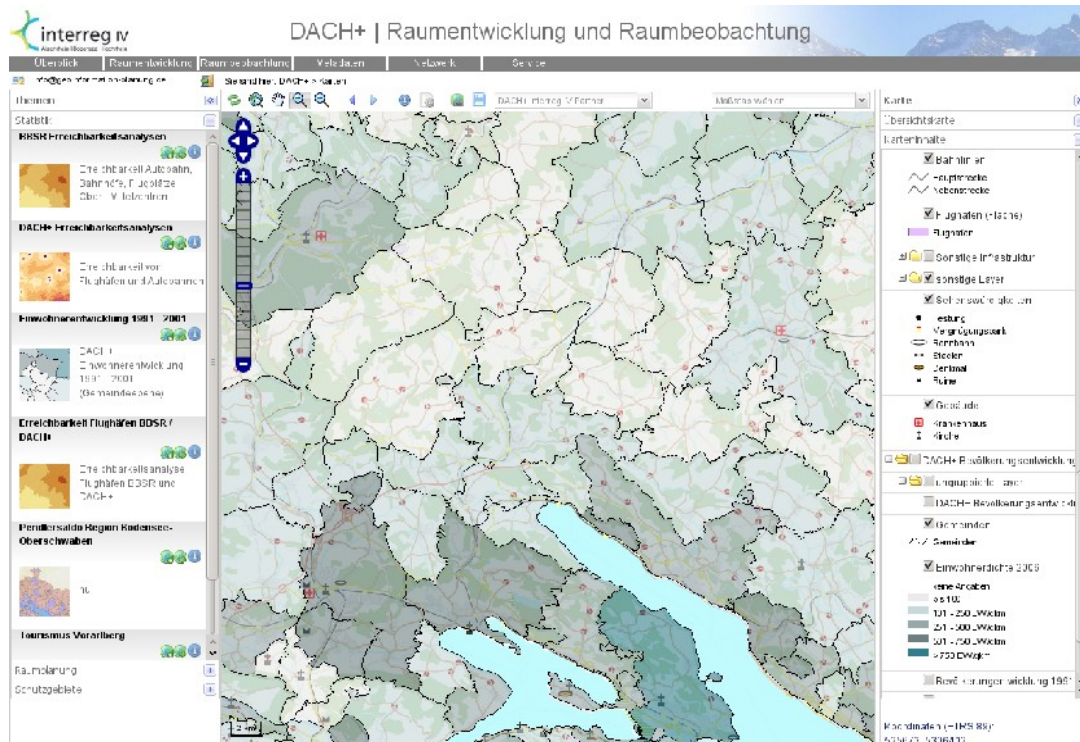


Abbildung 3: Neuer Kartenviewer

Der Vortrag fokussiert den Einsatz von OpenLayers und GeoExt im Zusammenspiel mit Drupal für eine dynamische Themengenerierung und Übergabe an den Kartenviewer. Zudem wird auf die Metadatenhaltung im CMS Drupal und Übergabe der Metadaten an den Metadatenkatalog GeoNetwork eingegangen.

Kontakt zum Autor:

Dipl.-Geogr. David Arndt
geoinformation+planung
Gölzstraße 22
72072 Tübingen
arndt@geoinformation-planung.de

Wie wird das Kartenportal sichtbar für die Suchmaschinen?

Untertitel: Eine Symbiose von WebMapping und Content-Management-System (CMS).

Anja Sigesmund, Christoph Lauber

Technische Komponenten: PostgreSQL mit PostGIS, MapServer, MapFish, OpenLayers, Typo 3

Onlinere Ressourcen: www.regiofreizeit.de, www.chemieatlas.de, www.energieatlas.org

Ausgangslage:

Der Kreis Recklinghausen betreibt mehrere Kartenportale, diese werden in regelmäßigen Abständen inhaltlich und technisch aktualisiert. Hierzu gehört u.a. das WebMapping-Portal „www.regiofreizeit.de“, in dem freizeitrelevante Informationen aus elf Städten im Ruhrgebiet präsentiert werden. Um die Nutzerfreundlichkeit zu verbessern wurde 2011 die Kartenanwendung vollständig auf MapFish und OpenLayers umgestellt und erweitert. Die Besucherzahlen veränderten sich trotz der Umgestaltung des Angebotes nur geringfügig und auch die Suchmaschinenplatzierung war relativ schlecht.

Dies ließ sich u.a. dadurch erklären, dass das Kartenportal nur eine Hauptseite enthielt, die gleichzeitig als Homepage fungierte und zu verschiedenen Themen in der Karte verlinkte, aber nahezu keinen indizierbaren Inhalt enthielt. Die Datenbankinformationen, die zu den interessanten Orten, den „Points Of Interest“ (POIs) vorlagen, konnten ausschließlich über die Karte erreicht werden und waren für Suchmaschinen nicht „sichtbar“. Für die Nutzer ergab sich der Nachteil, dass bestimmte Informationen nur über die Karte gesucht werden konnten und eine Freitextsuche von der Homepage aus nicht möglich war.

Um die Bekanntheit des Kartenportals zu vergrößern und die Inhalte einem größeren Nutzerkreis zugänglich zu machen, sollten nun auch die Haupt- und Unterseiten umstrukturiert und für die Suchmaschinen optimiert werden.

Überlegungen:

- Wie kann der „Karten“-Content aus der Datenbank so aufbereitet werden, dass er indizierbar für die Suchmaschinen wird?
- Wie können die Informationen medial so bereitgestellt werden, dass sie für die Nutzer übersichtlicher und intuitiver auffindbar sind?
- Wie können die Nutzerzugriffe gesteigert werden?
- Wie kann man ohne Programmierkenntnisse für die Pflege der Haupt- und Unterseiten befähigt werden?

Technische Infrastruktur & Umsetzung:

In technischer Hinsicht bestand die Anwendung bis dato aus einer PostgreSQL-Datenbank zur Verwaltung der Geodaten sowie dem Apache Webserver, um diese Informationen über das Netz zu publizieren. Die Erzeugung der geographischen Informationen übernahm der UMN MapServer. Um innerhalb des CMS auf die Geodaten inkl. deren Metadaten zugreifen zu können, galt es, die Verbindung zwischen Typo3 und der PostgreSQL-Datenbank sicherzustellen.

Durch Tests zeigte sich, dass die Performance der Kombination PostgreSQL/Typo3 sehr gering war, so dass Alternativen gesucht werden mussten. Aufgrund der guten Anbindung von MySQL an Typo3 wurde diese als Datenbank gewählt und ein System zur Synchronisierung der beiden Datenbanken in-

Wie wird das Kartenportal sichtbar für die Suchmaschinen?

tegriert. Hierbei werden sämtliche Datenbankinhalte zeitgesteuert in einen HotFolder exportiert und von dort in die CMS-Datenbank importiert (siehe Abb. 1- Technische Infrastruktur).

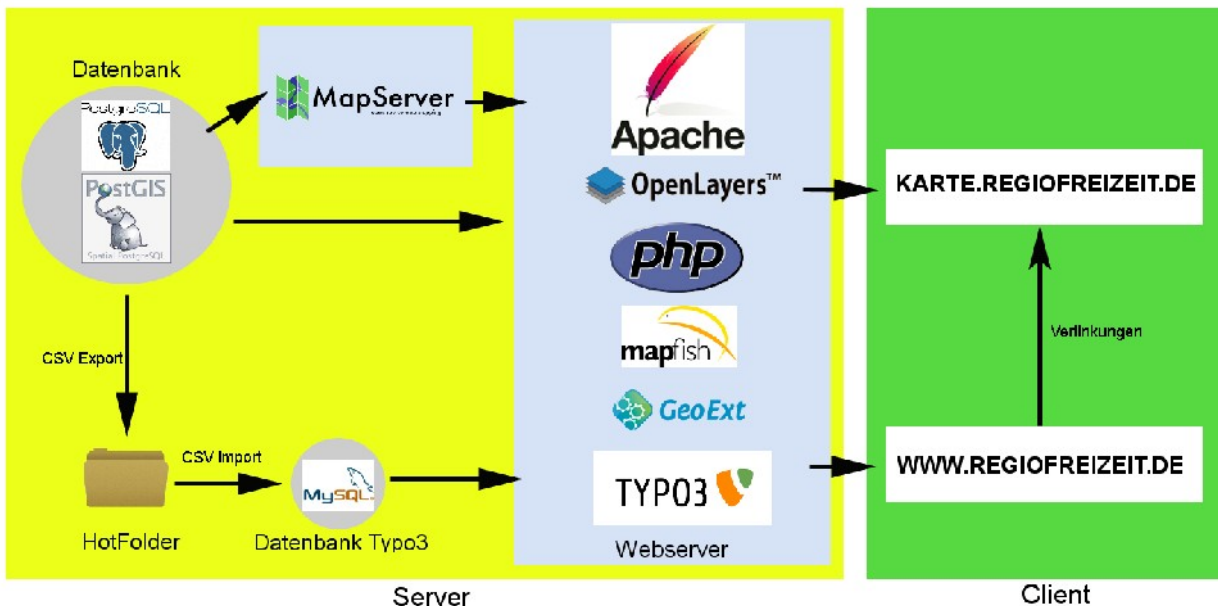


Abbildung 1: Technische Infrastruktur (Quelle: eigene Darstellung)

Neben dem Ziel, die Datenbankinhalte in die Unterseiten zu integrieren, war die Online-Pflege der Datenbankinhalte weiterhin zu gewährleisten. Daher war es notwendig, die bestehende Datenbank samt der zugehörigen Backendmasken als „führendes System“ zu erhalten.

Erweiterungen sind auch ohne Änderungen am Datenmodell möglich, indem auf den Ausgabeseiten Inhalte ergänzt werden. Dies schließt Texte, Bilder, zusätzliche Dateien und Bewegtbilder ein. Da die Daten im CMS nicht zwingend über ihren Raumzusammenhang dargestellt werden müssen, ergeben sich alle Vorteile der redaktionellen Aufarbeitung. Die Daten können unterschiedlich ausgelesen und sortiert werden, je nach Art der Anwendung. Zusätzlich können verschiedene Vorlagen zur Darstellung der Daten eingebunden werden. Der Bezug zur Kartenanwendung wird durch Links gewährleistet, über die die relevanten Abfrageparameter an den Mapserver übertragen werden.

Entwicklungsprozess

Die Objekte aus der Datenbank sollten mit ihren hinterlegten Informationen auf Unterseiten dargestellt werden. Der Umfang der Themengebiete in der Kartenanwendung ließ es sinnvoll erscheinen, die Inhalte mit Hilfe von verschachtelten Unterseiten zunächst zu gliedern. Hieraus ergab sich ein „3-schichtiger“ Aufbau.

In Abbildung 2 wird der Aufbau der Seiten schematisch dargestellt. Der Nutzer gelangt zunächst auf eine Startseite, wo eine Übersicht über die in der Anwendung enthaltenen Kategorien gegeben wird. Nach Auswahl einer Kategorie gelangt er auf die erste Unterseite, hier werden die Themen, welche in der Kategorie enthalten sind, aufgelistet. Je nach dem, ob zu den Objekten in einem Thema weitere Informationen vorliegen, gelangt der Nutzer nach Auswahl eines Themas entweder von hier direkt zur Kartenanwendung oder er gelangt zu einer zweiten Unterseite, auf der ausgewählter Datenbankinhalt aufgelistet ist.

Wie wird das Kartenportal sichtbar für die Suchmaschinen?

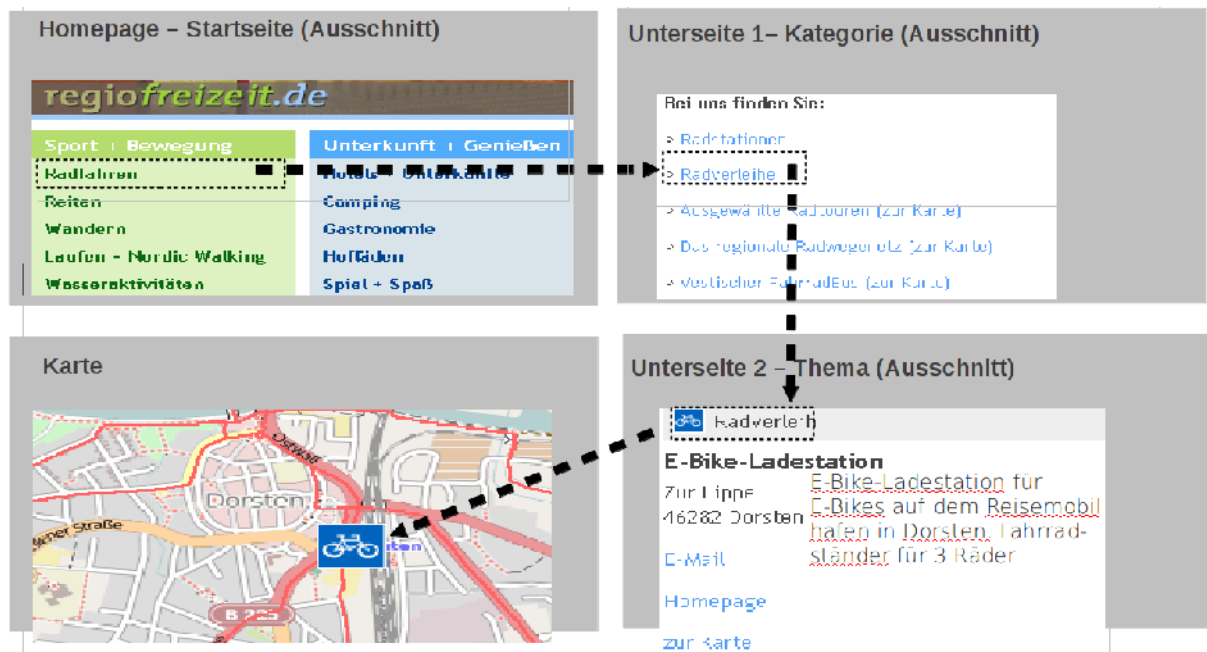


Abbildung 2: Schematische Darstellung der Haupt- und Unterseiten von www.regiofreizeit.de (Quelle: eigene Darstellung)

Durch die beschriebene Aufteilung konnte nicht nur eine Vervielfachung des dargebotenen Inhalts erreicht werden, sondern auch eine Verbesserung der Übersichtlichkeit für den Nutzer. Durch die Darstellung von Detailinformationen zu den Objekten direkt auf der Website waren Adressen und Objektbeschreibungen für die Suchmaschinen indizierbar und für den Nutzer ohne vorhergegangene Suche in der Karte verfügbar.

Ergebnis:

Die Webseiten wurden mit Unterseiten ausgestattet und von Grund auf neu gestaltet (s. Abb. 3). Der Datenbankinhalt wurde über das CMS in übersichtlicher Form dargestellt. Das Backend wurde für die redaktionelle Eingabe optimiert und im Layout an das Frontend angepasst (s. Abb. 4). Die Auffindbarkeit über die Suchmaschinen wurde deutlich verbessert, die Zugriffszahlen mehr als verdoppelt. Zudem ist es nun für den Nutzer möglich, die Inhalte der Datenbank auch direkt auf der Webseite zu durchsuchen und somit zielgerichtet die gewünschten Informationen zu erhalten ohne den „Umweg“ über die Karte zu gehen.

Wie wird das Kartenportal sichtbar für die Suchmaschinen?

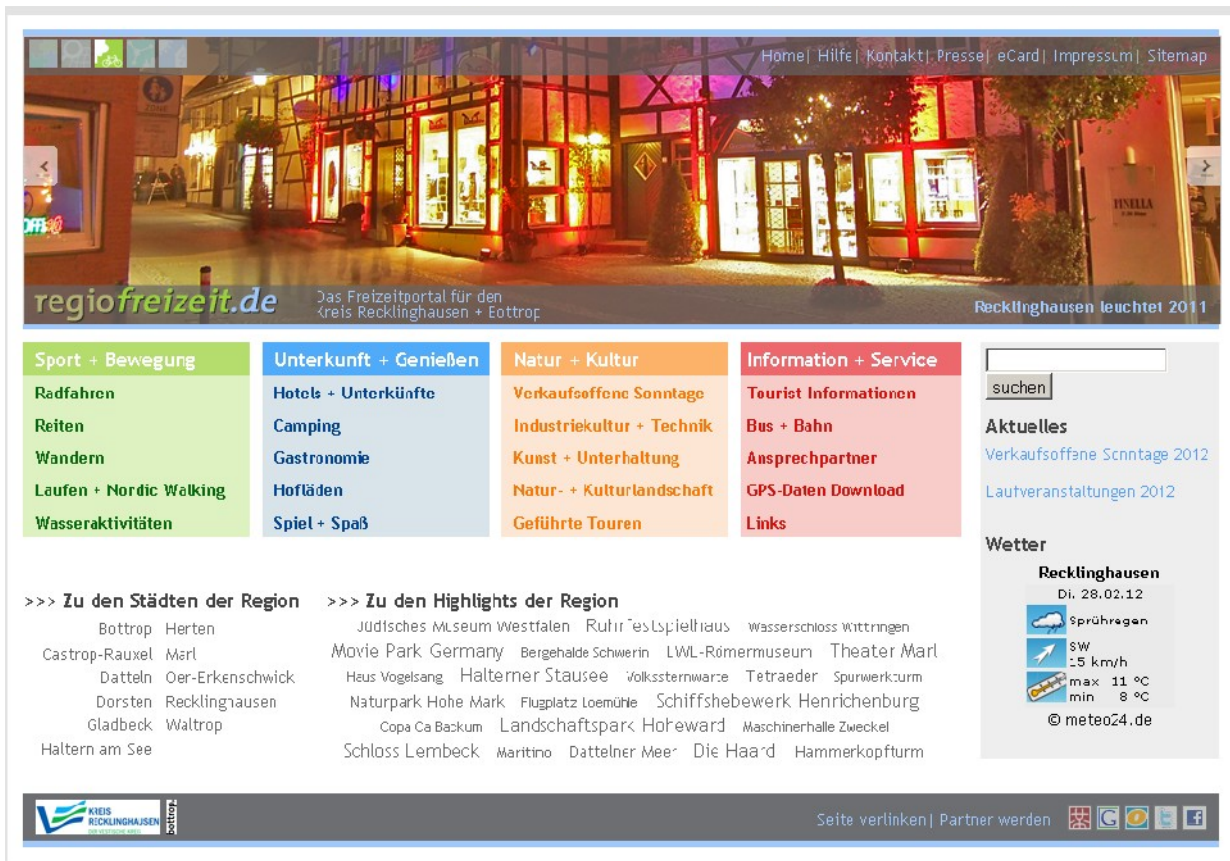


Abbildung 3: Startseite www.regiofreizeit.de

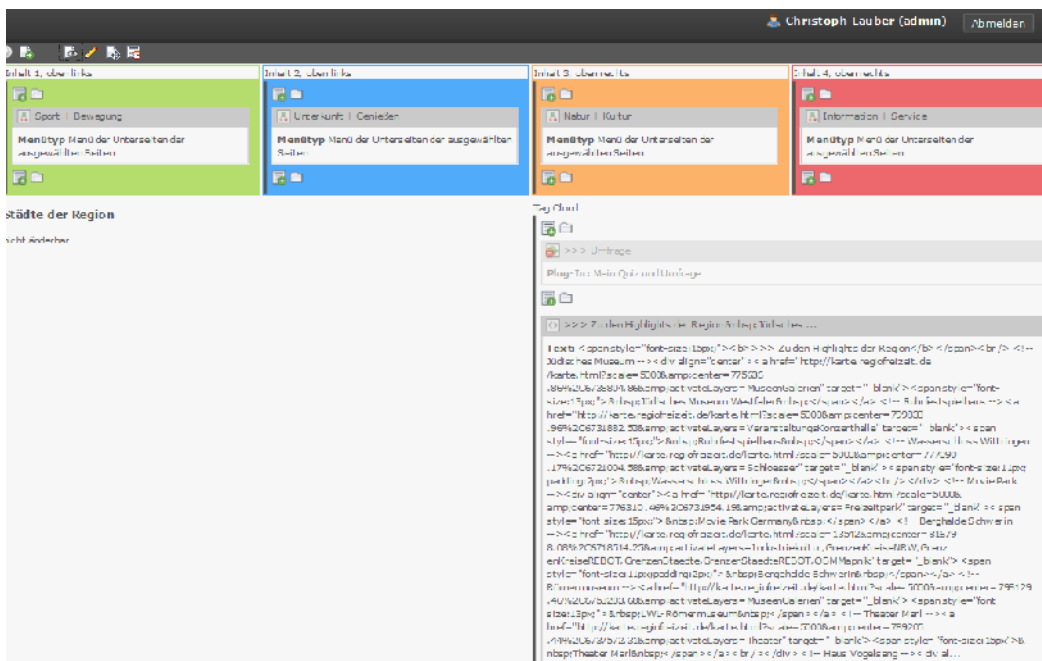


Abbildung 4: Pflege der Seiten über das Backend des CMS

Wie wird das Kartenportal sichtbar für die Suchmaschinen?

Fazit:

Die Umstellung der Seiten auf das neue System sowie die Anpassung der Inhalte und des Layouts der Webpräsenz kann als Erfolg gewertet werden. Sowohl das Ziel der Verbesserung der Suchmaschinenplatzierung als auch die Steigerung der Besucherzahlen konnte in kurzer Zeit erreicht werden. Hinzu kommt, dass sich die Übersichtlichkeit für die Besucher der Seite verbessert hat und eine leichtere Bearbeitbarkeit für die Redakteure resultierte.

Die Technische Umsetzung verlief nicht reibungslos, es war z.B. nicht möglich die bestehende PostgreSQL-Datenbank –wie geplant– als Datenquelle für das CMS zu verwenden, da sich die Performance beim Seitenaufruf extrem verschlechterte. Als Ausweg wurde eine MySQL-Datenbank aufgesetzt, welche laufend mit der eigentlichen Datenbank synchronisiert wird. Nachteilig ist, dass Änderungen in der Datenbank nur zeitverzögert im CMS sichtbar werden. Da die Verzögerung jedoch maximal 24 Stunden beträgt, ist dies zu vernachlässigen.

Da es sich bei Typo3 um ein CMS mit zahlreichen Funktionalitäten handelt, wurde für die Redakteure und Administratoren eine eintägige Schulung durchgeführt und mittels Wartungskontingent sichergestellt, dass Fragen zum CMS vom Fachmann beantwortet werden können. Vergleicht man dies jedoch mit dem Aufwand der notwendig ist, um Mitarbeiter (ohne Programmier-/ HTML-Kenntnisse) zur selbstständigen Erstellung komplexer Websites zu befähigen, ist dieser wesentlich geringer.

Allgemein wurde das System trotz der Erweiterung der Inhalte nicht verkompliziert. Viele Inhalte der Seite werden automatisch aus der Datenbank erstellt, zudem führt eine klare Trennung von administrativen Aufgaben und redaktioneller Arbeit dazu, dass auch Personen ohne spezielle Fachkenntnisse dazu befähigt sind, die Inhalte der Anwendung zu pflegen. Zudem ist es nun möglich, im CMS mit sehr wenig Aufwand neue Domains zu erstellen oder die vorhandenen zu erweitern oder zu individualisieren.

Kontakt zu den Autoren:

Anja Sigesmund	Christoph Lauber
Kreisverwaltung Recklinghausen, FD 18.3	virtual value
Kurt-Schumacher-Allee 1	In der Hohl 7
45657 Recklinghausen	35043 Marburg
Tel.: +49 2361 534500	Tel.: +49 6421 9683490
	Tel.: +49 157 71432425
E-Mail: as@kreis-re.de	E-Mail: c.lauber@virtual-value.de

[1] <http://www.regiofreizeit.de>

[2] <http://www.virtual-value.de/virtual-value.html>

[3] <http://www.chemieatlas.de/>

[4] <http://www.energieatlas.org/>

HTML5-Editor für OpenStreetMap

Paul-Fiete Hartmann

Der Markt für OSM-Editoren scheint recht gut abgedeckt: Potlatch bietet ein schlüssiges Bedienkonzept, JOSM besticht als Offline-Editor mit vielen Funktionen und Merkaartor findet als native Anwendung seine Anhänger. Dazu kommen Apps für mobile Geräte. Eine Nische, die es noch auszufüllen gilt, ist ein Editor, der ohne Java und Flash als JavaScript Anwendung im Browser funktioniert.

Für einen reinen HTML-Editor sind Webtechniken wie das Canvas-Element (zur Erzeugung dynamischer Grafiken) und SVG (Skalierbare Vektor-Grafik) entscheidende Voraussetzungen. Ein Problem war bisher die mangelnde Unterstützung durch einzelne Browser-Hersteller, aber Besserung ist in Sicht, da die aktuellen Versionen der verbreitetsten Browser mittlerweile die benötigten Funktionen bereitstellen.

In diesem Vortrag möchte ich einen Prototyp vorstellen, der sich im frühen Entwicklungsstadium befindet, aber bereits die prinzipielle Machbarkeit beweist. Unter anderem ist der komplette Editier-Zyklus implementiert: Daten lassen sich vom OSM-Server Herunterladen, Rendern, Bearbeiten und wieder Hochladen. Technisch basiert dieser Ansatz wesentlich auf diversen bestehenden Projekten: Teile des JOSM-Quelltexts werden mit Hilfe des Google Web Toolkits nach JavaScript übersetzt und mit einer HTML/JavaScript Benutzeroberfläche angesteuert. Für die Anzeige von Luftbildern im Hintergrund kommt OpenLayers zum Einsatz.

Video-Mapping / Des Mappens letzter Schluss?

Matthias Meißer

Die Erstellung eines detaillierten globalen Geo-Datensatzes, ist ein Vorhaben, dass sich das Open-StreetMap (OSM) Projekt genauso wie die kommerzielle Anbieter verschrieben hat. Navteq und Teleatlas nutzen gleichwohl ganze Fahrzeugflotten, um das weltweite Straßennetz optisch zu erfassen [1] und auch Google Maps schickt im Rahmen des „Google Streetview“-Programmes, eigene mobile Erfassungssysteme durch das ganze Land. Diese Erweiterungen gewöhnlicher PKWs, umfassen neben optischer Sensoren wie (Panorama-)Kameras und Lidar, insbesondere auch Differentielles GPS, Inertiale Navigationssysteme und weitere Sensorik zur Bestimmung der lokalen Bewegungsvektoren [2]. Dies Zusammenspiel ermöglicht es mit einem minimalen Personaleinsatz, in kürzester Zeit automatisch eine Vielzahl von Daten zu sammeln, die für die Unternehmen Grundlage für verschiedenste Produkte darstellen. So können dank Nahbereichs-Photogrammetrie interaktive Panoramafotos oder 3D Stadtmodelle erstellt, sowie natürlich die Erfassung des Verlaufs, Bebauung und des Zustandes von Straßen dokumentiert werden. Eine Auswertung der gesammelten Informationen erfolgt hier meist off-board und nur semiautomatisch durch externe Dienstleister im Ausland.

Die Datenerfassung im Rahmen von Volunteered Geographic informations (VGI), folgt prinzipbedingt jedoch anderen Regeln. So kostet ein Besuch eines einzelnen Mappers global gesehen de facto kein Geld. Denn auch wenn es natürlich wünschenswert ist möglichst viele Details in einem Durchgang zu erfassen, so besteht innerhalb der Projekte doch die Überzeugung, dass eine Gegend auch regelmäßig aufgesucht werden muss, um auf lokale Veränderungen zeitnah reagieren zu können. Da das Produkt im Falle von OSM lediglich reine Geodaten sind, muss auch kein all zu hoher technischer Aufwand betrieben werden, um photogrammetrisch auswertbare Informationen zu erfassen. Um die Massenbewegung des Projektes darüber hinaus nicht zu gefährden, ist es außerdem nötig, dass Geräte für eine Erfassung preiswert zu beschaffen und die Verfahren alltagstauglich sind. Bisher wurden dafür ausschließlich Digicams oder Voice-Recorder genutzt, deren Aufzeichnungen mit Hilfe der mitgeführten GPS Empfänger georeferenziert wurden. Alternativ reicht in Gebieten mit gut ausgestatteter OSM Datenbasis auch bereits ein Kartenausdruck, welcher dann händisch mit Stiften ergänzt wird. All diesen Verfahren ist jedoch gemein, dass man jedes zu erfassende Objekt speziell fokussieren muss um es zu erfassen, was eine schnelle flächendeckende Erfassung natürlich unmöglich macht. Ausgehend von den Erfahrungen der professionellen GIS-Branche, wurde deshalb untersucht, inwiefern Video-Technik auch für OpenStreetMap nutzbar gemacht werden kann, denn dies würde die Vorteile des digitalen Fotos (hoher Detailreichtum) mit denen des Diktiergerätes kombinieren (durchgängige Dokumentation) [3].

Da für normale Verbraucher nur Camcorder des Consumer Segmentes zu Verfügung stehen, wurden verschiedene Modell von Herstellern angefordert und miteinander verglichen. Diese Auswahl führt zu einigen Einschränkungen bezüglich Qualität der Optik und der Videokompression. Grob gliedert sich das Feld in Helmkameras, Pocket-Camcorder sowie gewöhnliche Hand-Camcorder, die sich jedoch nur in wenigen technischen Details unterscheiden (Öffnungswinkel, Videoformat, ...) und deren gemeinsame Einschränkung die geringe Akku-Kapazität darstellt, welche für kurze mehrstündige Ausflüge jedoch ausreichend ist. Um die Eignung des Equipments für die einfachen Vermessungsarbeiten des Projektes zu testen, wurden verschiedene Teststrecken in der Rostocker Innenstadt untersucht. Diese waren so gewählt, dass einerseits ein gleichbleibender Abschnitt einen Vergleich der Geräte zuließ, auf der anderen Seite wurden bewusst spontan und an unterschiedlichen Orten gedreht. Dies ermöglichte es die Grenzen des Verfahrens aufzuzeigen, sowie unterschiedliche Transportmethoden und Objektdichten zu testen. Die Montage erfolgte dabei so möglich mit einem Stativ, andernfalls erfolgte eine Führung durch die Hand. Eine Sichtung der Ergebnisse erfolgte mit gewöhnlichen Videoplayern und die dargestellten Objekte wurden parallel mit dem OSM Editor JOSM eingearbeitet. Es zeigte sich dabei, dass ein enorm hoher Detailgrad für die meisten für OSM relevante Kategorien aufgezeichnet werden konnte (StVo, Geschäfte, Wegqualität, Einschränkungen, ...) und eine Bewegung mit ÖPNV oder Fahrrad keine Probleme bei der Sichtung bereiteten, während die Geschwindigkeiten von PKWs oder Zügen durchaus kritisch sein kann und es zu Unschärfe kommen kann. Insbesondere

Video-Mapping / Des Mappens letzter Schluss?

zeigte sich auch das mangelnde Zusammenspiel von Editor und Videosoftware, so dass es häufig zu Fehlbedienungen und fehlender Orientierung kam.

Um diese Probleme zu beheben, wurde eine Erweiterung des Editors konzipiert, die ähnlich der bestehenden Funktionalität zur Georeferenzierung von Audio-Dateien, die Wiedergabe eines Videos mit dem GPS Track synchronisiert. In dem neu entstandenen „Videomapping“-Plugin ist nach einer Synchronisation des Videomaterials mit dem GPS Spur (etwa durch filmen des Empfängers mit aktueller Zeit) eine Steuerung der Videowiedergabe durch klicken auf den Track möglich, so dass nach Objekten gezielt recherchiert werden kann. Außerdem ist es mit Tastenkürzeln möglich den Player ohne das Verlassen des Editors zu steuern, was sich in den Tests ebenfalls als sehr hinderlich herausstellte [4].

Man kann zusammenfassen, dass das Verfahren die Erwartungen an die Aufzeichnung eines hohen Informationsflusses durchaus gerecht wird und die Zeit zur Extraktion von möglichen Map features durch die neu entwickelte Erweiterung des Editors gesenkt werden konnte. Dennoch bleibt das Eintragen der Objekte mit dieser Technik sehr zeitintensiv, da viel Material gesichtet werden muss, wobei sich herausstellte das in etwa das zwei bis dreifache der Videolänge für die Auswertung veranschlagt werden muss. Da das Verfahren derzeit den geringsten Informationsverlust unterliegt, ist es sogar möglich, dass zwei völlig verschiedene Personen die Daten im Feld erfassen und später zu OpenStreetMap eintragen. Es könnte also eine Lösung für Hakley's Problem „Tyrannei des Ortes“ darstellen [5]. Die Qualität der Kamera ist hierbei nicht zwingend entscheidend, auch wenn die Untersuchungen zu dem Schluss kommen, dass all zu preiswerte Geräte vermieden werden sollten, wobei jedoch bereits heutige Smartphones eine ausreichende Bildqualität liefern. Ein limitierender Faktor stellt dabei der Sichtbereich da, so dass vor Ort ein Kompromiss zwischen dem Abstand zum Objekt und der damit einhergehenden Auflösung von Details gefunden werden muss. Angebracht ist das Verfahren in erster Linie für Szenarios in denen die Informationsdichte nur schwerlich mit bisherigen Techniken erfasst werden kann, wie etwa Hausnummern oder Einkaufsstraßen mit einer Vielzahl von Geschäften. Inwiefern die OSM Community bereit ist die verkürzte Zeit während der Exkursion gegen den nicht unerheblichen zeitlichen Aufwand bei der Auswertung am PC zu investieren, muss sich erst noch erweisen.

Kontakt zum Autor:

Matthias Meißer
Universität Rostock
Max Planck Straße 2a
0162 706 1305
matthias.meisser@uni-rostock.de

Literatur

- [1] *Monika Rech*, : GisBusiness, NAVTEQ investiert in Daten , Berlin, 2010.
- [2] *Ralf Bill*, : Grundlagen der Geo-Informationssysteme , Rostock, 2010.
- [3] *Matthias Meißer*, : Videomapping im OpenStreetMap Umfeld , Rostock, 2010.
- [4] <http://wiki.openstreetmap.org/wiki/JOSM/Plugins/VideoMapping>
- [5] *Mukki Hakley*,: The Tyranny of Place and OpenStreetMap , Girona, 2010

Geocoding mit OpenStreetMap

Marcel Radischat

Geocoding behandelt das Parsen einer Adresse oder Ortsbeschreibung und die Zerlegung der Adresse nach bestimmten Kriterien, sodass innerhalb eines Referenzdatensatzes nach diesen Kriterien gesucht werden kann. Jeder Adresse wird eine bestimmte Position im Raum gegeben, dabei erfolgt die Positionsangabe durch Koordinaten in einem Bezugskoordinatensystem. Mit dem Prozess des Geocodings werden geographische Informationen gesucht, die die Orte oder Adressen am besten beschreiben, damit diese letztlich auf einer Karte angezeigt werden können oder selbst wieder Bestandteil räumlicher Analysen sind.

Kommerzielle Kartenanbieter wie Google oder Microsoft bieten hierfür kostenlose APIs an. Eine Verwendung dieser Such-APIs mit eigenen Karten oder im kommerziellen Umfeld ist allerdings durch die Nutzungsbedingungen dieser Dienste nur eingeschränkt möglich. Abhilfe schaffen hierfür die OpenStreetMap-Daten, die als Grundlage für eigene Geocodierungsdienste genutzt werden können.

Es gibt bereits diverse Anbieter, die eine Geocodierung mit OpenStreetMap-Daten durchführen und deren APIs kostenlos verwendet werden können. Problematisch daran ist, dass keine eigenen OpenStreetMap-Daten benutzt werden, sondern man immer auf die jeweiligen OpenStreetMap-Daten des Anbieters zurückgreift.

Abhilfe schafft hier das Projekt Nominatim. Es ermöglicht den Einsatz eines freien und offenen Geocodierers für eigene Karten auf Grundlage von OpenStreetMap-Daten, die je nach Verwendungszweck individuell bearbeitet werden können. Zudem wird Nominatim auch auf der OpenStreetMap-Website als Suchfunktion für Adressen, Orte und Ähnlichem offiziell seit 2010 eingesetzt.

Nominatim gewährleistet eine schnelle Suche und bietet diverse Ausgabeformate wie JSON, XML und HTML, sodass die Suchergebnisse auch für weitere Analysen eingesetzt werden können. Zudem bietet Nominatim die Eingabe der Suchbegriffe, durch Kommata getrennt, in ein sogenanntes Freiform-Adress-Feld an. Diese Begriffe werden anschließend mittels einer exakten Suche in der Datenbank gesucht.

Dennoch ist Nominatim nicht für kleinräumigere Gebiete/Karten geeignet. Denn die Installation der Software und die Indizierung der OpenStreetMap-Daten nimmt viel Zeit in Anspruch. Laut der Installationsanweisung aus dem Nominatim-Wiki benötigt ein Computer mit 12 Rechenkernen, 32GB RAM und Standard SATA-Festplatten 20 Stunden für den initialen Import der kompletten OpenStreetMap-Daten und weitere 250 Stunden für deren Indizierung.

Auch die scharfe Suche für die Geocodierung kann problematisch für den Nutzer sein, da einerseits Daten in der Datenbank falsch benannt sein können oder der Nutzer sich bei der Eingabe der Begriffe verschreiben kann und somit keine Ergebnisse gefunden werden.

Eine neue Möglichkeit für einen Geocodierungsdienst für eigene Karten anzubieten, wird durch den Imposm Geocoder bereitgestellt. Diese Anwendung basiert



Abb. 1: Erfolgreiche Suche mit Nominatim

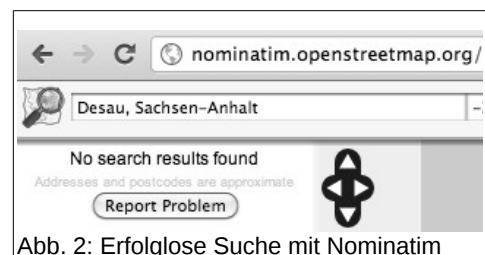


Abb. 2: Erfolglose Suche mit Nominatim

Geocoding mit OpenStreetMap

auf den sehr schnellen und individuell anpassbaren Import der OpenStreetMap-Daten in eine Geodatenbank durch das Import-Tool Imposm.

Der Geocoder entstand in einer Bachelorarbeit bei dem Unternehmen Omniscale GmbH & Co. KG aus Oldenburg. Das Ergebnis dieser Arbeit wurde als Open-Source Bibliothek bereitgestellt und ist unter der Apache Licence 2.0 veröffentlicht.

Der Geocodierungsprozess des Imposm Geocoders muss dabei durch verschiedene Suchmethoden realisiert werden, da nicht überall die gleiche Qualität und Detailreichtum an OpenStreetMap-Daten vorhanden ist. Hinzu kommt, dass die OpenStreetMap-Daten lediglich eine geographische Information besitzen, nämlich die Koordinaten, wo sich das jeweilige Objekt befindet. Anhand dieser Koordinaten müssen weitere geometrische Verschneidungsoperationen und Distanzmessungen durchgeführt werden, die zur Informationsgewinnung dienen. Diese Methoden bewerkstelligen letztlich, dass eine gesuchte Straße auch dem korrektem Ort zugewiesen werden kann, da die OpenStreetMap-Daten keine topologischen Informationen auf vertikaler Ebene aufweisen.

Der Imposm Geocoder bietet folgende Merkmale:

- schneller Datenimport und nur eine sehr geringe Datenaufbereitung
- geringerer Speicherplatzverbrauch als Nominatim
- Erweiterung von PostgreSQL/PostGIS Funktionen, um Straßensegmente zu einem Element zusammenzufügen
- unscharfe Suche nach Adressen, sodass falsche Schreibweisen berücksichtigt werden
- als Bibliothek entwickelt und somit vielseitig einsetzbar
- leicht erweiterbar



Abb. 3: Beispielhafte Einbindung der Bibliothek in eine Webanwendung und Darstellung der Suchergebnisse

Durch einen direkten Vergleich von Nominatim und dem Imposm Geocoder in den Bereichen Datenimport und -aufbereitung, Speicherplatzbedarf, Ergebnisse und Geschwindigkeit der Ergebnisberechnung ist eine qualitative Beurteilung möglich. Lediglich durch die lange Indizierung der Daten durch Nominatim ist die Ergebnisberechnung schneller als bei dem Imposm-Geocoder, der allerdings bei allen anderen Kriterien besser oder mindestens gleich gut abgeschnitten hat.

Geocoding mit OpenStreetMap

Letztlich stellt der Imposm-Geocoder eine Bibliothek dar, mit der eine Geocodierung von Adresse, Straßen, Orten, Postleitzahlen und Ländern möglich ist.

Kontakt zum Autor:

(B. Sc.) Marcel Radischat
Omniscale GmbH & Co. KG
Nadorster Straße 60, 26123 Oldenburg
0441/93927740
radischat@omniscale.de

Literatur

[1] Radischat, Marcel: Geocoding mit OpenStreetMap-Daten – Analyse, Konzeption und Entwicklung von Methoden zur Suche in Geodaten. Oldenburg. 2012

Synchronisation zwischen OpenAddresses und OpenStreetMap

Hans-Jörg STARK

Zusammenfassung

OpenAddresses (OA) hat zwei wesentliche Neuerungen, die in diesem Beitrag vorgestellt werden. Zum einen wird das Qualitätssicherungskonzept vorgestellt, zum anderen das Konzept zur Synchronisation mit OpenStreetMap.

Einführung

Das Open Geodata Projekt OpenAddresses wurde 2007 am Institut Vermessung und Geoinformation der Fachhochschule Nordwestschweiz FHNW gegründet. Im Wesentlichen ist es ein Projekt, das geokodierte Gebäudeadressen über eine Webanwendung sammelt und sie allen Interessierten zur Verfügung stellt. Nebst einer einfachen Benutzeroberfläche bietet das Projekt auch Geokodierdienste [1] an. Im Jahr 2010 erfolgte ein wesentlicher Entwicklungsschritt durch die Migration des Systems nach MapFish. Das Projekt und seine Entwicklungsschritte wurden bereits mehrfach vorgestellt ([2], [3], [4]).

Qualitätssicherung

Im Jahr 2010 wurde im Rahmen einer Masterarbeit ein Qualitätssicherungskonzept für OpenAddresses konzipiert und in einer Testumgebung realisiert ([5]). Die Grundidee dabei ist, dass neu erfasste Gebäudeadressen mit den Geokodierungsergebnissen von freien GeoWeb-Services verglichen werden. Der Ansatz im Vergleich der Ergebnisse lehnt sich an die ISO Normen der ISO/TC 211 19xxx Familie an. Der Vergleich bezieht sich sowohl auf die Adressattribute wie Strassenname, Hausnummer, Postleitzahl und Ort als auch auf die Lage der Gebäudeadresse, welche geometrisch als Punkt definiert wird.

Hinsichtlich des Vergleichs der Attributwerte, wird ein binärer Vergleich auf die Schreibweise durchgeführt. Hinsichtlich der Lagegenauigkeit wurde im Rahmen der Masterarbeit eine fundierte statistische Analyse zur Ermittlung von Grenzwerten ausgearbeitet, welche bei der Detektion grober Ausreisser helfen.

Das Qualitätssicherungskonzept wurde weiterentwickelt und bietet nun folgendes an:

- Die Ergebnisse des Vergleichs einer in OpenAddresses erfassten Adresse mit freien GeoWeb-Services, wird in einer tabellarischen Ansicht dem Anwender zur Verfügung gestellt (=QA-Report) (vgl. Abbildungen 1 & 2). Mittels Einfärbung wird sogleich klar, wo möglicherweise Lagefehler oder falsche Adressattributwerte vorliegen.
- Der Inhalt der Vergleichsübersicht kann angepasst werden. Es kann gefiltert werden nach Ersteller der Daten, Datum, Anzahl der Ergebnisse, Lageabweichung usw. Eine detaillierte Beschreibung ist unter [6] zu finden.
- Ersteller von Adressen können per E-Mail eine Nachricht erhalten, wenn eine ihrer erfassten Adressen verändert wurde.
- Es besteht die Möglichkeit, dass sich QualityManager mit einem gewissen räumlichen Gebiet, für welches sie verantwortlich sein möchten, in der Datenbank eintragen lassen. Dann wird ein solcher QualityManager auf Wunsch per E-Mail informiert, wenn in seinem Gebiet Adressen hinzugefügt, verändert oder gelöscht wurden. Ebenso kann der QA-Report entsprechend gefiltert werden.

Synchronisation zwischen OpenAddresses und OpenStreetMap

Diese Option ist insbesondere für Organisationen aus der Verwaltung, die ihre Adressen spenden von Interesse: sie erhalten mit dieser Option die Möglichkeit, automatisch über Fehler in ihren Daten und deren Verbesserung informiert zu werden.

Quality Report for 20 OpenAddresses Objects

This page shows results of comparison of OpenAddresses.org addresses to OpenWebMapServices from Bing, Google and Yahoo.
Distance values as deviations are in [m]. 'True' and 'False' values indicate the results of a binary comparison of the user entered values to the ones from the mentioned OWMs.
Precision indicates as binary value whether the compared address values could be geocoded by the OWMs to address level.

In order to change the position or address values of an address simply click on its id. OpenAddresses launches in a new window at the address' location.

id	User Street	Number	Zip code	City	Country	Bing Distance	Bing Address	Bing Zip	Bing City	Bing Precision	Google Distance	Google Address	Google Zip	Google City	Google Precision	Yahoo Distance	Yahoo Address	Yahoo Zip	Yahoo City	Yahoo Precision	Date
14655757	stark Holckerstrasse	56	4132	Müllanz	CH	31.3743	True	True	True	True	64.033	False	True	True	True	45.5488	True	True	True	True	2012-02-24 08:57:01
14020349	stark Hinningerstrasse 2		4140	Münchenstein CH		113.7465	True	True	True	True	86.0295	True	True	True	True	52.8005	True	True	True	True	2012-02-21 14:44:04
14655757	stark Emil Frey-Strasse	91	4142	Münchenstein CH		27.3891	True	True	True	True	2.5911	True	True	True	True	25.0851	True	True	True	True	2012-02-21 12:22:08
14655750	stark Route du Port	29	1009	Pully	CH	19.3397	True	True	True	True	3.208	True	True	True	True	18.2309	True	True	True	True	2012-02-22 21:37:06
14655748	stark Hochstrasse	6	4053	Basel	CH	9.5167	True	True	True	True	2.119	True	True	True	True	7.5573	True	True	True	True	2012-02-22 10:59:39
14655747	stark Emil Frey-Strasse	112	4142	Münchenstein CH		242.2685	True	True	True	True	719.722	False	True	True	True	42.8658	True	True	True	True	2012-02-22 16:10:11
14020382	stark Hardstrasse	46a	4142	Münchenstein CH		14.3644	False	True	True	True	15.7423	False	True	True	True	16.9393	False	True	True	True	2012-02-20 19:18:12
14020381	stark Hardstrasse	46	4142	Münchenstein CH		30.3096	True	True	True	True	1.9017	True	True	True	True	30.3272	True	True	True	True	2012-02-20 19:18:10
14020380	stark Hardstrasse	42	4142	Münchenstein CH		20.9064	True	True	True	True	2.7112	True	True	True	True	24.9404	True	True	True	True	2012-02-20 19:18:10
14020379	stark Hardstrasse	38	4142	Münchenstein CH		16.0452	True	True	True	True	1.6698	True	True	True	True	9.4447	True	True	True	True	2012-02-20 19:18:10
14020378	stark Hardstrasse	30b	4142	Münchenstein CH		42.6783	False	True	True	True	3.4853	True	True	True	True	34.5375	False	True	True	True	2012-02-20 19:18:08
14020377	stark Hardstrasse	30c	4142	Münchenstein CH		35.2561	False	True	True	True	16.6313	False	True	True	True	13.4015	False	True	True	True	2012-02-20 19:18:04
14020376	stark Hardstrasse	32a	4142	Münchenstein CH		58.7021	False	True	True	True	14.5825	False	True	True	True	20.9361	False	True	True	True	2012-02-20 19:18:00
14020375	stark Hardstrasse	32	4142	Münchenstein CH		48.812	True	True	True	True	3.6003	True	True	True	True	16.7035	True	True	True	True	2012-02-20 19:17:55
14020374	stark Hardstrasse	34	4142	Münchenstein CH		54.1058	True	True	True	True	1.2297	True	True	True	True	19.1405	True	True	True	True	2012-02-20 19:17:51
14020373	stark Hardstrasse	35	4142	Münchenstein CH		46.4218	True	True	True	True	26.9405	True	True	True	True	43.2981	True	True	True	True	2012-02-20 19:17:47
14020372	stark Basenstrasse	35	4142	Münchenstein CH		126.351	True	True	True	True	6.1477	True	True	True	True	61.5832	True	True	True	True	2012-02-20 19:17:45
14020371	stark Basenstrasse	41	4142	Münchenstein CH		46.001	False	True	True	False	26.7035	False	True	True	True	114.2442	False	True	True	False	2012-02-20 19:17:38
14020370	stark Basenstrasse	32	4142	Münchenstein CH		29.6875	True	True	True	True	2.1408	True	True	True	True	61.3014	True	True	True	True	2012-02-20 19:17:33
14020369	stark Dammstrasse	30	4142	Münchenstein CH		33.1329	True	True	True	True	5.546	True	True	True	True	34.322	True	True	True	True	2012-02-20 19:17:30

Abb. 1: Übersicht QA-Report

Route du Port	29	1009	Pully	CH	19.3397	True	True	True	True	3.208	True	True	True
Hochstrasse	6	4053	Basel	CH	9.5167	True	True	True	True	2.119	True	True	True
Emil Frey-Strasse	112	4142	Münchenstein	CH	242.2685	True	True	True	True	719.722	False	True	True
Hardstrasse	46a	4142	Münchenstein	CH	14.3644	False	True	True	True	15.7423	False	True	True
Hardstrasse	46	4142	Münchenstein	CH	30.3096	True	True	True	True	1.9017	True	True	True
Hardstrasse	42	4142	Münchenstein	CH	20.9064	True	True	True	True	2.7112	True	True	True
Hardstrasse	38	4142	Münchenstein	CH	16.0452	True	True	True	True	1.6698	True	True	True
Hardstrasse	30b	4142	Münchenstein	CH	42.6783	False	True	True	True	3.4853	True	True	True

Abb. 2: Detailsansicht QA-Report

Synchronisierung mit OpenStreetMap (OSM)

Schon seit Beginn des Projekts kam immer wieder die Frage auf, wie OA mit OSM zusammenpasst. Da zu Beginn von OA die Adressen als Google Maps Mashup erfasst wurden, war seitens der OSM

Synchronisation zwischen OpenAddresses und OpenStreetMap

Community keine Offenheit da, solche Daten aufzunehmen. Seit 2010, als auf MapFish und Yahoo! Maps im Hintergrund umgestellt wurde, wurden jedoch alle auf Google Maps Basis erhobenen Daten aus der OA Datenbank gelöscht. Da OA dieselbe Lizenz wie OSM nutzt, steht – rein juristisch – einer Datenharmonisierung also nichts mehr im Wege. Seitens einzelner OSM-Mitglieder wurde auch immer wieder Sympathie für die einfache Benutzerschnittstelle von OA geäußert. Daher wurde am Institut Vermessung und Geoinformation der Fachhochschule Nordwestschweiz FHNW im GIS Team nach der Konzeptstudie im Rahmen einer Bachelorthesis ([7]) ein Synchronisationsmechanismus entworfen und eine Testimplementierung erstellt.

Die Kernelemente des Mechanismus sind wie folgt:

- Aus OSM werden Adresspunkte (Nodes) und geschlossene Linien (Ways) als Gebäudeumrisse übernommen. Aus letzteren wird ein Schwerpunkt gerechnet, der innerhalb des Gebäudegrundrisses liegt. Es werden keine Adressen interpoliert, die anhand von Ways in OSM durch Adressangaben des Start- und Endknotens bekannt sind.
- Aus OSM werden in regelmässigen Abständen die OsmChange-Dateien (OSC) mit OSMOSIS auf die erwähnten Objekte untersucht und ggf. in OA übernommen. Änderungen werden über die OSM-ID nachgeführt.
- Wird in OA eine Adresse erfasst, so wird sie automatisch über die OSM API in die OSM Datenbank eingetragen. Ein Adressknoten von OA wird in OSM nur gelöscht, wenn er mittels OA ursprünglich erstellt und nie von OSM verändert wurde.

Der Synchronisationsmechanismus ist graphisch in Abbildung 3 dargestellt.

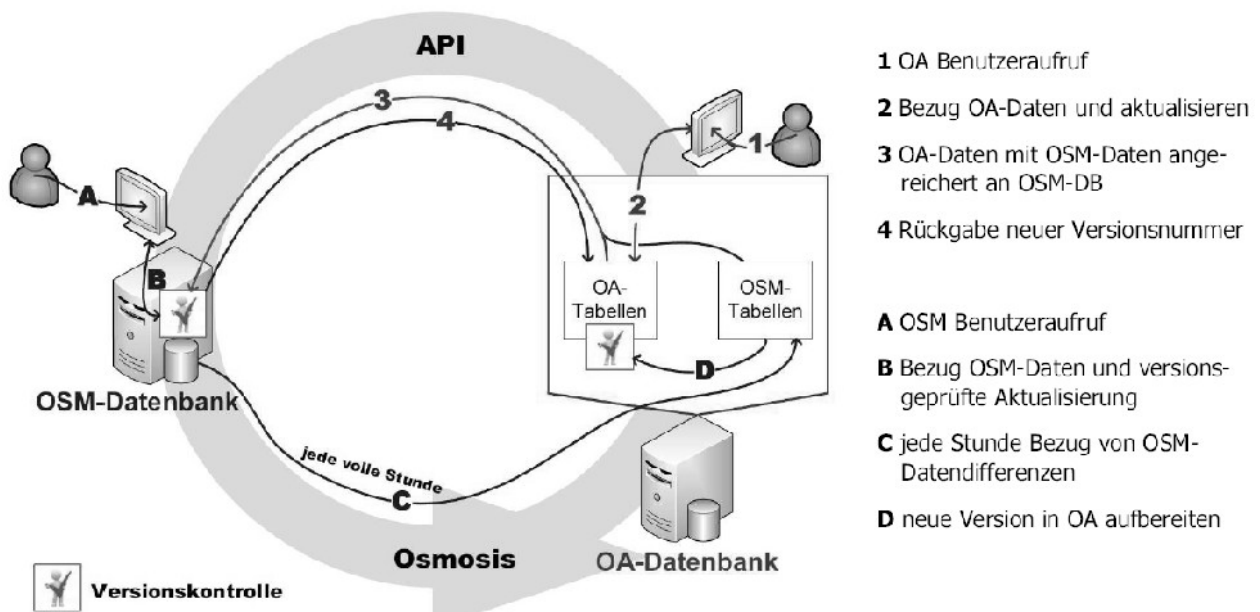


Abb. 3: OA Oberfläche mit OSM Adressen als Dreiecke und OA Adressen als Punkte (aus [7])

Die Adressen aus OSM werden in der Karte in OA durch Dreiecke dargestellt, native OA Adressen durch Kreise (vgl. Abbildung 4).

Synchronisation zwischen OpenAddresses und OpenStreetMap

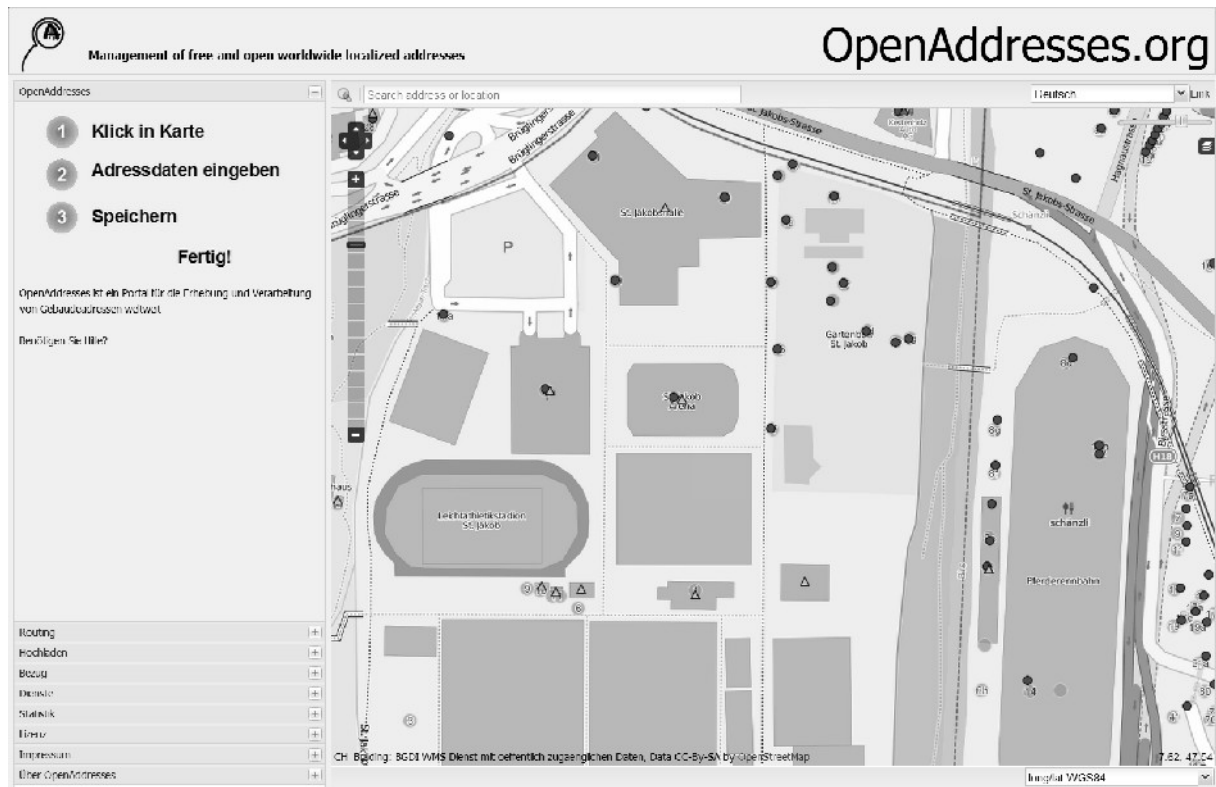


Abb. 4: OA Oberfläche mit OSM Adressen als Dreiecke und OA Adressen als Punkte

Wie in Abbildung 4 ersichtlich wurden die beiden Datenquellen noch nicht harmonisiert. D.h. es gibt zurzeit noch Gebäude, welche originär sowohl Daten aus OSM als auch aus OA enthalten. Nach Rücksprache mit Geofabrik wollten wir zuerst die Reaktion der OSM Community auf unseren Ansatz abwarten und gemeinsam mit der OSM Community das weitere Vorgehen besprechen.

Aus Sicht des Autors und der OA Community wird OA zunehmend als einfacher und guter Editor für Adressen gesehen, der nebst einer einfachen Verwaltung der Gebäudeadressen auch gute Geokodierdienste anbietet. Dabei tritt die Datenhaltung im Sinne einer von OSM unabhängigen Quelle in den Hintergrund und OA ist offen, OSM als Hauptdatenspeicherort zu sehen.

Fazit und Ausblick

Die jüngsten Erweiterungen von OA nähern sich OSM immer mehr an und lassen die Rolle von OA eher als Gebäudeadress-Interface für OSM erscheinen, welches nebst einer einfachen Benutzeroberfläche auch gute Geokodierdienste und ein umfangreiches Qualitätssicherungskonzept bietet. In naher Zukunft werden die Geokodierdienste weiter verbessert, so dass auch Adressen mit unterschiedlicher Schreibweise zu Treffern führen und bei einem Treffer zusätzlich die Güte des Treffers als Qualitätsindikator mitgegeben wird.

Synchronisation zwischen OpenAddresses und OpenStreetMap

Kontakt zum Autor:

Prof. Hans-Jörg Stark
Fachhochschule Nordwestschweiz
Hochschule für Architektur, Bau und Geomatik
Institut Vermessung und Geoinformation
Gründenstrasse 40
4132 Muttenz
+41 61 467 4605
hansjoerg.stark@fhnw.ch

Literatur

- [1] <http://code.google.com/p/openaddresses/wiki/RESTService>. [Zugriff: 24.2.2012]
- [2] Stark, Hans-Jörg, 2008. OpenAddresses.ch, in: Tagungsband Konferenz für Freie und Open Source Software für Geoinformationssysteme. Freiburg i.Brsg., 2008.
- [3] Bähler, Lukas, Stark Hans-Jörg, 2009. OpenAddresses - Evolution. in: Tagungsband Konferenz für Freie und Open Source Software für Geoinformationssysteme. Hannover, 2009.
- [4] Stark, Hans-Jörg, 2010. OpenAddresses - reloaded. In: Geomatik Schweiz 05/2010.
- [5] Stark, HJ. 2010. Quality assessment of volunteered geographic information (vgi) based on open web map services and ISO/TC 211 19100-family standards. Geoinformatics, Vol 7, 2010, pp.28-30.
- [6] <http://code.google.com/p/openaddresses/wiki/qa>. [Zugriff: 24.2.2012]
- [7] Bröchin, Manuel, 2010. OpenAddresses goes OpenStreetMap. Bachelor Thesis FHNW.

JOSM: Der Java OpenStreetMap Editor

Simon Legner

(Nicht automatisierte) Bearbeitungen werden in OpenStreetMap typischerweise mit einem Editor durchgeführt. Diese Programme sind auf den Umgang mit OSM-bezogenen Geodaten spezialisiert und bieten eine Vielzahl von Funktionen aus dem OSM-Workflow: Laden von GPS-Tracks, Anzeigen von Hintergrundbildern, Download von Ausschnitten der OSM-Datenbank, Bearbeiten der Daten, Vergeben von Attributen, Hochladen der Änderungen, Umgang mit Bearbeitungskonflikten, ... JOSM ist ein in Java entwickelter Offline-Editor, der (von Anfang an) auf erfahrene Nutzer abzielt. Derzeit wird das Projekt von Dirk Stöcker geleitet; ursprünglich wurde JOSM von Immanuel Scholz programmiert.

In diesem Vortrag wird auf die aktuelle Entwicklung von JOSM eingegangen. Neben dem Entwicklungskonzept wird auf relevante Funktionen/Neuerungen eingegangen. Anhand von einfachen Beispielen soll gezeigt werden, wie zusätzliche Features im Bezug auf die Validation von Daten, Tagging-Vorlagen und Styles umgesetzt werden können.

Nutzung amtlicher Geodaten - aktueller Stand

Joachim Kast

Im Vergleich zur Republik Österreich, welche seit Juli 2011 die Nutzung ihrer amtlichen Luftbilder in höchster Qualität (12,5 - 25 cm) zur Erzeugung von Open Data Folgeprodukten gestattet, gab es in der Bundesrepublik Deutschland zunächst keine wesentlichen Fortschritte. Lediglich der Freistaat Bayern erweiterte im Dezember 2011 sein Angebot an offenen Geodaten [2] neben dem bisherigen DOP 2m um Freizeitwege, Verwaltungsgrenzen und DGM 200 unter der Lizenz CC-BY.

Es sei erwähnt, dass es innerhalb der OSM-Community Diskussionen gibt, ob unter CC-BY veröffentlichte Dienste oder Daten zur Aktualisierung des OSM Datenbestandes verwendet werden dürfen. Wir können allenfalls die korrekte Quellenangabe in der ersten Version garantieren und die Datenquelle im Wiki erwähnen. Bei Folgeprodukten ist dies nicht mehr gewährleistet, da diese meistens nur auf „OpenStreetMap und Mitwirkende“ verweisen.

Der im November unter der Schirmherrschaft des Bundesministers des Innern gestartete Open-Data-Wettbewerb „Apps für Deutschland“ [3] war geodatenmäßig zunächst enttäuschend: Standortdaten von Hundekotbehältern, Toiletten und Altglascontainern, welche teilweise bis zu 700 Meter vom tatsächlichen Standort abwichen. Zum Jahreswechsel kam still und heimlich MAPS4DE hinzu, sozusagen ein amtliches Google Maps mit ATKIS-DLM-Daten und Gebäudeumringen aus ALKIS der Bundesländer Baden-Württemberg, Nordrhein-Westfalen, Rheinland-Pfalz sowie Saarland. Obwohl seitens des Veranstalters als „Open Data“ gekennzeichnet, ist nach Aussage des LGL Baden-Württemberg *eine Nutzung des Datenbestands WebAtlasDE zur Aktualisierung von OpenStreetMap ausgeschlossen*.

Daher wurde MAPS4DEbugs [5] als Wettbewerbsbeitrag eingereicht und so bis Ende Februar 1.000 Fehler in dem amtlichen Kartenwerk an die zuständigen Landesvermessungsämter gemeldet. Die Anwendung wird zwar vermutlich keinen Preis gewinnen, aber sie war bereits Anlass für ein ausführliches Gespräch mit dem Vorsitzenden des Lenkungsausschusses Geobasis des AdV.

Es bleibt also spannend, was sich bis zur FOSSGIS-Konferenz noch alles ereignet.

Links:

[1] www.geoimage.at/geodatendienste.html

[2] vermessung.bayern.de/opendata

[3] www.apps4deutschland.de

[4] offenedaten.de/dataset/bw-maps4de

[5] maps4debugs.openstreetmap.de

Am Puls der Zeit: Minütliche Daten-Updates bei OpenStreetMap

Frederik Ramm

Einmal pro Woche veröffentlicht das OpenStreetMap-Projekt einen Abzug der gesamten Datenbank, das sogenannte „planet file“. Mit einem ausreichend starken Rechner lassen sich die grob 200 GB XML in mehr oder weniger einem Tag einlesen und weiterverarbeiten. Von solchen Update-Zyklen können die Nutzer anderer Geodatenquellen nur träumen – aber es geht noch schneller: OpenStreet-Map veröffentlicht jede Minute eine Datei, die alle Änderungen der letzten Minute enthält.

Warum minütliche Updates?

Es gibt verschiedene Gründe, warum minütliche Updates attraktiv sind. Der offensichtlichste ist natürlich, dass man stets brandaktuelle Daten zur Verfügung hat. Für den von OpenStreetMap selbst betriebenen Tile-Server sind diese schnellen Updates wichtig, damit die Mapper schnelles Feedback und eine Qualitätskontrolle für Ihre Arbeit haben. In einer 2010 veröffentlichten Studie konnte wissenschaftlich nachgewiesen werden, dass Menschen bessere Arbeit machen, wenn sie erwarten, dass sie schnelles Feedback erhalten[1].

Für viele Anwendungsfälle scheint das zunächst nicht unbedingt wichtig, aber Updates sind ja nicht nur dazu da, um neu hinzugekommene Daten zu verteilen, sondern auch, um Fehler zu korrigieren. Stellen Sie sich vor, Sie betreiben einen Kartenserver und ein Nutzer beschwert sich über einen Tippfehler beim Namen seiner Straße. In wenigen Minuten können Sie sich bei OSM eingeloggt und den Fehler korrigiert haben; die automatische minütliche Aktualisierung Ihres Servers besorgt den Rest.

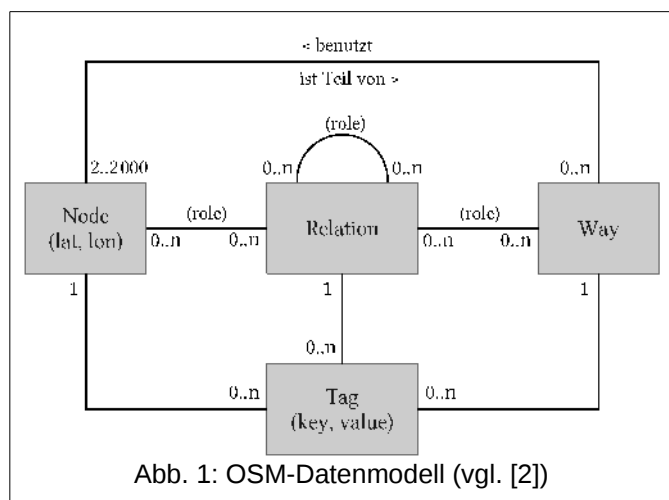
Minütliche Updates haben aber auch bei der technischen Verarbeitung einige Vorteile. Die summierten minütlichen Updates einer Woche haben zusammen ein Datenvolumen von rund 300 MB; mindestens 15 GB (komprimiert) müssen allerdings heruntergeladen werden, wenn man den kompletten „planet“ importiert – das Fünffzigfache. Hinzu kommt, dass diese Datenmenge bei wöchentlichem Download ja auch in einem Stück bearbeitet werden muss, während die Verarbeitung bei minütlichen Downloads gleichmäßig über die Woche verteilt werden kann.

Das OSM-Datenformat und der Aufbau von Update-Dateien

OpenStreetMap kennt drei Basis-Datentypen: Nodes (Punkte), Ways (Linien oder, wenn sie geschlossen sind, auch Flächen) und Relationen (diese können je nach Beschaffenheit wie ein Multipolygon oder ein Multilinestring oder eine andere Beziehung zwischen Objekten interpretiert werden).

Diese Datentypen stehen untereinander in Beziehung. Nur Nodes verfügen über eine eigene Geometrie; Ways beziehen sich auf eine Sequenz von Nodes, und Relation können sich auf jede Objektart als „Geometriequelle“ beziehen.

Eine Update-Datei folgt dem sogenannten „OsmChange“-Format. Sie kann beliebig viele Blöcke enthalten, und jeder Block fasst eine Reihe von „create“-, „modify“- oder „delete“-Operationen zusammen. Für neu erstellte oder veränderte Objekte findet sich in der Datei das vollständige neue Objekt; für gelöschte Objekte wird in der Regel nur die Objektidentifikation übertragen.



Am Puls der Zeit: Minütliche Daten-Updates bei OpenStreetMap

Diese Art der Übertragung in Zusammenhang mit der Bezugnahme von Objekten aufeinander kann die Verarbeitung von Update-Dateien in manchen Fällen deutlich erschweren, weil zum Beispiel ein neu erstelltes „Way“-Objekt durchaus auch ohne Geometrie daherkommen kann – seine Geometrie bezieht der Way unter Umständen aus Nodes, die schon längst existierten und gar nicht neu angelegt werden müssen.

Mit Updates umgehen

Die einfachste Möglichkeit, mit Updates umzugehen, ist die, mit Hilfe eines geeigneten Tools wie Osmosis eine lokal gespeicherte OpenStreetMap-Datei einfach um das Update zu ergänzen. Danach hat man eine aktuelle Datei und kann jede Art von gewöhnlicher Verarbeitung damit durchführen. Allerdings muss man schon rund eine Stunde Zeit veranschlagen, bis ein kompletter „Planet“ von der Platte gelesen und um ein Update ergänzt ist – die weitere Verarbeitung noch gar nicht berücksichtigt. Mit Osmosis geht das etwa so:

```
osmosis --read-xml-change update.osc --read-pbf alt.pbf --apply-change --write-pbf neu.pbf
```

Leichter geht das, wenn man nur einen Ausschnitt der Daten benutzt; zum Beispiel einen regionalen Auszug. Zwar sind keine regionalen Update-Dateien erhältlich, aber man kann z.B. mit Osmosis eine weltweite Update-Datei auf einen regionalen Ausschnitt anwenden und danach im gleichen Arbeitsschritt alle eventuell ausserhalb des interessanten Bereichs hinzugekommenen Objekte gleich wieder wegwerfen:

```
osmosis --read-xml-change update.osc --read-pbf alt.pbf --apply-change --bound-box left=10.0 right=15.0 top=55.0 bottom=48.0 --write-pbf neu.pbf
```

Die beste Performance erzielt man allerdings, wenn man die Updates direkt verarbeiten kann – ohne also überhaupt erst eine aktualisierte OSM-Datei zu erzeugen.

Beispiele

Ein sehr einfaches Beispiel für die direkte Verwendung von minütlichen Updates ist eine Anwendung, die schlicht die Anzahl der Nodes in der OSM-Datenbank zählt. Hierfür ermittelt man anfänglich die Anzahl aus einer Planet-Datei und zieht dann jede Minute für jeden gelöschten Node eins ab und zählt für jeden neu erzeugten Node eins hinzu:

vorher	OsmChange	nachher
Anzahl Nodes = 191	<pre><delete> <node id="182" ... /> </delete> <create> <node id="1082" ... /> <node id="1083" ... /> </create></pre>	Anzahl Nodes = 193

Am Puls der Zeit: Minütliche Daten-Updates bei OpenStreetMap

Möchte man hingegen bestimmte Nodes zählen – nehmen wir zum Beispiel an, es ginge um Apotheken, die in OpenStreetMap das Tag „amenity=pharmacy“ tragen – so wird es etwas aufwendiger:

vorher	OsmChange	nachher
Anzahl Apotheken = 13	<pre><delete> <node id="182" ... /> </delete> <create> <node id="1082" ... > <tag k="amenity" v="pharmacy" /> </node> </create></pre>	Anzahl Apotheken = ?

Es ist offensichtlich eine Apotheke hinzugekommen – aber was ist mit dem gelöschten Node, war das vielleicht auch eine Apotheke? Um also eine solche Zählung durchzuführen, reicht es bereits nicht mehr, sich nur die Zahlen zu merken, sondern man muss die IDs aller Apotheken-Nodes ebenfalls abspeichern.

Noch komplizierter wird es, wenn Ways und ihre Geometrien ins Spiel kommen. Angenommen, man möchte zu jedem Zeitpunkt wissen, wieviele Kilometer Autobahn sich in OpenStreetMap befinden. Nun trifft eine Änderungsmitteilung ein, die nur einen einzigen Node betrifft; seine Position habe sich verändert. Um nun zu bestimmen, ob dies einen Effekt auf die Summe aller Autobahnkilometer hat, muss man herausfinden, ob der gerade verschobene Node Teil einer Autobahn ist (und man muss zu diesem Zweck mindestens alle Ways, die als Autobahn gekennzeichnet sind, vorhalten), und wenn ja, muss man die Länge der nun veränderten Liniengeometrie der Summe hinzuaddieren (und die Länge der alten Geometrie abziehen).

Während also einfache, und vor allem auf Nodes beruhende Auswertungen relativ einfach umzusetzen sind, wird eine flexible Verarbeitung von Updates schnell kompliziert und erfordert oft, dass man eine vollständige OSM-Datenbank in einer geeignet zugreifbaren Form vorhält. Dies wird durch existierende Tools vereinfacht.

Der Update-Werkzeugkasten

Die wichtigsten Hilfsprogramme zum Umgang mit Updates sind „Osmosis“ und „osm2pgsql“. Beide können OpenStreetMap-Daten in eine PostGIS-Datenbank importieren; es kommt aber ein leicht unterschiedliches Datenschema zur Anwendung.

Der Import mit osm2pgsql ist speziell auf das Zeichnen von Karten abgestimmt. osm2pgsql erzeugt einen Satz von Geometrietabellen (je eine Tabelle für Punkte, Linien und Polygone) und zusätzlich noch einen Satz von Rohdaten-Tabellen (für Nodes, Ways und Relationen) und kann so bei jeder neu hereinkommenden Änderung sofort feststellen, welche Geometrieobjekte von dieser Änderung betroffen sind und aktualisiert werden müssen.

Osmosis erzeugt eine andere Datenstruktur (das sogenannte „Snapshot-Schema“); hier sind die Geometrie-Objekte direkt in den Tabellen mit den Rohdaten untergebracht, und man kann wählen, ob man für Ways nur Bounding Boxes oder vollständige Linestrings erstellt haben möchte. Die Multipolygon-Behandlung ist Osmosis nicht inbegriffen und muss selbst aufgebaut werden; dafür gibt es aber eine „action“-Tabelle, in der Osmosis nach einem Update vermerkt, welche Objekte verändert wurden und die verwendet werden kann, um weitere, eigene Verarbeitungsschritte anzustoßen.

Am Puls der Zeit: Minütliche Daten-Updates bei OpenStreetMap

Für Anwendungsfälle, bei denen es hauptsächlich auf das Kartenzeichnen ankommt, empfiehlt sich meist eher osm2pgsql; für weitergehende eigene Verarbeitung ist Osmosis oftmals besser geeignet.

Kontakt zum Autor:

Frederik Ramm
Geofabrik GmbH
Scheffelstr. 17a
76135 Karlsruhe

Teile dieses Vortrags basieren, mit freundlicher Genehmigung des Autors, auf einem Vortrag von Matt Amos zum gleichen Thema [3].

Literatur

- [1] Kettle et al. Motivation by Anticipation: Expecting Rapid Feedback Enhances Performance. Psychological Science, 2010; DOI: 10.1177/0956797610363541
- [2] Ramm, Topf: OpenStreetMap – die freie Weltkarte benutzen und mitgestalten. Berlin, 2010.
- [3] Amos, Matt: <http://www.asklater.com/matt/wordpress/2010/07/12/osm-without-delay-sotm10-talk/>, Girona, 2010.

History-Informationen in der OSM Datenbank, Reloaded

Peter Körner

Jede Änderung an jedem Objekt wird gespeichert und ist öffentlich. Doch wie kommt man an diese Daten heran? Welche Quellen gibt es und wofür sind sie geeignet? Welche Probleme ergeben sich beim Auswerten dieser Informationen und wie kann man diese überwinden? Welche tollen und nützlichen Anwendungen lassen sich mit diesen Daten bauen und wie können solche Daten der Community helfen?

Begriffsklärung:

In der Diskussion über die Vergangenheit in OpenStreetMap kommt es häufiger zu Missverständnissen. Die Meisten rühren daher, dass es verschiedene Zeitlinien gibt, die miteinander nicht verbunden werden können. Insbesondere muss zwischen folgenden „Vergangenheiten“ unterschieden werden:

- Die echte Vergangenheit
manifestiert sich beispielsweise in der Frage, wann ein Geschäft eröffnet hat und wann es wieder schloss
- Die Vergangenheit des OpenStreetMap-Projektes
welche sich beispielsweise auf Änderungen in der Gestaltung der Karte auf openstreetmap.org bezieht
- Vergangene Versionen von Daten
die in Form von Versionen als Teil der OpenStreetMap-Datenbank betrachtet werden

Der Vortrag bezieht sich auf die letzte Art von Vergangenheit.

Fragestellungen

Für die Kartographen des Projektes stehen Fragen nach der direkten Vergangenheit der bearbeiteten Daten im Vordergrund. Dies sind beispielsweise „Wer hat ... zuletzt bearbeitet?“, „Wie sah ... am ... aus?“ und in letzter Instanz „Wie kann ich ... wieder rückgängig machen?“.

Gleichzeitig sind jedoch auch andere Personengruppen an den vergangenen Versionen der Daten interessiert. Insbesondere können die Informationen von Mathematikern und Statistikern ausgewertet werden, um das Verhalten und das Wachstum der Community Zeit- und Ortsbezogen zu analysieren. Ebenso interessant ist die Datenbasis für genaue Berechnungen im Zusammenhang mit dem Lizenzwechsel und den damit verbundenen Maßnahmen [1].

Notwendigkeit

Warum, außer aus wissenschaftlicher Neugier, könnten solche Betrachtungen interessant sein? Für die Funktionsweise einer Community in der Größenordnung von OpenStreetMap und für gemeinschaftlich gepflegte Datenbestände in dieser Größenordnung gibt es nicht allzu viele Vergleichsmöglichkeiten. Gerne wird die Wikipedia heran gezogen, was auch in Fragen der Communitybildung und – Pflege gut funktioniert, allerdings beim Umgang mit Vandalismus oder Uneinigkeit schnell an seine Grenzen stößt. Hintergrund der Vandalismusbekämpfung bei der Wikipedia ist die Grundregel, dass „kaputt machen“ genau so einfach geht, wie „wieder reparieren“. Dieses zurückspielen auf einen alten Stand kostet ist nur einige Klicks und steht sehr vielen Mitgliedern offen. Dadurch können Manipulationen und mutwillige Zerstörungen schnell und auf viele Schultern verteilt behoben werden.

History-Informationen in der OSM Datenbank, Reloaded

Bei OpenStreetMap ist dies nicht so. Das Zurücknehmen von Änderungen ist ein komplizierter Prozess, der nur wenigen „eingeweihten“ zugänglich ist. Oft sind die Änderungen nur komplett zurücknehmbar, wobei nicht selten wesentliche Teile der Arbeit anderer mit vernichtet werden. Manchmal kann der Zerstörung auch nur durch Handarbeit entgegen gewirkt werden.

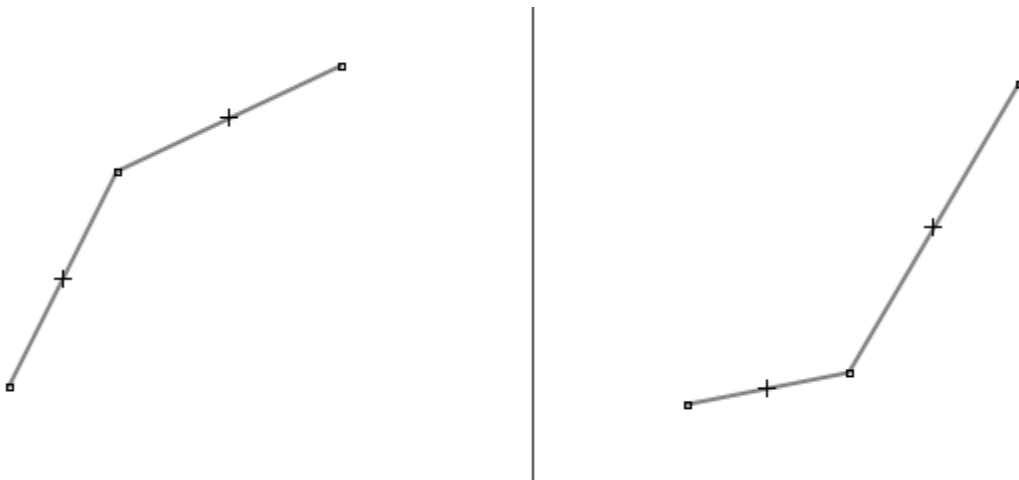
Hier können bessere Methoden in der Verarbeitung der Vergangenheit eingreifen und jedem Community-Mitglied diese Möglichkeiten der Qualitätskontrolle zugänglich machen.

Ein weiterer, wenig erforschter aber viel diskutierter Aspekt, welcher sich mit Hilfe der Auswertung der Datenvergangenheit untermauern ließe, ist die Auswirkung von Importen auf das Communityleben. Aus den Versionsinformationen lässt sich nicht nur Anzahl der Bearbeitungen von importierten Daten mit den Charakteristiken von manuell erfassten Informationen vergleichen, es ist ebenso möglich die Zu- und Abwanderung von Mappern oder die Verlagerung bestimmter Nutzergruppen auf andere Teile des Landes oder zu anderen Arten von Objekten zu quantifizieren. Auch das Verhalten von Neulingen in Regionen mit sowie in solchen ohne Importe könnte verglichen werden.

Dies seien nur zwei Beispiele für die praktische Anwendung der Vergangenheitsinformationen der OpenStreetMap-Datenbank, denen viele noch ungenannte Möglichkeiten gegenüber stehen.

Schema

Das Datenschema, dem alle Datenquellen rund um OpenStreetMap folgen, beinhaltet die Elemente Node (Stütz- und Datenpunkt), Way (Verbindung von zwei oder mehr Punkten) und Relation (Verbindung von einer oder mehreren Nodes oder Ways), jeweils mit angehängten Schlüssel-Wert-Paaren. Für jedes Element wird eine eigene und unabhängige Versionsgeschichte geführt. Wichtig ist hier zu beachten, dass nur Nodes geographische Informationen codieren. Ein Way enthält Referenzen auf Nodes, ohne deren Koordinaten zu reflektieren. In der Auswertung der Versionsgeschichte ergibt sich daher ein Problem, welches sich an einem Beispiel erläutern lässt.



Auf der linken Seite sieht man einen Way mit drei Nodes als Stützpunkte. In der rechten Hälfte wurde die mittlere Node nach unten verschoben. Aus Sicht der Datenbank wurde die Koordinate der Node verändert, weshalb dieser nach der Änderung in einer neuen Version vorliegt. Der Way wurde hingegen, aus Sicht des Datenmodells, nicht verändert.

Würde dem Way hingegen ein Stützpunkt hinzu gefügt oder würden seine Key-Value Informationen (beispielsweise sein Name) geändert, hätte diese Änderung er eine neue Version begründet; die Nodes wären in diesem Beispiel nicht betroffen gewesen.

Zusammenfassend lässt sich sagen, dass Geometrieänderungen in den Nodes abgebildet werden, während Änderungen an den Eigenschaften eines Ways auch in dessen Versionsgeschichte gespei-

chert werden. Um also beispielsweise die Frage „Wie sah ... am ... aus?“ beantworten zu können, muss beides betrachtet werden.

Weitere Probleme

Neben den Bürden durch das Datenbankschema begegnen einem Interessierten noch weitere Probleme. Beispielsweise sind die zu verarbeitenden Datenmengen enorm. Ein Kompletter Auszug der OpenStreetMap-Datenbank mit allen Versionen belegt derzeit 32 GB [2].

Einige andere Probleme wie eine unklare Lage bezüglich des Datenformats sowie das Fehlen von entsprechenden Bibliotheken zum lesen und schreiben dieser Daten haben sich im letzten Jahr verbessert, hier fehlt es aber noch bei der Adaption der Datenformate für verschiedenen Programmiersprachen.

Tools & Bibliotheken

Bei der Suche nach Programmen zum Verarbeiten der Versionsinformationen kommt man derzeit am Jochen Topfs Framework „osmium“ nicht vorbei [3]. Vor allem wird hier eine in C++ geschriebene Bibliothek bereit gestellt, mit welcher ein Programmierer ohne Mühe die gesamte Bandbreite von verwendeten Datenformaten lesen, verarbeiten und schreiben kann. Daneben ist auch das erstellen von Zwischenformaten (Shapefiles, CSV-Dateien) möglich.

Für Nicht-Programmierer hält „osmium“ einen Strauß von Beispielprogrammen z.B. zum konvertieren zwischen den verschiedenen Datenformaten bereit. Mit „osmjs“ wird eine Möglichkeit angeboten, die Verarbeitung in Javascript statt in C++ durchzuführen, wodurch für viele ein einfacherer Einstieg in die Verarbeitung von OpenStreetMap-Daten möglich wird.

Eines der Werkzeuge, welche die Arbeit mit Versionsinformationen im vergangenen Jahr stark voran gebracht haben, ist der „OpenStreetMap History Splitter“ [4]. Mithilfe dieses Schneidwerkzeuges lassen sich aus den gewaltigen Datenmengen eines Planeten-Dumps handliche Stücke in geographischer Abgrenzung erzeugen, welche aufgrund der verringerten Dateigröße ohne Weiteres auf handelsüblichen Notebooks oder PCs in angenehmer Zeit verarbeitet werden können.

Ein weiteres Tool aus dem letzten Jahr ist der „OpenStreetMap History Renderer“ [5], welcher es mit einem eigenen Datenbankschema samt Importer ermöglicht, aus einer einzelnen Datenbank Karten für jeden beliebigen Zeitpunkt zu erstellen. Dieses Programm stellt jedoch auch für viele weitere Auswertungen einen guten Ansatzpunkt dar, weil sich die erstellte Datenbank gleichermaßen auch für viele andere Abfragen eignet. In jedem Fall kann der Importer als Beispiel für die Auswertung und Aufbereitung der Datenhistorien dienen.

Datenquellen

Neben dem unregelmäßig erscheinenden kompletten Export aller Daten inklusiver der Vergangenheitsinformation [2] gibt es wenigstens zwei Quellen für vorgeschnittene Extrakte [6]. Desweiteren enthalten auch die minütlich, stündlich und täglich publizierte Diff-Dateien [7] vollständige Versionsinformationen, mit welchen eine Datenbank oder ein Extrakt auf dem neusten Stand gehalten werden könnte. Hier mangelt es aber zum Zeitpunkt des Erscheinens an geeigneten Werkzeugen.

History-Informationen in der OSM Datenbank, Reloaded

Kontakt zum Autor:

Peter Körner
Gustav-Heinemann Straße 1
55239 Gau-Odernheim
0152/5167822
osm@mazdermind.de

Verweise & Links

- [1] Vortrag „Die neue OpenStreetMap-Lizenz“ von Frederik Ramm auf der FOSSGIS 2011
- [2] Full-Planet-Dump vom 13.02.2012 auf <http://planet.osm.org>
- [3] Jochen Topfs Framework „osmium“ <https://github.com/joto/osmium>
- [4] Peter Körners “OpenStreetMap History Splitter” <https://github.com/MaZderMind/osm-history-splitter>
- [5] Peter Körners “OpenStreetMap History Renderer” <https://github.com/MaZderMind/osm-history-renderer>
- [6] geographische Extrakte <http://ftp5.gwdg.de/pub/misc/openstreetmap/osm-full-history-extracts/> und <http://odbl.poole.ch/extracts/>
- [7] vollständige Diffs der Datenbank <http://planet.osm.org/minute-replicate/>, <http://planet.osm.org/hour-replicate/> und <http://planet.osm.org/day-replicate/>

Das Osmium-Framework

Jochen Topf

Osmium ist eine performante und flexible C++- und Javascript-Bibliothek für die Arbeit mit OSM-Daten. Osmium kann OSM-Dateien (XML- und PBF-Format) lesen und schreiben, Geometrien aus OSM-Objekten zusammensetzen (Linestrings aus Ways, Multipolygone aus Relations, usw.), OSM-Change-Dateien anwenden, OSM-Objekte und Node-Geometrien zwischenspeichern, OSM-Daten in Shapefiles und viele OGR-Formate wandeln und vieles mehr. Osmium kann sowohl "normale" OSM-Daten als auch OSM-History-Daten verarbeiten.

Seine volle Funktionalität stellt Osmium unter C++ zur Verfügung. Dank der eingebundenen Google V8 Javascript-Engine ist es aber auch möglich, ganz ohne C++-Kenntnisse nur mit Javascript viele seiner Fähigkeiten zu nutzen.

Dieser Vortrag stellt Osmium und das Tool osmjs vor und erklärt, wie man die Funktionalität in C++ und Javascript nutzen und erweitern kann.

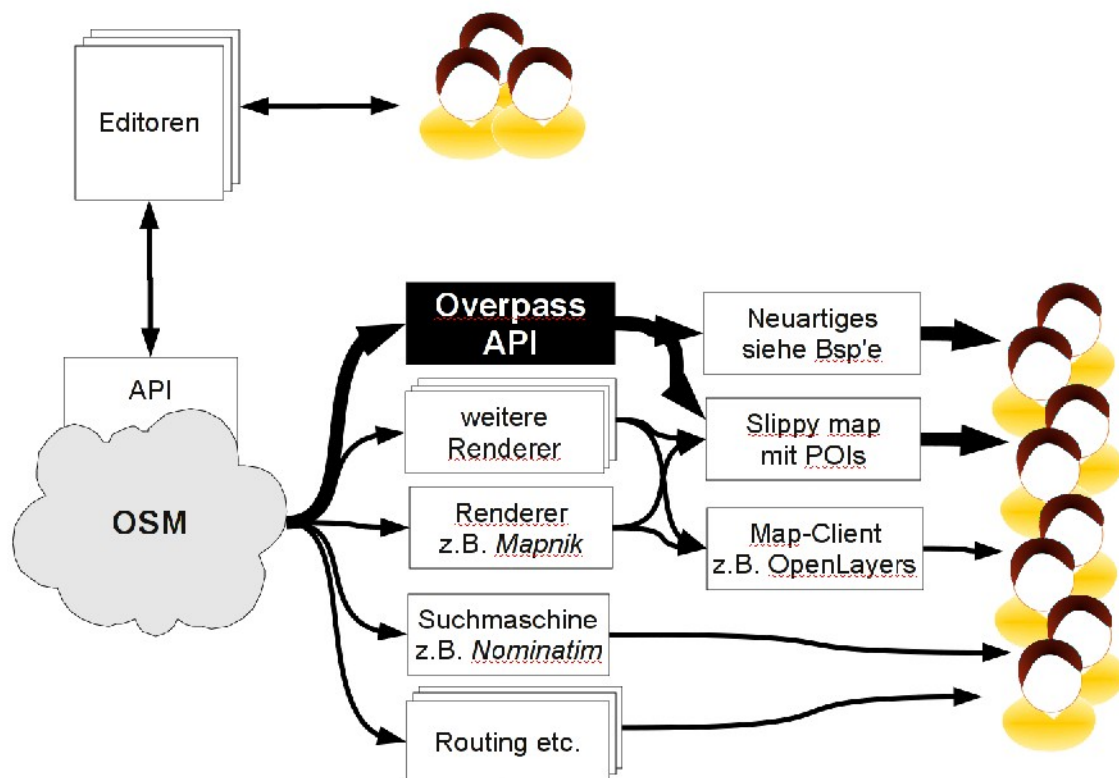
Osmium ist Open Source. Weitere Informationen gibt es unter <http://wiki.openstreetmap.org/wiki/Osmium>.

Overpass API

Roland Olbricht

Einleitung

Overpass API [7] ist eine Datenbank für Daten des Projekts *OpenStreetMap* (OSM) [11], die effektiv über das Web abgefragt werden kann und die gesamten weltweiten OSM-Daten mit nur wenigen Minuten Verzögerung vorhält.



Eines der Ziele von OpenStreetMap ist es, gerade auch neuartige und überraschende Nutzungen von Geodaten zu ermöglichen. Wie die später angeführten Beispiele zeigen, benötigen einige dieser Dienste so ungewöhnliche Abfragen auf die OSM-Daten, dass diese nicht mehr von den APIs, die auf Karten-Renderer und Geodaten-Editoren ausgelegt sind, mit beantwortet werden können. Dieses Problem ist Anlass für die Entwicklung von Overpass API gewesen.

Mittlerweile profitieren aber auch einige bereits vorher funktionsfähige Dienste von Overpass API: Für eine Slippy Map mit POIs ist dann keine eigene Datenhaltung für die POIs mehr nötig, sondern die Daten aus Overpass API und die Tiles aus dem Karten-Renderer können in einem Map-Client wie OpenLayers [9] zur Laufzeit zusammengefügt werden.

Ziele

Die Abfragesprache muss einerseits flexibel genug sein, um interessante Abfragen auch tatsächlich formulieren zu können. Andererseits muss die Abfragesprache beschränkt genug sein, um übergroße Abfragen, egal ob sie aus Vorsatz oder einem Versehen entstanden sind, abzufangen, bevor sie den Server überlasten. Daher hat Overpass API eine eigene Abfragesprache, *Overpass QL*, erhalten.

Das Sicherheitskonzept von Overpass API fußt als Kernidee darauf, dass keine Komponente Root-Rechte erfordert, sondern alle Bestandteile von einem normalen Nutzer betrieben werden können. Insbesondere kann dies auch ein eigens eingerichteter Benutzer mit entsprechend zugeschnittenen Rechten sein.

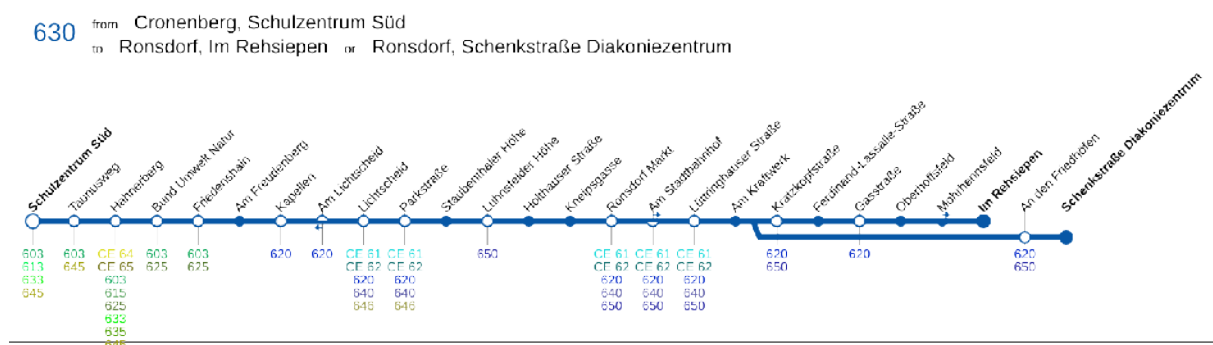
Anwendungsbeispiele

Karten mit POI-Markierungen

Technisch gesehen ist dies ein Map Client, der eine von Mapnik gerenderte OpenStreetMap-Karte als Grundlage anzeigt. Darauf werden POIs, nämlich Nodes aus der OSM-Datenbank mit bestimmten Tags, eingetragen. Diese Nodes werden dynamisch aus Overpass API geliefert. Wesentlich für die Effektivität ist hier, die recht einfache Abfrage stets ausreichend schnell bedienen zu können. Die Reaktionszeit liegt hier zuverlässig bei etwa einer Sekunde.

Linienband-Generator

<http://overpass-api.de/api/sketch-line?network=VRR&ref=630&style=wuppertal> zurückgeliefert:



Overpass API

Der Linienband-Generator wird weltweit angeboten und erstellt bei Vorgabe des *Liniennetzes* und der *Liniennummer* vollautomatisch ein Diagramm ähnlich dem obigen. Der Service hat seinen hohen Reifegrad erst durch wiederholte Abfragen nach Feinschliff durch Tiziano D'Angelo aus Padua erhalten.

Im Gegensatz zur Slippy Map mit POIs ist hier die Abfrage weitaus komplexer: Finde alle Relationen mit gegebenen Werten für die Tags *network* und *ref*, und finde alle dort referenzierten Nodes. Finde dann zu diesen Nodes alle weiteren Nodes in der Nähe, die Haltestellen oder Bahnhöfe oder sonst interessant sind (z.B. Park&Ride-Plätze, lässt sich konfigurieren). Finde dann alle Relationen, die diese Nodes referenzieren und trage sie als Umsteigeverbindungen ein. Es sind Abfragen dieser Komplexität, für die die Abfragesprache *Overpass QL* intendiert ist, auch wenn die Abfragezeit zur Zeit etwa 10 Sekunden erreicht.

Ohne Overpass API wäre ein solcher Service nur mit einer aufwendigen und spezialisierten Datenbank realisierbar, die dann unflexibel gegenüber spontanen Änderungswünschen wie z.B. der Einbindung von Park&Ride-Plätzen wäre.

Entlastung der API

Ein überraschender Verwendungszweck für Overpass API ist die Entlastung der Read-API der zentralen OSM-Datenbank.

Konkret kann für den Editor JOSM [12] das Plugin *mirrored_download* die minutenaktuellen Rohdaten zum Bearbeiten statt von der API des zentralen OSM-Servers von Overpass API aus anfordern.

Als Beispiel soll das Stadtzentrum von Bonn editiert werden. Statt der URL

<http://api.openstreetmap.org/api/0.6/map?bbox=7.08,50.73,7.11,50.75>

wird die URL
aufgerufen.

<http://overpass.osm.rambler.ru/cgi/xapi?map?bbox=7.08,50.73,7.11,50.75>

Downloadzeiten für Rohdaten aus Bonn-Zentrum	
zentrale OSM API	3 m 20 s
Overpass API	0 m 15 s

Bonn ist eine sehr intensiv gemappte Stadt mit einer dementsprechend hohen Datendichte. Die abgefragte Bounding Box enthielt zum Abfragezeitpunkt 16455 *Nodes*, 3564 *Ways* und 251 *Relations*. Als empirisch belegt kann die einmalige Abfrage nicht gelten. Sie zeigt aber auf, dass Overpass API durch seine effiziente Architektur als alternative Zugriffsmöglichkeit zur API des zentralen OSM-Servers ebenfalls die Arbeit im Geodaten-Editor deutlich flüssiger gestalten kann.

Abfragesprache

Anforderungen

Eine Datenbank, die einerseits über das Web abfragbar sein soll, andererseits sehr große Datenmengen verwaltet, muss ihre Abfragemöglichkeiten sorgfältig so gestalten, dass nicht der Server durch aufwendige Abfragen einzelner Nutzer überlastet werden kann. Wir greifen dafür eine Idee von Stroustrup [14] auf: Im besten Fall sollte eine günstige Abfrage stets kurz und hübsch, aber eine aufwendige Abfrage stets lang und hässlich zu formulieren sein; dies gibt Benutzern eine Intuition, ob die Abfragestrategie gut ist. Darüber hinaus sollte die Abfragesprache kanonisch mit sämtlichen Sonderzeichen umgehen können, da diese mitunter zahlreich in geographischen Daten vorkommen.

Overpass API

Ein naheliegender Kandidat wäre eine spezielle SQL-Variante. Allerdings passt SQL nicht kanonisch zum Datenmodell von OSM, ist anfällig für irrtümlich übergroße Abfragen und weist Sicherheits-Schwachstellen auf.

Die beiden übrigen Kandidaten, die Abfragesprache API 0.6 der zentralen OSM-Datenbank sowie die Abfragesprache des Services XAPI sind bei weitem nicht ausreichend ausdrucksstark, um Abfragen, wie sie im Beispiel Linienband-Generator aufgeführt sind, abzudecken.

Overpass API besitzt daher eine eigene Abfragesprache, *Overpass QL*, und zwar in einer neueren, C-artigen Syntax und in einer älteren und umständlicheren XML-Syntax. Wir stellen beide im Folgenden vor.

Overpass QL ist eine imperative Sprache, d.h. eine Abfrage besteht aus abgrenzbaren Anweisungen, die nacheinander ausgeführt werden. Solche Anweisungen sind entweder die Kombination einer oder mehrerer Prädikate, eine Kontrollflussanweisung oder eine Ausgabeanweisung. Prädikate können absolut sein (z.B. die Suche nach Elementen, die ein Tag mit bestimmten *Key* und *Value* besitzen) oder relativ zum Ergebnis der letzten Abfrage (z.B. alle Nodes, die höchstens einen Abstand von 100 Metern zu den bereits gefundenen Nodes haben). Um endliche Laufzeit garantieren zu können und die Laufzeit vorhersagen zu können, ist die Sprache nicht Turing-vollständig. Sie kennt Variablen von nur einem Typ, nämlich Zwischenergebnissen.

XML

Als Beispiel sei die Abfrage gezeigt, alle Nodes zu finden, die für das Tag *name* den Wert „Gielgen“ haben (ein Stadtteil und eine Bushaltestelle in Bonn).

```
<query type="node">
  <has-kv k="name" v="Gielgen"/>
</query>
<print mode="meta"/>
```

Es ist gut zu erkennen, dass die Abfragen recht schnell eine beachtliche Länge erreichen. Wegen Schlusstags wie z.B. von *query* enthält die Sprache auch ein hohes Maß an Redundanz. Dafür bestehen aber dank der sehr präzisen XML-Spezifikation [15] und ausgereiften Parsern wie Expat [2] weder Lücken noch Fehler im Umgang mit Sonderzeichen.

Eine größere Anzahl von Nutzern hat auf verschiedenen Kommunikationskanälen die Syntax als unschön und kompliziert eingeschätzt. Dennoch zeigen die Abfrage-Statistiken, dass die korrekte Verwendung der Abfragesprache über deren fehlerhafte deutlich überwiegt.

Overpass QL

Die für XML als Beispiel gezeigte Abfrage lässt sich nun kürzer formulieren.

```
node["name" = "Gielgen"];out meta;
```

Die Verkürzung wird über das Weglassen redundanter Elemente erreicht. Über den Erfolg der Sprache lässt sich noch keine Aussage treffen, da diese erst Mitte Februar 2012 eingeführt worden ist.

Die Sprache erfüllt insbesondere den Wunsch, übliche Abfragen so knapp formulieren zu können, dass sie in eine HTTP-GET-URLs eingebettet werden können. In diesem Fall muss nur das Leerzeichen zwischen *out* und *meta* durch %20 maskiert werden [3].

XAPI Compability Layer

Der XAPI Compability Layer ist erst auf Nachfrage zahlreicher Benutzer entstanden. Er ermöglicht, statt Overpass QL die primitivere XAPI-Syntax zu verwenden.

XAPI ist eine ältere Server-Komponente im OpenStreetMap-Ökosystem, die eine ähnliche Aufgabe wie Overpass API erfüllen sollte. Das Konzept sah jedoch nur einfache Abfragen vor. Zwischen 2008 und 2011 haben sich die Instanzen wiederholt als unzuverlässig erwiesen. Zudem hat die Verwendung der exotischen Plattform GT-M für die Server eine Portierung so sehr erschwert, dass schließlich eine Neuimplementierung in Java erfolgt ist [4].

Heute stehen zwar keine leistungsfähigen öffentlichen XAPI-Server zur Verfügung, aber dennoch nutzen zahlreiche Software-Komponenten die XAPI-Syntax. Diesen steht nun über den XAPI Compability Layer auf Overpass API wieder eine leistungsfähige XAPI-Abfragemöglichkeit zur Verfügung.

Gemäß der Benutzerstatistik machen Abfragen im XAPI-Format zwar nur einen Bruchteil des Datenvolumens aus, sorgen aber für eine große Zahl Benutzer. Eine detaillierte Analyse erfolgt im Abschnitt zur Benutzerstatistik.

Technik

Datenbank

Zu meiner Überraschung haben sich das Backend von MySQL [6], hier MyISAM, als ungeeignet erwiesen. Erste Tests haben gezeigt, dass die Zielgröße für einen großen Test, 1.000.000 Tabellenzeilen zu lesen, selbst bei aggressiver Optimierung in MySQL Zeiten von 600 Sekunden an aufwärts benötigt (auf einem PC mit Intel Core Duo, 4 GB RAM und einer üblichen 2,5"-Festplatte). Das eigens entwickelte Backend *Template DB* benötigt hierfür auf der gleichen Hardware nur etwa 3 bis 5 Sekunden.

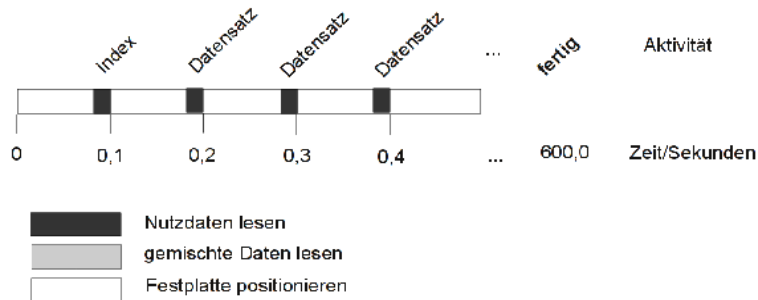
Um den Unterschied für diesen speziellen Einsatzfall aufklären zu können, sind einige Details über die Funktionsweise relationaler Datenbanksysteme erforderlich: Die eigentlichen Daten werden je Zeile an einem jeweils festen Platz in einer zur Tabelle assoziierten Datei gespeichert. Neue Zeilen werden in neu auf der Festplatte reservierten Speicher gelegt oder ersetzen gelöschte Zeilen. Um nach verschiedenen Suchkriterien effizient auf einige Dutzend oder hundert Zeilen zugreifen zu können, werden in speziellen Dateien Indizes vorgehalten, die zu jedem Suchkriterium die in der Tabelle vorkommenden Werte und die Position der zugehörigen Zeile in der Datei speichert. Zur Vereinfachung gehen wir hier davon aus, dass es nur einen Index gibt.

Dabei dominiert die Zeit zum Springen an Lesepositionen innerhalb der Datei deutlich gegenüber der Zeit für das eigentliche Lesen. Ziel einer relationalen Datenbank ist es daher, asymptotisch nur einen Festplattenzugriff pro Zeile zu benötigen, d.h. möglichst nur wenige Festplattenzugriffe für das Lesen der Indizes zu benötigen. Würde jede Kombination von Zeilen gleich wahrscheinlich als Suchergebnis auftreten können, wäre dies auch optimal.

Overpass API

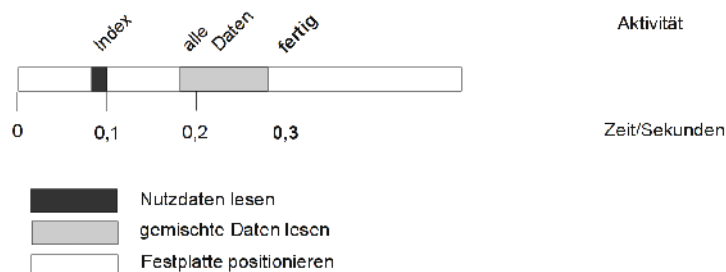
Im Falle der OSM-Daten trifft diese Annahme allerdings nicht zu. Typischerweise werden aus einer Gruppe räumlich beieinander liegender Nodes alle oder zumindest sehr viele Nodes abgefragt. In diesem Fall kann ein Festplattenzugriff pro Node bzw. Zeile deutlich unterboten werden, wenn erreicht wird, dass Daten mit gleichem Index gemeinsam von der Festplatte gelesen werden.

Template DB organisiert daher Daten in Blöcken. Aufgrund der Verhältnisse der Zeitaufwände fürs



Positionieren (0,1 s pro Zugriff) und Lesen (50 MB/s) lohnt sich dies selbst dann noch, wenn nur 0,1% der gelesenen Daten tatsächliche Nutzdaten sind.

Um die räumliche Nähe nutzen zu können, werden die OSM-Daten in Overpass API mit einer raumfüllenden Kurve indiziert; Details folgen im nächsten Abschnitt. Mittels Benchmarks hat sich eine Blockgröße von 512 KB als optimal erwiesen.



Raumfüllende Kurve

Die in OpenStreetMap vereinbarten Genauigkeiten für Geokoordinaten legen eine bestimmte Implementierung nahe: für Breiten- und Längengrad werden jeweils 32-Bit-Integer verwendet, so dass die Schrittweite 1 einer Weite von 10^{-7} Grad entspricht. Damit werden tatsächlich nur 31 Bit für den Breitengrad und volle 32 Bit für den Längengrad verwendet.

Die raumfüllende Kurve entsteht, indem die beiden 32-Bit-Integer zu einem 64-Bit-Integer bitweise verschränkt werden. Die zugehörige Idee stammt aus dem OpenStreetMap-Wiki [10] zum Stichwort Quadtiles. Als Index werden die oberen 32 Bit verwendet.

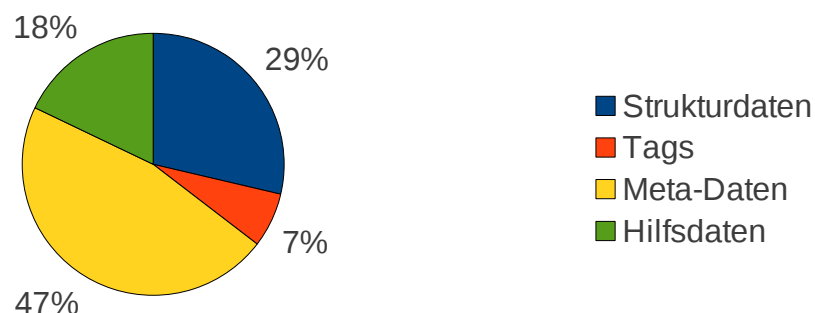
Für die Indizierung von Objekten mit räumlicher Ausdehnung ist es nötig, dieses Verfahren geeignet zu erweitern. Dies geschieht durch die Einführung überlappender Sammelindizes. Wir beschränken uns hier auf Ways. Bei Relations wird das gleiche Verfahren angewendet auf die Member vom Typ Way oder Node, Member vom Typ Relation werden ignoriert.

Haben alle Nodes eines Ways den gleichen 32-Bit-Index, so wird dieser auch als Index des Ways verwendet. Ansonsten wird der ungefähre der Index der südwestlichen Ecke des umschließenden Rechtecks abgerundet, und zwar umso stärker, je größer das umschließende Rechteck ist, und zusammen mit einer Größenangabe der Bbox als Index verwendet. Damit gibt es zu jedem Node-Index immer nur wenige Orte, an denen Ways liegen können, die diese Node referenzieren.

Gliederung nach Typ

Eine erhebliche Rolle für die Geschwindigkeit spielt die Auswahl der verarbeiteten Daten. Da Overpass API nur dann für alle Zwecke benutzbar sein kann, wenn es auch alle OSM-Daten vorhält, folgt es dem Paradigma, dass in einer Abfrage nicht benutzte Daten auch nicht die Geschwindigkeit der Abfrage beeinträchtigen sollen.

Auf der geographischen Ebene wird dies durch die raumfüllende Kurve als Index bereits wirksam erreicht. Aber die verschiedenen Aufgaben erfordern auch inhaltlich unterschiedliche Arten von Daten. Beispielsweise werden für das Rendern der Geometrie weder Tags noch Metadaten benötigt, und für alle gängigen Möglichkeiten der Weiterverarbeitung (z.B. Rendern, Routing, Suche, Linienband-Diagramme) werden zumindest keine Metadaten benötigt. Diese machen jedoch einen Großteil des Datenvolumens aus (Größe der Datenbank vom 23.02.2012):



Die als redundant ausgewiesenen Daten werden dabei grundsätzlich nur benutzt, wenn sie einen Geschwindigkeitsgewinn erbringen. Etwa eine Beschleunigung um den Faktor 3 bringt es aufgrund der Datenmengen allerdings, die Metadaten (Zeitstempel, Changeset und User-ID der letzten Änderung) zu vermeiden. Gleiches gilt für die Tags. Diese Beobachtung ist der eigentliche Anlass dafür gewesen, die Detailstufe der auszugebenden Daten steuerbar zu machen.

Transaktionsalität

Transaktionsalität für Overpass API bedeutet, dass alle Daten einer Abfrage den gleichen Datenstand haben, damit z.B. Verweise zwischen Objekten immer wohldefiniert sind. Die gewählte Implementierung ist darauf ausgelegt, dass pro Datei nur aus je einem zentralen Schreibprozess Daten geschrieben werden.

Dazu dienen *Shadow Copies*: Jeder lesende Prozess speichert zu Beginn der Abfrage eine private Kopie aller Indizes. Zu schreibende Blöcke werden nun an andere, entweder frei gewordene oder neu anzulegende Positionen geschrieben. Ein Block wird erst dann zum Überschreiben freigegeben, wenn der letzte Prozess, der mit seinem Index auf den Block verweist, beendet ist.

Die Serialisierung erfolgt durch einen zentralen Daemon-Prozess. Dieser hält nach, welcher Prozess welche Blöcke kennt und begrenzt als zusätzliche Leistung die Anzahl der maximal parallel möglichen Abfragen.

Geschichte

OSM Server Side Script

Begonnen hat Overpass API unter dem unhandlichen Namen „OSM Server Side Script“ im Dezember 2008. Der Name war Programm: Zu dieser Zeit war der einzige funktionstüchtige Weg, mit OSM-Daten zu hantieren, ein Planet-File herunterzuladen und mehrmals mit Osmosis zu filtern. Planet-Files waren im Dezember 2008 gerade einmal 5 GB groß, aber pro Durchlauf durch die Daten hat Osmosis damals einige Stunden gebraucht, und für viele Aufgaben sind mehrere Durchläufe unerlässlich gewesen.

Als Alternative gab es die Extrakte der Geofabrik. Mein Vertrauen in die Verlässlichkeit von Extrakten aus dritter Quelle war allerdings erschüttert, als der besonders nützliche Extrakt „Welt ohne USA“ mit nur 1,3 GB aus Effizienzgründen weggefallen ist.

Planet-Files gab es einmal pro Woche, Diffs waren zwar möglich, aber hätten einen damit vollkommen ausgelasteten Server an einer ständigen Internet-Verbindung erfordert. Das Planet-File in eine lokale Instanz des Rails-Port zu laden, erwies sich als völlig undurchführbar. Nach gut 2 Tagen Indexerzeugung ist meine Festplatte voll gewesen, aber noch nicht einmal der Import abgeschlossen.

Für etwas anspruchsvollere Abfragen wurde XAPI empfohlen, aber ich habe in mehreren Versuchen es nie geschafft, Daten zu bekommen. Zudem hätte ich auch hier deutlich mehr Daten laden und diese dann mit spezialisierter Software nachbearbeiten müssen.

Ziel waren damals eigentlich die Linienband-Diagramme, aber auf keinem der bestehenden Werkzeuge hätte ein solcher Service aufsetzen können. Daher brauchte es einen Service, der komplexere Abfragen als XAPI verarbeiten konnte und durch eine Skript-Sprache erlaubte, flexibel auf Änderungen im schon damals stark umstrittenen Tagging der ÖPNV-Daten reagieren zu können.

Zunächst habe ich ein vereinfachtes Datenbankschema für MySQL [6] aufgesetzt und wollte dies mit der Skriptverarbeitung als Frontend kombinieren. Die Erwartung war, dass sich mit einer ausgereiften relationalen Datenbank der Rechner gut bis an seine physikalischen Grenzen ausnutzen ließe, nötigenfalls durch Feineinstellung der Datenbank-Parameter. Das Ergebnis war ernüchternd: Der Service war enttäuschend langsam, selbst auf einer damals guten Hardware mit schnellem Prozessor und 4 GB RAM. Eine umfangreiche Analyse im März und April 2009 hat dann zu dem heute verwendeten Konzept geführt. Eigentlich hatte ich die Hoffnung, dies in MySQL realisieren zu können, aber weder in MySQL noch PostgreSQL [13] lässt sich steuern, welche Zeilen physisch zusammenhängend abgespeichert werden.

Overpass API

Einige untergeordnete Aufgaben hat MySQL sogar noch bis Anfang 2011 wahrgenommen: OSM Server Side Script hat wie heute Overpass API als Feature angeboten, Areas vorzuberechnen, um eine Abfrage zu ermöglichen, in welchen Gebieten eine Koordinate liegt. Zusätzlich wurde damals noch nachgehalten, welche Fehler bei der Berechnung aufgetreten sind. Das Feature wurde allerdings nie nennenswert genutzt und ist daher nach der Abtrennung von MySQL im März 2011 nicht wieder neu implementiert worden.

Öffentlich in Betrieb gegangen ist OSM Server Side Script am 28. Mai 2009 mit der Ankündigung auf der Mailingliste talk@openstreetmap.org, auf dem unbenannten Server 78.46.81.38.

Die Nachfrage ist zunächst schwach geblieben. Neben einer doch zurückhaltenden Werbung für den Server und der viel Einarbeitung erfordernden XML-Abfragesprache könnte dafür auch ein anderes, kurz vorher aufgetretenes Ereignis verantwortlich sein: mit Einführung der API 0.6 sind unter anderem Meta-Daten zu den editierten Objekten im Standard-XML-Format sichtbar geworden – und damit ein Service, der keine Meta-Daten anbietet, ausgerechnet mehrere Wochen nach deren Einführung gestartet.

Im Rest des Jahres 2010 habe ich viel am Linienband-Generator gearbeitet: Dank der Anregungen einiger Benutzer hat der Linienband-Generator am Ende ein ansehnliches Niveau erreicht, auch wenn er seine Grenzen in den nicht vereinbaren Vorstellungen verschiedener Teile der OSM-Community über das ÖPNV-Tagging gefunden hat.

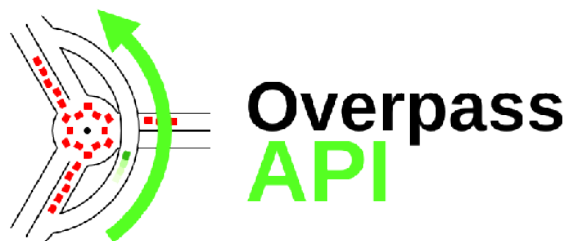
Overpass API

Ebenfalls im Jahr 2010 hat sich abgezeichnet, dass eine komplette Neuentwicklung des Quellcodes unumgänglich ist: Das Backend basierte noch auf dem Testcode, um auszuloten, ob sich ein Ersatz von MySQL durch eine Eigenentwicklung gelohnt hätte. Die Ursache einiger Bugs ließ sich nicht mehr mit verhältnismäßigem Aufwand finden.

Das neue Backend ist dann gleich darauf ausgelegt worden, generisch für beliebige Datentypen und auch unabhängig von Overpass API zu funktionieren. Auch bei den übrigen Komponenten ist es mit dem Wissen, welchen Leistungsumfang die Software am Ende haben soll, möglich geworden, eine besser passende Architektur zu wählen.

Eine vorläufigen Abschluss hat die Neuimplementierung im März 2011 mit der Veröffentlichung der Version 0.6 gefunden: endlich hatte der Quellcode einen Reifegrad erreicht, bei dem einem Dritten zugemutet werden konnte, mit dem Quellcode zu hantieren. Wesentlich ist dabei nicht so sehr ein ästhetischer Quellcode gewesen (bis heute ist der Quellcode deutlich zu wenig kommentiert), sondern vorwiegend systematische Unit- und Integrations-Tests, damit sich auch nach Refaktorisierungen und Erweiterungen noch nachvollziehen lässt, dass der Code keine neuen Fehler enthält.

Einen Aufbruch hat es dagegen bedeutet, über das Projekt auf der SOTM-EU im Juli 2011 in Wien vorzutragen; dies ist zugleich Anlass gewesen, dem Projekt ein Logo zu geben:



Overpass API ist vielen erst seitdem bekannt geworden und überraschend positiv aufgenommen worden. Vor allem hat es aber eine Menge mehr oder weniger umsetzbarer Wünsche gegeben. Den stärksten Nachfrageschub hat dabei der XAPI Compatibility Layer bewirkt, aber auch dieser hat nicht

Overpass API

gereicht, um die Lastgrenzen des Servers auszuloten; vermutlich haben die häufigen Meldungen über die Überlastung der alten XAPI-Server zu einer wesentlichen Überschätzung des XAPI-Bedarfs im Projekt geführt.

Seitdem ist das Ziel die Version 0.7: Anspruch an diese Version ist, alle auf der SOTM aufgeworfenen Ideen in praktisch nutzbare Features umzusetzen. Mit der Einführung einer kompakteren Abfragesyntax als die XML-Syntax sind dabei alle Ideen prinzipiell umgesetzt; als letzten Schritt bleibt noch die Suche nach Ways und Relations in einer Bounding Box zu beschleunigen.

Zukunft

OpenStreetMap stößt dank seiner großen Popularität bald an eine technische Grenze: die Anzahl der Nodes ist dann zu groß für die heutzutage üblichen 32-Bit-Zahlen mit Vorzeichen. Da die Umstellung auf 64-Bit-Ids Arbeit in fast allen Teilen des Quellcodes erfordert und eine spürbare Verlangsamung bewirken könnte, wird der Versionssprung von 0.7 auf 0.7.1 ausschließlich diese Änderung beinhalten.

Nach dieser rein technischen Maßnahme wird das übergreifende Thema der 0.7.x-Serie die Öffnung für weitere Quellen und ausführbare Tag-Semantik sein. Die Öffnung für weitere Quellen löst dabei eine Reihe chronisch wiederkehrender Probleme der Community:

Sollen große Imports (z.B. Stadtgrenzen, Gebäudeumrisse) durchgeführt werden? Einerseits macht dies diese Daten für die OSM-Infrastruktur verfügbar, erlaubt also das Rendering in zahlreichen Karten, die Aufnahme in viele Suchmaschinen, und vereinfacht so manche kreative Idee so sehr, dass sie tatsächlich umgesetzt wird. Andererseits sind solche Imports häufig ihrem Wesen nach für individuelle Mapper nicht bearbeitbar – eine Stadtgrenze ist weder auf einem Luftbild noch vor Ort sichtbar. Durch das Mischen auf dem Server wird der Konflikt entschärft: über die Schnittstelle sind die Importe so gut wie genuine Daten der OSM-Datenbank verfügbar, beeinträchtigen aber nicht das Mapping, weil sie beim Mappen nicht sichtbar sind.

Overpass API wird insbesondere aber auch zwei weitere nicht naheliegende Quellen erschließen. Zum einen bietet es sich für komplizierte Gefechtslagen beim Tagging an, verschiedene Taggings bei der Ausgabe einer Query durch Berechnung zu bereinigen – die Originaldaten werden nicht nur nicht verändert; jede der Parteien kann auch die Daten in dem für sie gewünschten Format erhalten. Dies ist oben als ausführbare Tag-Semantik umschrieben.

Zum zweiten ist es bis heute schwierig, Änderungen in einem kleinen Gebiet zu überwachen oder nachzuvollziehen. Dafür sind Diffs für beliebige Abfragen zwischen zwei Zuständen der Datenbank zu zwei beliebigen verschiedenen Zeitpunkten sowie Feeds zu Änderungen für beliebige Abfragen geplant, aber noch nicht auf Umsetzbarkeit geprüft.

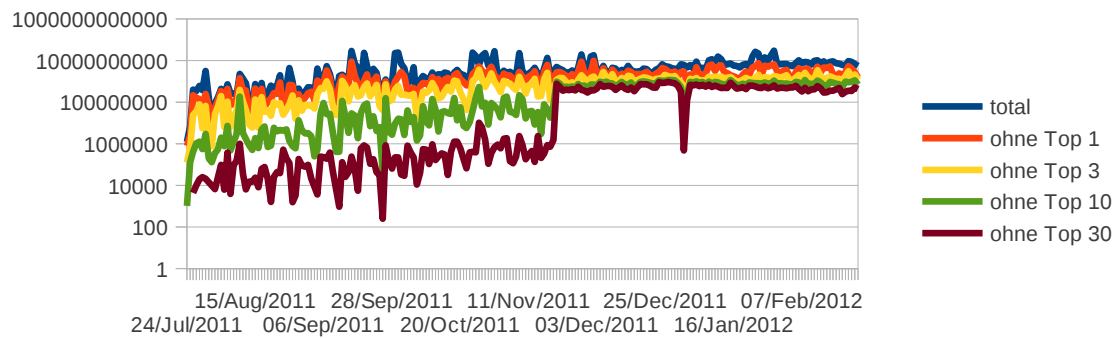
Ich erwarte mindestens ein Jahr Dauer für die vollständige Umsetzung dieser Ziele in der Version 0.8.

Nutzung

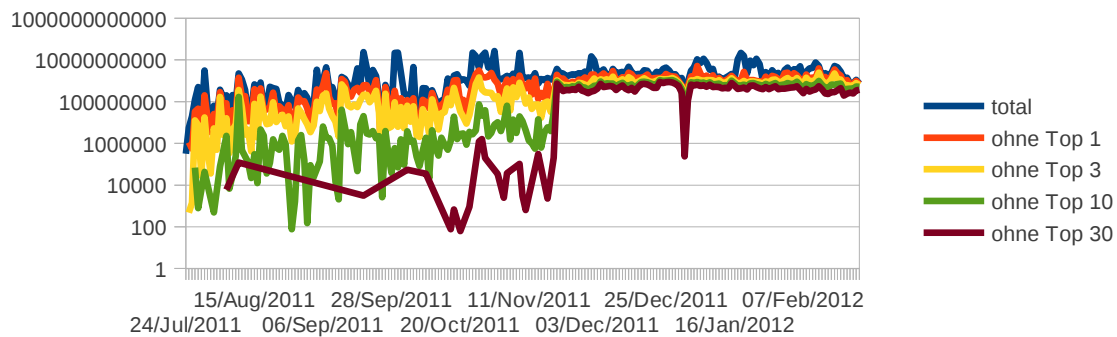
Overpass API soll experimentelle Nutzungen der Daten ausdrücklich ermuntern. Daher sind regelmäßige Nutzungen mit geringem Datenvolumen ebenso legitim wie spontane Nutzungen mit sehr großen Datenvolumina. Sollte es allerdings zu einer Ressourcenknappheit kommen, ist es leichter, mit wenigen Intensivnutzern zügig andere Nutzungsmuster zu vereinbaren als eine große Anzahl Wenignutzer auszusperrern. Die folgenden Diagramme enthalten daher stets mehrere Linien: Die oberste Linie enthält den Gesamtwert, die nächste den Wert ohne den intensivsten Nutzer, die dritte ohne die drei intensivsten Nutzer, die vierte ohne die zehn intensivsten Nutzer und die unterste ohne die dreißig intensivsten Nutzer. Alle Werte sind pro Tag ausgewiesen, als Benutzer wird jede IP-Adresse pro Tag einmal gezählt. Die Diagramme sind logarithmisch skaliert:

Zunächst die über die API ausgelieferten Datenmengen summiert auf beiden Servern overpass-api.de und overpass.osm.rambler.ru:

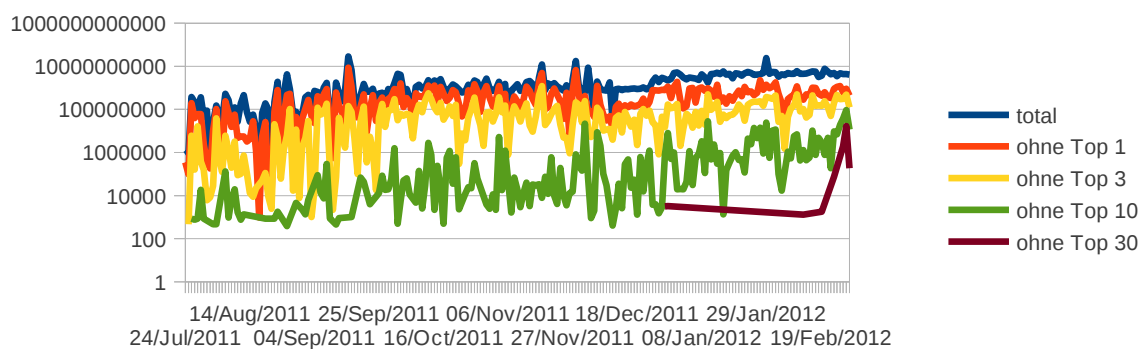
Overpass API



Darunter entfällt auf den XAPI Compatibility Layer:

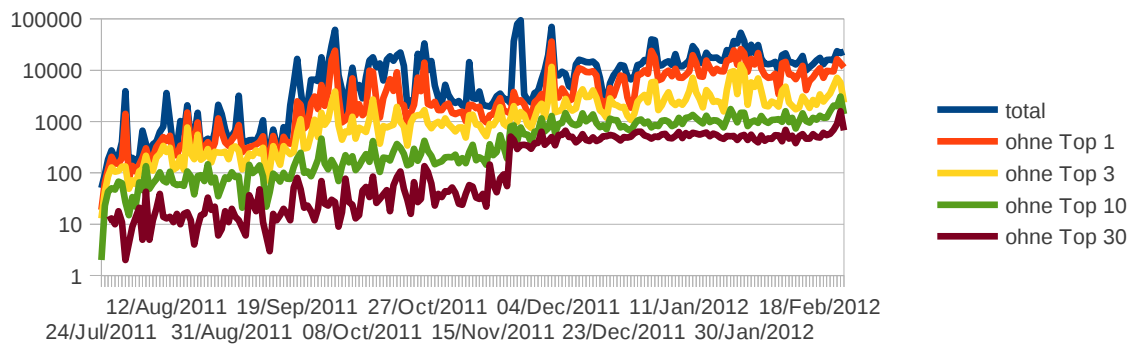


Auf die nativen Abfragesprachen entfallen:

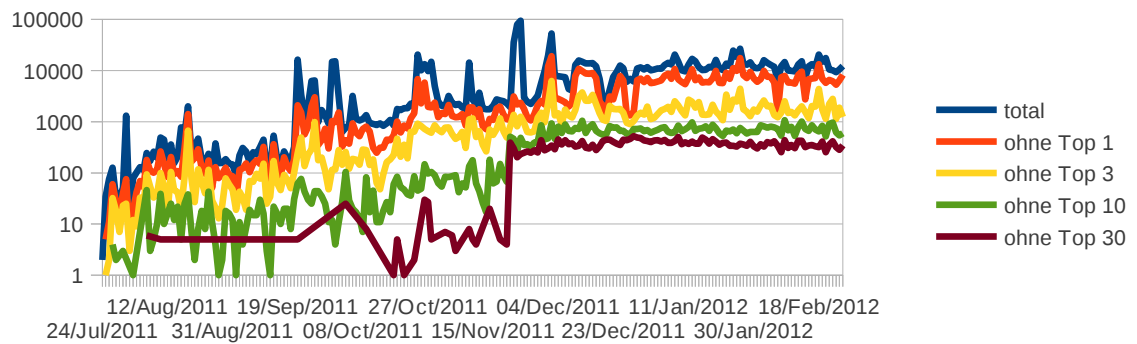


Anzahl der Abfragen gesamt:

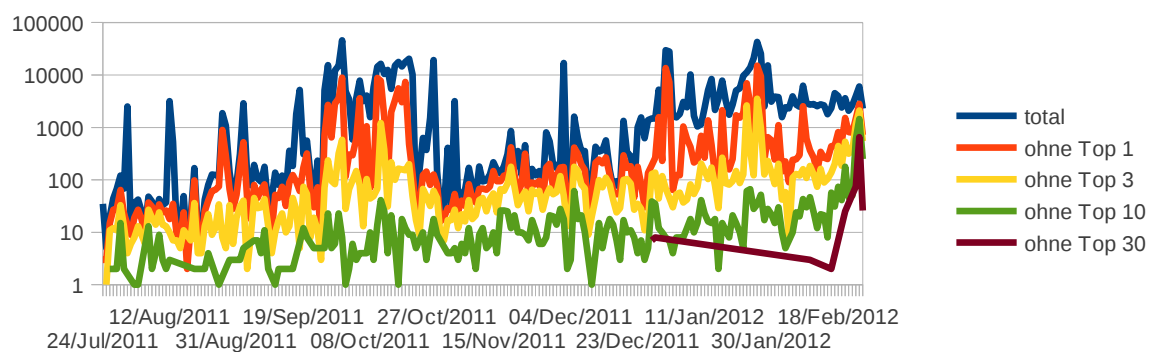
Overpass API



Anzahl der Abfragen im XAPI-Format:

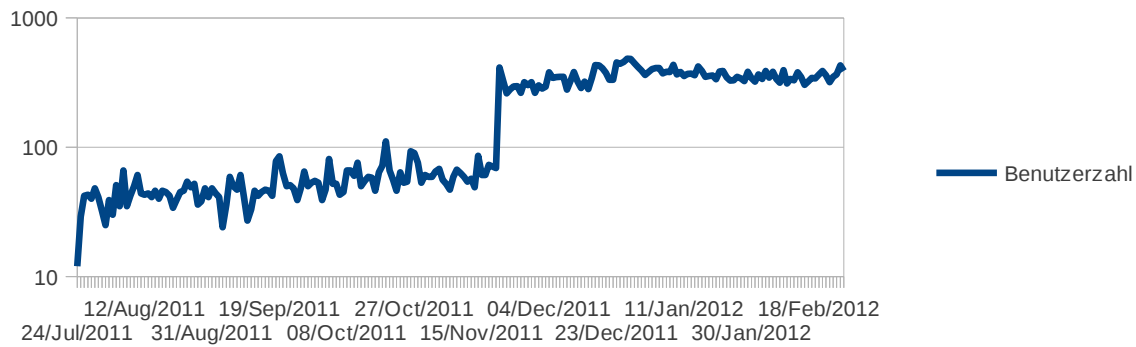


Anzahl der nativen Abfragen:

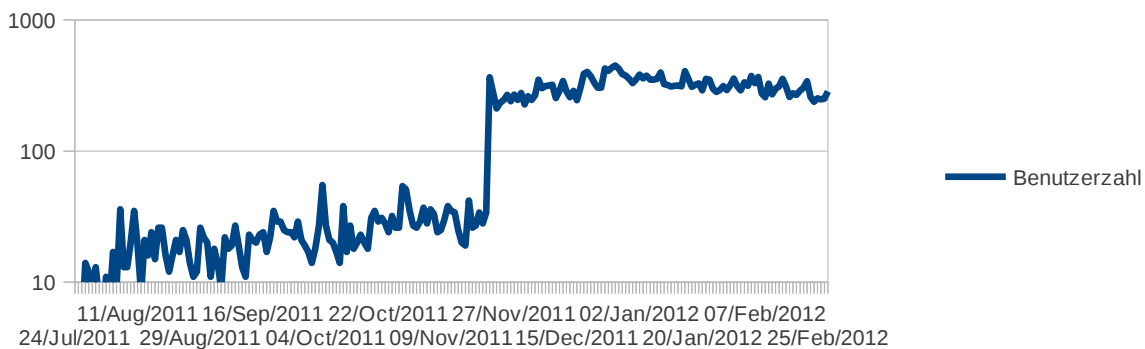


Bei den Benutzerzahlen ergibt sich als Gesamtbild:

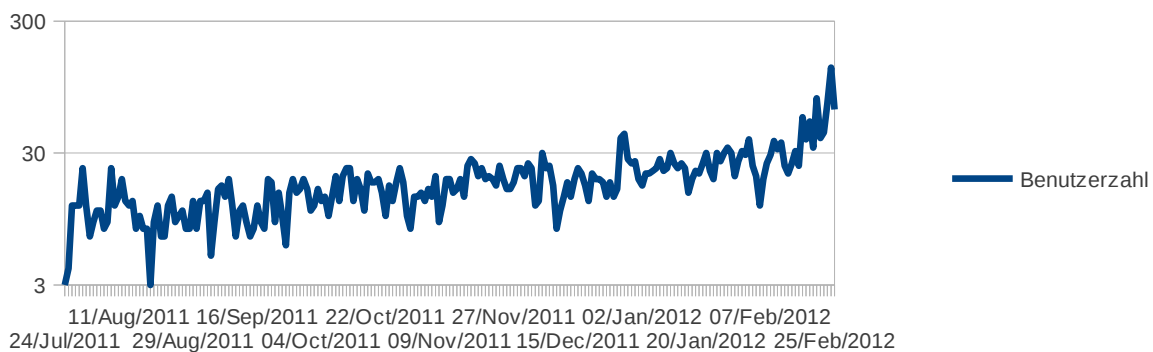
Overpass API



Für XAPI ergibt sich:



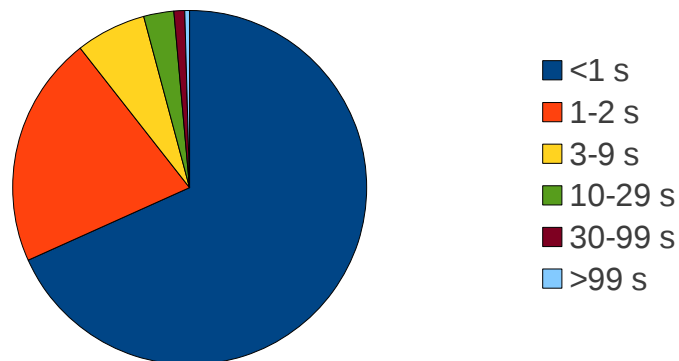
Für native Abfragen ergibt sich:



Zwar leistet der XAPI Compability Layer als Brückentechnologie hin zur nativen Syntax gute Dienste; die Nutzung der nativen Syntax hat mittlerweile ein breites Nutzerfundament. Dass beim XAPI Compability Layer der Schwerpunkt auf Wenignutzern liegt, dürfte darauf zurückzuführen sein, dass mehrere, mir nur teilweise bekannte Drittanwendungen XAPI-Abfragen generieren. Es bedeutet aber auch, dass der XAPI Compability Layer wohl auch mittelfristig weiter gewartet werden muss.

Overpass API

Zuguterletzt sei die Leistungsfähigkeit untersucht (Summe über die Daten im Januar), auf overpass-api.de. Für overpass.osm.rambler.ru wird die Laufzeit leider nicht mitgeloggt.



Über 90% aller Abfragen werden in weniger als 3 Sekunden und über 99% aller Abfragen in weniger als 30 Sekunden beantwortet. Bei der derzeitigen Nutzungsmustern ist also keine Überlastung zu befürchten.

Kontakt zum Autor:

Dr. Roland Olbricht
Hohe Straße 2
53119 Bonn
0228/36006956
roland.olbricht@gmx.de

Literatur

- [1] *Böhme, Christoph*: NOVAM-Viewer, <http://mappa-mercica.org/novam/>
- [2] *Clark, James et al.*: Expat, <http://expat.sourceforge.net/>, 2007
- [3] *Coar, K., Robinson, D.*: CGI 1.1, RFC 3875, <http://tools.ietf.org/html/rfc3875>
- [4] *Dees, Ian*: jXAPI, <https://github.com/iandees/xapi-servlet>
- [5] *FSF (Hrsg.)*: GNU Autotools, <http://www.gnu.org/software/automake/>
- [6] *MySQL AB, Sun Microsystems, Oracle Corporation*: MySQL 5.0.75, 2005
- [7] *Olbricht, Roland*: Overpass API, Quellcode: [git@github.com:drolbr/Overpass-API](https://github.com/drolbr/Overpass-API), 2012
Dokumentation: http://wiki.openstreetmap.org/wiki/Overpass_API, laufend aktualisiert
- [8] *Olbricht, Roland*: Linienband-Generator, http://overpass-api.de/public_transport.html
- [9] *OSGeo (Hrsg.)*: OpenLayers, <http://openlayers.org/>
- [10] *OSMF (Hrsg.)*: OpenStreetMap-Wiki, <http://wiki.openstreetmap.org/>, laufend aktualisiert
- [11] *Ramm, Frederik; Topf, Jochen*: OpenStreetMap, Berlin, 2008.
- [12] *Scholz, Immanuel et al.*: JOSM, <http://josm.openstreetmap.de/>
- [13] *Stonebraker, Michael et al.*: PostgreSQL, <http://www.postgresql.org/>
- [14] *Stroustrup, Bjarne*: The design and evolution of C++, 1994
- [15] *W3C (Hrsg.)*: XML-Spezifikation, <http://www.w3.org/TR/REC-xml/>, 2008

OpenStreetMap und R

Thomas

Integration freier Geodaten in die Lingua franca der Statistik

Dieser Vortrag stellt eine Schnittstelle zwischen der freien Statistiksoftware R und der OpenStreetMap-Datenbank vor. Das R Add-On Paket "osmar" ermöglicht einen einfachen Zugriff auf OSM-Daten aus verschiedenen Quellen, ein -- für den R-Benutzer -- gewohntes Arbeiten mit diesen Daten und das Konvertieren der OSM-Daten in Objekte anderer R Add-On Pakete.

Das freie Projekt R hat sich in den letzten Jahren zu einer ernstzunehmenden Softwareumgebung in der akademischen Statistik entwickelt. Durch die Möglichkeit Add-On Pakete zu schreiben, bietet R inzwischen einen gewaltigen Funktionsumfang in allen möglichen Teilbereichen der Statistik und Datenanalyse. Die Verknüpfung eigener Daten mit Geodaten stellt in vielen Fällen eine Bereicherung der Analyse dar. Und auch durch die statistische Auswertung der Geodaten ansich, können sich spannende Analysen und neue Erkenntnisse ergeben. Aus diesem Grund, bietet das Add-On Paket "osmar" eine Integration der OpenStreetMap-Daten in das R-Projekt an.

Dieser Vortrag gibt eine kurze allgemeine Einführung in das R-Projekt und stellt dann das Add-On Paket "osmar" vor. Anhand von Beispielanwendung werden die drei Kernpunkte des Paketes dargestellt: (1) Der Zugriff auf OSM-Daten aus unterschiedlichen Quellen (API v0.6 und Planet files). (2) Das einfache Arbeiten mit diesen Daten in gewohnter R Manier; d.h., das Begutachten, Zusammenfassen und Visualisieren, sowie das Suchen und Bilden von Teildatensätzen. (3) Das Konvertieren der OSM-Daten in Objekte anderer Add-On Pakete; zum Beispiel in die Pakete "sp" und "igraph", welche Strukturen und Analysemethoden für räumliche Daten bzw. Graphen bereitstellen.

MapFish, ein OSGeo WebGIS 2.0 Framework

ebeleo

MapFish ist ein Framework, welches die Bildung von individuell gestalteten und erweiterbaren Web-GIS Anwendungen vereinfacht.

MapFish ist so gestaltet, dass es einfach zu benutzen ist, entweder als selbständige Anwendung oder als Bestandteil einer bestehenden Webseite. Als selbständige Anwendung ist MapFish via einige Parameter konfigurierbar und bietet schnell ein funktionales Web-GIS. MapFish kann auch in bestehenden Webseiten wie CMS oder Informations System orientierte Anwendungen ohne Aufwand integriert werden.

Der Client Teil ermöglicht es via OpenLayers2, ExtJS3 und GeoExt4, weiterführende WEB2.0 Funktionalitäten einzubauen.

Der Server Teil stellt in verschiedene Programmiersprachen flexible Prozesse zur Verfügung (Routing, Suchmaschine, thematische Kartografie, usw).

Mapbender3 – Next Generation PHP Mapping

Christian Wygoda

Mit Mapbender hat sich in den letzten neun Jahren eine PHP-basierte WebGIS-Anwendung erfolgreich etabliert. Mit den Jahren ist natürlich auch die Komplexität der Anwendungen und des zugrundeliegenden Codes gewachsen, so dass Mapbender in der Version 2 inzwischen aus etwas über einer Million Zeilen Code besteht.

Die Architektur spiegelt dabei die Ansichten des frühen ersten Jahrzehnts wieder, und hat dazu geführt, dass die Entscheidung für einen Neustart auf einem leeren Blatt Papier fast schon logisch erschien.

Anfang 2011 wurde diese Entscheidung dann auch in die Tat umgesetzt und die grundlegenden Entscheidungen zur Architektur von *Mapbender3* gefällt. Die weitreichendste Entscheidung war sicherlich die Wahl eines Web Application Frameworks, welche zugunsten von *Symfony 2* getroffen wurde. Damit steht ein modernes und schnelles PHP-Framework als Grundlage für die Kernkomponenten als auch Erweiterungen – seien diese als Plugins für den Mapbender3-Kern gedacht oder projektspezifisch – von Mapbender3 zur Verfügung. Mit seiner Bundle-Architektur zur Kapselung von Funktionalitäten [1], dem durchdachten Konzept und der verständlichen Dokumentation [2] erleichtert es den Zugang zur Entwicklung für „alte“ und neue Entwickler.

Ein Ziel ist es dabei, die im Mapbender 2 vorhandene Abhängigkeit zu PostGIS als Backend für Datenhaltung und Geometrieoperationen zu lösen. Mit der Integration von Doctrine DBAL und ORM [3] als Datenbankabstraktionsebene steht eine ganze Reihe von Datenbanksystemen zur Datenhaltung zur Verfügung, so z.B.

- PostgreSQL,
- MySQL
- SQLite
- Oracle.

Für den Bereich der Geometrieoperationen steht momentan noch keine native Lösung für PHP zur Verfügung, die Entwicklungen der PHP-Bindings von GEOS [4] und PROJ.4 bzw. deren Existenz in Mapscript [5] werden aber evaluiert und beobachtet.

Clientseitig wurde die mit Mapbender 2.5 begonnene Einführung von jQuery als JavaScript-Framework konsequent umgesetzt: Elemente einer Mapbender3-Anwendung benutzen die jQuery-UI Widget Factory [6] zur Erzeugung von clientseitiger Funktionalität. Damit steht auch die komplette Widget-Bibliothek von jQuery UI – wie Dialogen, Tabs – in allen neuen Anwendungen zur Verfügung.

Als Kartenviewerkomponente wird OpenLayers mit den jQuery-Bindings von MapQuery [7] verwendet, was die Nutzung der teilweise unterschiedlichen API-Konzepte von jQuery und OpenLayers erleichtert.

Mit den Entwurfsmustern von Symfony und der Widget Factory von jQueryUI sind damit immer wiederkehrende Programmiermuster vorgegeben, die zwar einen grundlegenden Paradigmenwechsel zur Mapbender-2-Entwicklung darstellen, aber nach kurzer Zeit das Verständnis für alle Bestandteile einer Mapbender3-Anwendung erleichtern.

Mapbender3 – Next Generation PHP Mapping

Bis Anfang 2011 wurden die wichtigsten Grundkomponenten für Mapbender3-Anwendungen entwickelt, so dass mit den Mapbender3-Bundles bereits Anwendungen umgesetzt sind, so zum Beispiel auch das Geoportal.DE:

- CoreBundle: Die gesamte Grundfunktionalität für Mapbender3-Anwendungen findet sich im CoreBundle, ebenso die Verknüpfungspunkte für Erweiterungsbundles.
- WmsBundle: WMS-Dienste stellen weiterhin die am häufigsten genutzten Kartendienste für Mapbender3-Anwendungen, die Funktionalität für Nutzung und Verwaltungskomponenten finden sich daher in diesem Erweiterungsbundle.
- WmtsBundle: Gerade im letzten Jahr mehrten sich Anfragen für die Verwendung von WMTS-Diensten für performante Kartendienste, gerade für Bildschirmfüllende Kartenviewer. Daher ist das WmtsBundle ähnlich dem WmsBundle in stetiger Entwicklung.

Die bisherige Entwicklung konzentrierte sich dabei auf die Frontendentwicklung, so dass eine der Stärken von Mapbender 2 noch nicht in Mapbender3 wieder umgesetzt ist – die Möglichkeit zur Verwaltung von Anwendungen im Browser. Bisher werden Anwendungen noch in YAML-Dateien konfiguriert. Seit Anfang 2011 befindet sich jedoch mit dem ManagerBundle eine umfassende Backendanwendung zur Verwaltung einer Mapbender-Instanz in Entwicklung. Das Hauptaugenmerk liegt dabei auf einer verbesserten Benutzerführung.

Wo in Mapbender 2 für jedes Element, egal ob es „auch“ nur ein Button mit Text und Icon war, ein umfassendes Formular mit 21 Feldern angezeigt wurde, sollen in Mapbender3 die Elemente spezifische Formulare anbieten, die nur die Informationen abfragen, die auch nötig sind. Um einen Wildwuchs bei der Gestaltung von Element-Formularen zu vermeiden, wird es neben den Beispielen der Kernelemente noch einen passenden Styleguide geben.

Ähnlich werden die Verwaltungstools für Dienste und Benutzer gestaltet werden, um ein einheitliches Verwaltungswerkzeug für Mapbender3-Instanzen zu präsentieren. Das Mapbender-Dev-Team plant, vor der FOSS4G 2012 die erste integrierte Fassung des ManagerBundles zu präsentieren, um damit die technischen Grundlagen für die aus Mapbender 2 bekannten Möglichkeiten für Mapbender3 komplett gelegt zu haben, und dann diese weiter zu entwickeln.

Kontakt zum Autor:

Christian Wygoda
WhereGroup GmbH & Co. KG
Eifelstr. 7, 53119 Bonn
0228/9090380
christian.wygoda@wherogroup.com

Quellen

- [1] http://symfony.com/doc/current/book/page_creation.html#the-bundle-system
- [2] <http://symfony.com/doc/current/index.html>
- [3] <http://symfony.com/doc/current/book/doctrine.html>
- [4] <https://github.com/strk/geos/tree/master/php>
- [5] <http://mapserver.org/mapscript/php/index.html>
- [6] <http://wiki.jqueryui.com/w/page/12138135/Widget%20factory>
- [7] <http://mapquery.org/>

SHOGun - Spring, Hibernate, OpenLayers, GeoExt und weitere

Christian Mayer, terrestris GmbH & Co. KG

Zusammenfassung

Der Vortrag stellt ein komplettes WebGIS aus einem Guss basierend auf modernen Open Source Web-Frameworks vor.

Präsentiert wird die Umsetzung einer komplexen WebGIS-Lösung bestehend aus einer Java-Middleware, einem konfigurierbarem WebGIS-Client auf Basis von JavaScript sowie einer Nutzerverwaltung und diversen Sicherheitsmechanismen. Die Kommunikation zwischen Client und Server wird komplett über AJAX und (Geo)-JSON realisiert. Die Anwendung wurde unter anderem im Rahmen des Forschungsprojektes MoMo [1] entwickelt, wo die Software für integriertes Wasserressourcen-Management in Zentralasien eingesetzt werden soll.

SHOGun selber setzt konsequent auf die folgenden anerkannten Open Source-Frameworks:

- Spring Framework [2]
- Hibernate [3]
- Hibernate Spatial [4]
- ExtJS [5]
- OpenLayers [6]
- GeoExt [7]

Durch den Einsatz dieser renommierten und hochqualitativen Frameworks wird die Nachhaltigkeit, Wartbarkeit sowie die Qualität der Anwendung gewährleistet. Dazu basiert die Anwendung auf modernen Sicherheitsstandards und das clientseitige Framework ExtJS sorgt für eine weitestgehende Browserkompatibilität mit Standardbrowsern.

Einleitung

In einer verteilten, dienstebasierten Web-Architektur ist es heutzutage üblich, dass die (Web-)Clientanwendungen auf eine geeignete Middleware zugreifen und nicht direkt mit der Datenbank kommunizieren. Solch eine Middleware agiert hierbei als zentrale Komponente zur Steuerung aller Anfragen und Antworten, die zwischen Client und Datenbank ablaufen. Dennoch können die Vorteile einer verteilten Architektur, d. h. die Nutzung weiterer dezentralen Dienste genutzt werden. Zusätzlich existiert durch die Middleware eine effektive und sichere Zugangskontrolle sowie die Einbindung benutzerspezifischer Filter.

Dazu kann über SHOGun ein benutzergesteuerter WebGIS Client bestehend aus entsprechenden Werkzeugen, Panels und Diensten, zusammengestellt werden.

SHOGun

Die Architekturskizze zeigt den Aufbau von SHOGun. Es handelt sich um eine klassische Client-Server Anwendung. Die eigentliche Middleware ist im hellblauen Bereich rechts zu sehen. SHOGun arbeitet als Servlet im Apache-Tomcat ab Version 7. In SHOGun können beliebige weitere Servlets integriert werden. Die gesamte Kommunikation erfolgt über das HTTP-Protokoll, mit XML oder auch (Geo)JSON als Austauschformate, welche im Client direkt verarbeitet werden können. SHOGun sichert

SHOGun - Spring, Hibernate, OpenLayers, GeoExt und weitere

dabei alle Anfragen, Daten und Funktionen zwischen Client und Server ab. Außerdem können auch externe (OGC-)Dienste, wie beispielsweise WMS und WFS auch gegen unberechtigte Zugriffe abgesichert werden. Diese Funktion befindet sich z. Zt. In der Implementierung.

Die Eingliederung externer Dienste auf Java-Basis (z.B. GeoServer, vgl. oder einem CSW-Metadatenknoten, z.B. Geonetwork Open Source, vgl. Abbildung) direkt in das Setup ist theoretisch möglich.

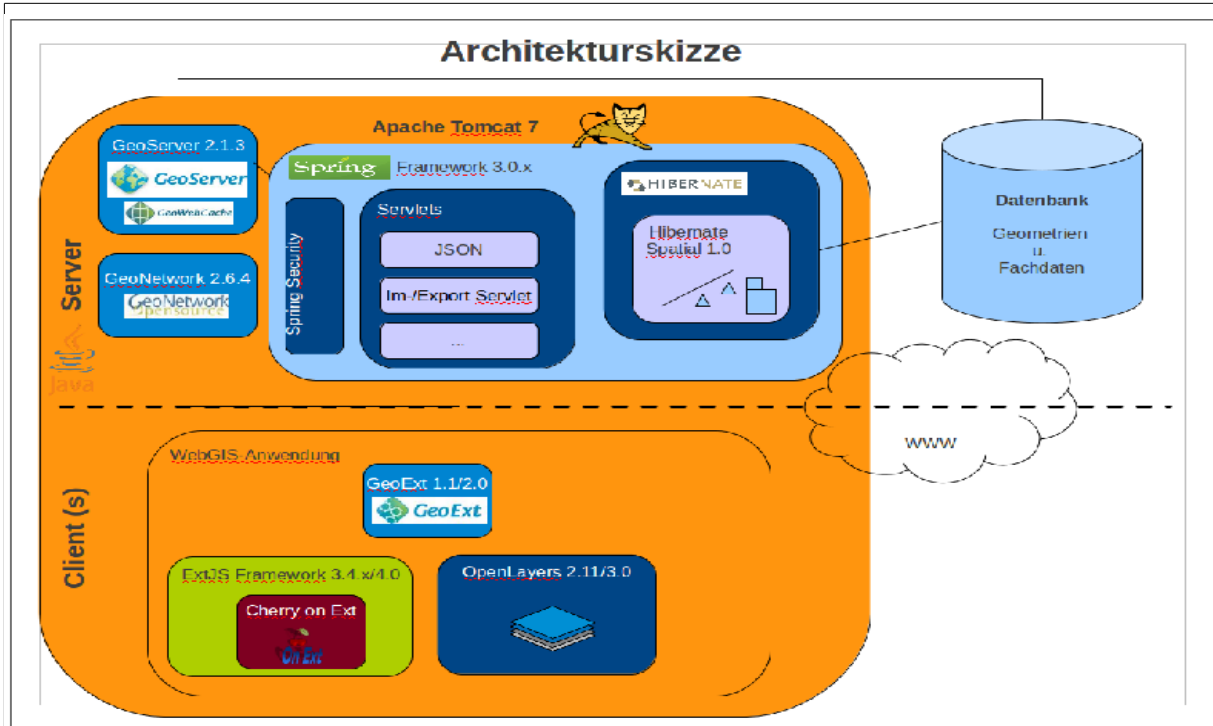


Abbildung 1: schematische Skizze einer Web-Architektur mit Middleware und Client

Aufgrund der offenen Architektur sind die Komponenten, wie Client oder Datenbank ebenfalls austauschbar.

Spring / Spring MVC

Die in SHOGun beinhaltete Middleware ist auf Basis des Java Spring MVC Frameworks entwickelt. Das Spring-Framework, kurz Spring, vereinfacht die Entwicklung von professionellen, ganzheitlichen Java EE Anwendungen und stellt dabei die Entkopplung der einzelnen Komponenten in den Vordergrund. Spring MVC im Speziellen stellt eine Erweiterung von Spring dar, um Webanwendungen zu implementieren, welche auf Servlets basieren. Mit Spring Security existiert eine zusätzliche Erweiterung des Frameworks, welche zur Absicherung sowie zur Benutzer-, Rechte- und Rollenverwaltung dient und ist ebenfalls in SHOGun inbegriffen.

Hibernate & Hibernate Spatial

Der Datenzugriff erfolgt innerhalb von SHOGun über die Datenbankabstraktionsschichten Hibernate/Hibernate Spatial.

Hibernate ist ein Persistenz- und ORM (object-relational-mapping) – Framework. Es ermöglicht die Abbildung von Datenbankstrukturen in Java-Klassen und umgekehrt. Dadurch können gewohnte Java-Objekte (POJOs) in Datenbanken gespeichert werden, bzw. aus Datenbankobjekten POJOs erzeugt werden. Durch die Kompatibilität von Hibernate mit diversen Datenbanken (z.B. PostgreSQL, Oracle,

SHOGun - Spring, Hibernate, OpenLayers, GeoExt und weitere

etc.) erfolgt eine Abstraktion des Datenbankzugriffes, was die Flexibilität der gesamten Architektur erhöht.

Gegenüber der direkten Kommunikation mit der Datenbank hat Hibernate den weiteren Vorteil des Caching. Die Datenbankobjekte werden im Arbeitsspeicher vorgehalten und für entsprechende Abfragen ist so keine direkte Kommunikation mit der Datenbank mehr nötig. Über Hibernate Spatial stehen auch räumliche Datentypen sowie GIS-Operationen zur Verfügung, wie sie beispielsweise in PostGIS bekannt sind.

Client

Der WebGIS-Client basiert auf den Frameworks ExtJS 3.4.x, OpenLayers 2.11 und GeoExt 1.1 und ist nahtlos in SHOGun integriert. Derzeit ist eine Eigenlösung der Firma terrestris auf Basis der zuvor genannten Bibliotheken im Einsatz. Aufgrund der Austauschbarkeit der Komponenten in SHOGun können auch alternative JavaScript-Clients eingesetzt werden (siehe dazu das Kapitel „Ausblick“).

terrestris

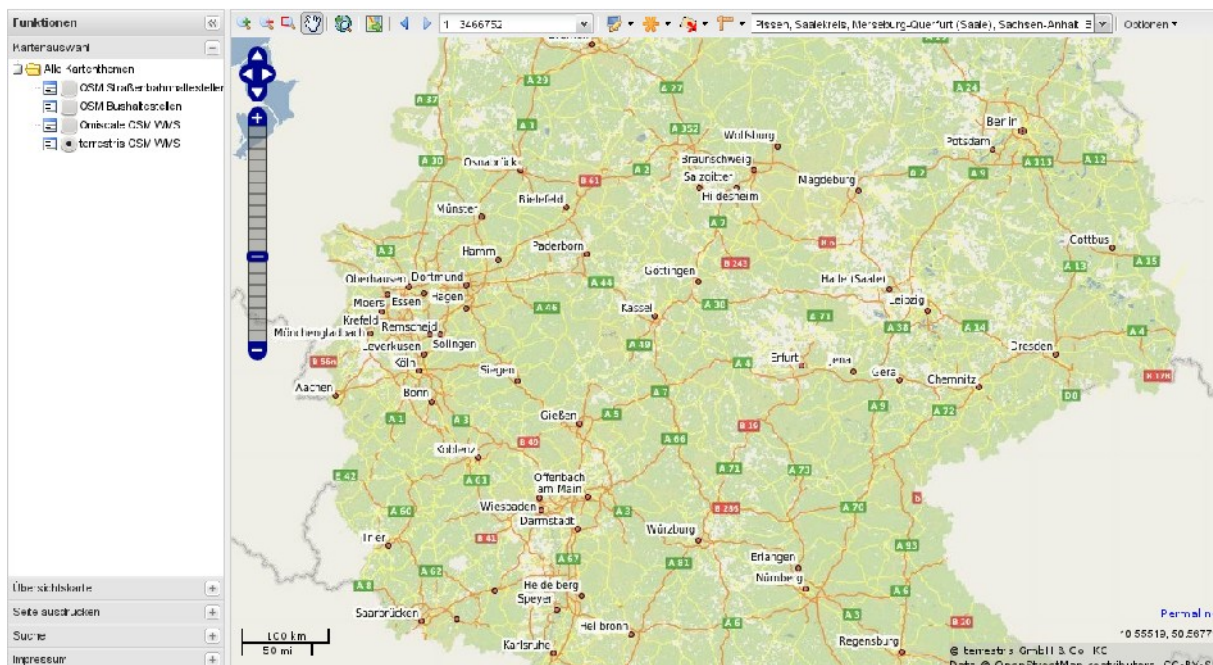


Abbildung 2: Screenshot des SHOGun WebGIS Clienten , Anwendung unter "ows.terrestris.de"

Zur Konfiguration existiert eine Administrationsoberfläche, in der neue Benutzer und Benutzergruppen angelegt und Rechte vergeben werden können. Somit können sämtliche Client-Funktionen und daran geknüpfte Interaktionen mit der Datenbank über SHOGun gesteuert werden. Das heißt Benutzer sehen nach dem Einloggen im Client nur die Funktionen, Dienste und Daten, die Sie gemäß der Ihnen zugewiesenen Rechte sehen und benutzen dürfen. Natürlich existiert auch ein freier Zugang, der einen WebGIS-Client ohne die Notwendigkeit sich einzuloggen anbietet.

Ausblick

SHOGun wird derzeit im Rahmen des Projektes Momo [1] weiterentwickelt. Aktuell wird die OWS-Absicherungsschicht implementiert. Dazu wird eine Komponente entwickelt, die es erlauben wird, Geodaten hochzuladen (bzw. auf einem Server abzulegen) und über die GeoServer REST-API daraus automatisch einen WMS/WFS Dienst zu erstellen. Clientseitig wird dabei ein auf ExtJS basierender

SHOGun - Spring, Hibernate, OpenLayers, GeoExt und weitere

SLD-Styler weiterentwickelt und integriert. Dadurch erhält der Benutzer die Möglichkeit, Kartendienste selber auszugestalten.

Des Weiteren wird z.Zt. geprüft, ob clientseitig das Framework GeoExplorer [8] integriert werden kann und somit eine gesteigerte Nachhaltigkeit sowie weitere Synergieeffekte erzielt werden können.

SHOGun wird als Open Source Software veröffentlicht, derzeit wird noch geprüft unter welcher Lizenz die Software gestellt wird.

Kontakt zum Autor:

Christian Mayer
terrestris GmbH & Co KG
Pützchens Chaussee 56, 53227 Bonn
+49 – (0)228 962 899 53
mayer@terrestris.de

Literatur

- [1] <http://www.iwrm-momo.de/>
- [2] <http://www.springsource.org/>
- [3] <http://hibernate.org>
- [4] <http://www.hibernate.org>
- [5] <http://www.sencha.com>
- [6] <http://www.openlayers.org>
- [7] <http://www.geoext.org>
- [8] <https://github.com/opengeo/GeoExplorer>

Ausschreibungen und Open Source Software

Arnulf Christl

Wie Open Source Software in Ausschreibungen berücksichtigt werden kann

Dieser Vortrag gibt eine kurze Einführung in die Besonderheiten von Open Source (Lizenzen, Gewährleistung, Kosten, Investitionssicherheit) und wie diese bei der Erstellung von Ausschreibungen berücksichtigt werden.

Ausschreibungspraktiken sowohl in der öffentlichen Verwaltung als auch in der Privatwirtschaft sind in einer Zeit entstanden, als quelloffene Software noch etwas Besonderes war. Heute ist Open Source zwar bereits überall in der Standard-IT im Einsatz, wird aber vor allem in Nischenbereichen (wie der Geoinformationsverarbeitung) wenig berücksichtigt. Es wird immer wieder diskutiert, ob in einer Ausschreibung explizit Open Source oder auch spezielle Software-Produkte angefordert werden können oder nicht. Des weiteren ist die Preiskalkulation und damit die Vergleichbarkeit von Angeboten unterschiedlicher Hersteller oder Dienstleister wesentlich schwieriger, als es mit festen, sogenannten "Commercial Off-the-Shelf" Produkten (einsatzfertige Produkte aus dem Regal) der Fall ist.

Inzwischen gibt es jedoch eine Reihe von Open Source Projekten, die durchaus wie "Commercial Off-the-Shelf" Produkte eingesetzt werden können. Auch kann eine durchdachte Ausschreibung wesentlich besser die Dienstleistungskomponente in Wert setzen, als dies bei herkömmlicher proprietärer Software der Fall ist.

Der Vortrag stellt unterschiedliche Möglichkeiten vor, wie die Vorteile von Open Source auch in Ausschreibungen berücksichtigt werden können, ohne sich darauf festlegen zu müssen.

Neuigkeiten aus dem Open Geospatial Consortium – Rückblick 2011 und eine Zusammenfassung der letzten OGC Technical Committee Meetings

Athina Trakas

Im September 1994 wurde das Open Geospatial Consortium[1] (OGC – bis 2004 bekannt unter Open GIS Consortium), von 8 Mitgliedern in Leben gerufen. Mittlerweile ist das internationale Konsortium mit weltweit mehr als 440 Mitgliedern aus Industrie, Behörden, öffentlichen Verwaltungen und Universitäten aus der Entwicklung von Standards im Geo-Bereich nicht mehr wegzudenken.

Das OGC hat sich zum Ziel gesetzt, die Verarbeitung von raumbezogenen Informationen über Hersteller-, System- und Organisationsgrenzen hinaus zu vereinfachen und dadurch eine breitere Nutzung von Geoinformationen zu ermöglichen. Um dies zu erreichen und um Interoperabilität zwischen den verschiedensten Systemen zu ermöglichen, erarbeiten die Mitglieder des OGC allgemeingültige Standards und Spezifikationen.

Die Standards des Open Geospatial Consortium (OGC) bilden die Grundlagen für viele GDI- und Geoportal-Anwendungen und WebGIS Lösungen in den verschiedensten Bereichen. Zusätzlich nutzen immer mehr „Communities“ die OGC Standards, um fachspezifische Anforderungen zu lösen. Dies zeigt sich in den verschiedenen Standards- und Domain-Arbeitsgruppen, die in den letzten Jahren von OGC-Mitgliedern ins Leben gerufen wurden. Standards aus dem Sensor Web Bereich (für z.B. Frühwarnsysteme oder Internet of Things), der Open GeoSMS Standard (z.B. für Crowded Sourcing Anwendungen) oder die Arbeiten in den Bereichen Augmented Reality und 3D Visualisierung (Smart Cities, Städteplanung etc.) sind nur einige wenige Beispiele der aktuellen Tätigkeiten des OGC.

OGC Standards entstehen im Standards Program [2] durch einen offenen, konsensbasierten Prozess, an dem sich die Mitglieder beteiligen. In den Arbeitsgruppen (Domain- und Standards-Working Groups DWG/SWG) [3/4] werden entweder verschiedene thematische Aspekte beleuchtet oder aber an den Standards selbst gearbeitet. Im Interoperability Program [3] werden Standards dann in realen Szenarien getestet. Das Programm unterstützt sogenanntes rapid prototyping, das Testen und Validieren neuer Standards durch sogenannte Testbeds, Interoperabilitäts-Experimente und Pilot-Initiativen.

Aufbauend auf dem Vortrag der FOSSGIS 2011 [5] und entsprechenden Fragen aus dem Publikum, werden in diesem Vortrag Neuigkeiten aus dem OGC vorgestellt. Zusätzlich wird erläutert, welche Möglichkeiten es für Interessierte gibt, sich am OGC Prozess zu beteiligen.

Der Vortrag gibt einen Überblick über folgende Aspekte:

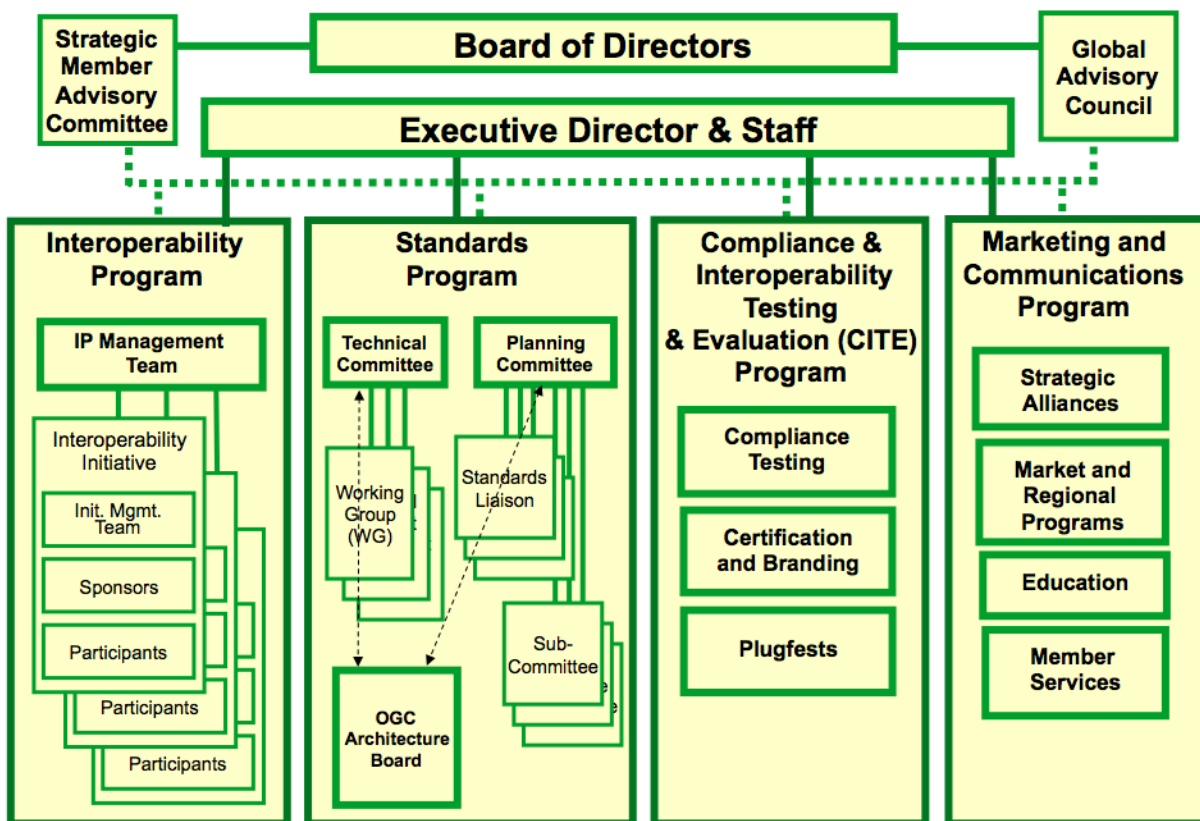
- Aktuelles der letzten OGC Technical Committee Meetings
- Neuigkeiten aus den einzelnen OGC Programmen und Arbeitsgruppen, wie z.B. welche neuen Arbeitsgruppen wurden in den vergangenen 12 Monaten in Leben gerufen, wie ist der Standard der Dinge der OGC Web Services Testbeds (OWS-8 und OWS-9).
- Welche „Requests for Comments“ und „Requests for Participations“ gibt es und wie können sich Interessierte einbringen?
- Ausblick auf OGC TC Meeting in Europa in 2012 / 2013

Neuigkeiten aus dem Open Geospatial Consortium – Rückblick 2011 und eine Zusammenfassung der letzten OGC Technical Committee Meetings

Kontakt zum Autor:

Athina Trakas
OGC Director European Services
Open Geospatial Consortium
Heerstraße 162
53111 Bonn
Telefon +49 228 54 8899 42
eMail atrakas@opengeospatial.org

Programme des OGC - <http://www.opengeospatial.org/ogc/programs>



Links

- [1] OGC Webseite: <http://www.opengeospatial.org/>
- [2] OGC Standards Programm: <http://www.opengeospatial.org/ogc/programs/spec>
- [3] OGC Somain Working Groups: <http://www.opengeospatial.org/projects/groups/wg>
- [4] OGC Standards Working Groups: <http://www.opengeospatial.org/projects/groups/swg>
- [5] <http://www.fossgis.de/konferenz/2011/programm/events/302.de.html>

Open Source GIS Software in der UNIGIS Fernlehre

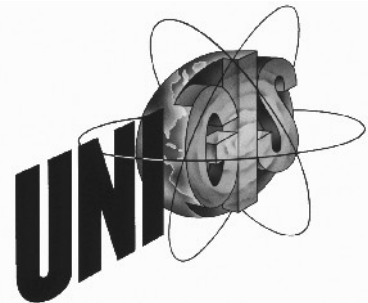
Cornelius Roth, Gudrun Wallentin

Kurzfassung

In diesem Vortrag geht es um den Einsatz von Open Source GIS Software im Rahmen des universitären Fernlehrgang UNIGIS mit einem Einblick über verwendete FOSSGIS Software und Lernmodule, den thematischen Schwerpunkten in denen Open Source und Open Content in der Fernlehre eingesetzt werden und nicht zuletzt der Bedarfsimpuls in Richtung FOSSGIS Community für eine effektive und nachhaltige Fernlehre aus der mehrjährigen Erfahrung seitens UNIGIS.

Einführung - UNIGIS Salzburg und FOSSGIS

UNIGIS ist ein weltweites Netzwerk an Universitäten für berufsbegleitende Fernstudien im Bereich Geoinformatik. Im deutschsprachigen Raum ist die Universität Salzburg, Zentrum für Geoinformatik, als UNIGIS Gründungsmitglied seit 1994 Träger dieser GI-Ausbildung. Es werden zielgruppenangepasst zwei Studiengänge angeboten, das 2-jährige postgraduale MSc Studium und die stärker Praxis-orientierte 1-jährige Variante mit Abschluss „Akademische/r GeoinformatikerIn“. Zusätzlich werden beide Studiengänge auch mit englischsprachigen Materialien an etwa 10 Studienzentren in Osteuropa, Süd-Ostasien und Afrika angeboten. Insgesamt haben 1400 Studenten erfolgreich einen UNIGIS Abschluss an der Universität Salzburg erworben. |



Der akademische Anspruch von UNIGIS ist es, Konzepte und Methoden der Geoinformatik, zu vermitteln. Es geht also um den richtigen Umgang mit GI-Werkzeugen, nicht um die Werkzeuge selbst. Dies bedeutet, dass Übungs- und Prüfungsaufgaben im Bereich der Pflichtmodule grundsätzlich mit jeder passenden GI-Software bearbeitet werden können, unabhängig davon, ob es sich dabei um eine proprietäre oder eine Open Source Software handelt, sofern der Lösungsweg entsprechend dokumentiert wird. Einen wesentlich stärkeren Akzent auf FOSSGIS setzt UNIGIS im Bereich des Wahlfaches, wo aktiv die Trends und Bedürfnisse der GI Community aufgegriffen werden. Es überrascht daher nicht, dass gerade in den letzten Jahren Open Source Software immer stärker im Angebot integriert wurde und sich zum Beispiel in Form von Übungsaufgaben, bis hin zu eigenen Lernmodulen für spezifische Softwareprodukte, wiederfindet.

Im Bereich der Studienzentren, die wie eingangs beschrieben in wirtschaftlich schwächeren Regionen lehren, steht die zentrale Bedeutung von Open Source Software seit jeher außer Frage. Doch auch im deutschsprachigen Raum sind die Zeiten in der die Frage im Raum stand: - „Kann man Open Source Produkte im beruflichen Kontext ernsthaft verwenden?“-oder -„Macht es Sinn sie in der Lehre einzubinden?“ schon lange vorbei. Heute besteht die Nachfrage nach Fortbildung um Open Source GIS Software und betreffenden Architekturen aus Wirtschaft und von Kommunen mehr denn je. Ein gutes Gradmaß für diese Entwicklung sind die MSc Master Thesis Arbeiten, die zu einem bedeutenden Teil explizit unter Verwendung von offenen GI-Software Produkten entwickelt werden. Das besondere an Open Source GIS Software ist, dass sie nicht nur lizenzgünstig zu verwenden ist, sondern auch meist OGC-Standards in hohem Maße unterstützt und implementiert hat. Sprich: „Will man nun offene Standards erlernen, geht das mittlerweile ganz besonders gut mit FOSSGIS Software.“

Open Source GIS Software in der UNIGIS Fernlehre

UNIGIS stellt im Rahmen seiner Lehrgänge schon seit einiger Zeit mit Hilfe von Experten aus den Communities optionale Lernmodule unter Verwendung von offenen Softwareprodukten zur Verfügung

Open Source GIS Software in der UNIGIS Fernlehre

und will dies weiter ausbauen, was die Ende 2011 ins Leben gerufene Kooperation mit der FOSS-Academy unterstreicht [FOSS-Academy, 2011]

In welchen Themenbereichen verwenden wir derzeit FOSSGIS in UNIGIS:

- OGC-bezogene Standardisierungskonzepte, wo IT-Methoden (Webserver, Applikationsserver) mit bestimmten Geoapplikations-Architekturen kombiniert erlernt werden sollen (WEBGIS und OGC-Grundlagen, OGC-basierte Applikationsentwicklung)
- Open CONTENT / Crowded Sourcing / Volunteered Geographic Information (Developing Applications with OSM)
- softwarespezifische Schwerpunktmodule als softwarenahe Umsetzung von Software Manuals (Lernmodule: GIS-Analysemethoden mit gvSIG und Application Development)

Im Folgenden werden 2 UNIGIS Modul kurz vorgestellt in denen FOSSGIS im Jahr 2012 verwendet wird:

GIS-Analysemethoden mit gvSIG ,in a nutshell'

Dieses in UNIGIS angebotene Lernmodul wurde von der Firma CSGIS entwickelt und bietet in insgesamt 9 Lektionen die Möglichkeit, die freie Software gvSIG und Sextante im Detail kennen zu lernen. gvSIG verfügt über zahlreiche Funktionen und kommt oftmals als Alternative zu proprietärer Software zum Einsatz. Im Rahmen dieses Moduls lernen die Teilnehmer Grundsätze von Open Source Software kennen und erhalten auch Einblick in die Anbieter hinter diesen Softwareanbietern. Der freien Java-Bibliothek Sextante wird eine eigene Lektion gewidmet. Sie lässt sich problemlos in gvSIG verwenden und erweitert mit zahlreichen Algorithmen zur Vektor- und Rasterdatenanalyse, einer GRASS- und SAGA-GIS-Schnittstelle den Funktionsumfang von gvSIG erheblich. [UNIGIS Salzburg, Modul gvSIG, 2011]

Developing applications with OSM ,in a nutshell'

Volunteered Geographic Information, GeoWeb 2.0 & Co – a lot of terms sprang up to describe the previous developments to use free geographic information systems (GIS) together with users who actively generate geographic content similar to the Wikipedia principle. User-generated knowledge cannot be compared with classic forms of data acquisition and implies an enormous potential for future applications in the field of free geographic information. The UNIGIS Module "Developing applications with OSM" corresponds to this recent trend and is supposed to provide a latest state of the art overview towards the collaborative, free editable word map project - OpenStreetMap. But what exactly is OpenStreetMap? Why are users motivated to voluntarily contribute and how can these geodatasets be mapped and edited? How good is the OSM data coverage in general and what can OSM data used for? We're going to answer these questions upon module's completion. [Steiger, E. Modul OSM, 2011]

Offene Software und offene Standards als Chance für die Lehre

Die Gemeinsamkeit „offen“ führt schon in der Terminologie den besonderen Schwerpunkt auf Geo-standards und entsprechendem Hintergrundwissen und Architekturen mit sich. Die Verquickung von FOSSGIS und OPENGIS führt nicht nur zu Verwechslungen, sondern auch zu der Chance zeitgemäße und neue GI-Konzepte zu lehren und damit FOSSGIS-Software nachhaltig weiter in den Fachdomänen als ‚primary‘ GIS-Werkzeug zu etablieren. Zweitens sollte die Community auf ein ebenfalls nachhaltiges und solides Angebot aufbauen, um mit proprietärer Software konkurrieren zu können.

Trainingsumgebungen und Betriebssystembedarf

Aus technischer Sicht im Sinne der Betriebssystemwahl sind FOSSGIS Produkte mittlerweile zwar oft OS unabhängig aber nicht zuletzt aus Kostengründen leichter und eher mit Open Source Betriebssystemen (Linux, etc.) gekoppelt oder kommen oft auch von FOSS-Entwicklerseite aus dieser Ecke.

Für die Lehre ergibt sich daraus häufig die Notwendigkeit sich bei der gewählten Lernumgebung zwischen proprietären und offenen Betriebssystemen zu entscheiden, bzw. so weit als möglich unabhän-

Open Source GIS Software in der UNIGIS Fernlehre

gig von Produkten zu sein. Im UNIGIS Modul: Developing applications with OSM wird beispielsweise mit einem Virtualbox¹³-Image mit Ubuntu Linux gearbeitet, wobei eine virtuelle Client- als auch eine virtuelle OSM-Servermaschine für das Lernen zur Verfügung gestellt werden. Damit können Konzepte ausreichend veranschaulicht, als auch die Umsetzung praktisch geübt werden, ohne zeitaufwändiges Aufsetzen von (zu Beginn für die Studierende fremde und neue) Server-Tools auf einem zu dem noch „fremden“ Betriebssystem. Ein weiterer Vorteil liegt in der durch die Virtualisierung erreichten Einheitlichkeit der Softwareumgebung für das Training und zweitens in der leichten Implementierung auf verschiedenen Host-Betriebssystemen. Eine wichtige Rahmenbedingung für Onlinelehre muss es auch sein so weit wie möglich flexibel zu sein, was die Trainingsbetriebssystemumgebung angeht. In diesem Zusammenhang ist das Verhältnis der verwendeten Betriebssysteme in UNIGIS interessant, um die derzeitige Unausgeglichenheit zu veranschaulichen.

Das Verhältnis der verwendeten Desktop-Betriebssysteme aus der Gruppe der UNIGIS Studenten der vergangenen 3 Jahre liegt im Schnitt bei MS Windows (84%), Linux-basierte Betriebssysteme (12%), sonstige Betriebssysteme (4%). [interne Befragung, 2011]. Durch die Anforderungen seitens der im Studium verwendeten proprietären GIS-Softwareprodukte liegt der Anteil bei der Modulbearbeitung mit MS-Windows sogar bei knapp 100%. Zu dem stellt sich immer mehr die Anforderung die Geo-Serverseite samt Architekturkonzepten und verwendeter Standardisierungsmethoden (WMS, WFS, WPS, etc.) praktisch zu erlernen. In diesem Kontext sind gerade OGC-Webservices (zB. Geoserver, etc.) auf Basis von Linux als Server-Umgebung das Mittel der Wahl darstellen.

Was braucht es nun aus Sicht der Geoinformatik-Ausbildung, um FOSSGIS Software in der Ausbildung nachhaltig und ressourcen-effektiv einzusetzen?

- Usability: schnell zu erlernender Umgang mit der Software
- vollständige Funktionsdokumentation
- detaillierte Algorithmen-Dokumentation
- methodisch fokussierte (fortgeschrittene) Tutorials
- solide Weiterentwicklungsschiene, Funktionssicherheit/Rückwärtskompatibilität
- Trainings-Softwareumgebungen Desktop und GIS-Server (Virtuelle Maschinen-Images mit vorausgesetzten GIS-Serverfunktionalitäten)¹⁴
- Softwaresupport für Lehrbeauftragte und gute Dokumentation von Trainings-Softwareumgebungen
- aktuelle Übungsdaten und -umgebungen (zB. WMS-Dienste, WPS, OSM.org, etc.)

Zusammenfassung

Insgesamt wird im Rahmen der UNIGIS Fernlehre FOSSGIS Software gerne eingesetzt, weil die Kombination mit Geo-Standards eine ausgezeichnete Chance darstellt wichtige Konzepte von OGC-Grundlagen, *Implementierung* von Geo-Standards bis hin zu ‚bleeding-edge‘ Forschungsthemen und -algorithmen als auch frisch verabschiedeten Geo-Standards angemessen flexibel zu lehren. Will man die Lehre als gerechtfertigtes Mittel zur Etablierung von FOSSGIS Software verwenden, gibt es seitens der UNIGIS Fernlehre bestimmte Bedarfe an die Community, um dieses Ziel nachhaltig zu erreichen.

Kontakt zu den Autoren:

Cornelius Roth, Gudrun Wallentin
Universität Salzburg, Zentrum für Geoinformatik
Schillerstraße 30
+43 662 8044 5210

13 Virtualbox Virtualisierung: www.virtualbox.org

14 zB. wie bereits mit der virtuellen Maschine der OSGEO Live DVD verwirklicht

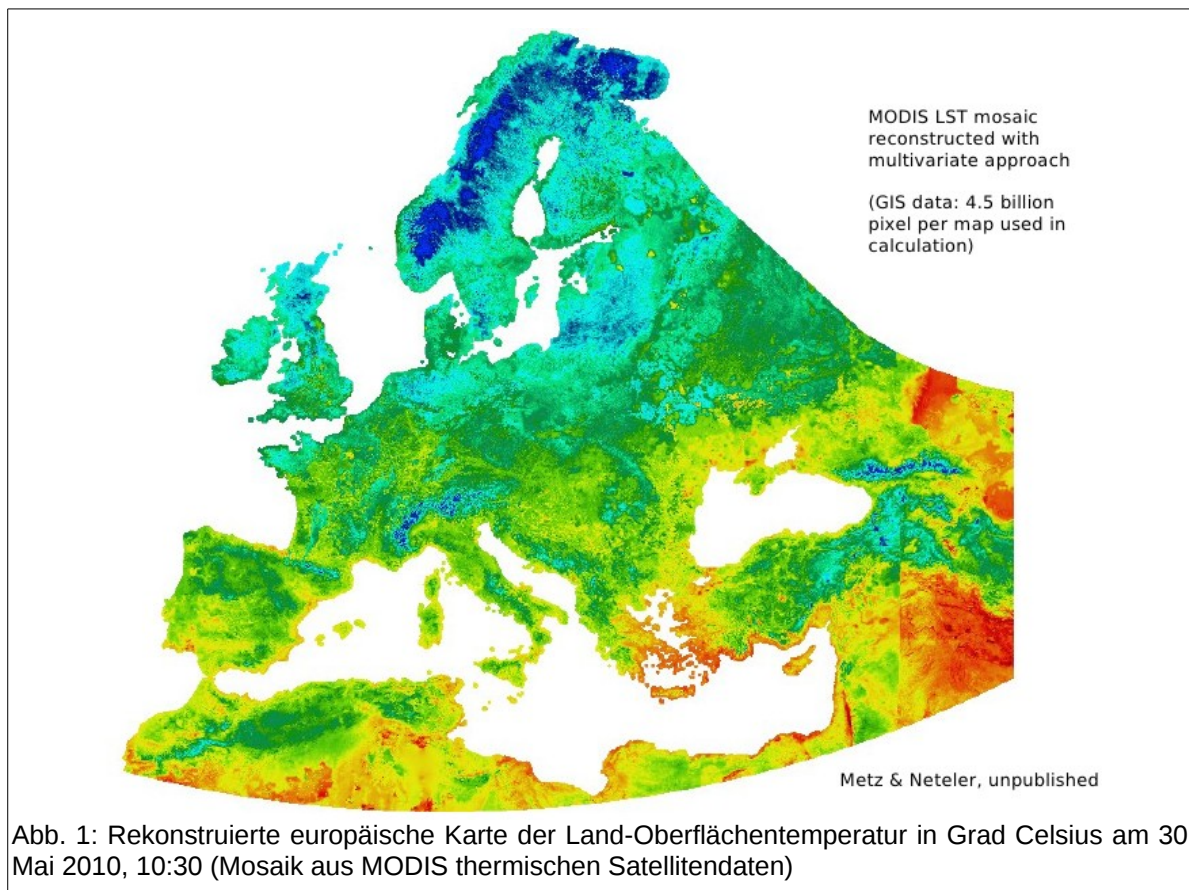
Literatur

- [1] FOSS Academy: FA und UNIGIS kooperieren bei Fortbildung: <http://www.foss-academy.eu/>, [last accessed 2011-12-18], 2011.
- [2] *Schönbuchner, R., Canalejo, José*: CSGIS Schulungen: GIS – Analysemethoden mit gvSIG: http://csgis.de/joomla/index.php?option=com_content&view=article&id=149&Itemid=152, [last accessed 2011-12-18].
- [3] *Schönbuchner, R., Canalejo, José*: GIS-Analysemethoden mit gvSIG. In: UNIGIS Offline, Ausgabe 4/11, http://www.unigis.ac.at/club/offline/archiv/UNIGISoffline_4_11.pdf, [last accessed 2011-12-18].
- [4] *Steiger, Enrico, Krampe, Stefan*: Developing applications with OSM. In: UNIGIS Offline, Ausgabe 3/11, Salzburg, http://www.unigis.ac.at/club/offline/archiv/UNIGISoffline_3_11.pdf, [last accessed 2011-12-18].
- [5] *Trakas, A.*: Osgo and OGC, FossGIS, <http://reinout.vanrees.org/weblog/2011/04/07/fossGIS-osgeo-ogc.html>, [last accessed 2011-12-18], 2011.
- [6] *UNIGIS Salzburg*: Optionales Modul - GIS-Analysemethoden mit gvSIG: http://www.unigis.ac.at/club/weiterbildung/optmod_gvsig.pdf, [last accessed 2011-12-18].
- [7] *UNIGIS Salzburg*: Optionales Modul - Developing applications with OSM: http://www.unigis.ac.at/club/weiterbildung/optmod_osm.pdf, [last accessed 2011-12-18].

Freie Software und infektiöse Krankheiten: Von Rohdaten zu ökologischen Indikatoren

Markus Neteler

Zoonosen sind vom Tier (Vektoren) auf den Menschen übertragene Infektionskrankheiten, die in vielen Ländern ein großes Gesundheitsproblem darstellen. GIS-Modellierung wird routinemäßig eingesetzt, um die räumliche Verteilung der insbesondere von Nagetieren, Zecken und Stechmücken übertragenen Zoonosen besser zu verstehen. Die Eingangsdaten sind üblicherweise heterogene GIS-Datenquellen sowie neue, zeitlich hochauflösende Fernerkundungsdaten. Besonders letztere helfen, die räumlichen und zeitlichen Muster von vektorübertragenen Krankheiten zu verstehen. Obwohl Satellitendaten seit mehr als zwei Jahrzehnten erfolgreich in epidemiologischen Anwendungen eingesetzt wurden, gibt es erst seit kurzem zeitlich hochauflösende Zeitreihen (insb. MODIS Sensor: vier Abdeckungen der Erde pro Tag in verschiedenen Auflösungen (250m-1000m)). Mittels GIS werden aus diesen neuen Fernerkundungszeitreihen, u.a. Land-Oberflächentemperatur (LST, siehe Abb. 1; [1, 2]), Vegetationsindizes (NDVI/EVI) und Schneebedeckung, ökologische Indikatoren extrahiert.



Die Datenaufbereitung (Geocodierung, Projektion, GIS-Integration und Qualitätsbewertung) ist oft mit erheblichem Aufwand verbunden. Hier bieten Freie GIS Software-Tools (PROJ4, GDAL/OGR, GRASS GIS etc., [3]) alle notwendigen Methoden und Algorithmen zur Datenaufbereitung und -analyse an.

Aus den täglichen LST Karten sind aggregierte Temperaturwerte ableitbar, die u.a. zur Identifizierung von Spätfrost, heißen Sommern, und Gradzahltagen dienen. Der EVI erlaubt, saisonale und regionale

Freie Software und infektiöse Krankheiten: Von Rohdaten zu ökologischen Indikatoren

Unterschiede des Vegetationszustandes oder Frühlings- bzw. Herbst-Anfang zu erkennen. Der Vortrag geht auf einige abgeleitete Indizes ein und zeigt deren Anwendung im Hinblick auf potenzielle Verbreitungskarten zu Zecken und Tigermücken.

Die schiere Menge an Daten erfordert ausreichende Rechenkapazitäten. Typischerweise werden, wie auch in der Fondazione Edmund Mach, Computer-Cluster verwendet, um parallelisiert große Datenmengen zu verarbeiten. Freie GIS Software-Tools wie insbesondere GRASS GIS erlauben, parallel Batch-Jobs auszuführen, um große Mengen von Satellitendaten in High Performance Computing (HPC) Einrichtungen zu bearbeiten.

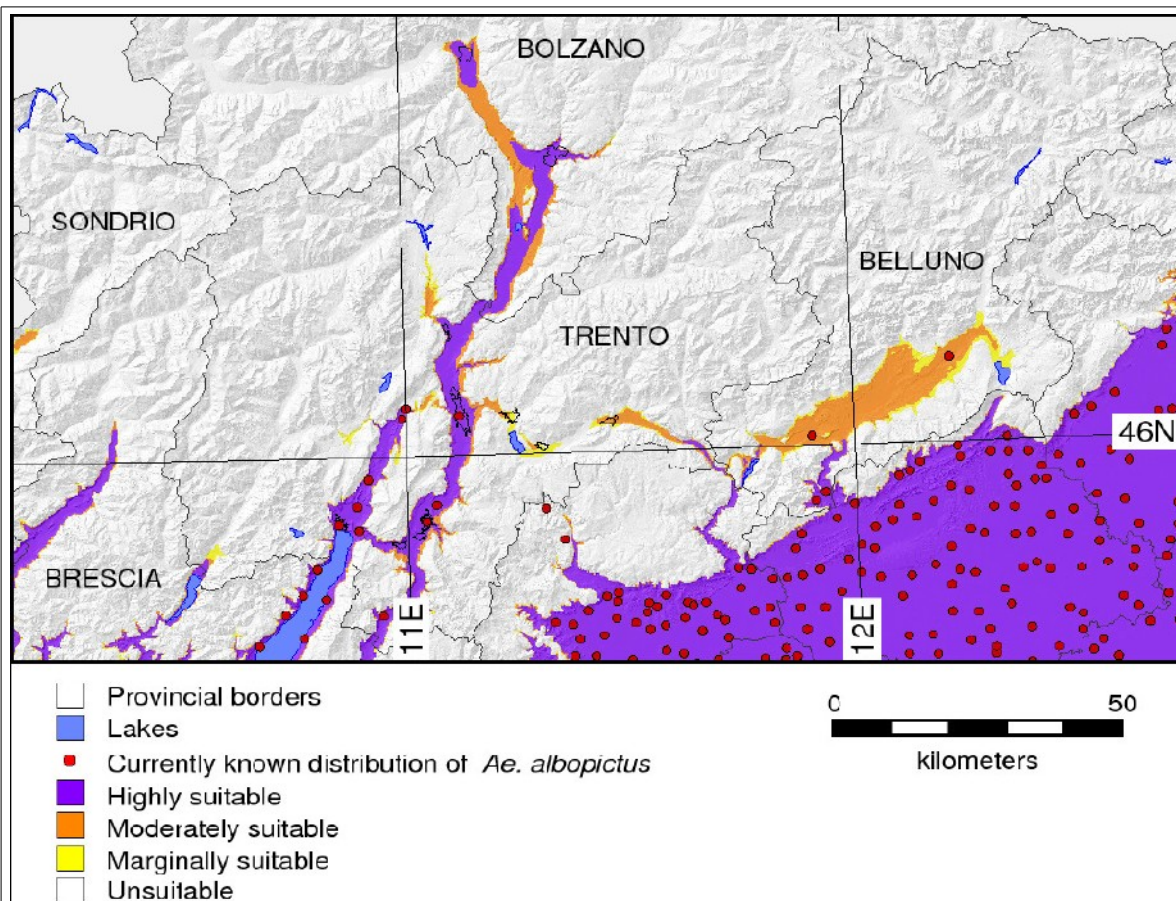


Abb. 2: Habitategnungskarte von *Ae. albopictus* (Asiatische Tigermücke) im Nordosten Italiens. Die Karte basiert auf einer Klassifizierung des Winter der Tigermückeneier, der Überlebens der adulten Tiere und der Chance, den Lebenszyklus erfolgreich abzuschließen (zitiert aus [1])

Kontakt zum Autor:

Dr. Markus Neteler
Fondazione Edmund Mach
Via Edmund Mach 1
38010 S. Michele all'Adige (TN), Italien
+39-0461-615-570
markus.neteler@fmach.it

Literatur

- [1] Neteler, M., Roiz, D., Rocchini, D., Castellani, C. and Rizzoli, A.: Terra and Aqua satellites track tiger mosquito invasion: modeling the potential distribution of *Aedes albopictus* in north-eastern Italy. *International Journal of Health Geographics*, 10:49, 2011 [[DOI](#) | [PDF](#)].
- [2] Roiz D., Neteler M., Castellani C., Arnoldi D., Rizzoli A.: Climatic factors driving invasion of the tiger mosquito (*Aedes albopictus*) into new areas of Trentino, Northern Italy. *PLoS ONE*. 6(4): e14800, 2011 [[DOI](#) | [PDF](#)].
- [3] Neteler, M., Bowman, M.H., Landa, M. and Metz, M.: GRASS GIS: a multi-purpose Open Source GIS. *Environmental Modelling & Software*, 31: 124-130, 2012 [[DOI](#) | [PDF](#)].

Verarbeitung sehr großer räumlicher Datensätze mit GRASS GIS

Markus Metz

Sowohl die Ausdehnung als auch der Detailreichtum räumlicher Daten wächst beständig mit verfeinerten Technologien und dem Bedarf für hochaufgelöste Daten. Beispiele für sehr große Datensätze sind globale administrative Gebiete (GADM), OpenStreetMap, Landbedeckungs- und Landnutzungskarten sowohl als Raster als auch als Vektor, digitale Höhenmodelle und Fernerkundungsdaten. Solche großen Datensätze stellen besondere Anforderungen für die Verarbeitung mit GIS Software, um z.B. Speicherüberläufe und extrem lange Bearbeitungszeiten zu vermeiden. Zeit- und Arbeitsspeicheranforderungen können die Analyse großer Datensätze ineffektiv bzw. Unmöglich machen, teils weil viele GIS Pakete zur Bearbeitung alle Daten in den Arbeitsspeicher laden, was leicht die vorhandenen Ressourcen selbst moderner Workstations überschreiten kann. Bedauerlicherweise geht eine Reduzierung der Arbeitsspeicheranforderungen oft mit verlängerten Bearbeitungszeiten einher. Besonders anspruchsvolle Analysen sind z.B. Kostenoberflächen und Suche nach kürzesten Pfaden, hydrologische Modellierung, Georektifizierung, Umprojektionen, Erstellung und Bereinigung von Vektortopologie. Um in der Lage zu sein, große Datensätze zu verarbeiten, müssen Algorithmen derart angepasst werden, dass Limitierungen durch Hardware-Ressourcen berücksichtigt werden, sie aber gleichzeitig effizient und schnell bleiben. Übliche Ansätze sind Kacheln, die Verwendung externen Speichers, und schnelle Such- und Sortierungsalgorithmen. Diese Mechanismen werden bereits seit langem in GRASS GIS benutzt. Die nächste GRASS-Version wird diese Mechanismen verstärkt benutzen, die zum Teil überarbeitet wurden und werden. Die Fähigkeiten der zukünftigen GRASS-Version zur Verarbeitung großer Datensätze werden dargestellt und mit der aktuellen stabilen Version verglichen.

SEXTANTE in gvSIG CE

geodata analysis for everyone

Referentin: Ruth Schönbuchner <ruth.schoenbuchner@csgis.de>

Sextante ist eine freie Bibliothek, die neben gvSIG CE auch unter jedem anderen Open Source Java-basierten GIS-System (uDig, OpenJump) oder proprietärer Software wie ArcGIS verwendet werden kann. Durch die Anbindung von Grass GIS und SAGA bietet Sextante eine mächtige Toolbox mit mehr als 700 Funktionalitäten für die Vektor- und Rasterdatenverarbeitung. Im Jahr 2008 wurde die Ausführung von Geoprozessen durch die Integration eines Modelbuilders sowie der batch-Processing-Methode vereinfacht. Für bestimmte Fragestellungen ist es oft notwendig, verschiedene Prozesse spezifisch zu kombinieren. Dafür wurde ein Modeller integriert, mit dem sich verschiedene Funktionalitäten durch einfache Prozessketten abbilden lassen. Die Software ist sehr leicht bedienbar und mit einer Kontexthilfe, die eine sehr gute Hilfestellung bei der Arbeit mit Sextante bietet, ausgestattet. Ein eigener Videokanal bietet zudem zahlreiche Videos, die den korrekten Umgang mit der Software veranschaulichen. Da in gvSIG CE keine Installation oder Konfiguration von Sextante durchgeführt werden muss, fällt der Umgang mit dieser Software sehr leicht. Sextante wird in englischer Sprache entwickelt, um das Projekt für jedermann zugänglich zu machen. Eine kurze Live-Einführung der aktuellen Version sowie die Vorstellung der konkreten Web-Organisation des Projektes runden den Vortrag ab. Ansprechpartner CSGIS José Canalejo & Ruth Schönbuchner GbR Innere Wiener Straße 32 D-81667 München Telf.: 0049/(0)89 37415227 Mobil: 0049/(0)178 6931412 Fax: 0049/(0)8323 986407 Email: ruth.schoenbuchner@csgis.de Web: <http://www.csgis.de>

Mehr Information über Sextante erhalten Sie durch den nachfolgenden Artikel:

- Sextante - eine freie Java-Bibliothek zur Geodatenanalyse: GIS Business - Geoinformationstechnologie für die Praxis 4/2011- ISSN: 1869-9286 (http://csgis.de/joomla/images/stories/pdf/Sextante_GB.pdf)

MapProxy - das "Schweizer-Taschenmesser" für Kartenserver

Dominik Helle

Zwei Jahre nach der ersten Veröffentlichung des MapProxy auf der FOSSGIS Konferenz 2010 in Os-nabrück, hat sich das MapProxy Projekt etabliert und ist um viele Funktionen erweitert worden.

Die Grundfunktion von MapProxy ist es bestehende Kartendienste durch Zwischenspeichern – so ge-nanntes Caching – zu beschleunigen. Beim Caching speichert MapProxy einmal beantwortete Anfra-gen in einem Kachelformat in einem Zwischenspeicher ab. Erfolgt nun eine gleichartige Anfrage wird diese aus dem vorhandenen Zwischenspeicher beantwortet. So können bestehende Dienste einer ak-tiven Geschwindigkeitsverbesserung unterzogen werden.

Eine Besonderheit des MapProxy ist es, dass trotz Caching der WMS-Standard komplett unterstützt wird. Die über MapProxy beschleunigten Dienste können wie gewohnt, z.B. in einem Desktop-GIS, ge-nutzt werden. Neben dem On-the-Fly Caching, bei dem der Speicher gefüllt wird wenn der Benutzer den Dienst verwendet, verfügt MapProxy über die Möglichkeit den Zwischenspeicher automatisiert zu füllen. Dieses Füllen wird in der Fachsprache Seeding genannt.

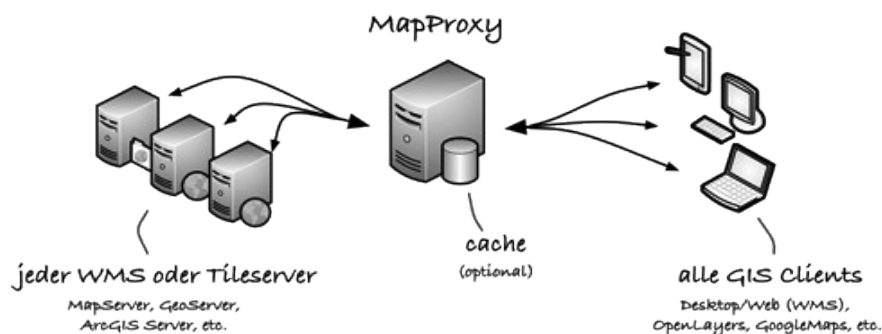


Abb. 1: Integration von MapProxy

MapProxy unterstützt unterschiedliche offene Standards und kann diese als Quelle einbinden oder ausliefern:

- OGC WMS 1.0.0 - 1.3.0
- OGC WMTS
- OSGeo TMS

Neben der Verwendung von Standards, können einige Quellen auch direkt in MapProxy eingebunden werden. Hierzu zählen unter anderem MapServer und Mapnik.

MapProxy verfügt neben der Beschleunigung über eine Vielzahl von weiteren Funktionen, die in den letzten Versionen hinzugefügt wurden. Einige ausgewählte Funktionen werden im Folgenden Abschnitt näher erläutert.

Coverages

Unter Coverages versteht man im MapProxy eine Geometrie, die einen Geltungsbereich definiert. Die Verwendung der Coverages zieht sich durch die gesamte Struktur des MapProxy. So kann über die Verwendung von Coverages angegeben werden wo eine Quelle gültig ist, welche Daten vorgeneriert werden oder welcher Bereich für bestimmte Benutzer sichtbar sind.

Für eine einfache Handhabung gibt es drei Möglichkeiten um Coverages im MapProxy zu definieren:

MapProxy - das "Schweizer-Taschenmesser" für Kartenserver

- als einfache rechteckige Bounding-Box
- als Textdatei mit einem oder mehreren Polygonen im WKT-Format
- (Multi-)Polygone von jeder Quelle die über OGR gelesen werden kann
z.B. Shapefile, PostGIS

Umprojizieren & Kaskadieren

Eine weitere Funktion des MapProxy ist es Kartendienste On-the-fly umzuprojizieren. Eine Besonderheit hierbei ist, dass der Quell-Dienst die neue Projektion nicht unterstützen muss. MapProxy projiziert die Daten selbst.

So ist es zum Beispiel möglich einen Dienst mit OpenStreetMap Daten – welcher nur die Google-Projektion beherrscht – mit einem Dienst der in UTM32 vorliegt zu überlagern und wie gewohnt in einem Desktop-GIS zu verwenden.

Neben dem Umprojizieren kann MapProxy auch dazu verwendet werden, unterschiedliche Dienste zusammenzufügen und somit ein so genanntes kaskadieren durchzuführen. Beim kaskadieren werden zwei oder mehrere Dienste zusammengefasst und sind für den Benutzer nur noch als ein Dienst zusammen sichtbar.

Das Kaskadieren vom MapProxy wird so durchgeführt das weiterhin der komplette WMS-Standard zur Verfügung steht. Auch Anfragen über GetLegendGraphic oder GetFeatureInfo sind bei kaskadierten WMS weiterhin möglich.

Sicherheit

Durch die zentrale Position kann MapProxy als Sicherheitsschicht für vorhandene Kartendienste genutzt werden. Zum einen kann MapProxy dazu verwendet werden, dass die Ursprungs-Server nicht mehr von aussen zu erreichen sind.

Des weiteren können mit MapProxy auch gezielt einzelne Layer für bestimmte Nutzergruppen sichtbar gemacht oder vor ungewollten Zugriffen geschützt werden. Bei WMS-Diensten werden bei GetCapabilities-Anfragen nur die Layer angezeigt die für den Benutzer sichtbar und auswählbar sind.

Durch die Verwendung der oben beschriebenen Coverages ist MapProxy in der Lage, die Karte so zu beschneiden, dass nur die Gebiete ausgeliefert werden, die für den Benutzer freigegeben sind. So sieht der Benutzer nur exakt den Kartenausschnitt der für ihn bestimmt ist.



Abb. 2: Absicherung mit MapProxy – auf Deutschland limitierter OpenStreetMap Layer

MapProxy - das "Schweizer-Taschenmesser" für Kartenserver

Die Sicherheitsschicht wurde als offene Schnittstelle entwickelt. So besteht die Möglichkeit Sie an unterschiedliche - auch schon bestehende - Benutzerdatenbanken anzugliedern.

Allgemein

MapProxy wird von einer aktiven Community unter Leitung von Omniscale weiterentwickelt. Das Projekt setzt dabei auf einen regelmässigen Release-Zyklus. So wurden innerhalb der letzten zwei Jahre über 10 Releases mit neuen Features und Bugfixes veröffentlicht.

Daneben setzt das MapProxy Projekt sehr auf Qualität, was sich in über 1100 automatisierten Tests und in einer ausführlichen Dokumentation für das Projekt widerspiegelt. Seit der Version 1.0.0 wurden die Entwicklung so koordiniert, dass MapProxy abwärtskompatibel zu älteren Konfigurationen ist.

Weitere Informationen zum MapProxy, die Dokumentation und ein Tutorial zum Einstieg finden sie unter <http://mapproxy.org>.

Kontakt zum Autor:

Dipl.-Ing. (FH) Dominik Helle
Omniscale GmbH & Co. KG
Nadorster Straße 60
26121 Oldenburg
0441 / 9392 7740
helle@omniscale.de

Literatur

[1] <http://mapproxy.org>

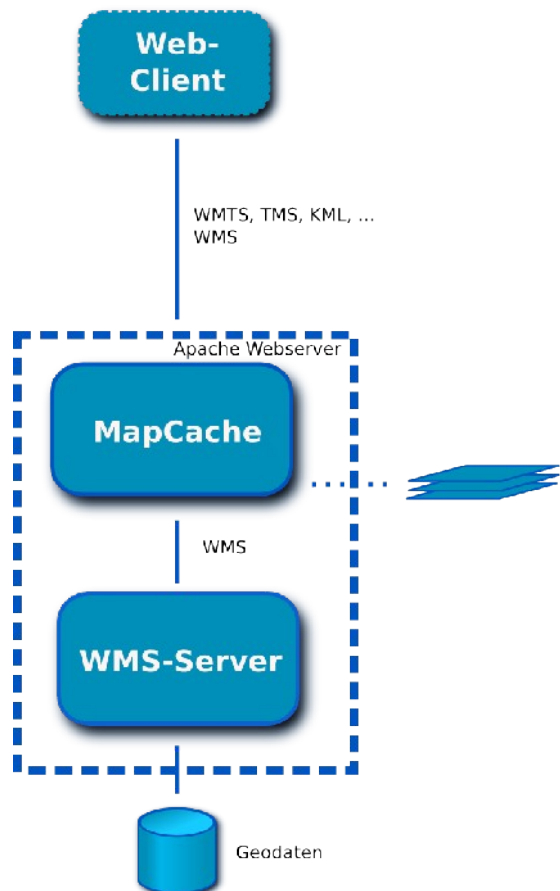
[2] <http://omniscale.de>

MapServer MapCache - der neue WMTS Tile Cache

Pirmin Kalberer

Der MapServer MapCache ist ein leistungsfähiger Tile Cache für den Apache Web Server mit umfangreicher Funktionalität und höchster Performance.

- Unterstützte Dienste: WMS, WMTS, TMS, VirtualEarth/Bing and Google Maps kompatible Tiles
- Schneller WMS Server mittels Kombination von Kacheln aus dem Tile Cache
- Unterstützt WMS und WMTS GetFeatureInfo Anfragen (Weiterleitung an WMS Backend)
- KML Superoverlay Generierung
- Beliebiger WMS Server als Backend. Z.B. UMN MapServer oder QGIS Server
- Cache Speicherarten: Dateisystem, SQLite-DB, Memcached, Tiled TIFFs
- Konfigurierbares Meta-Tiling
- On-the-fly Tile Kombination mehrerer Tile-Layer in ein Einzelbild
- Post-Processing von Images: Kompression, Wasserzeichen, etc.
- Seeding-Applikation mit paralleler Tile-Cache (Konfigurierbare Zoomstufen und sogar geografische Bereiche)
- Optimierte Behandlung von leeren Tiles (Symlinks)



Beispiel-URLs eines WMTS-Servers:

- <http://tiles.geodienste.ch/gws/wmts/1.0.0/WMTSCapabilities.xml>
- <http://tiles.geodienste.ch/gws/wmts/1.0.0/gewaesserschutz/default/CH1903/19/28/47.png>

Kontakt zum Autor:

Pirmin Kalberer
Sourcepole AG
Churerstrasse 22
CH-8808 Pfäffikon SZ
+41 44 440 77 11
pka@sourcepole.ch

Links:

- <http://mapserver.org/trunk/mapcache/>

WMS2GO – Der WMS zum Mitnehmen

Johannes Weskamm

Sie brauchen einen WMS, sind aber gerade nicht im Büro und haben kein Internet? Mit WMS2GO existiert eine Open Source Software, die WMS offline in einem vollwertigen WebGIS verfügbar macht. WMS2GO ist ein Gemeinschaftsprojekt der Firmen terrestris GmbH & Co. KG und WhereGroup GmbH & Co. KG sowie dem Dienstleistungszentrum IT (DLZIT) der Bundesanstalt für Wasserbau.

Der Anwender von WMS2GO hat zunächst über einen WebGIS-Client die Möglichkeit, die gewünschten WMS-Dienste auszuwählen. Anschließend kann grafisch über die Karte ein Begrenzungsbereich (BoundingBox) festgelegt werden, der das Interessengebiet abdeckt. Daraufhin wird im Hintergrund eine Prozessierung gestartet, die die Kacheln und Informationen der Dienste abrufen und diese zusammen mit einem offline nutzbaren WebGIS-Client konfigurieren. Das Endprodukt stellt einen, vom System dynamisch konfigurierten und offline nutzbaren WebGIS-Client auf Basis der Bibliotheken Ext JS, GeoExt und OpenLayers dar. Dieser Client kann vom Benutzer z.B. auf ein Notebook kopiert werden und anschließend vollständig offline, also ohne Internetanbindung, genutzt werden. Der Anwender hat somit die Möglichkeit, WMS-Dienste auch in Bereichen zu nutzen, in denen keine oder nur eine sehr langsame Internetverbindung zur Verfügung steht. Als typisches Anwendungsbeispiel wäre etwa die Nutzung „im Freien“ zur Digitalisierung oder zur Orientierung zu nennen. Der Vortrag wird den Gesamtaufbau und die Funktionsweise des WMS2GO-Systems erläutern sowie natürlich die Quellen zum Code und der Dokumentation des Projekts nennen.

Couchbase Mobile, GeoCouch und MapQuery

Mobile Anwendungen auch ohne Konnektivität

Volker Mische

Die moderne mobile Welt ermöglicht uns fast grenzenlosen Zugriff auf Informationen und Daten. Problematisch wird es jedoch, wenn die Verbindung zum Internet unterbrochen ist, beispielsweise in einem Tunnel oder in Gegenden, in denen das Netz noch nicht ausgebaut wurde.

Funklöcher sind immer ärgerlich, besonders jedoch, wenn man eine Verbindung zu einem zentralen Server benötigt, etwa zum Übertragen der gerade gesammelten Informationen im Feld. Die Lösung des Problems besteht darin, die Daten zunächst lokal auf dem mobilen Gerät zu speichern und erst dann zu übertragen wenn eine Internetverbindung besteht.

Damit sich Entwickler um die Applikation selbst und nicht um die Synchronisation kümmern müssen, kommt Couchbase Mobile zum Einsatz. Diese Datenbank liefert bereits alles, um eine solche Offline-Anwendung zu schreiben.

Dabei ist Couchbase Mobile mehr als eine Datenbank. Sie kann nicht nur Daten im JSON Format speichern, sondern auch HTML5-Anwendungen direkt ausführen. Es genügen also bereits Grundkenntnisse im Webbereich, um eine Anwendung zu entwickeln. Durch GeoCouch ist es möglich, räumliche Anfragen lokal auf dem Gerät zu bearbeiten.

MapQuery, ein Web-Mapping-Framework das auf OpenLayers und jQuery basiert, vereinfacht die Entwicklung zusätzlich.

Dieser Vortrag gibt zunächst einen kurzen Überblick über die verwendeten Technologien Couchbase Mobile, GeoCouch und MapQuery. Danach wird demonstriert, wie leicht eine Anwendung gebaut werden kann, die die oben beschriebene Datensynchronisation ermöglicht.

Alle erwähnten Softwareprodukte sind Open Source. Couchbase Mobile und GeoCouch stehen unter der Apache License Version 2.0, MapQuery steht unter der MIT Lizenz.

Mobiler Haltestellenclient zur Erfassung barrierespezifischer Mobilitätskriterien und Angebotsqualität des ÖV auf Basis von OSM-Daten

Enrico Steiger

Das Mobilitätsverhalten und die Mobilitätsmöglichkeiten von Personen werden wesentlich von demographischen und gesellschaftlichen Entwicklungen wie auch von raumstrukturellen Veränderungen geprägt. Essentielle Zielstellung bei der ÖPNV Planung sind daher Fragen nach Barrierefreiheit, Altersgerechtigkeit der ÖV-Infrastruktur zielgruppenspezifischen Mobilitätsbedürfnissen. Im Rahmen des vom österreichischen Bundesministeriums für Verkehr, Innovation und Technologie (Bmvit) geförderten Projekts „PublicTransportScreener“ wurden neue Methoden für raumbezogene Planungsinstrumente zur zielgruppenorientierten Bewertung und Planung der Zugänglichkeit von Haltestellen und der Angebotsqualität im Öffentlichen Verkehr entwickelt. Die technologiebasierte Grundlage bildete dabei die mobile Erfassung der Ausstattung, Zugänglichkeit und Barrierefreiheit von Haltestellen. Nach einem intern erarbeiteten Grundkonzept zur Erfassung der Haltestelleninfrastruktur erfolgt durch entsprechende Attributierung im OSM Schema (z.B. Umgebungsplan (Stadt/Gemeinde) pts:area_map) eine Annotierung zusätzlicher Haltestelleninformationen in der Testregion Salzburg. Damit lassen sich neben den klassischen Mobilitätskriterien im ÖV (Beförderungsangebot, Bedienungshäufigkeit) Erreichbarkeiten auch für Verkehrsteilnehmer mit speziellen Mobilitätsbedürfnissen visualisieren. Für das OpenStreetMap Projekt ist eine mobile Editierung zusätzlicher Attribute nach dem im letzten Jahr vorgeschlagenen Oxomoa/ÖPNV-Schema möglich. Hier lassen sich auch streckenabhängige Steigungen (Längs- und Quergefälle) über die im Smartphone integrierte Neigungssensorik automatisiert erfassen. Die bisherigen Projektaktivitäten des Aufbaus eines Routingportals für Rollstuhlfahrer (werden durch den Ausbau und die Verbesserung der barrierespezifischen Informationen entscheidend unterstützt. Vor allem punktuelle Haltestelleninfrastrukturen werden hinsichtlich ihrer Datenqualität erweitert und verbessert.

GeoExt Mobile (GXM)

Marc Jansen, Till Adams

Zusammenfassung

GeoExt Mobile [1] [2] ist eine noch junge Open Source Bibliothek, welche die Funktionalitäten von OpenLayers [3] und Sencha Touch [4] verknüpft, um Entwickler in die Lage zu versetzen, mobile Web-GIS-Anwendungen (WebApps) zu erstellen. Hierbei können die vielfältigen Möglichkeiten der Basisbibliotheken voll ausgenutzt werden:

- Verwendung aller Kartentypen und Interaktionswerkzeuge, die OpenLayers unterstützt
- Oberflächenelemente von Sencha Touch, die natives Look and Feel im mobilen Browser gewährleisten

Mobiles WebGIS

Smartphones wohin das Auge blickt. Dabei ist es egal, ob man im Zug sitzt oder auf den Bus wartet. Der Zugriff auf das Internet erfolgt zunehmend von mobilen Geräten.

Diesem Trend folgend werden Webseiten zunehmend auch mobil aufgerufen und dadurch ergeben sich auch für webbasierte Kartenanwendungen neue Anwendungsfelder. Insbesondere durch die Möglichkeit sich selber verorten zu können, gewinnt WebGIS eine neue Qualität denn damit ist die Voraussetzung geschaffen, im Raum auf Karten und Geodaten zuzugreifen und gleichzeitig diesen Raum auf den eigenen, aktuellen Standort einzugrenzen. Die Frage nach „Was befindet sich im Umkreis von...?“ wird so zu „Was befindet sich im Umkreis von meinem aktuellen Standort?“.

Um diese neuen Anwendungsfelder bedienen zu können, wurde die Freie WebMapping Bibliothek GeoExt Mobile (GXM) speziell für Browser auf Smartphones entwickelt.

GXM

GeoExt Mobile oder kurz GXM ist eine JavaScript Bibliothek, die als Basis für einen WebGIS Client dient, der optimiert ist für die meistgenutzte App auf jedem Smartphone: Den Browser.

Die erste Version von GXM wurde als Auftragsarbeit von terrestris für die Schweizer Landestopographie swisstopo entwickelt. Ziel der swisstopo war es zu diesem Zeitpunkt, die bestehende Geodateninfrastruktur um eine mobile Komponente zu erweitern.

Im Rahmen der Arbeiten für swisstopo sollten folgende Komponenten entwickelt werden:

- eine Open Source Basisbibliothek analog zu GeoExt [5] (vgl. unten), das ist GXM
- eine swisstopo-interne API um zum Beispiel auf eigene Daten zugreifen zu können
- ein mobiler Client aufbauend auf GXM und der vorgenannten API, der neben der Karte auch andere Elemente, wie z.B. ein Feedback-Formular oder eine Sprachauswahl enthält

GXM in der aktuellen Version 0.1 (zur Konferenz vermutlich schon 0.2, das auf Sencha Touch 2.0 aufsetzt) unterstützt aktuell die WebKit-basierten Browser der Betriebssysteme von Iphone und Ipad, Android-Phones sowie von Blackberry ab der Version 6, die Beschränkung ergibt sich aus den durch Sencha Touch unterstützten Browsern. Laut mehreren Funden im Web können so über 90% aller Smartphones Sencha Touch nutzen [7].

Das interessante am Projekt für swisstopo war, das vor Schaffung einer projektspezifischen Lösung zunächst die dafür fehlende Basisbibliothek geschaffen werden sollte.

GeoExt Mobile (GXM)

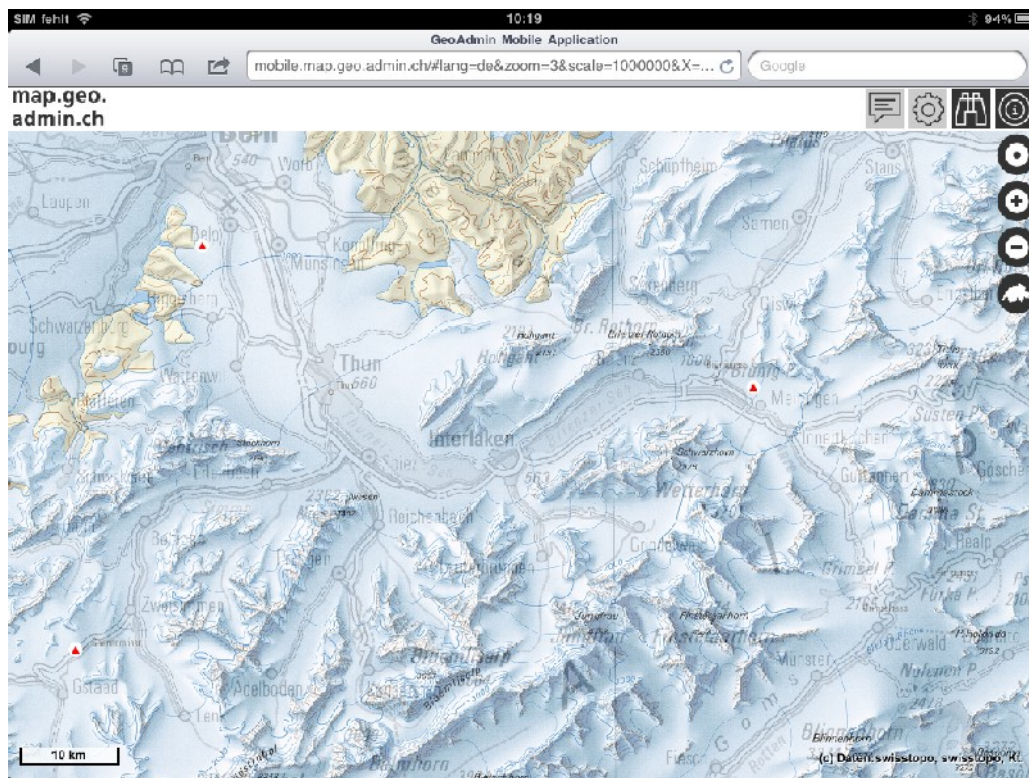


Abbildung 1: Im mobilen WebGIS der swisstopo wird GXM verwendet.

Die Abbildung zeigt eine Ansicht des Projektergebnisses aus [6]. Zwar besitzt OpenLayers alleine seit dem Codesprint in Lausanne im Frühjahr 2011 für mobile Endgeräte taugliche Bedienelemente. Hierzu zählen Unterstützung von Karteninteraktionen via Touchscreen (Drag, Pinch, Tap, DoubleTap, usw.), Implementierung einer Geolocation Control (HTML5-API um den Standort des Gerätes auszulesen) und Performance Optimierung.

Die alleinige Verwendung von OpenLayers ist aber wenig komfortabel und die Resultate wenig intuitiv bedienbar.

Es wurde also beschlossen, eine Schwester-Anwendung zu GeoExt aufzusetzen, die sich aus den Bibliotheken OpenLayers und Sencha Touch zusammensetzen sollte. Die Wahl fiel auf Sencha Touch, da diese Bibliothek aus demselben Hause wie ExtJS kommt und künftige Versionen sowohl von ExtJS als auch von Sencha Touch auf derselben Kern, der „Ext Base“, basieren werden.

Sencha Touch

Auf der Basis von ExtJS und HTML5 hat die Firma Sencha mit Sencha Touch eine JavaScript Bibliothek für die Entwicklung von mobilen, auf Touchscreens optimierte Anwendungen unter entwickelt. Die Bibliothek bietet viele Bedienelemente, AJAX-Komponenten sowie Möglichkeiten zur Dom-Manipulation, ebenso wie die für Web-Anwendungen wichtigen CRUD Operationen (Create, Read, Update, Delete), die mit relationalen Datenbankservern implementiert werden können.

Die Open Source Version von Sencha Touch steht der GNU GPL Lizenz Version 3 und bietet somit die Voraussetzung zur Schaffung einer Freien darauf aufbauenden WebMapping Bibliothek.

GeoExt Mobile (GXM)

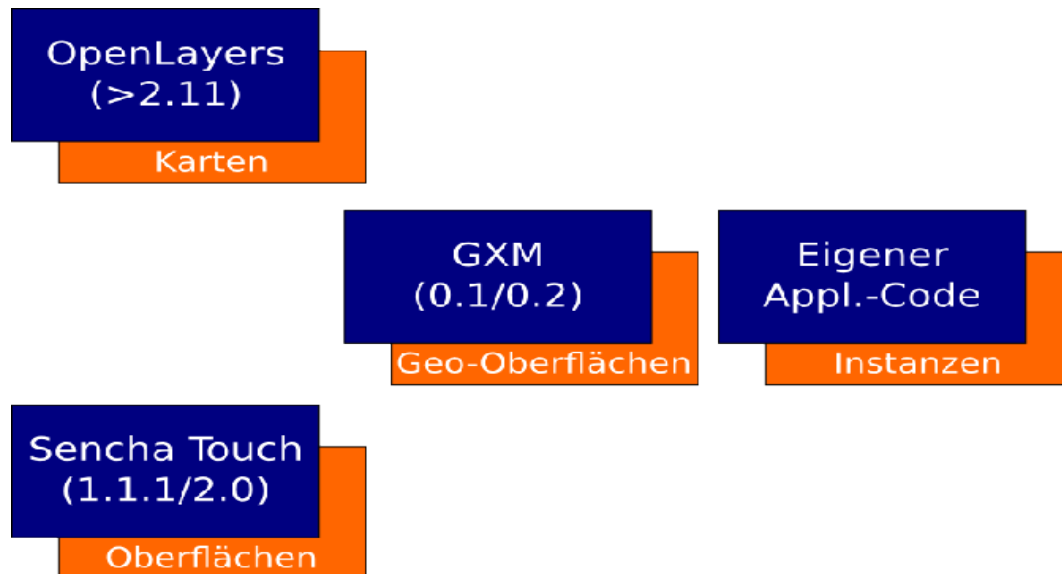


Abbildung 2: schematischer Aufbau eines mobilen WebGIS Clienten mit GXM

Die Abbildung oben skizziert schematisch den Aufbau eines mobilen WebGIS clienten auf Basis von GXM. Daraus ist ersichtlich das eigener Applikations-Code erforderlich ist, um eine eigene Client-Instanz zu entwickeln.

Komponenten von GXM

GXM besteht aus den folgenden Hauptkomponenten:

- Die Karte („MapPanel“), welches bereits jede Instanz einer OpenLayers Karte beinhaltet (die natürlich alle Layertypen beinhaltet, die OpenLayers auch unterstützt). Dieses wird in einen Panel gerendert, das in ein durch Sencha Touch bereitgestelltes Layout integriert werden kann.
- Eine Layerliste („LayerList“), die die Interaktion mit den unterschiedlichen thematischen Layern einer Mapping-Anwendung vereinfacht. Die Liste selber kann in jeden Container, der durch Sencha Touch bereitgestellt wird, integriert werden (dies kann ein Popup oder z.B. ein sogenanntes Tab-Panel sein).
- Eine Bedienelementklasse (Button class), welche die Interaktion mit OpenLayers Controls wie beispielsweise ein „zoom in“ erlaubt. Diese kann ebenso genutzt werden, um über die Geolokations-API (als Teil von HTML5) des Browsers die geographische Position des Gerätes abzufragen.

Anwendungsbeispiel - das Geoportal des Schweizer Bundes

Das Geodatenportal des Schweizerischen Bundes wird täglich von rund 10.000 Nutzern frequentiert. Über das Geoportal, welches clientseitig aus den Komponenten OpenLayers und GeoExt aufgebaut ist, können Geodaten betrachtet, ausgedruckt und bestellt werden. Seit letztem Herbst existiert nun auch die Möglichkeit des Zugriffs von mobilen Geräten (Smartphones).

Nutzer können sich zum Beispiel an Ort und Stelle darüber informieren, ob sie sich auf einem historischen Verkehrsweg befinden oder Informationen über historische Gebäude in ihrem Blickfeld abrufen. Es kann ebenso nach Gemeinden, Ortschaften, Gewässern oder Bergen gesucht werden („Welcher Berg ist das?“). Weiter erhältlich sind unter anderem Informationen über Gefahrenzonen („Befinde ich mich hier in einer Gefahrenzone?“), Naturschutzgebiete („Ist das hier ein Amphibienlaichgebiet?“) so wie Stadt- und Siedlungsentwicklung („Wie sah es hier vor 100 Jahren aus?“).

Zusätzlich stehen folgende Funktionen zur Verfügung:

GeoExt Mobile (GXM)

- eine Adresssuche (aktuelle Adressen sowie im Bau befindliche Gebäude)
- eine Suche nach Orts- und Flurnamen
- eine Feedbackfunktion, mit der Korrekturen von Dateninhalten gemeldet werden können (z.B. Wanderwege, die nicht wie angegeben verlaufen, oder falsche Gebäudeadressen samt Angabe der richtigen Adresse.)

In Planung oder in Umsetzung sind zudem mobile Clients für das Geoportal Brandenburg [8], des Geoportal BergIS [9] und das Freizeitportal Regiofreizeit [10].

Ausblick - Mögliche Anwendungsfälle

Das oben skizzierte Beispiel sowie die nachfolgenden Projekte zeigen bereits einen Trend: Für große Portallösungen ist es unerlässlich, neben dem reinen WebGIS Clienten auch einen mobilen Clienten zu haben.

Es lassen sich aber leicht weitere Anwendungsfälle, vom Restaurantfinder oder auch insbesondere für den kommunalen Bereich entwickeln. Dabei kann eine Anwendung geschaffen werden, die es unter dem populären Stichwort „offene Verwaltung“ erlaubt, ein direktes Feedback von Seiten der Bürger zur Verwaltung zu geben. Dies könnte einerseits eine Art Mängelmelder für BürgerInnen einer Stadt sein, so dass auf Missstände wie ein Schlagloch oder einen verdreckten Spielplatz vor Ort direkt hingewiesen werden kann. Dies könnte auch eine Anwendung für ein Baumkataster sein, wo Mitarbeiter vom Betriebshof der Stadt Ihre Bestandsaufnahmen direkt vor Ort in eine zentrale Datenbank dokumentieren können.

Andere Anwendungsbereiche auch für Firmen, deren Liegenschaften verteilt liegen, lassen sich entwickeln. So können nicht nur Prüfungen vor Ort dokumentiert werden, sondern entsprechende Prüfer auch durch das Gerät zum zu prüfenden Objekt geleitet werden.

Kontakt zum Autor:

Marc Jansen & Till Adams
terrestris GmbH & Co KG
Pützchens Chaussee 56, 53227 Bonn
+49 – (0)228 962 899 54
jansen@terrestris.de / adams@terrestris.de

Literatur

- [1] <https://github.com/geoext/GXM>
- [2] <http://trac.geoext.org/wiki/mobile>
- [3] <http://openlayers.org>
- [4] <http://www.sencha.com/products/touch>
- [5] <http://www.geoext.org>
- [6] <http://mobile.map.geoadmin.ch>
- [7] <http://www.openxcell.com/sencha-touch.html>
- [8] <http://geoportal.brandenburg.de>
- [9] <http://www.bergis.at>
- [10] <http://www.regiofreizeit.de>

WPS-Appstore

Markplatz für freie Algorithmen

Bastian Schäffer

Seit einiger Zeit ändert sich das Paradigma zur Verarbeitung von Geodaten: Weg von monolithischen GIS Systemen hin zu verteilten Prozessierungsdiensten mit offenen Schnittstellen (i.e. OGC Web Processing Service (WPS)). Dieser Vortrag stellt eine Weiterentwicklung vor, die es ermöglicht freie Verarbeitungslogiken (Algorithmen und Modelle) in einfacher Weise aufzufinden und in seiner eigenen vertrauenswürdigen und performanten Infrastruktur (Desktop, Server, Private Cloud) auszuführen. Mittels eines Markplatzes „WPS-Appstore“ lassen sich Algorithmen in einfachster Weise auffinden und austauschen. Revolutionär ist hier das angewandte Prinzip des „Moving Code“ bei dem die Prozesse zu den Daten kommen und nicht umgekehrt. Dies ermöglicht die rasante Verarbeitung auch von großen Datenmengen, da sie nicht transferiert werden müssen. Weiterhin ist man so nicht auf fremde, fragwürdige oder nicht verfügbare Dienste angewiesen. Der Vortrag stellt das zugrundeliegende Konzept und Anwendungsbeispiele da und zeigt in anschaulicher Weise die Vorteile gegenüber klassischen Geoprocessingmethoden auf.

Marble - ein Schweizer Taschenmesser für Karten

Neues vom freien virtuellen Globus

Thorsten Rahn

Auf der letzten FOSSGIS wurde Marble vorgestellt: ein freier virtueller Globus und ein "Schweizer Taschenmesser" für Kartenmaterial. Mit Marble kann der Anwender spielerisch die Welt erkunden oder er kann Marble als kleines Navi verwenden. In diesem Vortrag präsentieren wir den Entwicklungssprung, den Marble im vergangenen Jahr gemacht hat: Marble's Routing verfügt inzwischen über Sprachnavigation, die den Anwender in Sprachen wie Deutsch, Englisch, Spanisch oder Hindi zum Ziel führen kann. Als Teil von Marble 1.3 wird "Marble Touch" ausgeliefert: eine spezielle Version von Marble, die vor allem an Smartphones und Tablet-Geräte angepasst ist. Außerdem zeigen wir Neuerungen in Marble's Unterstützung von OpenStreetMap: Dazu gehören die ersten Schritte des OSM Vektorrenderings und die Unterstützung von OSM basierten Kartenthemen, wie zum Beispiel OpenSeaMap. Schließlich werfen wir noch einen Blick ins All: Dort stellen wir die Neuerungen vor, die im Rahmen des "ESA Summer of Code in Space" umgesetzt wurden.

OSM 3D

Der Gemeinschaftsvortrag von Tobias Kneer, Matthias Meißer und Matthias Uden geht auf aktuelle und zukünftige Entwicklungen rund um das Thema OSM 3D ein.

OSM2World erzeugt dreidimensionale Szenen aus OpenStreetMap-Daten. Das flexible Konzept der Software erlaubt die Erstellung von Standbildern oder Kartenkacheln ebenso wie den Export von Dateien in gängigen 3D-Modellformaten oder das Echtzeitrendering mit OpenGL. Der Vortrag informiert über Anwendungsmöglichkeiten, den aktuellen Stand der Entwicklung und Pläne für die Zukunft.

Das Thema 3D ist gerade sehr in Bewegung bei OSM. Es gibt bereits seit längerem einige Lösungen wie OSM3D OSM2pov oder OSM2World die Bereiche der Weltkarte nehmen und zum Beispiel Gebäude in die dritte Dimension extrudieren. Doch für viele Anwendungen wie Simulationen, Spiele und natürlich die Argumented- und Virtual-Reality sind es gerade die kleinen Objekte, die eine Wiedererkennung ausmachen. Außerdem bietet eine Erfassung mittels des OSM Datenmodells nur sehr eingeschränkte Möglichkeiten für 3D Modellierung.

O3DM soll dagegen ein Portal werden, dass es erlaubt 3D Modelle gegen OSM Objekte zu linken. So können generelle Objekte (Poller, Ampeln, ...) modelliert werden und natürlich ebenso Gebäude mit einem hohen Detaillierungsgrad (Schlösser, Denkmäler,...)

So soll eine weitreichende Alternative zu dem Google 3D Warehouse entstehen, über die für OSM hoffentlich noch weitere Designer und Entwickler aus dem klassischen 3D Umfeld gewonnen werden können. Den bisherigen Entwicklern, die sich mit der Thematik 3D beschäftigen, soll so ein neues Detaillevel gemeinsam nutzbar gemacht werden. Eine größere Projektvorstellung folgt in diesem Talk.

OpenStreetMap hat sich in letzter Zeit von einer reinen, zwei-dimensionalen Straßenkarte zu einer umfangreichen Sammlung von verschiedensten Datentypen und Inhalten entwickelt. Längst geht die abgelegte Information über eine einfache Straßenkarte hinaus. Die erstellten Geometrien werden immer detaillierter und beschreiben in manchen Regionen kleinräumige Phänomene wie einzelne Wanderwege oder U-Bahn-Trassen. Darüber hinaus wird auch die Semantik verbessert, indem vielfältige Zusatzbeschreibungen in Form von verschiedenen tags abgelegt werden. Dazu zählen vermehrt auch dreidimensionale Informationen, vor allem im Bezug auf kartierte Gebäude (z.B. Höhe, Dachform usw.). Die Anzahl der Gebäudegrundrisse hat vor kurzem die Anzahl der Straßen in OSM überholt. Dies zeigt, dass der Fokus zunehmend von den Straßen und Wegen hin zu Gebäuden und anderen Objekten unserer Umgebung geht. Die Welt besteht aus drei Dimensionen und die zusätzliche Höheninformation ermöglicht verschiedene neue Anwendungsszenarien. Beispiele sind in der Stadtplanung (Lärmkartierung, Sichtbarkeitsanalyse, Telekommunikation etc.), dem Katastrophenmanagement (Hochwasseranalyse, Evakuierung etc.) und vor allem der mobilen Fußgängernavigation zu finden. Es ist daher zu begrüßen, dass auch OSM immer mehr dreidimensionale Daten enthält. Diese Daten können genutzt werden, um 3D-Modelle von Gebäuden und dem Gelände zu erstellen, um somit der virtuellen Welt (Digital Earth) näher zu kommen. Projekte wie OSM-3D (www.osm-3d.de) und zahlreiche Diskussionen im OSM Wiki zeigen, dass das Interesse in der Community an 3D schnell zunimmt und auf große Zustimmung stößt. Die Entwicklung steckt allerdings noch in den Kinderschuhen und es gibt viele offene Fragen hinsichtlich der Erfassung, Verarbeitung und Präsentation der 3D-Daten in OSM. Der Vortrag zeigt die neuesten Entwicklungen rund um die dritte Dimension in der Community auf und beschreibt insbesondere die neuesten Arbeiten an der Universität Heidelberg im Bereich OSM-3D. Dabei reichen die Ideen und Möglichkeiten von der Erweiterung der 3D-Tags, über die Verfeinerung/Erweiterung von 3D Viewern für OSM Daten, über die Entwicklung von 3D Indoor OSM für Innenräume, bis hin zur kollaborativen Entwicklung, Bereitstellung und Nutzung einer Datenbank mit vollständigen 3D Modellen von Gebäuden oder Gegenständen in unserer Umgebung. Auch auf die Verknüpfung von OSM mit offenen Geodatenstandards wird eingegangen.

OpenIndoorMap – Erfassung von Innenräumen mittels Smartphone

Frederic Kerber, Florian Daiber, Antonio Krüger

Zusammenfassung

In den vergangenen Jahren erfreut sich das Projekt OpenStreetMap (OSM) wachsender Beliebtheit sowohl auf Seiten der Anwender wie auch der Mapper. Bislang beziehen sich die erfassten Daten fast ausschließlich auf Gebiete im Freien (z.B. Straßendaten). Allerdings könnte von Anwendungen wie Routenplanung oder Navigation auch in Innenräumen profitiert werden. Da traditionelle Erfassungsmethoden (GPS, Satellitenbilder) in Innenräumen nicht zuverlässig funktionieren, existieren bisher nur wenige Gebäude in OSM, deren innere Strukturen erfasst wurden. Um Erkenntnisse über das Interesse an der Erfassung von Innenräumen zu erlangen, wurde zunächst eine Online-Umfrage durchgeführt, in der das Projekt vorgestellt wurde. Im Zuge dessen wurde unter anderem auch erfragt, ob die Teilnehmer bereit wären, an einem entsprechenden Projekt mitzuarbeiten. Damit dies realisiert werden kann, bedarf es Methoden zur Innenraumerfassung, die intuitiv nutzbar sind. Um eine Vielzahl von Nutzern ansprechen zu können, sollten diese Ansätze ohne spezielle und unter Umständen teure Hardware realisierbar sein.

Moderne Smartphones, die mit einer Vielzahl von Sensoren (z.B. Kompass, Beschleunigungssensor, Kamera) ausgestattet sind, finden zunehmend Verbreitung. Es werden zwei Smartphone-gestützte Verfahren vorgestellt, die eine Erfassung von Innenräumen ermöglichen und so ein OpenIndoorMap (OIM) realisierbar machen. Der erste Ansatz nutzt Sensordaten, um die Bewegungen des Nutzers mitzuverfolgen. Dies erlaubt die Erfassung des Grundrisses einzelner Räume bzw. des gesamten Gebäudes. Dieser kann durch Zusatzeingaben während der Erfassung (z.B. Position von Türen) ergänzt werden. Der zweite Ansatz erlaubt eine Vermessung von Innenräumen mittels Kameraunterstützung und Neigungsbestimmung des Smartphones. Zur Erfassung muss der Nutzer lediglich mit einem Zielkreuz im Kamerabild die Ecken des Raumes anpeilen, um die Raumdimensionen zu ermitteln.

Während der erste Ansatz eine schnelle, aber grobe Erfassung ermöglicht, bietet der zweite Ansatz auch dann gute Ergebnisse, wenn Teile des Raumes verstellt sind (z.B. durch Schränke). Eine höhere Genauigkeit wird erzielt, da die Erfassung mittels Anpeilung der Raumecken an der Decke ermöglicht wird. Die beiden vorgestellten Verfahren ermöglichen so ein Mapping von Innenräumen, ohne auf externe Systeme wie GPS oder eine Instrumentierung der Umgebung angewiesen zu sein.

Motivation

Das Projekt OpenStreetMap (OSM) erfreut sich speziell in den letzten Jahren immer größerer Beliebtheit. Aktuell (Februar 2012) gibt es über 500.000 registrierte Nutzer, die dazu beitragen, den Datenbestand einerseits zu ergänzen, andererseits bestehende Daten zu pflegen und zu aktualisieren. Die Standard-Verfahren zum Mapping neuer Gebiete vertrauen dabei in erster Linie auf Techniken wie GPS oder das Vorhandensein entsprechender Luftbilder. Bei Verwendung geeigneter GPS-Geräte wird das neu zu erfassende Gebiet abgegangen bzw. abgefahren und die aufgezeichnete Route dann am Computer genutzt, um entsprechende Straßen in den Datenbestand von OSM einfügen zu können. Demgegenüber steht das Mapping ohne direkt vor Ort sein zu müssen, beispielsweise durch das Abzeichnen geeigneter Luftaufnahmen. Somit ist es gelungen, eine sehr große Abdeckung von Straßen, Fußwegen und ähnlichem zu erreichen [4]. Die Daten lassen sich für zahlreiche Zwecke einsetzen: neben Straßenkarten gibt es auch spezielle Karten für Radfahrer oder auch Routing-Applikationen, die eine Routenplanung mit diesen Daten ermöglichen.

In den letzten Jahren hat die Zahl erfasster Gebäudeumrisse enorm zugenommen [6]. Dies wirft die Frage auf, inwieweit Applikationen auch von detaillierten Daten über das Gebäudeinnere profitieren könnten. Werden komplexe Gebäudestrukturen betrachtet, wie sie beispielsweise in großen Firmen, an Flughäfen oder in Einkaufszentren vorkommen, so wird erkennbar, dass Nutzer hier auch profitieren könnten, wenn sie nicht nur bis zum Gebäudeeingang navigieren können, sondern direkt bis zum gewünschten Büro oder Geschäft. Auch für Rettungskräfte sind unübersichtliche Gebäude ein Problem [19], sodass auch hier ein positiver Effekt erzielt werden könnte. Eines der Hauptprobleme bei der Erfassung von Innenräumen stellt dabei die Tatsache dar, dass etablierte Mapping-Methoden nicht angewandt werden können. GPS-Signale stehen meist nicht in ausreichender Genauigkeit zur Verfügung, ebenso entfällt die Möglichkeit, Luftbildaufnahmen nutzen zu können. Daher müssen Alternativen zur Erfassung von Innenräumen gefunden werden, ohne auf externe Systeme angewiesen zu sein. Dabei sind Lösungen wie das manuelle Abzeichnen von Gebäudeplänen zwar grundsätzlich denkbar, aber einerseits sehr zeitintensiv und andererseits nur schwer umsetzbar, da maßstabsgetreue Gebäudepläne nur selten zur freien Verwendung verfügbar und wenn, meist nur einzelnen Personen zugänglich sind. Dem gegenüber steht der Gedanke des Crowdsourcings, bei dem Freiwillige verteilt Daten erfassen und so schnell einen großen Datenbestand erzeugen können, wie es auch bei OSM der Fall ist. In diesem Zusammenhang spricht man auch von Volunteered Geographic Information (VGI) [8]. Auch Google Maps erweiterte kürzlich das Angebot um Innenraum-Karten¹⁵ und setzt ebenfalls auf einen Crowdsourcing-Ansatz¹⁶, der es jedem erlaubt, neue Gebäudepläne hinzuzufügen.

Im folgenden Abschnitt werden zunächst verwandte Arbeiten vorgestellt, die sich mit Verfahren beschäftigen, die für eine Innenraumerfassung ohne Instrumentierung oder Verwendung externer Systeme in Frage kommen. Anschließend werden die Ergebnisse einer vorab durchgeführten Umfrage zu den Themenkomplexen OpenStreetMap und OpenIndoorMap präsentiert. Der vierte Abschnitt beschreibt die Implementierung der beiden Smartphone-gestützten Erfassungsverfahren. Abschließend werden erste Ergebnisse präsentiert sowie erläutert, wie das Projekt zukünftig weitergeführt werden kann.

Verwandte Arbeiten

Es existiert eine Vielzahl von Arbeiten, die sich mit der Positionierung von Nutzern in Innenräumen befassen. Die dort verwendeten Ansätze sind oftmals ebenso verwendbar, wenn es darum geht, die Maße eines Raumes zu erfassen. Dabei gibt es einerseits Systeme, die die Position des Benutzers mittels externer Referenzen (z.B. Infrarot-Baken [2] oder W-Lan [7]) bestimmen. Da dies allerdings eine vorherige Instrumentierung der Umgebung erfordert, die einerseits zeit- und andererseits kostenintensiv sein kann, werden diese Systeme hier nicht weiter betrachtet. Demgegenüber stehen Systeme, die lediglich mit Hilfe von Sensoren wie Beschleunigungssensoren, Gyroskopen oder Magnetfeldsensoren die Nachverfolgung der Nutzer-Position erlauben. Dieser Prozess des Bestimmens der aktuellen Position auf Basis einer bekannten Position sowie Betrachtung von Richtung und Geschwindigkeit wird als Dead-Reckoning bezeichnet [16]. Dieser Ansatz findet sich in zahlreichen Arbeiten wieder:

Cho und Park entwickelten ein System [3] auf Basis eines Schrittzählers. Mit Hilfe der Werte des Beschleunigungssensors, den der Nutzer an seinen Schuhen trägt, werden einzelne Schritte identifiziert und dann mit der Schrittlänge multipliziert um den zurückgelegten Weg zu bestimmen. Dabei ist insbesondere die Bestimmung der Schrittlänge ein kritischer Punkt, der von verschiedenen Faktoren abhängen kann (vergleiche auch [9, 10, 12]). Cho und Park nutzen für ihr System ein neuronales Netzwerk zur Bestimmung der Schrittlänge sowie einen Kalman-Filter zur Korrektur von Fehlern, die beispielsweise durch Störungen des Magnetfeldes auftreten können. In ihrem Personal Odometry System

15 <http://googleblog.blogspot.com/2011/11/new-frontier-for-google-maps-mapping.html>, abgerufen am 20.02.2012

16 <http://maps.google.com/help/maps/floorplans/>, abgerufen am 20.02.2012

(POS) [14, 15] verfolgen Ojeda und Borenstein einen Ansatz, der sich des sogenannten Zero Velocity Update (ZUPT)-Verfahrens bedient. Dabei wird sich zu Nutze gemacht, dass bei einer Schrittbewegung die Schuhsohle zu einem Zeitpunkt Kontakt mit dem Boden hat. Währenddessen dürften keine Beschleunigungswerte gemessen werden – Abweichungen davon sind das Resultat von Störungen und können so korrigiert werden. Der Nachteil dieser Methode ist, dass ihre Anwendung eine Anbringung der Sensoren an den Schuhen des Nutzers erfordert. Bei anderen Befestigungspunkten wie beispielsweise am Kopf [1] oder an der Hüfte [11] sind daher andere Verfahren notwendig. Dabei wird in vielen Fällen darauf zurückgegriffen, die Messwerte verschiedenartiger Sensor-Typen zu kombinieren, um mit den Vorteilen des einen Typs, Nachteile eines anderen auszugleichen. Da auch handelsübliche Smartphones mit einer Vielzahl von Sensoren ausgestattet sind, haben sich verschiedene Arbeiten damit beschäftigt, wie entsprechende Ergebnisse mit diesen Geräten erzielt werden können. Das DRec-System [5] von Dekel und Schiller zeigt eine Implementierung des Dead-Reckoning-Ansatzes auf einem iPhone 3GS. Dabei konnten erste vielversprechende Ergebnisse (Fehlerrate $\leq 6.3\%$) erzielt werden. Das PINwl-System von Löchtfeld et al. [13] zeigt ebenfalls mit einem iPhone 3GS, dass es unter Verwendung der Messwerte von Beschleunigungs- und Magnetfeld-Sensor möglich ist, die Positions- und Richtungsveränderungen eines Nutzers auf einer zugrundeliegenden Karte eines Stockwerks zu visualisieren. Auch das Indoor Pedestrian Navigation System [17] von Serra et al. zeigt, dass sowohl Veränderungen der Nutzer-Position wie auch seiner Orientierung für Strecken bis zu 100 Metern in Innenräumen zuverlässig erkannt werden können.

Neben den angesprochenen Dead-Reckoning-Ansätzen besteht auch die Möglichkeit, die Orientierung eines Smartphones zu nutzen, um damit die Maße eines Raumes bestimmen zu können, ohne ihn abgehen zu müssen. Unserer Kenntnis nach existieren in diesem Bereich keine wissenschaftlichen Veröffentlichungen. Allerdings gibt es eine App für iOS (iPhone 4/4S, iPod (4. Generation) und iPad 2), die diesen Ansatz umsetzt: MagicPlan¹⁷ erlaubt das Vermessen eines Raumes indem Bilder der Ecken des Raumes (mit oder gegen den Uhrzeigersinn) angefertigt werden. Hierzu werden die Werte des Gyroskops genutzt. Mehrere Videos auf der Herstellerseite zeigen den Funktionsreichtum der App sowie die grundsätzliche Funktionalität. Die zahlreichen Nutzerkommentare zeigen das große Interesse an einer solchen Anwendung. Für das Android-Betriebssystem existiert bislang keine Anwendung, die das direkte Vermessen eines Raumes erlaubt. Allerdings gibt es mehrere Anwendungen (z.B. SmartMeasure¹⁸ oder Camera Arc Calculation¹⁹), die das Ausmessen einzelner Gegenstände ermöglichen. Somit ist es theoretisch möglich, einzelne Wände zu vermessen. Allerdings muss dabei auf das automatische Zeichnen eines Raumplanes verzichtet werden.

Ergebnisse einer Vorab-Umfrage zu OSM/OIM

Wie im ersten Abschnitt angesprochen, hängt der Erfolg eines Projektes wie OSM oder auch OIM sehr davon ab, dass sich viele Freiwillige an der Erfassung und Pflege der Daten beteiligen. Daher sollte im Vorfeld untersucht werden, inwieweit ein Interesse für OIM besteht und auf welche Aspekte bei der Umsetzung besonders Wert gelegt werden sollte. Die dazu erstellte Online-Umfrage wurde dabei zunächst über soziale Netzwerke verbreitet, um eine breite Teilnehmerschicht ansprechen zu können. Dabei zeigte sich, dass bislang nur etwa die Hälfte der Teilnehmer das OSM-Projekt kannte. Lediglich 23 % der Teilnehmer hatten bereits eigene Erfahrungen mit OSM gesammelt. Daher wurde die gleiche Umfrage nochmals gezielt über die deutsche OSM-Mailingliste talk-de publiziert. Die Ergebnisse werden im Folgenden betrachtet. Dabei wird zwischen den Internet-Nutzern unterschieden, die die Umfrage bei der ersten Verteilung beantworten haben, und den OSM-Experten, die über die Mailingliste erreicht wurden.

17 <http://www.sensopia.com>, abgerufen am 20.02.2012

18 <https://market.android.com/details?id=kr.sira.measure>, abgerufen am 20.02.2012

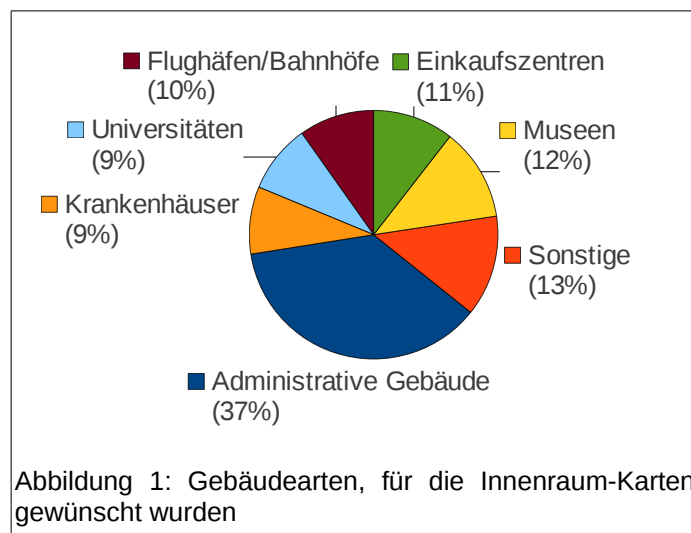
19 <https://market.android.com/details?id=de.javaresearch.android.camCalc>, abgerufen am 20.02.12

Ergebnisse der Umfrage unter Standard-Nutzern

Über einen Zeitraum von 14 Wochen haben sich insgesamt 765 Teilnehmer an der Umfrage beteiligt, wobei 256 Teilnehmer (33,4 %) die Umfrage vorzeitig abgebrochen haben. Die verbliebenen 509 Teilnehmer (381 männliche, 127 weibliche) wurden zunächst danach befragt, welche der zur Auswahl stehenden Kartenanbieter (Google Maps, Bing Maps, Yahoo Maps, OpenStreetMap) sie kennen. Während 98,4 % angaben, Google Maps zu kennen, kannten nur 53 % auch OpenStreetMap.

Lediglich 27 Teilnehmer äußerten, bereits Daten in OpenStreetMap editiert oder ergänzt zu haben. Im Folgenden wurden diesen Teilnehmern spezifische Fragen bezüglich der OpenIndoorMap-Idee gestellt. Zwei Drittel von ihnen fanden die Idee OpenIndoorMap sinnvoll oder sehr sinnvoll. 18 Teilnehmer gaben an, auch an dem Projekt mitarbeiten zu wollen. Die Mehrheit wäre bereit zur Erfassung von neuen Karten ein mobiles Endgerät (Tablet-PC oder Smartphone) zu nutzen. 19 Teilnehmer, die bislang nichts zum Datenbestand von OpenStreetMap beigetragen haben, äußerten, dass sie lieber an OpenIndoorMap mitarbeiten würden.

Die Befragten haben 18 verschiedene Gebäudearten angegeben, für die es Innenraum-Karten geben sollte (siehe Abbildung 1). Am häufigsten genannt wurden dabei administrative Gebäude wie Rathäuser (210 Nennungen), Museen (69 Nennungen), Einkaufszentren (60 Nennungen), Flughäfen und Bahnhöfe (56 Nennungen) sowie Universitäten (51 Nennungen), dicht gefolgt von Krankenhäusern (50 Nennungen). Insgesamt 28,5 % der Befragten hatten auch bereits mindestens einmal den Wunsch, solche Karten nutzen zu können.



Ergebnisse der Umfrage unter Experten

Durch die zielgerichtete Verteilung der Umfrage konnten 149 erfolgreich abgeschlossene Teilnahmen (69,6% der begonnenen 214 Teilnahmen) von Mitgliedern aus der OSM-Community verzeichnet werden. Im Gegensatz zu anderen Umfragen im Kontext von OSM (beispielsweise [18]) zeigt sich die stärkste Altersgruppe im Bereich der 20- bis 29-Jährigen. Dieser Effekt könnte durch die hauptsächliche Verbreitung über die Mailingliste hervorgerufen sein. Lediglich 3,4% der Befragten haben bislang noch keine Veränderungen am Datenbestand von OSM durchgeführt.

Mehr als 63% der Teilnehmer fanden dabei die Idee von OpenIndoorMap sinnvoll oder sehr sinnvoll. Eine große Zahl der Befragten (26,1%) gab an, noch unentschieden zu sein wohingegen nur 7,3% äußerten, die Idee weniger sinnvoll oder unnötig zu finden. Knapp 64% der Umfrageteilnehmer erklärten, auch Daten zum Bestand von OpenIndoorMap hinzufügen beziehungsweise vorhandene pflegen zu wollen. Dabei wären sie im Schnitt bereit, 45 Minuten pro Woche an diesem Projekt zu arbeiten. Die 74 Befragten (51%), die sich schon einmal Karten für Innenräume gewünscht hatten, gaben ähnliche Anwendungsbereiche an, wie sie auch schon in Abschnitt 3.1 genannt wurden. Lediglich 40,2% der Teilnehmer (60 Personen) gaben an, ein Smartphone oder ein Tablet zu besitzen. 45% von ihnen würden dabei das Erfassen neuer Innenräume ausschließlich oder zumindest überwiegend mit Hilfe einer PC-Applikation erledigen wollen. Demgegenüber stehen weitere 45%, die sowohl eine PC-Applikation als auch eine mobile Anwendung nutzen würden sowie weitere 10%, die eine Erfassung mit einem mobilen Endgerät vorziehen würden. Hierbei sollte erwähnt werden, dass einige Teilnehmer Bedenken geäußert haben, eine Vor-Ort-Erfassung sei aufgrund mangelnder GPS-Signale nicht möglich. Diese

OpenIndoorMap – Erfassung von Innenräumen mittels Smartphone

Befürchtungen könnten mit ein Grund dafür sein, dass eine mobile Erfassung direkt im Objekt nicht als bevorzugte Lösung angesehen wird.

Aus den Ergebnissen der beiden Umfragen lassen sich einige Schlüsse ziehen. Zum Einen hat sich gezeigt, dass die Idee von OIM bei aktiven OSM-Nutzern grundsätzlich Anklang findet und die Umfrageteilnehmer bereit wären, auch an diesem Projekt mitzuwirken beziehungsweise erstmalig sich an einem solchen Projekt zu beteiligen. Ebenso hat sich gezeigt, dass sich viele Teilnehmer bereits Karten für Innenräume gewünscht haben, gerade in Bereichen wie der öffentlichen Verwaltung, Museen oder Einkaufszentren. Zum Anderen hat sich aber auch gezeigt, dass eine häufiger angesprochene Hürde die Erfassung vor Ort darstellt.

Im Folgenden werden daher zwei Verfahren vorgestellt, mit denen eine Erfassung mittels Smartphone direkt vor Ort erfolgen kann, ohne dass hierbei auf externe Systeme wie GPS zurückgegriffen werden muss und ohne, dass eine vorherige Instrumentierung der zu erfassenden Umgebung erforderlich ist.

Erfassungsmethoden für Innenräume mittels Smartphone

Im Rahmen dieser Arbeit wurden zwei verschiedene Verfahren entwickelt, die jeweils in unterschiedlichen Situationen ihre Vorteile ausspielen können. Auf beide Verfahren wird in den folgenden Abschnitten näher eingegangen.

Beim ersten Ansatz übernimmt das Smartphone einerseits die Aufgabe eines Schrittzählers, ermittelt aber andererseits auch die Richtung, in die der Nutzer sich bewegt. Dieser Ansatz erlaubt eher eine Erfassung von freien Räumen wie Fluren oder Hallen, da es erforderlich ist, die Außenwände des zu erfassenden Raumes abzugehen. Gleichzeitig bietet der Ansatz auch die Möglichkeit, Details wie die Anzahl der Treppenstufen einer Treppe leicht zu erfassen. Der zweite, kameragestützte, Ansatz ist eher für kleinere Räume geeignet, da er dort sehr präzise Ergebnisse liefern kann. Er eignet sich beispielsweise gut für Büroräume, in denen ein Abgehen der Außenwände aufgrund von Möbeln oftmals gar nicht möglich ist und somit eine Anwendung des ersten Ansatzes ausscheidet.

Innenraumerfassung mit Schrittzähleransatz

Bei initial durchgeführten Tests mit einem Smartphone und drei Testpersonen hat sich gezeigt, dass sich insbesondere die Werte des Beschleunigungssensors auf der Z-Achse des Smartphones eignen, um einzelne Schritte identifizieren zu können. Wie in Abschnitt 2 angesprochen ist die korrekte Bestimmung der Schrittlänge ein mit entscheidender Faktor für die Qualität des Gesamtergebnisses. Da für den hier betrachteten Fall einige Annahmen getroffen werden können, wird zunächst von einer konstanten Schrittlänge ausgegangen. Zu diesen Annahmen zählt insbesondere die Tatsache, dass die Erfasser die Aufgabe gezielt durchführen, also auch bewusst auf eine entsprechend gleichmäßige Schrittgeschwindigkeit und -länge achten können. Faktoren wie unterschiedliche Laufgeschwindigkeiten oder Gangarten (zum Beispiel schleichen oder geduckt gehen) sind daher nicht zu berücksichtigen. Eine anschließende Evaluation wird zeigen müssen, inwieweit diese Annahmen zutreffend sind und ob die daraus geschlossene Betrachtung einer konstanten Schrittlänge haltbar ist. Zur Bestimmung der zurückgelegten Wegstrecke ist unter dieser Annahme also lediglich die vorherige Kalibrierung des Endgerätes auf die Schrittlänge des Erfassers erforderlich.

Zur Bestimmung der Orientierung stehen unterschiedliche Möglichkeiten zur Verfügung. Zunächst können die Werte des Magnetfeld-Sensors betrachtet werden, der jeweils eine absolute Orientierung des Gerätes angibt. Die Differenz zweier aufeinander folgender Messungen beschreibt dabei dann die Drehung, die der Nutzer zurückgelegt hat. Bei ersten Versuchen hat sich allerdings gezeigt, dass die Messungen, zumindest in Innenräumen, unzuverlässig sind. Oftmals kommt es bereits bei ruhendem Smartphone zu Abweichungen von bis zu 10° durch Störungen in der Umgebung, beispielsweise durch elektrische Geräte. Daher wurde für Geräte, die über ein Gyroskop verfügen, darüber hinaus eine Erkennung implementiert, die die Orientierung des Nutzers mittels dieser beiden Sensor-Typen

bestimmt. Da das Gyroskop die Drehgeschwindigkeit misst, ist es erforderlich, die gemessenen Werte zu integrieren, um den zurückgelegten Winkel zu ermitteln. Dies ist problematisch, da Störungen in den Messungen durch das fortwährende Integrieren zu einem unbegrenzt wachsenden Fehleranteil führen. Daher ist es nicht möglich, über einen längeren Zeitraum alleine auf die Messungen des Gyroskops zu vertrauen. Stattdessen müssen auch die Messwerte des Magnetfeld-Sensors einbezogen werden, um eine über längere Zeit stabile Orientierungsbestimmung zu gewährleisten. Für diese sogenannte Sensor-Fusion gibt es verschiedene Verfahren: beispielsweise wird in [11] ein Kalman-Filter eingesetzt. Da dessen Anwendung allerdings mit rechnerischem Aufwand einhergeht und somit für ein mobiles Endgerät problematisch sein könnte, wird zunächst ein einfacherer, komplementärer Filter eingesetzt. Bei dessen Anwendung werden die integrierten Messwerte des Gyroskops zu einem großen Anteil einbezogen, um schnell auf Änderungen reagieren zu können. Die geringere Einbeziehung der Magnetfeld-Messungen garantiert langfristig stabile Messungen, ohne dass kurzfristige Störungen im lokalen Magnetfeld zu verfälschten Ergebnissen führen. Auch in diesem Fall wird die Evaluation zeigen müssen, ob die Verwendung eines solchen, einfach gehaltenen Filters, in der praktischen Anwendung hinreichend gute Ergebnisse liefern kann.

Kameragestützte Innenraumerfassung

Der in Abschnitt 4.1 vorgestellte Ansatz ist nicht mehr anwendbar, wenn ein Raum beispielsweise durch Schränke oder Tische so zugestellt ist, dass seine Außenwände nicht mehr abgegangen werden können. Daher wurde ein zweiter Ansatz entwickelt, der auch in diesen Situationen eine zuverlässige Bestimmung der Raumdimensionen ermöglicht. Hierbei wird das Smartphone mit Hilfe eines Zielkreuzes im Kamerabild vom Erfasser so ausgerichtet, dass mittels trigonometrischer Berechnungen die Maße von Wänden oder Türen bestimmt werden können. Hierzu ist es lediglich erforderlich, im Vorfeld der Messungen die Anwendung einmalig auf den Nutzer zu kalibrieren, indem die Höhe eingegeben wird, in der der Erfasser das Gerät hält. Mit Hilfe dieser Information sowie den Winkeln, in denen das Smartphone gehalten beziehungsweise gedreht wird, können dann die Größen der Wände und Türen bestimmt werden. Dabei muss der Nutzer nur die Ecken des Raumes bzw. der Türen wahlweise im oder gegen den Uhrzeigersinn anvisieren und eine Messung ausführen. Dabei bietet die Applikation zusätzlich die Möglichkeit, den Messmodus so umzuschalten, dass die Ecken nicht auf Bodenniveau vermessen werden müssen, sondern dies auch in Deckenhöhe geschehen kann. So können auch Ecken vermessen werden, in denen sich beispielsweise Schränke befinden, was ein Anvisieren der tatsächlichen Raumecke erschwert. Zusätzlich bietet dieses Verfahren dementsprechend auch die Möglichkeit, die Raumhöhe zu bestimmen. Auch bei diesem Ansatz ist es von großer Bedeutung, die Ausrichtung des Smartphones exakt bestimmen zu können. Daher wird hier ebenfalls, wenn verfügbar, auf die Messwerte des Gyroskops zurückgegriffen, um so ungenaue Messungen durch Störungen im lokalen Magnetfeld auszuschließen.

Erste Tests bei der Vermessung dreier Innenräume (Fläche ca. 12-30m²) haben vielversprechende Ergebnisse geliefert, aber auch hier wird eine größer angelegte Evaluation mit mehreren Testpersonen zeigen müssen, ob der Ansatz unter verschiedenen Bedingungen zuverlässige Ergebnisse erzielen kann.

Zusammenfassung und Ausblick

Diese Arbeit diskutiert die Möglichkeiten digitaler Karten für Innenräume und deren Erfassung mittels eines Smartphone-gestützten Ansatzes. Die durchgeführte Online-Umfrage hat gezeigt, dass die Erfassung von Innenräumen als problematisch eingestuft wird, da klassische Hilfsmittel wie GPS oder Satellitenbilder nicht zur Verfügung stehen. Um die Idee des Crowdsourcings, welche bei OSM gut funktioniert, auch auf OIM anwenden zu können, müssen daher andere Erfassungsmethoden gefunden werden, die ohne großen Aufwand und Kosten für eine Vielzahl von Nutzern anwendbar sind. Die beiden vorgestellten Methoden erfüllen diese Anforderungen und haben in ersten Tests vielversprechende Ergebnisse geliefert. Trotzdem wird erst eine systematische Evaluation zeigen können, wie

OpenIndoorMap – Erfassung von Innenräumen mittels Smartphone

zuverlässig die Ansätze in der Praxis funktionieren. Insbesondere Faktoren wie Störungen durch Metall in der Umgebung oder unterschiedliche Schrittverhalten können die Ergebnisse negativ beeinflussen. Wie bereits angesprochen muss auch untersucht werden, ob der verwendete Algorithmus zur Sensor-Fusion ausreichend gute Ergebnisse liefert oder ob eventuell andere Verfahren bessere Ergebnisse liefern können und trotzdem noch für die Verwendung auf einem Smartphone geeignet sind.

Kontakt zu den Autoren:

Frederic Kerber, Florian Daiber, Antonio Krüger
Deutsches Forschungszentrum für Künstliche Intelligenz GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken

Email: {Frederic.Kerber, Florian.Daiber, krueger}@dfki.de

Literatur

- [1] *Beauregard, S.*: A Helmet-Mounted Pedestrian Dead Reckoning System. In: Proceedings of the 3rd International Forum on Applied Wearable Computing (IFAWC), 2006, S. 1–11
- [2] *Butz, A.; Baus, J. und Krüger, A.*: Augmenting Buildings with Infrared Information. In: Information Proceedings of the International Symposium on Augmented Reality ISAR 2000, S. 93–96
- [3] *Cho, S. Y. und Park, C. G.*: MEMS Based Pedestrian Navigation System. The Journal of Navigation 59, 01, 2006, S. 135–163
- [4] *Ciepluch, B.; Jacob, R.; Mooney, P. und Winstanley, A.*: Comparison of the accuracy of OpenStreetMap for Ireland with Google Maps and Bing Maps. In: Proceedings of the Ninth International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences 2010, S. 337–340
- [5] *Dekel, A. und Schiller, E.*: DRec: exploring indoor navigation with an un-augmented smart phone. In: Proceedings of the 12th international conference on Human computer interaction with mobile devices and services, MobileHCI, 2010, S. 393–394
- [6] *Götz, M. und Zipf, A.*: Extending OpenStreetMap to Indoor Environments: Bringing Volunteered Geographic Information to the Next Level. In: Urban and Regional Data Management: Udms Annual 2011, S. 47–58
- [7] *Golden, S. A. und Bateman, S. S.*: Sensor Measurements for Wi-Fi Location with Emphasis on Time-of-Arrival Ranging. Mobile Computing, IEEE Transactions on 6, 10, 2007, S. 1185–1198
- [8] *Goodchild, M. F.*: Citizens as sensors: the world of volunteered geography. In: GeoJournal, 2007, S. 211–221
- [9] *Käppi, J.; Saarinen, J. und Syrjine, J.*: Mems-IMU Based Pedestrian Navigator for Handheld devices. In: Proceedings of the 14th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS), 2001, S. 1369–1373
- [10] *Ladetto, Q.*: On foot navigation: continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering. In: Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS), 2000, S. 1735–1740

- [11] *Ladetto, Q. und Merminod, B.*: In Step with INS – Navigation for the Blind, Tracking Emergency Crews. GPS World, 10, 2002, S. 30–38
- [12] *Lee, S. W. und Mase, K.*: Recognition of walking behaviors for pedestrian navigation. In: Proceedings of the IEEE International Conference on Control Applications, 2001, S. 1152–1155
- [13] *Löchtefeld, M.; Gehring, S.; Schöning, J. und Krüger, A.*: PINwl: pedestrian indoor navigation without infrastructure. In: Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries, NordiCHI, 2010, S. 731–734.
- [14] *Ojeda, L. und Borenstein, J.*: Non-GPS Navigation for Emergency Responders. In: International Joint Topical Meeting: "Sharing Solutions for Emergencies and Hazardous Environments", 2006
- [15] *Ojeda, L. und Borenstein, J.*: Non-GPS navigation with the personal dead-reckoning system. In: Unmanned Systems Technology IX, Vol. 6561 of Proceedings of Spie, 2007
- [16] *Randell, C.; Djiallis, C. und Muller, H. L.*: Personal Position Measurement Using Dead Reckoning. In: Proceedings of the Seventh International Symposium on Wearable Computers, 2003, S. 166–173
- [17] *Serra, A.; Carboni, D. und Marotto, V.*: Indoor pedestrian navigation system using a modern smartphone. In: Proceedings of the 12th international conference on Human computer interaction with mobile devices and services, MobileHCI, 2010, S. 397–398
- [18] *Stark, H.-J.*: Umfrage zur Motivation von Freiwilligen im Engagement in Open Geo-Data Projekten, In: FOSSGIS Tagungsband, 2010, S. 173–177
- [19] SWAT team blames Gehry architecture for delay in trapping Cleveland shooter. online, http://greg.org/archive/2003/05/10/swat_team_blames_gehry_architecture_for_delay_in_trapping_cleveland_shooter.html, 2003, zuletzt abgerufen: 20.02.2012

INSPIRE View Service in MapServer

Stephan Meißl

Dieser Vortrag gibt einen Überblick über den derzeitigen Stand der Implementierung in <http://mapserver.org> zur Unterstützung des von INSPIRE geforderten Darstellungsdienstes. Anhand von konkreten Beispielen werden verschiedene Möglichkeiten einer Umsetzung nach der technischen Anleitung (Technical Guidance) von INSPIRE gezeigt.

Diese technische Anleitung empfiehlt die Verwendung und Erweiterung des Web Map Service (WMS), ein offener Standard des Open Geospatial Consortium (OGC). MapServer unterstützt bereits alle gängigen WMS Versionen sowie einige weitere OGC Standards wie den Web Feature Service (WFS) und den Web Coverage Service (WCS).

Die Umsetzung der von INSPIRE empfohlenen Erweiterungen des bestehenden WMS in MapServer ist im <http://mapserver.org/development/rfc/ms-rfc-75.html> beschrieben.

Das OGC Coverage Konzept: Daten und Dienste

Konzepte, Standards, Umsetzung - von OGC bis INSPIRE

Peter Baumann

In diesem Vortrag stellen wir "Coverages" vor, als Verallgemeinerung von Rasterdaten wie z.B. Orthophotos. Mit dem WCS 2.0 Standard wurde das Coverage-Konzept neu definiert, mit GML und Sensor Web Enablement harmonisiert und vom WCS-Dienst unabhängig gemacht. Damit sind Coverages als universelle Datenstruktur für multi-dimensionale Sensor-, Bild- und Statistikdaten zwischen den OGC-Diensten austauschbar. Der Web Coverage Service (WCS) ist ein solcher Dienst, er erlaubt den Zugriff auf die originalen, weiterverarbeitbaren Daten (im Unterschied zum bildgebenden WMS).

Im Vortrag stellen wir das Coverage-Konzept sowie die aktuelle WCS 2.0 Suite vor und beleuchten Einfluss auf INSPIRE. Schließlich werden Open-Source Implementierungen vorgestellt.

Der Referent ist Chair der OGC WCS-Gruppe, Editor der Coverage-Standards sowie eingeladener Experte bei den INSPIRE-Arbeitsgruppen.

EOxServer, GDAL, MapServer - Zugang zu großen Archiven von Erdbeobachtungsdaten

Joachim Ungar



Abb. 1: EOxServer Logo

Infrastruktur für Erdbeobachtungsdaten spielt aufgrund des steigenden Angebots und Nachfrage eine immer wichtigere Rolle. Seit Beginn der satellitengestützten Erdbeobachtung vor knapp 30 Jahren steigt die Anzahl der Anbieter, sowie die räumliche und zeitliche Auflösung der Daten signifikant an. Dieses Angebot auf der einen sowie die steigende Nachfrage von Forschung und Industrie auf der anderen Seite, erfordern eine gut strukturierte, übersichtliche und hochperformante Infrastruktur.

EOxServer [1] ist ein Open Source Projekt, welches sich zum Ziel gemacht hat, Software zu entwickeln um große Bestände von Erdbeobachtungsdaten und deren Metadaten zu verwalten. Mittels MapServer werden diese Daten in Folge über offene Schnittstellen für Clients oder aber auch anderen Servern zur Verfügung gestellt.

Als standardisierte Schnittstelle für den Austausch Erdbeobachtungsdaten, oder Coverages, wurde WCS (Web Coverage Service) entwickelt. Das Ziel eines offenen Standards ist klar: je mehr Server und Clients offene Standards unterstützen, desto einfacher wird der Zugang zu verschiedensten Archiven und größer somit die Vorteile und Möglichkeiten der einzelnen Akteure.

Ein Teil der Weiterentwicklung von EOxServer wird von der ESA im Rahmen der Projekte HMA-FO [2] und O3S [3] finanziert.

Das EU Projekt "Earth-Server - European Scalable Earth Science Service Environment" [4] [5] versucht für ver-



Abb. 2: Beispielprojekte

schiedene Domänen der Erdwissenschaften mittels Open Source Softwarekomponenten Infrastruktur basierend auf offenen Standards zur Verfügung zu stellen. Während der dreijährigen Projektlaufzeit sollen Softwarekomponenten und Standards entwickelt beziehungsweise weiterentwickelt werden.

Dieser Vortrag gibt einen Überblick über das Potential einer derartigen Infrastruktur aus den Sichtweisen der Maintainer und der User sowie den Möglichkeiten von "User-Tailored Data Download" und serverseitigem "Geospatial Data Analytics".

Kontakt zum Autor:

Mag. Joachim Ungar
EOX IT Services GmbH
Thurngasse 8/4, 1040 Wien
+43 664 887 36 332
joachim.ungar@eox.at

Literatur

[1] <http://www.eoxserver.org>

[2] <http://wiki.services.eoportal.org/tiki-index.php?page=HMA-FO>

[3] <http://rssportal.esa.int/deepenandlearn/tiki-index.php?page=O3S+Project>

[4] EU FP7-INFRA Projekt 283610

[5] <http://earthserver.eu/>

djeo: eine einheitliche Open Source API für unterschiedliche geographische Web-Engines

Vladimir Elistratov

Im Vortrag wird die Javascript Bibliothek djeo vorgestellt. djeo ist für die Entwicklung geographischer Webanwendungen entworfen. djeo ist auf Dojo Toolkit basiert.

Man kann drei Besonderheiten von djeo unterscheiden. Erstens, wie mit anderen Open-Source und kommerziellen Bibliotheken, kann man Vektor-Objekte auf Bitmap-Tiles mit djeo darstellen. Zweitens, kann man das eigene Engine mit populären kommerziellen APIs (Google Maps, Google Earth, Yandex Maps) ersetzen. So derselbe Anwendungscode funktioniert sowohl mit dem eigenen Engine als auch mit anderen APIs. Drittens, statt grafischer Objekten auf einem Browserfenster kann mann eine textuelle Datei generieren. POV-Ray Format ist zur Zeit unterstützt, mit dem bekommt man photorealistische Darstellung von geographischen Objekten.

djeo hat eine Art Analogie mit HTML, CSS und Document Object Model. Mit djeo beschreibt man geographische Objekte als eine Baumstruktur. Ein Objekt erbt den Stil seines Vorfahren. Man kann ein Ereignis zu einem Objekt sowie zu einem Objektcontainer befestigen.

Webseite des Projekts mit dem Source-Code: <http://djeo.org> Demobeispiele: <https://github.com/vvoovv/djeo/wiki/demos>

SOS-Importer - Werkzeug zum Veröffentlichen von Daten im Sensor Web

Simon Jirka, Daniel Nüst

Sensordaten stellen eine immer wichtigere Informationsquelle dar, die von einer kontinuierlich wachsenden Anzahl von Nutzern verwendet wird, um Informationen über die Umwelt zu erhalten. Daher wurde die Idee des Sensor Webs entwickelt, dessen Ziel es ist, den interoperablen Austausch von Sensordaten zu ermöglichen und eine möglichst große Menge an Daten verfügbar zu machen. Typische Anwendungsbereiche sind die Umweltbeobachtung [1], der Verkehr [2] oder der Katastrophenschutz [3].

Die Sensor Web Enablement-Technologie (SWE) des Open Geospatial Consortiums stellt hierzu eine wichtige Basis dar [4]. Dieses aus verschiedenen Standards bestehende Framework bietet Definitionen von Schnittstellen zum Datenzugriff, zum Abonnieren von Messdaten und zum Steuern von Sensoren ebenso wie Datenformate zur Kodierung von Messungen und zugehörigen Metadaten.

Während die Anzahl von Dienst- und Client-Implementierungen stetig wächst, stellt die Veröffentlichung der Sensordaten für viele Datenanbieter noch eine Hürde dar. Dies liegt insbesondere an einem hohen Aufwand zur Einarbeitung in die verschiedenen SWE-Standards für potentielle Anbieter. Nach dem Studium der Standards ist dann oft eine Neustrukturierung der Daten für einen manuellen Import in einen Sensor Web-Dienst erforderlich. Erst dann ist er in der Lage seine Daten korrekt bereitzustellen.

Zur Vereinfachung dieses Prozesses wurde im Rahmen des EU-Projekts EO2HEAVEN (<http://www.eo2heaven.org/>, Grant Agreement 244100) die Entwicklung eines Werkzeugs begonnen, welches diese Problematik adressiert. Im Zuge von EO2HEAVEN sollen Fachanwender die Möglichkeit erhalten, ihre Daten anderen Fachkollegen interoperabel bereitzustellen. Allerdings ist von den Fachanwendern nicht zu erwarten, dass sie einen zusätzlichen Aufwand investieren können, um sich in die Sensor Web-Spezifikationen einzuarbeiten, bevor sie ihre Daten veröffentlichen. Stattdessen ist es wichtig, dass sie Werkzeuge erhalten, mit deren Hilfe die Veröffentlichung von Sensordaten auch ohne detaillierte Kenntnisse möglich ist. Insbesondere ist es wichtig, Sensor Web-spezifische Konzepte (z.B. „Observation Offering“ oder „Procedure“) weitgehend zu verbergen.

Das entwickelte Open Source-Werkzeug unterstützt Fachanwender durch einen Assistenten bei der Veröffentlichung von Sensordaten, welche in Form von Textdateien mit komma-separierten Werten vorliegen (sogenannte CSV-Dateien). Notwendige Metadaten für die Publikation im Sensor Web werden dabei vom Anwender abgefragt, wobei allgemeine Begriffe und domänenunabhängige Termini benutzt werden. Das Tool übernimmt die Transformation der Daten in die erforderlichen Datenmodelle und lädt die Informationen über standardisierte Schnittstellen zu bereits vorhandenen Serviceinstanzen hoch. Somit entfällt für Datenanbieter die Einarbeitung in die Sensor Web-Spezifikationen und der Aufwand, Datenbestände manuell in die Sensor Web-Datenstrukturen zu überführen.

Abb. 1 zeigt als Beispiel einen Schritt aus dem Importprozess, in welchem der Nutzer die Möglichkeit hat, die Inhalte der Spalten einer CSV-Datei zu beschreiben. Hierdurch erhält das Programm alle notwendigen Informationen um zu verstehen, welche Bedeutung die einzelnen Felder haben, und wie sie beim Importieren auf die passenden Sensor Web-Konzepte abgebildet werden können. Im dargestellten Fall wird die Bedeutung der ersten Spalte als Datums- und Zeitstempel festgelegt und das zugehörige Datums- und Zeitformat angegeben. In ähnlicher Weise werden nachfolgend die weiteren Spalten durch den Nutzer beschrieben (z.B. Bezeichnung des beobachteten Phänomens, Ort der Messung). Besonderes Augenmerk wurde auf die einfache Handhabung von Inkonsistenzen innerhalb der Daten und möglicherweise fehlerhafte Werte gelegt.

SOS-Importer - Werkzeug zum Veröffentlichen von Daten im Sensor Web

Nachdem der Inhalt einer CSV-Datei vollständig angegeben wurde und alle notwendigen Metadaten vorliegen, erfolgt das Einfügen der Daten in einen entsprechenden Sensor Web-Dienst, genauer gesagt einen sogenannten Sensor Observation Service (SOS), welcher das transaktionale Profil der SOS-Spezifikation unterstützt.

Step 3a: Choose Metadata for the selected column

???	n/a	n/a
Date & Time	SO2 (ppb)	TRS (ppb)
01/06/2010 00:00	1,3	3,9
01/06/2010 00:05	1,5	4,8
01/06/2010 00:10	1,6	5,3
01/06/2010 00:15	1,5	5,6
01/06/2010 00:20	1,3	5
01/06/2010 00:25	1,1	4,3
01/06/2010 00:30	1,4	3,9
01/06/2010 00:35	1,2	3,9
01/06/2010 00:40	1,2	4,2

☐ Undefined
☐ Measured Value
☒ Date & Time
☐ Position
☐ Feature of Interest
☐ Observed Property
☐ Unit of Measurement
☐ Sensor
☐ Do not export

☒ Combination
☐ UNIX time

Format: / X
Example: 05/08/2011 16:30 1 value not parseable.
Group:

Back Next

Abb. 1: Menü zum Beschreiben der vorliegenden Daten

Das vorgestellte Programm liegt in einer ersten Version vor, welche noch einen prototypischen Charakter besitzt. Wichtige Arbeitsschritte sind in der näheren Zukunft einerseits die Erweiterung des Werkzeugs um mit weiteren Datenformaten (z.B. netCDF) und Strukturvarianten von CSV-Dateien umgehen zu können. Hier sind die Autoren offen für weitere Anwendungsfälle und stets auf der Suche nach neuen Kooperationspartnern. Andererseits wird ein weiterer Schwerpunkt darauf liegen, die Nutzerfreundlichkeit, zum Beispiel durch Automation oder Mehrsprachigkeit der Benutzeroberfläche, weiter zu erhöhen und die zugrundeliegenden Sensor Web-Konzepte noch weiter zu verbergen.

Kontakt zum Autor:

Simon Jirka
52°North Initiative for Geospatial Open Source Software GmbH
Martin-Luther-King-Weg 24, 48155 Münster
Telefon: 0251-396371-31
E-Mail: jirka@52north.org

Literatur

SOS-Importer - Werkzeug zum Veröffentlichen von Daten im Sensor Web

- [1] Kjeld, Peter; Bliki, Jan; Jirka, Simon; Wytzisk, Andreas (2011): Sensor Web Technology for Sharing Environmental Data across Europe INSPIRE Conference 2011, Edinburgh, United Kingdom.
- [2] Jirka, Simon; Bredel, Henning (2011). Building Tracking Applications with Sensor Web Technology. Geoinformatik 2011, Münster.
- [3] Jirka, Simon, Bröring, Arne; Stasch, Christoph (2009). Applying OGC Sensor Web Enablement to Risk Monitoring and Disaster Management. GSDI 11 World Conference, Workshop: Sensorweb Enablement: Strengthening the SDI, Rotterdam, Niederlande.
- [4] Bröring, Arne; Echterhoff, Johannes; Jirka, Simon; Simonis, Ingo; Everding, Thomas; Stasch, Christoph; Liang Steve; Lemmens, Rob (2011). "New Generation Sensor Web Enablement." Sensors 11(3): 2652-2699.

OSGeo Jahresbericht 2011

Arnulf Christl

Der folgende Bericht von der Open Source Geospatial Foundation [0] fasst einige wichtige Entwicklungen des letzten Jahres zusammen (Download unter: [1]).

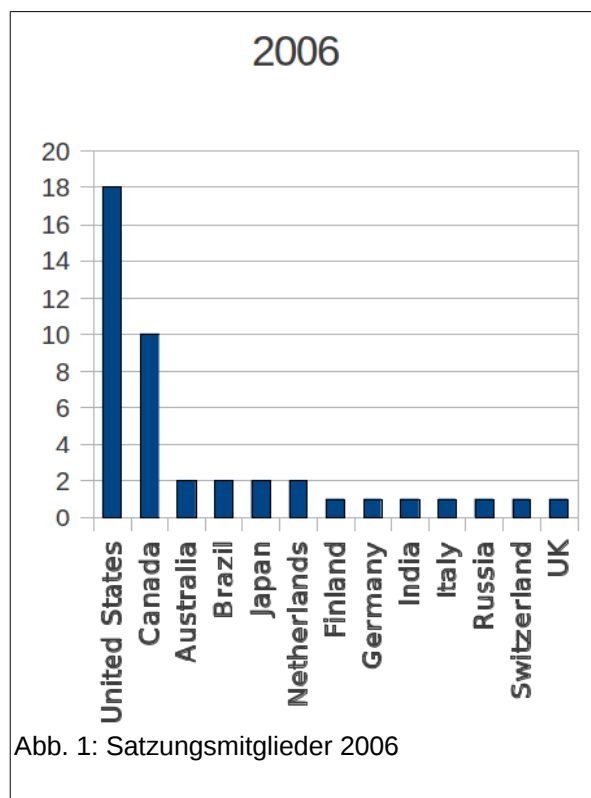
Einführung

Die OSGeo ist eine non-for-profit Organisation (engl. Foundation) die zum Ziel hat, die Entwicklung und Verbreitung von Open Source Software im Geo-Bereich zu fördern. Die Organisation wurde 2006 in Delaware, USA von 26 Aktiven aus der "Szene" gegründet, die als Satzungsmitglieder (engl: Charter Members) die rechtliche Basis der Organisation bilden. Diese ersten Satzungsmitglieder wählten aus ihrer Mitte 9 Direktoren, die den Vorstand (Board of Directors) bilden. Bereits von Anfang an ist die Ausrichtung der OSGeo global, die ersten Direktoren kamen aus 5 Nationen. Die Basis der Organisation bilden mehrere tausend Mitglieder aus aller Welt, die Open Source Geo- und GIS-Software sowohl entwickeln als auch nutzen. Inzwischen gibt es über 40 Organisationen, die OSGeo lokal vertreten, in Deutschland übernimmt der FOSSGIS e.V. die Funktion des deutschsprachigen OSGeo Local Chapters.

Wahl neuer Satzungsmitglieder

Um neue Impulse berücksichtigen zu können und am Puls der aktuellen Technologie zu bleiben, werden jedes Jahr zusätzlich neue Satzungsmitglieder gewählt. Wahlrecht haben alle bestehenden Charter Members. Jeder, auch nicht-Mitglieder, kann Kandidaten nominieren. Nach deren Zustimmung wird jeweils deren Lebenslauf auf einer Seite des OSGeo-Wikis veröffentlicht, so dass sich die bestehenden Satzungsmitglieder über sie informieren können. In den jährlichen Wahlen werden bis zu 20% neue Satzungsmitglieder gewählt [2].

Die Diagramme zeigen, wie sich die globale Verteilung der Mitglieder verändert hat. Im Gründungsjahr 2006 waren 28 der 43 also fast zwei Drittel der Gründungsmitglieder Nordamerikaner (Kanada und USA), insgesamt waren 13 Länder vertreten. Im Jahr 2011 sind bereits doppelt so viele Länder vertreten (26) und der Anteil Nordamerikanern beträgt jetzt weniger als die Hälfte aller Satzungsmitglieder (54 von 125).



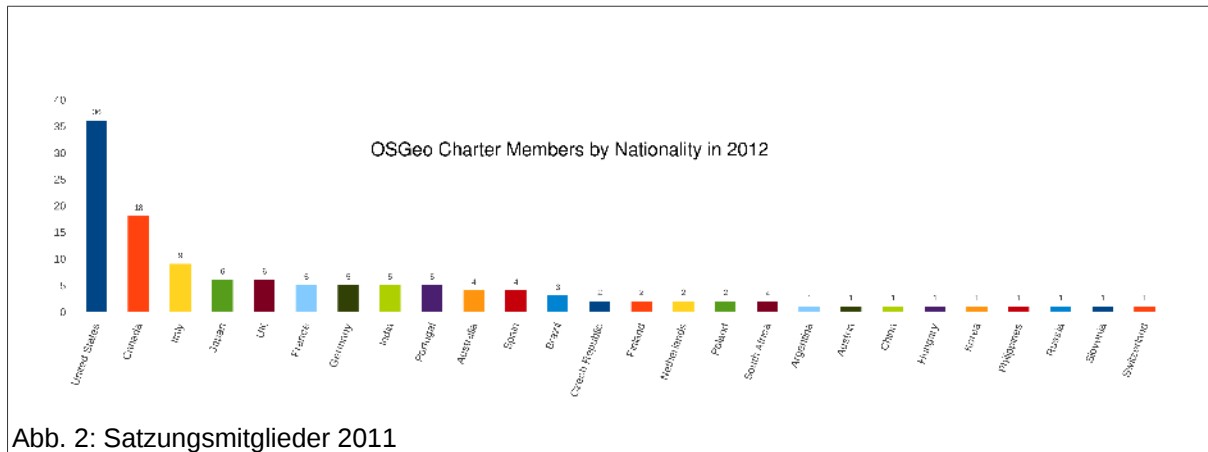


Abb. 2: Satzungsmitglieder 2011

Der Vorstand

Die Satzungsmitglieder der OSGeo wählen den ehrenamtlichen Vorstand [3], der aus neun Personen mit jeweils zweijähriger Amtszeit besteht. Um eine Kontinuität in der Führung der OSGeo zu gewährleisten, werden bei den jährlich stattfindenden Wahlen nur die Hälfte der Vorstandsmitglieder neu gewählt. Der Vorstand wiederum ernennt jedes Jahr den Präsidenten, der vor allem die Vorstandssitzungen leitet und die Organisation nach außen repräsentiert. Der Vorstand erstellt außerdem den Haushalt, trifft Entscheidungen, die nicht durch die Mitglieder der Organisation getroffen werden können und beauftragt Dienstleister mit Arbeiten außerhalb des Kernbereichs der OSGeo wie Design der Webseite, organisatorische Ausrichtung der Konferenzen etc. Der Vorstand verantwortet den Austragungsort der OSGeo-Konferenz FOSS4G und regelt im Allgemeinen das Alltagsgeschäft der OSGeo. Die Arbeit erfolgt meist in kleineren Gruppen, oft direkt mit aktiven Mitgliedern der OSGeo.

Die monatlich stattfindenden Besprechungen erfolgen meist über elektronische Medien und werden öffentlich protokolliert und archiviert. Ein- bis zweimal im Jahr trifft sich der Vorstand zusätzlich persönlich, um komplexere Themen zu besprechen und richtungsweisende Entscheidung zu treffen, die den persönlichen Austausch erfordern. Auch bei diesen Treffen werden oft OSGeo Mitglieder eingeladen um spezielle Fragen ausgiebig besprechen zu können. Sowohl die Agenda als auch das Protokoll dieser Sitzungen sind öffentlich einsehbar. Meist findet eines der Treffen parallel zur FOSS4G, der jährlichen Konferenz der OSGeo statt.

Neue Vorstandsmitglieder

Die letzten Vorstandswahlen erfolgten im August 2011 [4]. Zu den neu gewählten Direktoren zählen Peter Batty, Vorsitzender des letztjährigen FOSS4G Konferenz-Komitees in Denver, USA. Er bringt seine Erfahrungen als ehemaliger CTO (Chief Technology Officer) von Intergraph und Smallworld ein, er arbeitet derzeit bei Ubisense, einer Firma, die ihr Geschäftsmodell nach und nach auf Open Source Technologie umstellt. Jo Cook aus Großbritannien, derzeit die einzige Frau im Vorstand, ist Anwendungs- und Projektentwicklerin in der Firma Astun Technology. Sie ist auch Vorsitzende des UK Local Chapters und maßgeblich am Aufbau der dortigen lokalen Community beteiligt. Mark Lucas, Gründungsmitglied und bereits von 2006 bis 2008 im Direktorium wurde nach drei Jahren Pause wiedergewählt. Er ist Chief Scientist bei Radiant Blue Technologies Inc. und langjähriges Projektmitglied bei dem OSGeo Projekt OSSIM. Michael Gerlek, früher bei Lizardtech ist seit zwei Jahren selbständig mit seiner Firma Flaxen Geo Consulting und Schriftführer der OSGeo. Der Vorstand wird aus den ein weiteres Jahr im Amt verbleibenden Direktoren Arnulf Christl (Präsident), Daniel Morissette (Schatzmeister), Tim Schaub und Frank Warmerdam komplettiert.

Streichung der Stelle des Executive Director

Der Vorstand hat während des Treffens im September 2011 in Denver, USA nach eingehender Beratung beschlossen, die Stelle des Executive Directors (ED) zu streichen. Diese Entscheidung kam für Teile der Community überraschend und sorgte für einiges Aufsehen. Die Gründe für diese Entscheidung liegen jedoch auf der Hand, wenn die Fakten sorgsam abgewogen werden und lassen sich wie folgt zusammenfassen:

- Der ED hatte während der Gründungsphase der OSGeo eine wichtige Rolle, die aktuell in der Form nicht mehr benötigt wird
- Die finanziellen Mittel der OSGeo gingen zu einem großen Anteil in die Bezahlung dieser Stelle, wodurch andere Bereiche zu kurz gekommen sind. Eine Umverteilung des Haushalts soll dies ändern.
- Die Einnahmen der OSGeo durch Sponsoring sind hinter den Erwartungen zurückgeblieben.
- Die Rolle des ED hat keine zusätzlichen Sponsorengelder akquiriert, im Gegenteil, das Spendenaufkommen ist kontinuierlich geschwunden.
- Es wurde zunehmend schwieriger zu vertreten, warum ein einzelner Mitarbeiter der OSGeo den Großteil des Budgets der OSGeo verbraucht, während hunderte von Freiwilligen seit Jahren ebenfalls Arbeiten erbringen die sie nicht vergütet bekommen.
- Die gezielte Beauftragung von professionellen Dienstleistern spezieller Branchen (Medien, Design, Kommunikation, etc.) verspricht effektiver zu sein, als der Versuch alle Aufgaben mit einer einzigen Person zu erfüllen.

Zusätzlich wird die FOSS4G 2012 in China stattfinden und voraussichtlich weniger Einnahmen generieren, was ebenfalls einen verantwortlichen Umgang mit dem bestehenden Budget voraussetzt. Das neue Budget der OSGeo reflektiert die getroffenen Entscheidungen und plant Mehrausgaben in den Bereichen Marketing und Kommunikation und Hardware für die Ausweitung der Projekt-Infrastruktur. Des weiteren wird ein spezielles Budget für sogenannte Botschafter-Aufgaben bereitgestellt, das sind Reisekosten für Mitglieder, die die OSGeo offiziell auf Veranstaltungen, Konferenzen und strategischen Treffen vertreten. Damit wird auch dem globalen Anspruch der OSGeo Rechnung getragen, der mit einem einzigen Mitarbeiter im Nordwesten Kanadas nicht erfüllt wurde.

FOSS4G

Die Free and Open Source Software for Geospatial (FOSS4G [5]) ist die jährliche Konferenz der OSGeo. Ihre Vorläufer gehen zurück bis ins Jahr 2002 zur GRASS User Conference in Trento, Italien, der ersten UMN MapServer Konferenz in Minnesota, USA in 2003 und ins Jahr 2004 zur ersten FOSS4G unter diesem Namen in Bangkok, Thailand. Die FOSS4G zeichnete sich bisher vor allem dadurch aus, dass sie nur einmal im Jahr und an wechselnden Standorten auf der ganzen Welt stattfand. Die erste offizielle Konferenz der OSGeo fand 2006 in Lausanne in der Schweiz statt, die nächste 2007 in Victoria, Kanada, 2008 in Cape Town, Südafrika, 2009 in Sydney, Australien, 2010 in Barcelona, Spanien, 2011 in Denver und 2012 wird sie in Beijing in China stattfinden. Neben der FOSS4G entstanden lokale Konferenzen mit sehr ähnlicher Zielrichtung, oft unter eigenem Namen und meist in der eigenen, lokalen Sprache. Dazu zählen im deutschsprachigen Raum die FOSSGIS, gfoss.it in Italien, Reencontro de Usuarios de MapServer in Brasilien, SIGTE in Spanien oder FOSS4G in Japan und Portugal. Um eine globale Abdeckung zu erreichen, wurde der Austragungsort der offiziellen OSGeo Konferenz FOSS4G jedes Jahr international ausgeschrieben. Dabei sollte eine FOSS4G in Europa, die nächste in Nordamerika und eine weitere in einem bisher noch nicht besuchten Land/Kontinent stattfinden. Der Turnus der internationalen Konferenz, die nur alle drei Jahre in einem Kontinent stattfindet, reicht allerdings bei weitem nicht aus, um den Bedarf an Informationsaustausch und Netzwerken zu decken. Deshalb wurde beschlossen, dass das Konzept FOSS4G ausgeweitet werden und jedes Jahr in mehrfacher Auflage auf der ganzen Welt stattfinden soll. Die weiterhin international ausgeschriebene Hauptkonferenz wird auch in Zukunft das "Meeting of the Tribes" (das Treffen der (Software-(Indianer)-

Stämme) sein. Dort treffen sich die Entwickler und es werden die neuesten Technologien vorgestellt, Informationen ausgetauscht, man trifft sich. Auf der internationalen FOSS4G erfolgt auch die Jahreshauptversammlung der Mitglieder, die Vorstands-Konferenz und die Verleihung des Sol Katz Awards. Es wird sich zeigen müssen, ob die Indianer dem Ruf folgen werden und wieder um die halbe Welt reisen, um ihre Stammesgenossen zu treffen, oder ob sie es provinzieller bevorzugen werden und nur noch die lokale Ausgabe der FOSS4G besuchen.

2011 in Denver

Die FOSS4G in Denver, USA hat wieder alle Rekorde der vorherigen Konferenzen gebrochen, sowohl was Vortrags- und Workshop-Einreichungen, gehaltene Vorträge als auch Anzahl der Teilnehmer betrifft. Zusätzlich zeichnete sich diese Konferenz durch eine Vor-Konferenz für Open Source Neulinge aus, in der die Prinzipien von Open Source und Freie Software Lizenzen beleuchtet wurden, sowie die Projektarbeit in für Laien verständlicher Sprache erläutert wurden.

2012 in Beijing

Die kommende OSGeo Haupt-Konferenz findet in Beijing, China statt und ist eine echte Herausforderung für das OSGeo Konferenz-Komitee, das bisher in fast allen Fällen mit einem durchweg englischsprachigen Local Organizing Committee (LOC) zusammenarbeiten konnte. Dieses Jahr spricht das LOC weitestgehend nur chinesisch, was die Kommunikation ungemein erschwert. Beijing stellt als Austragungsort einige zusätzlich Hürden für Besucher aus dem Ausland auf, angefangen vom Visum bis zur Fortbewegung im Land. Trotzdem bietet Beijing als Austragungsort der FOSS4G 2012 eine Vielzahl von interessanten Facetten, die es sich zu sehen lohnt. Wir erwarten eine weitere FOSS4G der Superlative, wie diese ausfallen, da lassen wir uns überraschen.

Regionale FOSS4G

Die ersten regionalen FOSS4G die offiziell durch die OSGeo unterstützt werden, finden in Nordamerika (FOSS4G NA [6]) und Zentral- und Osteuropa (FOSS4G CEE [7]) statt. Die nordamerikanische Ausgabe ist schon recht zeitnah im April 2012, die FOSS4G CEE im Mai. Alle drei Konferenzen setzen vor allem für die lokale Industrie und Dienstleistungsbranche regionale Marktschwerpunkte.

OGC und OSM meetings

Denver war in 2012 das Geo-Mekka schlechthin. Kurz vor der FOSS4G traf sich die OpenStreetMap Community zur dritten großen "State of the Map" [8], der Konferenz der Mapper. Nach der FOSS4G traf sich die OGC Community eine Stunde nördlich von Denver zu ihrem 78. Technical Committee [9] meeting in Boulder, dem vier Mal im Jahr stattfindenden, einwöchigen Arbeitstreffen der Geo-Standardisierer. Auch wenn es nur wenige geschafft haben alle drei Treffen nacheinander zu besuchen, gab es doch Überlappungen, sowohl von OSM nach OSGeo, als auch von OSGeo nach OGC. Auch hier hat sich die OSGeo bereits als ganz zentrales Instrument der Verständigung zwischen diesen sehr unterschiedlichen Communities etabliert.

Vielfalt von Veranstaltungen

Zusätzlich zur FOSS4G ist die OSGeo auch auf anderen Veranstaltungen präsent. Im Jahr 2011 gab allein der derzeitige Präsident der Organisation Vorträge [10] auf über 30 Veranstaltungen von Hyderabad bis Freiburg. OSGeo stellte auf mehreren Fachmessen aus, im deutschsprachigen Raum unter anderem auf der Intergeo [11] und der AGIT [12]. Im Jahr 2012 ist ein Stand auf dem GEOSummit in der Schweiz geplant

United Nations

Ein Highlight der Arbeit der OSGeo im Jahr 2011 war die Einladung auf den ersten High Level Global Geospatial Information Management (GGIM [13]) Kongress der Vereinten Nationen in Südkorea. Hier wurde OSGeo zusammen mit den Global Players der Geo- und GIS Software eingeladen, um die Mitgliedsstaaten der UN über aktuelle Trends in der Software-Branche zu informieren. Unter den etwa 20 geladenen Vertretern der Industrie waren neben OSGeo, Google, Oracle, Esri, Smallworld, ERDAS, Bentley, Samsung SDS und weiteren anzutreffen.

Local Chapters

Neues lässt sich auch von den Local Chapters berichten, die in 2011 wieder besonders aktiv waren. Es fanden dutzende lokale Veranstaltungen in über 30 Ländern statt, angefangen von kleinen Stammtischen, über Code Sprints, Anwendertreffen bis zu vollständigen Konferenzen wie oben schon beschrieben wurde. Weitere 8 Communities sind im Laufe des Jahres zu offiziellen OSGeo Local Chapters aufgestiegen und repräsentieren das zunehmend multinationale Unternehmen OSGeo in aller Welt. Und dieser Tagungsband ist ein eindrucksvoller Nachweis der Aktivitäten des FOSSGIS e.V., des deutschsprachigen Local Chapter der OSGeo.

Kontakt zum Autor:

Arnulf Christl
metaspatial
Heerstr. 162
0172 2958 004
arnulf.christl@metaspatial.net

Literatur

- [0] <http://www.osgeo.org/>
- [1] http://arnulf.us/publications/bericht_von_der_osgeo_2011_arnulf-christl.odt
- [2] http://www.osgeo.org/charter_members
- [3] http://www.osgeo.org/content/foundation/board_and_officers.html
- [4] http://wiki.osgeo.org/wiki/Election_2011
- [5] <http://wiki.osgeo.org/wiki/FOSS4G>
- [6] http://wiki.osgeo.org/wiki/FOSS4G_NA_2012
- [7] <http://foss4g-cee.org/>
- [8] http://wiki.openstreetmap.org/wiki/State_Of_The_Map_2011
- [9] <http://www.opengeospatial.org/event/1109tc>
- [10] <http://arnulf.us/Events>
- [11] <http://www.intergeo.de/>
- [12] <http://www.intergeo.de/>
- [13] <http://ggim.un.org/>

Fallbeispiel Katzenbuckel – Optimierung der TETRA Funknetzabdeckung als Gruppenentscheidung mit GRASS GIS und KML

Peter Löwe, Michael Hahl

Zusammenfassung

Das vorgestellte Projekt repräsentiert ein best-practice-Beispiel einer gruppengetriebenen kommunalen Raumnutzungsoptimierung durch eine FOSS GIS-gestützte Studie. Zur Optimierung der notwendigen Gebietsabdeckung durch den digitalen Behördenfunk im FFH-Gebiet „Odenwald Eberbach“ und Landschaftsschutzgebiet "Winterhauch-Katzenbuckel" (unter Wahrung der Bedarfe des Landschaftsschutzes) wurden mit GRASS GIS „line of sight“ (LOS) Analysen für mehrere Alternativstandorte und Abdeckungskonfigurationen durchgeführt. Die Ergebnisse wurden in Form von thematischen Karten für Google Earth veröffentlicht, auf deren Basis die Standortentscheidung transparent erfolgte. Dabei konnten der Landschaftsverbrauch durch Einbeziehung bereits bestehender Sendemastinfrastrukturen minimiert, besonders sensible Lebensräume geschützt und Nutzungskonflikte ausgeräumt werden.

Sendemaststandorte als GIS-gestützter kommunalpolitischer Entscheidungsprozess

Der flächendeckende Ausbau des digitalen Funknetzes für Behörden und Organisationen mit Sicherheitsaufgaben (BOS) erfordert die Ausweisung zusätzlicher Sendemaststandorte, um eine möglichst flächendeckende Abdeckung sicherzustellen.

Bei der Auswahl potenzieller Sendemaststandorte kann es zu Interessenkonflikten bezüglich Parametern wie erreichbarer Funkabdeckung, Baukosten, Landschaftserhalt und Bürgerinteressen kommen.

Der kommunalpolitische Auswahl- und Entscheidungsprozess für optimale Sendemaststandorte kann durch GIS-basierte Gebietsabdeckungsanalysen unterstützt werden. Für den Diskussionsprozess mit lokalen Interessengruppen bieten webgestützte Globusbrowser wie Google Earth eine neue Dimension der interaktiven Situationsbeurteilung und Diskussion.

Fallbeispiel Katzenbuckel

Der **Katzenbuckel** gehört zur Gemeinde Waldbrunn im baden-württembergischen Neckar-Odenwald-Kreis, ca. 40 km östlich von Heidelberg. Er ist mit 626 m ü. NN die höchste Erhebung des Odenwaldes und gehört zu dem 3360 ha großen FFH-Gebiet „Odenwald Eberbach und somit zum Schutzgebietsnetz Natura 2000 der Europäischen Union; hierdurch sind gemäß Fauna-Flora-Habitat-Richtlinie sowohl Tier- und Pflanzenwelt als auch deren Lebensräume („Habitats“) geschützt.

Der nachweislich vor über 70 Millionen Jahren ausgebrochene, längst erloschene und durch Erosion herauspräparierte „Vulkanhärtling“ wurde am Südosthang noch bis ins Jahr 1974 als Steinbruch genutzt. Seither konnte sich die Kulturlandschaft mit ihren Wiesen und Buchen-dominierten Mischwäldern und einem renaturierten Steinbruchsee zu einem weit über die Region hinaus bekannten wander- und geotouristischen Anziehungspunkt mit gleichwohl ausgeprägter Artenvielfalt entwickeln; auch wintersportliche Aktivitäten spielen hier traditionell eine Rolle.

Für die Abdeckung des baden-württembergischen Gebiets bis zur hessischen Landesgrenze wurde initial die Errichtung eines 45 m hohen Sendemastes auf dem Katzenbuckel geplant. Die-

Fallbeispiel Katzenbuckel – Optimierung der TETRA Funknetzabdeckung als Gruppenentscheidung mit GRASS GIS und KML

se Planung führte zu Bedenken innerhalb der Bevölkerung, denn es stand zu befürchten, dass ein 45 m hoher Sendemast in unmittelbarer Nachbarschaft zum 18 m hohen historischen Aussichtsturm das Landschaftsbild und den Erlebniswert massiv stören könnte. Aus dieser Motivation veranlasste die Waldbrunner Ortsgruppe des Naturschutzbundes Deutschland (NABU) eine GRASS GIS-gestützte Analyse potenzieller Ersatzstandorte.

Um eine breite öffentliche Diskussion zu ermöglichen, wurden die Resultate als digitale Abdeckungskarten, als KML-Datensätze zur Nutzung mit Globusbrowsern [1] sowie in traditioneller Textform veröffentlicht [2].

Technische Durchführung

Ziel der GIS-Analyse war der Vergleich der Flächenabdeckung verschiedener Kombinationen von Sendemaststandorten und -höhen anhand thematischer Karten und ein Ranking der erreichten absoluten Abdeckungsflächen einzelner Varianten.

Als Ausgangsbasis wurde ein Ausschnitt des ASTER Globalen Geländemodells (ASTER GDEM) [3] gewählt. Die von lokalen Experten identifizierten alternativen Sendemaststandorte und -höhen wurden in GRASS GIS 6.4 als Basis für Sichtweiten-Analysen (r.los Modul) verwendet. Insgesamt wurden anhand von 8 Sendemaststandorten mit jeweils mehreren möglichen Antennenhöhen insgesamt drei Einzelstandortalternativen und 11 Kombinationen von teilweise schon bestehenden Sendemaststandorten als Alternative zum Einzelstandort auf dem Katzenbuckel ausgewertet. Die Ergebniskarten wurden mit dem GRASS-Modul r.out.kml zur Nutzung mit Globusbrowsern wie Google Earth aufbereitet.

Ergebnisse

Die durchgeführte Sichtweitenanalyse konnte drei landschaftsverträgliche Kombinationen von alternativen Sendemaststandorten identifizieren, deren jeweilige Flächenabdeckungen die der ursprünglichen Einzelstandort-Planung übertreffen. Eine dieser Alternativen wurde im Mai 2010 von der Gemeinde Waldbrunn für die Umsetzung der BOS-Abdeckung ausgewählt.

Kontakt zum Autor:

Dr. Peter Löwe
GISIX.com
Friedrich-Ebert-Straße 49
14469 Potsdam
loewe@gisix.com

Ko-Autor zu Geologie u. Naturraum:

Michael Hahl M.A.
Projektbüro proreg
Unterhöllgrund 3
69429 Waldbrunn
m.hahl@proreg.de

Literatur

[1] NABU Waldbrunn; Löwe, Peter: Standortalternativen für den Katzenbuckel-Sendemast – Digitale Karten; http://katzenbuckel.igude.com/nabu_waldbrunn.kmz

[2] NABU Waldbrunn; Löwe, Peter: Standortalternativen für den Katzenbuckel-Sendemast - Kurzfassung; <http://www.katzenpfad.de/wp-content/uploads/2010/05/kurzfassung-studie-pdf.pdf>

[3] ASTER Website: <http://www.gdem.aster.ersdac.or.jp/>