



Anwenderkonferenz für Freie
und Open Source Software für
Geoinformationssysteme,
Open Data und OpenStreetMap

06.-09.Juni



FOSSGIS 2021

ONLINE

RAPPERSWIL



Platinsponsoren:



Goldspensoren:



Silbersponsoren:



Bronzesponsoren:



Medienpartner:





Anwenderkonferenz für Freie
und Open Source Software
für Geoinformationssysteme,
Open Data und OpenStreetMap

Herausgeber:

FOSSGIS e.V.
Bundesallee 23
10717 Berlin

<http://www.fossgis.de>

ISBN 978-3-00-068644-3

Inhaltsverzeichnis

Keynote: Geodata on stereoids: open government geodata und OSS.....	7
Keynote: Open Whatever im Trend und im Clinch.....	8
Die Cloud hat Open Source gefressen.....	9
Warum Open Source-Projekte finanzielle Unterstützung benötigen.....	10
OpenLayers: Was ist neu und was kommt als nächstes.....	13
OpenLayers Feature Frenzy — Tolle Dinge, die OpenLayers kann, von denen aber zu wenige wissen.....	14
InkMap - Frontend-Lösung für Karten-Druck.....	15
Geoprozessierung und Change Detection: Wie kann die GIS Welt von fernerkundlichen Daten profitieren?.....	16
Sen2Europe – Ein einheitliches Bildmosaik Europas.....	18
Regelmäßiges nationales Monitoring der Landbedeckungsänderungen mit Copernicus-Daten.....	21
Maßgeschneiderte QGIS-Arbeitsumgebung mittels Python.....	22
Nominatim für den Hausgebrauch.....	24
10 Jahre Overpass API.....	25
Analyse Tool für Simple Indoor Tagging.....	26
FOSS in der Gewässerentwicklungsplanung – Erfahrungsbericht aus Praxis und Lehre.....	31
Wissenstransfer-Projekt SAPIENS: Satellitendaten für Planung, Industrie, Energiewirtschaft und Naturschutz	32
Konfiguration eines WFS-T mit MapServer und TinyOWS.....	34
Agile Entwicklung, Open Source und amtliche Daten.....	35
Erfassung und Visualisierung der Personenanzahl in Gebäuden unter Verwendung des Internets der Dinge	36
Einsatz von FOSS bei der Bewältigung der COVID-19 Krise in RLP.....	37
Mit digitalen Naturschutzinformationen und OpenStreetMap ein nachhaltiges Verhalten in der Natur ermöglichen.....	49
OpenStreetMap in Nationalparks - Chancen, Folgen und Herausforderungen.....	52
LiDAR langsam: Jetzt erst recht?.....	55
Soziale Medien, dezentrale Netzwerke und OpenStreetMap.....	56
Ist Ground Truth nun unten oder oben? Vergleich von Satellitenbild- mit OpenStreetMap-Daten.....	57
20 Jahre PostGIS - dazu 20 hilfreiche Tipps zu PostGIS und Neuigkeiten rund um das Projekt.....	58
GDI im Container.....	61
MapLibre – Out of the (Map)box. Wenn Open Source plötzlich Closed Source ist.....	64
Massenkonvertieren kartographischer Stildokumente mit GeoStyler-CLI.....	65
Der MapQonverter - Austausch zwischen ArcMap und QGIS.....	66
Erfahrungen mit QEP 179: Über Paketierung, Abhängigkeiten, ungenutztes technisches Potential sowie die Notwendigkeit von fundierten Programmierkenntnissen bei der Benutzung von GUI-Anwendungen.....	67
Vorteile einer Datenhaltung in PostgreSQL/PostGIS, die ein Shapefile nicht bietet.....	68
Skalierbare Geographische Datenverarbeitung mit Open Source.....	69
Aktuelles aus dem deegree-Projekt.....	70
Mapbender 3.2 - besser denn je.....	71
GeoMapFish und QGIS Server.....	73
Ein Ausblick auf das kommende GRASS GIS 8.....	74
Vector-Tiles - Stand der Technik 2021.....	75

swisstopo VectorTiles.....	76
VectorTiles mithilfe von Tilemaker erstellen und nutzen.....	77
FOSS hoch 5 - Praxisbericht eines Serversystems mit OpenSource-GIS.....	78
STAC - ein Ökosystem für raumzeitliche Daten.....	85
Erfahrungen mit Online-Schulungen freier GIS-Software in Zeiten der Pandemie.....	86
Site Reliability Engineering (SRE) in Geo-Diensten.....	87
Webassembly: Wo wir sind und wo es hingehen könnte.....	89
Neues aus dem OGC - Das Jahr 2021? Was lief schon? Was kommt noch?.....	90
Der neue OGC API Features Standard, wie ist der Stand und was ist mit Open Source Geospatial Software möglich.....	91
EASYDAB (Earth System Data Branding) - eine neue Kennzeichnung von qualitativ hochwertigen Geodaten	93
Baremaps: Kontextuelles Mapping in unerforschten Gebieten.....	94
Geo Engine: Explorative Datenanalyse mit raum-zeitlicher Workflowverarbeitung.....	97
Wer hat an der Uhr gedreht? - Zeitbasierte Geodaten modellieren, abfragen und visualisieren.....	99
SensorThings – Von der Sensordaten-Verarbeitung im Backend bis zur Visualisierung im Masterportal.....	100
Nicht nur Kinderwagen-Routing mit OpenStreetMap.....	103
KADAS Routing Plugin.....	104
Transitdaten und OpenStreetMap.....	105
Tivigi: Eine moderne Web-GIS-Komponentenbibliothek mit TypeScript, Vue.js und OpenLayers.....	106
Kommunales Geoportal zur dynamischen Bereitstellung INSPIRE-konformer Dienste - Beispiel einer inter-kommunalen Zusammenarbeit (IKZ).....	113
Docker-basierte GDI in der freien Wildbahn.....	117
Tricks und Kniffe rund um den QGIS-Atlas.....	118
3D-Geoapplikationen mit CesiumJS - Daten, Einsatzmöglichkeiten und Spezifikationen.....	119
swissgeol.ch - Geologische Daten im Web.....	120
3D-Punktwolken und Usability.....	121
Parkplatzzählung und Parkraumanalysen auf OSM-Basis.....	122
Import detaillierter Daten zur barrierefreien Nutzung von Bahnhöfen und Haltestellen.....	126
Quellen- und Lizenzangabe(n) – Vorgaben der Open Database License (ODbL).....	127
MapFish Print: Eine serverseitige Druck-Komponente.....	136
GeoNetwork, nächste Generation.....	137
GeoServer Microservices: der Weg von GeoServer in die Cloud.....	138
GBD WebSuite Demo-Session.....	139
Geodaten auf Smartphones – ein drittes Paradigma nach Desktop- und Web-GIS?.....	140
MrMap – GeoPortal.rlp Reloaded.....	144
SHOGun als Basis flexibler WebGIS-Lösungen — Oder: Weder "off-the-shelf" noch "fully handcrafted".....	154
Geodatenverarbeitung mit Node-RED.....	155
GDAL - Konzept und Werkzeuge.....	163
GIS mit der Kommandozeile.....	164
Effizientes many-to-many ÖPNV-Routing mit gtfsrouter.....	165
Barrierefreie Indoor-Karten.....	166
Aufbau eines Geoportals mit QGIS Server und QWC2.....	175

Neues von osm2pgsql.....	176
Die Nutzung öffentlicher Geodaten in OpenStreetMap am Beispiel Land Brandenburg.....	177
Ist die Zeit reif für permanente Identifikatoren in OpenStreetMap?.....	178
Ein Lastenfahrrad-Routing und -Index: CargoRocket.....	179
Glühstrumpf, Galgen und Bischofsstab - Berliner Gasleuchten in OpenStreetMap.....	180
SmartMapping: Amtliche Daten als VectorTiles.....	181
Interaktive Spaziergänge für die Erkundung von essbaren Pflanzen - Entwicklung einer OpenSource WebApp mit framework7.....	182
Kartographierung von Businessdaten.....	184
FORCE4Q Plugin - Ableitung wolkenfreier Bildkomposite aus frei verfügbaren Landsat und Sentinel-2 Satellitenbildern.....	185
Von Pixeln und Profilen: Die Nutzung von Spektralbibliotheken in QGIS mit der EnMAP-Box.....	186
Fortgeschrittene GIS-Analysen und Kartographie mit Geometrien-Ausdrücken in QGIS.....	187
Webmaps effizient mit Wegue erstellen.....	190
In 60 Minuten zur eigenen Geo-App mit YAGA leaflet-ng2.....	191
Geodaten im Einsatzleitsystem.....	195
Wie Behörden und bundesnahe Betriebe OpenStreetMap nutzen können.....	196
Open Data, Open Source – Open End?.....	197
Statistik der Bauzonenreserven mit QGIS und PostGIS.....	198
Teilautomatische Erkennung von Innenentwicklungspotenzialen mit QGIS und Python auf Basis von amtlichen Geodaten.....	200
Der Weg zur modernen Basiskarte – Potenziale und Herausforderungen von Kartographie und Datengrundlage.....	201

Keynote: Geodata on stereoids: open government geodata und OSS

Basisdaten Landeskarten, Höhenmodell und Luft-/Orthobilder sind ab 1.3.2021 schweizweit frei verfügbar – als Geodienste und Rohdatendownload. Wenn zu Open Data – als Dienst oder als Rohdatensatz – die entsprechenden OSS-Werkzeuge bereitgestellt werden: dann wird das Potential erst freigesetzt.

Es wird auf die Rolle von OSS (Openlayers, Maplibre, VueJS), OpenStandards (Spatial Asset Temporal Catalog, OGC) und das Open Mindset bezüglich Technologien und Partnerschaften (OSM, SW Provider) in der Bereitstellung und Nutzung von amtlichen Geodaten eingegangen.

David Oesch

redaktionelle Anmerkung: die Version dieses Betrags weicht vom gedruckten Tagungsband ab, er wurde nach Drucklegung aktualisiert.

Keynote: Open Whatever im Trend und im Clinch

Freie und Open Source Software (FOSS) und Open Data sind im Trend - aber auch im Clinch und zwar teilweise auf gleiche und teilweise auf unterschiedliche Weise. Und viele der Herausforderungen sind Missverständnisse und nur zum Teil branchenspezifisch. Dies sind einige Gedanken und Erfahrungen aus der "GISTech"-Branche und zwar zu FOSS und zu Open Data und dort jeweils aus Business- und aus Community-Sicht mit potentiell nachhaltigen und nicht-nachhaltigen FOSS- und Open Data-Projekten.

Zuerst FOSS aus Business-Sicht: Hier gilt das Sprichwort "open source isn't a business model - it is a go-to-market strategy". Das klingt nach Rechtfertigung - und ist es meist auch. Was aber vielleicht zusätzlich nötig ist, das sind Ergänzungen ethischer Lizenzen (Software und Daten). Dann FOSS aus Community-Sicht: Da gibt es einiges zu erzählen an Beispielen wie Mapbox, OGR, QGIS etc. Bei Open Data-Projekten aus Business-Sicht ist klar, dass Open Data allein kein "Business" sein kann - etwas was noch näher erläutert wird. Schliesslich Open Data-Projekte aus Community-Sicht: Da gibt es einiges zu erzählen mit Blick auf OpenStreetMap!

Stefan Keller

Die Cloud hat Open Source gefressen

Das Open Source Core Lizenzmodell scheitert an der Cloud. Was nun? Hintergründe zu Open Source Lizenz- und Geschäftsmodellen.

Letztes Beispiel eines gescheiterten Open Source Core Lizenzmodells in der Geowelt (zum Zeitpunkt der Einreichung) ist der Rückzug von MapBox. Gerade grundlegende Open Source Projekte haben es schwer, ein Geschäftsmodell jenseits der proprietären Lizenz zu finden. Woran liegt das und was können wir tun?

Arnulf Christl



Warum Open Source-Projekte finanzielle Unterstützung benötigen

ASTRID EMDE (OSGEO UND WHEREGROUP GMBH BONN)

Alle lieben Open Source Software und nutzen diese gerne. Open Source Software und Open Data sind aus unserem Alltag nicht mehr wegzudenken. Wir nutzen diese beruflich und auch im privaten Umfeld. Manche Unternehmen bauen ihr Geschäftsmodell auf durch Open Source auf.

Open Source Software kann leicht installiert und ausprobiert werden. Meist liegt eine gute Dokumentation vor und über Foren oder Mailinglisten können Fragen geklärt werden. Toll ist natürlich noch dazu, dass die Software nichts kostet.

Wir sollten uns aber immer darüber im Klaren sein, dass die meisten Softwareprojekte keine Freizeitprojekte sind, sondern Entwickler:innen und/oder Firmen in irgendeiner Weise von der Software leben.

Open Source Software ist fest in die eigene Architektur integriert und aus dieser nicht mehr wegzudenken. Projekte wie QGIS, PostgreSQL, PostGIS, GDAL, MapServer und viele weitere bauen die eigenen GDI auf, stellen die Geodaten zuverlässig bereit und ermöglichen deren Fortführung und bieten flexible Schnittstellen. Durch die Nutzung von freier Software konnten an vielen Stellen beachtliche Einsparungen erfolgen.

Zum Fortbestehen benötigt Freie Software eine finanzielle Unterstützung. Jede/r Einzelne sollte hier über die eigenen Möglichkeiten oder die Möglichkeiten des Unternehmens / der Behörde nachdenken.

Oft sind dabei die Nutzer:innen der Software nicht die Personen, die über das Budget entscheiden. Daher ist es wichtig, dass von Seiten der Nutzer:innen die Notwendigkeit der finanziellen Unterstützung an die Budgetverantwortlichen weitergegeben wird.

Die Nutzer:innen haben ein großes Interesse am Fortbestehen und der Weiterentwicklung der Software. Durch ein Sponsoring oder einen Wartungsvertrag kann eine finanzielle Unterstützung eines Projektes erfolgen.

GDAL Fundraising Initiative 2021

Das GDAL-Projekt ist ein sehr gutes und aktuelles Beispiel. GDAL ist eine zentrale Bibliothek, für die Handhabung von Vektor- und Rasterdaten, die standalone genutzt werden kann und darüber hinaus von zahlreichen Projekten verwendet wird. Die Liste der Software (freie und proprietäre Software), die GDAL nutzt, ist beeindruckend. In den letzten Jahren kommt GDAL zudem in unzähligen Installationen in der Cloud durch Microsoft, Google, Panet zum Einsatz.

Eigentlich ein sehr erfolgreiches – sogar systemrelevantes - Projekt. Doch die Finanzierung der Entwicklung und Unterhaltung war bis vor Kurzem nicht abgedeckt.

Erst durch die Fundraising-Initiative konnten Sponsoren gefunden werden und werden weiterhin gesucht.

https://gdal.org/software_using_gdal.html#software-using-gdal

<https://lists.osgeo.org/pipermail/gdal-dev/2021-April/053886.html>

Hören Sie auch gerne den MapScaping Podcast mit Paul Ramsey zu diesem Thema
<http://blog.cleverelephant.ca/2021/05/gdal-podcast.html>

Warum Open Source-Projekte finanzielle Unterstützung benötigen

Wie kann eine finanzielle Förderung erfolgen?

Die wenigsten Projekte verfügen über eine Sponsoringabteilung. So sind die Förderungsmöglichkeiten nicht immer offensichtlich. Es folgen ein paar Hinweise darauf, wie eine Förderung erfolgen kann.

Projekte direkt fördern

Manche Projekte können direkt gefördert werden wie z.B. GDAL oder QGIS siehe <https://github.com/sponsors/OSGeo>.

Förderung durch eine Vereinsmitgliedschaft

Eine finanzielle Förderung kann auch durch eine Vereinsmitgliedschaft z. Bsp. im FOSSGIS e.V. oder QGIS.DE Verein erfolgen.

<https://www.fossgis.de/verein/mitgliedschaft/>

Die Mitgliedschaft im QGIS.DE Verein für eine natürliche Person beträgt 50 €, für eine Firma 250 €. Der QGIS.DE verfügt über mehr als 70 Mitglieder und hat das Ziel, das QGIS-Projekt zu fördern, so dass viele der Gelder an das QGIS-Projekt fließen. Derzeit fördert der QGIS.DE das Projekt als großes Fördermitglied in einer Höhe von 9.000 €.

- <https://www.qgis.de/>

Förderung durch Sponsoring der OSGeo

Die OSGeo ist der Dachverband für viele Open-Source-Geospatial-Projekte. Die Unterstützung von freier Software kann daher durch ein Sponsoring der OSGeo erfolgen. Ein solches Sponsoring ist noch dazu werbewirksam, da das eigene Logo auf der Webseite aufgeführt wird.

- <https://www.osgeo.org/sponsors/>
- <https://github.com/sponsors/OSGeo>

Förderung durch Sponsoring von Veranstaltungen

Durch das Sponsoring von Konferenzen wie der FOSSGIS- oder der FOSS4G-Konferenz kann über Umwege eine Förderung von Software-Projekten erfolgen. Auch hier erfolgt durch die Präsenz auf der Konferenz-Webseite oder vor Ort Werbung für das eigene Unternehmen. Die Konferenzförderung kann zudem auch als OSGeo-Sponsoring angerechnet werden.

- <https://www.fossgis-konferenz.de>
- <https://2021.foss4g.org/>

Förderung von Einzelpersonen

Mehr und mehr können Projekte aber auch Einzelpersonen direkt über GitHub mit ein paar Klicks unterstützt werden.

Even Rouault GDAL-Entwickler <https://github.com/sponsors/rouault>

- OSGeo <https://github.com/sponsors/OSGeo>

Einzelpersonen können beispielsweise auch über Patreon gefördert werden.

- Regine Obe <https://www.patreon.com/reginaobe>

Warum Open Source-Projekte finanzielle Unterstützung benötigen

Zuwendungen an Organisationen

Organisationen wie Opencollective oder NumFocus dienen als Schnittstelle zu Projekten. NumFOCUS fördert beispielsweise viele Python-Projekte wie z.B. Pandas, Matplotlib, NumPy oder auch das Projekt Jupyter Notebooks. NumFOCUS dient dabei wie die OSGeo als steuerlicher Ansprechpartner (Fiscal Host).

<https://numfocus.org/>

<https://opencollective.com/>

Zusammenfassung

Mit diesem Beitrag sollte deutlich werden, dass es viele Wege gibt, um Open Source Software finanziell zu unterstützen. Dabei muss es sich auch nicht zwangsläufig um hohe Beträge handeln. Gehen Sie proaktiv vor und fragen Sie ggf. auch Ihren Dienstleister nach weiteren Möglichkeiten. Die Arbeit der Entwickler:innen kann durch Ihren Beitrag honoriert und das Fortbestehen der Software sichergestellt werden.

Darüber hinaus können Projekte natürlich auch auf andere Weise als durch die finanzielle Förderung unterstützt werden, beispielsweise durch Aktivität auf der Mailingliste, durch Bugmeldungen, Mithilfe bei der Dokumentation oder Werbung für und Verbreitung der Projekte.

Kontakt zur Autorin:

Astrid Emde
OSGeo - Open Source Geospatial Foundation
astrid_emde@osgeo.org

WhereGroup
Eifelstraße 7 53119 Bonn
+49 228 909038 22
astrid.emde@wheregroup.com

OpenLayers: Was ist neu und was kommt als nächstes

OpenLayers 6 profitiert von einer flexiblen Renderer-Architektur, einer Leistungssteigerung beim Rendering von Vektor-Kacheln und einem WebGL-Punkt-Renderer für Big-Data-Visualisierungszwecke. Seit der Veröffentlichung wurden zahlreiche Beiträge zur Bibliothek geleistet. Wir führen Sie durch die neuen Funktionen und zeigen anhand von Beispielen aus der Praxis, wie Sie von diesen profitieren können.

Am Ende des Vortrags werden wir heiße Themen für die zukünftige Entwicklung besprechen.

OpenLayers 6, das im September 2019 veröffentlicht wurde, profitiert von einer flexiblen Renderer-Architektur, einer Leistungssteigerung beim Rendering von Vektorkacheln und einem WebGL-Punkt-Renderer, der es ermöglicht, die Bibliothek für die Visualisierung von Big Data zu nutzen. Seit der Veröffentlichung wurden mehrere Beiträge zur Bibliothek geleistet, wie z. B.:

- WFS 2-Unterstützung
- WebGL-Punkt-Styling und Ausdrucks-Parsing
- Optimierte Treffererkennung-API
- Vektorielle Kachelebenen haben jetzt einen Vektor-Rendermodus
- Web workers
- Web-Komponenten

Wir führen Sie durch die neuen Funktionen und zeigen anhand von Beispielen aus der Praxis, wie Sie davon profitieren können.

Am Ende des Vortrags werden wir heiße Themen für die zukünftige Entwicklung besprechen.

Emmanuel Belo

OpenLayers Feature Frenzy — Tolle Dinge, die OpenLayers kann, von denen aber zu wenige wissen

OpenLayers ist eine sehr bekannte und ausgesprochen weit verbreitete OpenSource JavaScript Bibliothek, laut Webseite ist OpenLayers "A high-performance, feature-packed library for all your mapping needs". Der Vortrag möchte die Kernelemente dieser Aussage - also hohe Performanz, viele Features und geeignet für alle Webkartenanforderungen - anschaulich mit reichlich Beispielen, viel Code und ein wenig Theorie belegen.

OpenLayers ist eine sehr bekannte und ausgesprochen weit verbreitete OpenSource JavaScript Bibliothek, die in einer Vielzahl von Webanwendungen und auch in anderen OSGeo Projekten zum Einsatz kommt: Meist dann, wenn dynamische Karten dargestellt werden sollen, die eine Vielzahl von Geodaten in den unterschiedlichsten Formaten integrieren und mit denen ein Anwender dann wie gewünscht interagieren kann. Laut Webseite ist OpenLayers „A high-performance, feature-packed library for all your mapping needs“.

Der Vortrag möchte die Kernelemente dieser Aussage - also hohe Performanz, viele Features und geeignet für alle Webkartenanforderungen - anschaulich mit Beispielen, Code und ein wenig Theorie belegen. Wir werden einfache und auch komplexere Fragestellungen angehen und gemeinsam schauen, was im gut sortierten Werkzeugkasten von OpenLayers an Hilfsmitteln bereitgestellt wird, um ansprechende und funktionale Lösungen zu entwickeln. Wie kann man beispielsweise das Beste aus Mapbox Vector Tiles herausholen? Ist es wahr, das mit OpenLayers auch Rasteroperationen sowie Geo-Analysen mit Vektor- und Rasterformaten durchführbar sind? Wie kann man die neuesten OGC API Spezifikationen nutzbar machen? Diesen und weiteren Fragen wollen wir nachgehen.

Natürlich werden wir auch etwaige Neuerungen der bereits seit sehr vielen Jahren existierenden und sich konstant weiterentwickelnden Bibliothek vorstellen. Mit ein wenig Glück können wir einen unverbindlichen Ausblick auf die nächste Major-Version 7 werfen.

Marc Jansen

InkMap - Frontend-Lösung für Karten-Druck

Das Drucken von Web-Karten ist eine Herausforderung bezüglich Performanz, Genauigkeit und Flexibilität. Basierend auf Open-Source-Technologien wurde eine neue, ebenfalls quelloffene, Javascript-Library für hochauflösendes Front-End-seitiges Drucken entwickelt, die massgeschneiderte Anpassungen einfach macht und die Performanz des Browsers nicht einschränkt.

Backend-Lösungen für das Drucken von Karten sind zwar stabil und gut skalierbar aber, durch vordefinierte Vorlagen, unflexibel und auch das Styling ist eingeschränkt. Front-End-Lösungen hingegen sind einfach zu installieren und bieten eine bessere Übereinstimmung zwischen der Webkarte und dem Druckprodukt, aber sie kommen meist mit grösseren Datenvolumen nicht zurecht und haben eine schlechte Performanz. InkMap soll hier, durch die Nutzung moderner Webtechnologien, eine Verbesserung bringen.

InkMap benutzt OpenLayers für die Darstellung der Karte. Das Druckerzeugnis wird daher genauso aussehen, wie auf der Webkarte und es können die gleichen Styling-Möglichkeiten ausgeschöpft werden. Dem Print-Prozess wird eine JSON-Konfiguration mit Angabe von Layer, Auflösung, Layout, usw. übergeben. Damit wird eine Karte im PNG-Format generiert, die auch in ein PDF eingefügt werden kann.

Der Druckprozess wird an einen asynchron laufenden Prozess, einen Service-Worker delegiert, so dass die Webseite weiter zur Verfügung steht. Damit dieser den Canvas ausserhalb der Webseite darstellen kann, wird die OffscreenCanvas-Technologie benutzt. Für Browser, die diese noch nicht unterstützen gibt es eine Alternativlösung. Durch die Kombination dieser beiden Technologien wird eine ausgezeichnete Performanz möglich. Es können mehrere Druckaufträge gleichzeitig gestartet werden und es ist auch möglich deren jeweiligen Fortschritt anzuzeigen.

Die Präsentation gibt einen Überblick über die angewendeten Technologien und zeigt, wie die Software integriert und der Druck konfiguriert werden kann.

Renata Müller

Geoprozessierung und Change Detection: Wie kann die GIS Welt von fernerkundlichen Daten profitieren?

Im Vortrag zeige ich anhand von konkreten Anwendungsbeispielen aus der Praxis auf, welche Fragestellungen man konkret mit Daten aus der Fernerkundung beantworten kann. Im Fokus steht dabei nicht die Theorie, sondern die ganz konkreten Fragestellungen. Die Beispiele kommen aus dem Bereich Detektion von Oberflächenbeschaffenheiten, Landbedeckung und Change Detection sowie der Hochwassergefahrenkartierung.

Der Satz „your Software is useless – without data“ (eure Software ist nutzlos – ohne Daten), den ein gewisser Schuyler Erle in einem Lightning Talk vor vielen Jahren auf einer FOSS4G-Konferenz in Richtung der Entwickler von GIS-Software abfeuerte, wurde mit Sicherheit schon oft zitiert, was ihn aber nicht weniger wahr macht. Insgesamt ist die Datenlage mit Aufkommen von OpenStreetMap, aber auch mit zunehmender Freistellung von öffentlichen Geodaten seit diesem Ausspruch wesentlich besser geworden. Eine weitere Geodatenquelle, die insbesondere in den letzten Jahren eine große Dynamik entwickelt hat, kommt aus der Fernerkundung.

Mit weiterer Verfügbarkeit von Fernerkundungsdaten, haben sich auch die Disziplin der Fernerkundung und die GIS-Welt immer weiter angenähert. Dies war auch auf der FOSSGIS Konferenz spürbar, so wurde die Fernerkundung auf den vergangenen FOSSGIS-Konferenzen, speziell in Zusammenhang vom europäischen Copernicus-Programm und den damit oft im Zusammenhang gemeinten Sentinel Daten immer wieder thematisiert. Besucher der FOSSGIS haben gelernt, wie das Copernicus Programm aufgebaut ist und welche Arten von Daten es gibt, ebenso wurden Systeme vorgestellt mit denen sich diese Daten verarbeiten lassen. Um aber einen wirklichen Überblick darüber zu bekommen, welche Möglichkeiten und Grenzen Open Source GI-Software in Kombination mit Daten und Methoden der Fernerkundung bieten und welche Lösungen sich daraus entwickeln lassen, dazu fehlte es meist an konkreten Beispielen. Der Talk will diese Lücke schließen und stellt einige Beispiele aus der Praxis vor.

Beispiel 1: Detektion von Oberflächenbeschaffenheiten aus Orthophotos.

Im konkreten Fallbeispiel geht es darum, potentiell mögliche (Erd-)Kabeltrassen über eine Kostenoberfläche automatisch zu berechnen. Die Basis dafür, ist ein möglichst hoch aufgelöster Geodaten-satz, der Aufschluss über die Oberflächenbeschaffenheit, also Asphalt, Grünland, Wald (u.v.m.) gibt. Für die Oberflächendetektion spielt der sogenannte NDVI (normierter differenzierter Vegetationsindex) eine entscheidende Rolle, wenn es darum geht, Vegetation zu erkennen. Dieser lässt sich einfach mit GRASS GIS bestimmen und so entsprechende Flächen abgrenzen. Was aber tun, wenn die Luftbilder im Winter aufgenommen wurden? Hier kann die Kombination mit Sentinel-2 Daten aus dem Copernicus Programm helfen, auch wenn deren Auflösung wesentlich geringer ist, als die der Luftbilder. Natürlich können mit dem durch dieses Verfahren bereitgestellten Geodatensatz der Oberflächen auch andere Fragestellungen bearbeitet werden.

Beispiel 2: Landbedeckungsklassifizierung basierend auf Sentinel-2-Daten

Im Rahmen des Projekts „Inwertsetzung von Copernicus-Daten für die Raumbbeobachtung“ (incora) wurde auf Basis von GRASS GIS und actinia ein Verfahren zur Erstellung von jährlichen Landbedeckungsklassifikationen für ganz Deutschland entwickelt. Die Ergebnisdaten haben eine räumliche Auflösung von 10m und repräsentieren die Klassen bebautes Gebiet, Grünland, Wald, Landwirtschaft, vegetationsloser Boden und Wasser. Das Verfahren basiert auf Zeitreihen von Sentinel-2-Daten aus dem Copernicus-Programm. Die für die Klassifikation benötigten Trainingsdaten werden regelbasiert aus Referenzdaten (z.B. aus OpenStreetMap) und den Sentinel-2-Szenen selbst erstellt. So kann die Verarbeitung flexibel auf verschiedene Jahre übertragen und vollautomatisch durchgeführt werden. Berechnet man je eine Klassifikation für verschiedene Jahre, so lassen sich damit natürlich Veränderungen in der Landbedeckung feststellen. Hier zeigen sich insbesondere Veränderungen der Klasse „be-

Geoprozessierung und Change Detection: Wie kann die GIS Welt von fernerkundlichen Daten profitieren?

bautes Gebiet“, sodass sich Indikatoren zur Raumnutzung auf lokaler, regionaler und nationaler Ebene ableiten lassen. Die Klassifikationskarten sowie Ergebnisse der Veränderungsanalyse für die Jahre 2019 und 2016 stehen unter der Data licence Germany zum Download zur Verfügung.

Beispiel 3: Hochwassergefahrenkartierung mit Sentinel-1-Daten

Im Rahmen eines weiteren Projektes werden Prozessierketten in GRASS GIS erstellt, um unter Verwendung von Sentinel-1 Radardaten Hochwassergefahrenkarten am Beispiel Ecuadors zu erstellen. Zunächst werden für jede verfügbare Sentinel-1 Szene aus einem Zeitraum (bspw. ein Jahr) mit Hilfe eines automatischen Schwellenwertverfahrens Wasserflächen extrahiert und in GRASS GIS zu einem Data-Cube zusammengefasst. Daraus lassen sich Statistiken wie Hochwasserhäufigkeit und (gemeinsam mit Referenzdaten wie z.B. einem digitalen Geländemodell) mittlere Überflutungstiefe bestimmen. Die Unterscheidung zwischen permanenten Wasserflächen und Hochwassergebieten kann entweder automatisch über die Häufigkeit oder mit Hilfe eines Referenzdatensatzes erfolgen. Die Ergebniskarten dienen als wichtige Grundlage zum Informationsaustausch innerhalb des Sendai-Rahmenwerks zur Katastrophenvorsorge.

Till Adams, Guido Riembauer

Sen2Europe – Ein einheitliches Bildmosaik Europas

SANDRA BEYER, GREGOR TROSSET UND THOMAS WIATR

Sämtliche Orte unseres Planeten werden regelmäßig durch wiederholende Bildaufnahmen von Erdbeobachtungssatelliten abgebildet. Sie verschaffen dadurch, einen noch nie zuvor dagewesenen Einblick in die Dynamik von Umweltveränderung oder von endogenen Prozessen auf der Erde.

Zum Beispiel werden Dürreerscheinungen und Vegetationszustände erkennbar. Es können Prozesse an Küsten wie deren morphologische Dynamik erfasst werden. Auch können die Eisbilanzen in Gletscherregionen über die Jahre ermittelt werden.

Bevor jedoch die Dynamik und Veränderung durch Experten analysiert und interpretiert werden kann, ist eine essentielle Aufgabe der Fernerkundung, diese Erdbeobachtungsdaten in lesbare Informationen zu transformieren. Hierzu werden im Vorfeld einzelne aufeinanderfolgende Verarbeitungsschritte durchgeführt, sodass die Satellitenbildaufnahmen eine hilfreiche und unterstützende Datenquelle darstellen. Für kleinere Gebiete lassen sich diese Prozesse noch leicht realisieren. Der Schwierigkeitsgrad steigert sich jedoch mit der zu betrachtenden Fläche und dem dargestellten Detailgrad der Bilddaten. Je größer die Fläche und höher die räumliche Bodenauflösung ist, desto mehr Bilder von unterschiedlichen Zeitpunkten werden derzeit noch für eine flächendeckende Abdeckung benötigt.

Im Anbetracht des Copernicus-Programms und der damit verbundenen hohen temporalen Auflösung (ca. 5 Tagen für Deutschland) der Sentinel-2 Satelliten, sowie der Open Data Strategie von den Satellitenbilddaten, hat das BKG einen robusten und schnellen Prozess entwickelt der es erlaubt, Erdbeobachtungsdaten von großen Flächen und von hoch aufgelösten Bildern, schnell in ein scheinbar wolkenfreies Gesamtbild zu erstellen (siehe Abbildung 1).

Deutschlandweite Mosaike auf Basis von Sentinel-2-Daten werden seit Jahren durch das BKG jährlich frei zur Verfügung gestellt (2018, 2019, 2020). Durch die hohe Nutzung der Deutschlandmosaike (über 1800 verschiedene Nutzer mit mehr als 15 Millionen Zugriffen in 2020) inkl. den entsprechenden Nutzeranforderungen und aufbauend auf dieser Expertise des BKG, erstellte das BKG 2020, erstmals ein hoch qualitatives gesamteuropäisches und einheitliches Mosaik (Datengrundlagen überwiegend 2018). Dieses steht als interaktive Webkarte (WMS) mit einer Echtfarbendarstellung (R, G, B) jedem Nutzer frei zur Verfügung und wird alle drei Jahre aktualisiert werden (siehe Abbildung 2). Für dieses Produkt wurden Sentinel-2 Daten (1338 einzelne Bildkacheln mit einem Gesamtvolumen von rund 0,9 Terabyte) mit 10m Bodenauflösung aus dem europäischen Erdbeobachtungsprogramm Copernicus verwendet. Das BKG ist dadurch in der Lage, großräumige qualitativ hochwertige Bildverarbeitungsprodukte auf nationaler und internationaler Ebene bereitzustellen. Dadurch leistet das BKG unter anderem einen wichtigen Beitrag zu qualitätsgeprüften Open-Data-Produkten, zur Informationsfreiheit, zur europäischen Zusammenarbeit und der Digitalisierung von Verwaltungsaufgaben.

Anwendungen für das europäische Mosaik ergeben sich durch die Verschneidungen mit weiteren europäischen Datensätzen, sodass grenzübergreifende Veränderungsprozesse erfassbar werden. Diese Veränderungsprozesse der Umwelt, Gesellschaft und Wirtschaft können mithilfe dieses Prozesses und der daraus abgeleiteten Produkte in großen Gebieten dokumentiert werden.

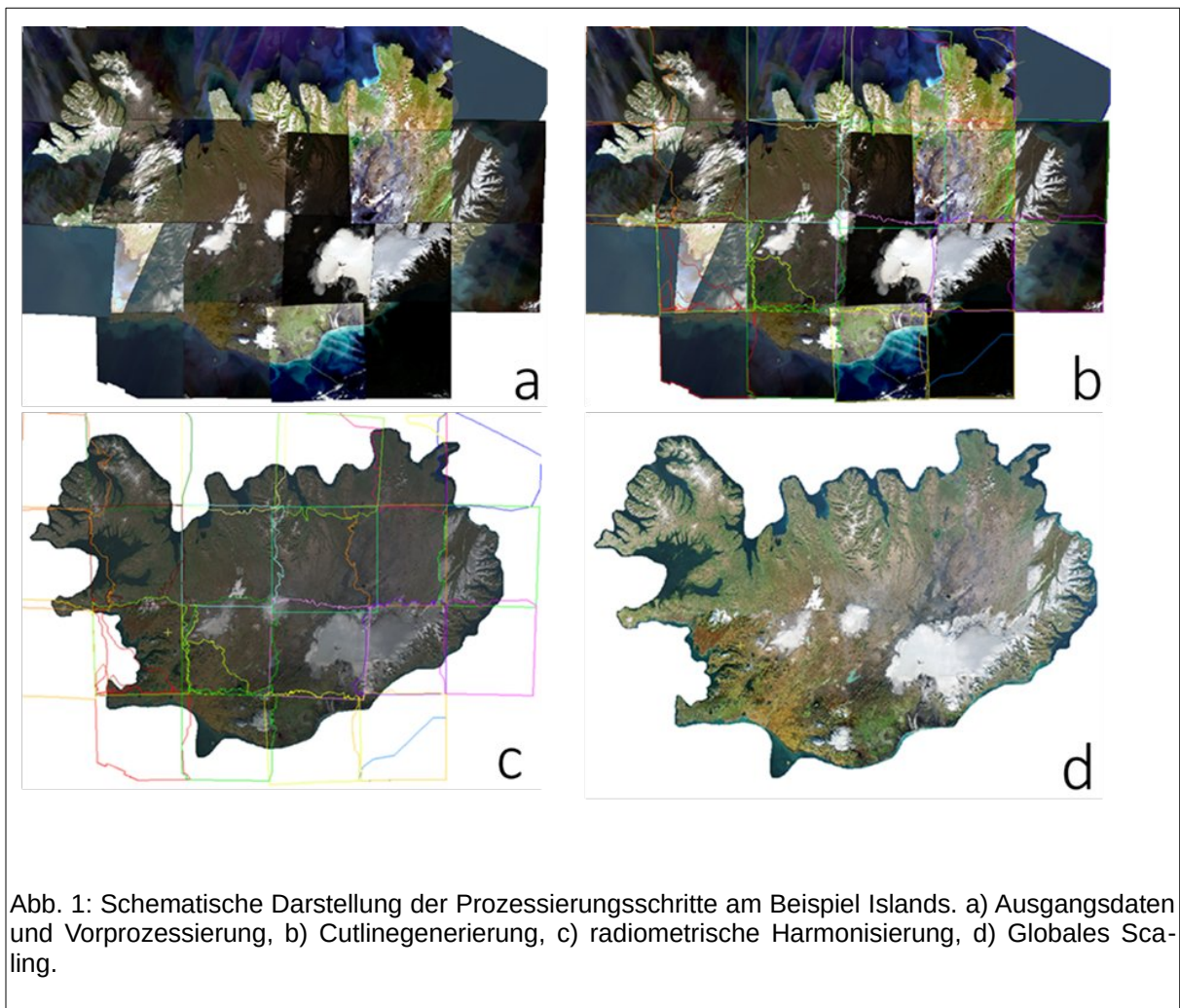


Abb. 1: Schematische Darstellung der Prozessierungsschritte am Beispiel Islands. a) Ausgangsdaten und Vorprozessierung, b) Cutlinegenerierung, c) radiometrische Harmonisierung, d) Globales Scaling.

Weiter Anwendungsbeispiele für hochwertige Satellitenbildverarbeitung sind unter anderem:

- Visuelle Kartierung von Veränderungen für bestimmte Zeitintervalle (Monitoring der Landbedeckung und Veränderungsanalysen)
- Datengrundlage für Flugsimulatoren in Verbindung mit 3D-Daten
- Grundlagendatensatz in GIS Projekten als Kartierungsgrundlage (Grenzharmoisierung)
- Segmentierung und geometrische Analyse von Landbedeckung
- Grenzübergreifende Referenzgrundlage für eine geometrische Korrektur sowie für die Georeferenzierung anderer Daten

Sen2Europe – Ein einheitliches Bildmosaik Europas



Abb. 2: Darstellung des Gesamtmosaiks Sen2Europe.

Kontakt zum Autor:

Sandra Beyer
Bundesamt für Kartographie und Geodäsie
Richard-Strauss-Allee 11
069 6333 395
Sandra.Beyer@bkg.bund.de

Gregor Trosset
Bundesamt für Kartographie und Geodäsie
Richard-Strauss-Allee 11
069 6333 2105
Gregor.Trosset@bkg.bund.de

Thomas Wiatr
Bundesamt für Kartographie und Geodäsie
Richard-Strauss-Allee 11
069 6333 358
Thomas.Wiatr@bkg.bund.de

Regelmäßiges nationales Monitoring der Landbedeckungsänderungen mit Copernicus-Daten

Das Projekt LandschaftsVeränderungsDienst (LaVerDi) wurde im Frühjahr 2017 initiiert und hatte die automatische Ableitung von Veränderungen mittels Copernicus Daten zur Unterstützung der Aktualisierung und Validierung BKG interner Produkte zum Ziel. Die detektierten Veränderungen sollen mit einer Genauigkeit von $\geq 80\%$ abgeleitet werden und als Aktualisierungshinweise für das Landbedeckungsmodell Deutschland fungieren. Der Dienst wurde auf der nationalen Copernicus Plattform CODE-DE implementiert.

Der „LandschaftsVeränderungsDienst“ (LaVerDi) ist eine kostenfreie Onlineanwendung zur Bereitstellung und Analyse von Veränderungshinweisen der Landbedeckungen Deutschlands. Hierzu werden Zeitreihen aus kostenfreien Sentinel2- Satellitendaten des Copernicus Programms verwendet, um automatisiert aktuelle und konsistente Hinweise zu Änderungen der Landbedeckung Deutschlands abzuleiten und ein regelmäßiges Monitoring zu gewährleisten. Es werden abgeleitete vegetationsbeschreibende Metriken, wie beispielsweise das Minimum oder der Mittelwert des NDVI (Normierter Differenzierter VegetationsIndex), der gleichen Periode zweier aufeinanderfolgender Referenzjahre (t_0 , t_1) miteinander verglichen. Die daraus resultierenden und auf 10m-Pixeln basierenden Veränderungen werden anschließend einer Filterung und Plausibilitätsprüfung unterzogen, um den finalen Änderungslayer zu generieren. Der endgültige Veränderungsdatensatz besitzt eine Genauigkeit der erkannten Veränderungen von über 80% und steht als Vektordatensatz inklusive der relevanten Veränderungsattribute als Shapefile zum Download bereit. Die Referenz für die Änderungshinweise ist das aktuelle digitale Landbedeckungsmodell (LBM-DE) des BKG. Der LaVerDi-Viewer ermöglicht es, vektorbasiert Veränderungshinweise interaktiv zu analysieren. Dabei stehen dem Nutzer eine Reihe von Onlineanalyse Werkzeugen zur Verfügung. Umgesetzt wurde die LaVerDi Webanwendung auf der nationalen Copernicus Plattform CODE-DE (Copernicus Data and Exploitation Platform – Deutschland).

Patrick Knöfel

Maßgeschneiderte QGIS-Arbeitsumgebung mittels Python

QGIS kommt schon von Hause aus mit einem beachtlichen Funktionsumfang daher. Trotzdem gibt es noch verschiedenste Stellen, an denen man mittels Python die Funktionalitäten von QGIS erweitern oder an eigene Bedürfnisse und Workflows anpassen kann.

Im Rahmen einer Demosession würde ich gerne die verschiedenen Stellen aufzeigen, an denen man mittels Python QGIS anpassen oder erweitern kann.

QGIS kommt schon von Hause aus mit einem beachtlichen Funktionsumfang daher. Trotzdem gibt es noch verschiedenste Stellen, an denen man mittels Python die Funktionalitäten von QGIS erweitern oder an eigene Bedürfnisse und Workflows anpassen kann.

Im Rahmen einer Demosession würde ich gerne die verschiedenen Stellen aufzeigen, an denen man mittels Python QGIS anpassen oder erweitern kann:

Ad-Hoc-Ausführen von Python Code (Python-Console, Script-Editor und ggf. Scriptrunner):

Ein unkomplizierter und schneller Weg um Python in QGIS auszuführen. In der Regel der Einstiegs-punkt für PyQGIS-Newcommer.

startup.py und PYQGIS_STARTUP:

Bei jedem Start von QGIS wird an einem bestimmten Pfad nach einer Datei startup.py gesucht und deren Inhalt ausgeführt wenn sie gefunden wird. Über die Umgebungsvariable PYQGIS_STARTUP kann man zusätzlich eine Python-Datei beim Start von QGIS ausführen lassen bevor die Initialisierung von QGIS beendet ist.

Benutzerdefinierte Funktionen:

Benutzerdefinierte Funktionen kann man an verschiedensten Stellen innerhalb von QGIS nutzen: Feldrechner, Datendefinierte Einstellungen („Übersteuerung“), Textfelder im Drucklayout/Atlas

Attributformular anpassen mit der Python Init-Funktion:

Wem die umfangreichen Einstellungsmöglichkeiten im Reiter „Attributformular“ eines Vektorlayers noch nicht reichen kann mittels Python eigene Logik zur Analyse der Nutzereingaben hinterlegen

Makros:

QGIS-Makros können beim Öffnen, Speichern und Schließen eines Projektes verwendet werden.

Bsp: Sicherstellen, dass niemand anderes gerade an meinem auf einem Netzlaufwerk abgelegten QGIS-Projekt arbeitet.

QGIS-Aktionen ("Actions")

Mittels QGIS-Aktionen kann man z.B. mit einem Klick auf ein Geometrieobjekt Python-Code ausführen

Beispiel: Action zum Anzeigen von Kind-Elementen für ein selektiertes Objekt.

Processing-Toolbox:

Mittels Python kann man eigene Processing-Algorithmen erzeugen oder bestehende Algorithmen an eigene Bedürfnisse anpassen

Maßgeschneiderte QGIS-Arbeitsumgebung mittels Python

„--code“-Parameter beim Aufruf von QGIS:

Über den code-Parameter kann man beim Start von QGIS direkt einen Python-Code übergeben, der direkt nach dem Laden von QGIS ausgeführt wird

Plugins:

Nicht zuletzt kann man auch über in Python geschriebene Plugins Python in QGIS nutzen.

Ich zeige einen Ausblick darauf, wie man:

- mit Hilfe eines einmal erzeugten Minimal-Plugins in kürzester Zeit eigene Funktionalitäten in QGIS testen und perspektivisch einbetten kann
- Eigene Processing-Algorithmen dauerhaft bereitstellt oder verteilt
- Eigene Locator Provider nutzt, um noch effektiver mit QGIS zu arbeiten

Ausblick: Standalone/ Kommandozeilen- PyQGIS-Skripte, die ausserhalb der QGIS-GUI laufen.

(Ab QGIS 3.14 auch Aufruf eigener Processing-Skripte über „qgis_process run“ außerhalb der GUI)

Thomas Baumann

Nominatim für den Hausgebrauch

Nominatim ist der derzeit bekannteste Geocoder für OpenStreetMap-Daten. Die meisten kennen ihn jedoch hauptsächlich als Online-Service, der von der OpenStreetMap Foundation unter openstreetmap.org betrieben wird. Aber Nominatim ist eigentlich eine Software, die man auch lokal installieren und nutzen kann. In dieser Session zeigen wir, wie man in nur 20 Minuten seinen eigenen Installation von Nominatim aufsetzen, Daten importieren und sofort nutzen kann.

Nominatim ist der derzeit bekannteste Geocoder für OpenStreetMap-Daten. Die meisten kennen ihn jedoch hauptsächlich als den Online-Service, der von der OpenStreetMap Foundation unter <https://nominatim.openstreetmap.org> betrieben wird. Aber Nominatim ist eigentlich eine Software, die man auch lokal installieren und nutzen kann. Das ist besonders dann von Interesse, wenn man grössere Mengen von Anfragen hat, die sich auf ein kleineres geografisches Gebiet beziehen. Mit einer eigenen Installation kann man das schnell erledigen, ohne auf die strikten Nutzungslimits des Online-Services achten zu müssen und vor allem ohne die unter Umständen privaten Adresslisten an einen öffentlichen Server zu schicken.

In den letzten Jahren hat sich viel getan, um die Installation von Nominatim einfacher zu machen. In dieser Demo-Session zeige ich, wie man in nur 20 Minuten seinen eigenen Installation von Nominatim aufsetzen, Daten einer Region importieren und eine erste Liste von Adressen geocodieren kann.

Sarah Hoffmann

10 Jahre Overpass API

Seit nunmehr 10 Jahren ermöglicht die Overpass API das Durchsuchen der OpenStreetMap-Daten. Grund genug zu zeigen, wie sich OpenStreetMap und die Overpass API über diesen Zeitraum gewandelt haben. Es gibt einen Überblick zu Nutzungsstatistiken der Overpass API, technischen Herausforderungen und Wünschen aller Art an mich als Entwickler.

Seit nunmehr 10 Jahren ermöglicht die Overpass API das Durchsuchen der OpenStreetMap-Daten. Grund genug, das OpenStreetMap von damals dem heutigen gegenüberzustellen und zu zeigen, wie sich OpenStreetMap und die Overpass API über diesen Zeitraum gewandelt oder auch nicht gewandelt haben.

Dazu gibt es einen Überblick, mit welchen Abfragen die Overpass API tatsächlich genutzt wird, wie sie sich gegen Überlastung geschützt hat und schützt, und welche Wünsche auf anderem Wege an mich als Entwickler herangetragen worden sind.

Dr. Roland Olbricht

Analyse Tool für Simple Indoor Tagging

JAN SCHMALFUSS-SCHWARZ, JULIAN STRIEGL

Abstract

Die Anzahl der vollständigen, mit Simple Indoor Tagging getaggten, Indoor-Karten ist gering und die darin enthaltenen Informationen sowie ihre Güte variieren stark. [1] Zur Unterstützung des Mappings wurde daher auf der Basis von OsmlnEdit ein Tool entwickelt, welches die Aspekte der Raumgeometrie, der Erreichbarkeit von Räumen sowie das Vorhandensein von benötigten Informationen für Menschen mit Einschränkungen analysiert und anhand von frei wählbaren Achievements Unvollständigkeiten oder Fehler innerhalb dieser darstellt. Die Mappenden haben diesbezüglich die freie Wahl, welche Achievements sie anstreben und werden durch das Tool unterstützt, indem sie beispielsweise auf sich überschneidende Räume oder fehlende Informationen hingewiesen werden.

OsmlnEdit

Die Kurzform OsmlnEdit [2] (*OpenStreetMap-Indoor-Editor*) steht für eine webbasierte Applikation, welche die Bearbeitung von Indoor-Karten im Format Simple Indoor Tagging [3] ermöglicht und von Adiren Pavier entwickelt wurde. Die Anwendung unterscheidet dabei zwischen einem Explorations- sowie Editier-Modus. Ferner bietet OsmlnEdit alle benötigten Grundfunktionalitäten wie beispielsweise das Abrufen der Gebäudedaten vom OSM-Server sowie deren Aufarbeitung. Da OsmlnEdit gleichzeitig unter einer Open-Source-Lizenz steht, wurde das entwickelte Tool namens "SIT-MapChecker" bewusst auf der Applikation aufgesetzt, wobei der Bearbeitungsmodus im ersten Entwicklungsschritt blockiert wurde, da die Benutzendenoberfläche stark angepasst werden musste. Die Funktionalitäten blieben aber erhalten, sodass dieser zu einem späteren Zeitpunkt an die neu geschaffene Benutzendenoberfläche angepasst werden kann.

Grundlage der Analyse von Indoor-Karten

Zur einfachen Verarbeitung der Gebäudedaten wurde OsmlnEdit dahingehend erweitert, dass eine Funktion alle zu einem Gebäude gehörenden Elemente innerhalb eines Objekts zusammenfasst. Dieses wird anschließend genutzt, um die verschiedenen Analyseschritte zu vereinfachen.

SIT-Konformität

Die Basis des entwickelten Tools bildet die Überprüfung der vorhandenen Daten auf spezifische Anforderungen. So wird kontrolliert, ob der Daten-Typ eines OSM-Objekts für das daran geknüpfte Key-Value-Paar zulässig ist. (vgl. Tabelle 1)

Tabelle 1: Übersicht zu den verschiedenen realen Objekten in Verbindung mit ihren Bezeichnern und erlaubten OSM-Objekten

Reales Objekt	Typbezeichner	OSM-Objekt-Art
Gebäude	building=*	<i>closed way</i> oder <i>relation:multipolygone</i>
Durchgang	door=*	<i>node</i>
Raum	indoor=room	<i>closed way</i> oder <i>relation:multipolygone</i>
Korridor	indoor=corridor	<i>closed way</i> oder <i>relation:multipolygone</i>
wandlose Fläche	indoor=area	<i>closed way</i> oder <i>relation:multipolygone</i>
Wand	indoor=wall	<i>way</i> aber keine <i>closed way</i>

Dabei werden drei Fehlerarten unterschieden, die sich in Objekte, die keinen Bezeichner besitzen, Objekte, die einen inkonsistenten Bezeichner besitzen sowie Objekte, die mehrere Bezeichner besitzen, aufgliedern.

Des Weiteren erfolgt eine Überprüfung der Geometrie einzelner Objekte und der innerhalb eines Gebäudes enthaltenen Elemente auf unzulässige Überschneidungen. Ferner wird auch betrachtet, ob Räume, Korridore, Wände oder sonstige Flächen die Gebäudegrenzen überragen, und ob die zu den Tags zugeordneten Werte zulässig sind. So wird beispielsweise für die Keys *max_level* und *min_level* überprüft, ob *max_level* größer oder mindestens gleich *min_level* ist und ob es sich jeweils um einen ganzzahligen Wert handelt.

Zusätzlich geschieht eine Evaluation der Vollständigkeit des Gebäudes anhand der Abdeckung des Grundrisses durch die gegebenen Indoor-Elemente sowie eine Überprüfung der Erreichbarkeit der Räume der einzelnen Etagen von den Eingängen des Gebäudes aus.

Barrierefreiheit

Im Zuge der Anreicherung der Karten-Daten mit Barrierefreiheitsinformationen wurde ein rudimentärer Ansatz geschaffen, der möglichst einfach zu erweitern ist. Dafür wurden in OSM bereits vorhandene Tags zur Darstellung von Barrierefreiheitsinformationen - wie der Wheelchair-Tag - analysiert und weitere für Menschen mit diversen Einschränkungen nötige Informationen erörtert und in Form von Achievements zusammengefasst. So wird innerhalb des Tools in *Basic*- sowie *Advanced*-Achievements unterschieden, wobei zum Erreichen der höheren Anforderungen spezifischere Angaben wie zum Beispiel die Breite einer Tür vonnöten sind.

Aktuelle Umsetzung

Die aktuelle grafische Nutzendenschnittstelle untergliedert sich in die drei Bereiche der Gebäudekarte, Gebäudeinformationen und den Bereich, in dem die aktuellen Warnungen und Fehler der einzelnen Achievements angezeigt werden (vgl. Abbildung 1).

Analyse Tool für Simple Indoor Tagging

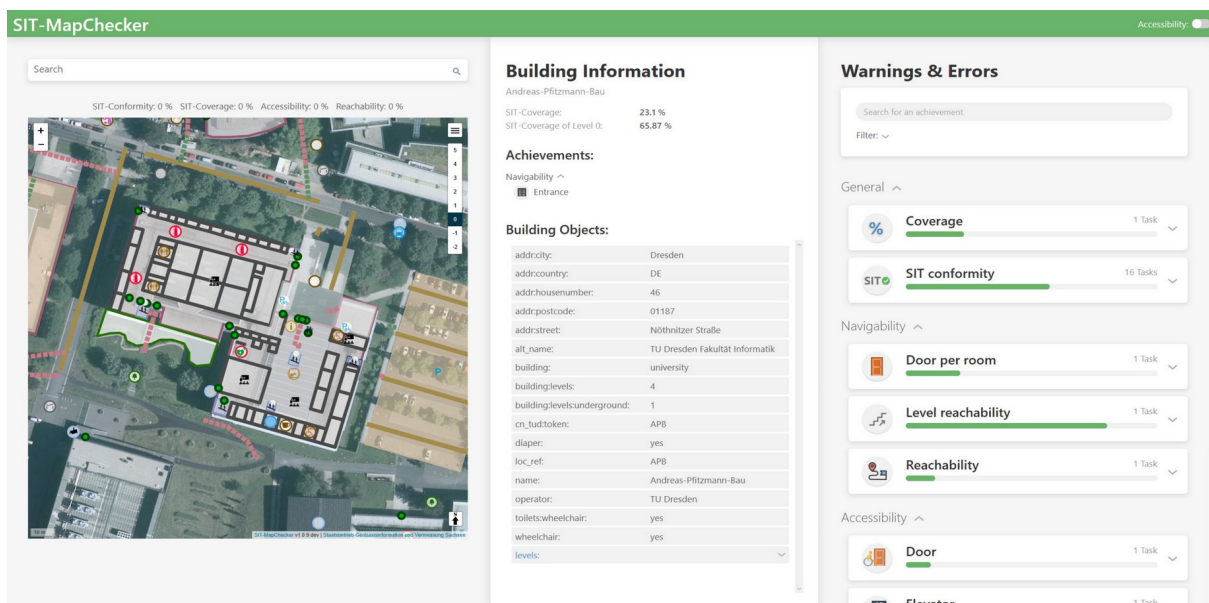


Abbildung 1: Aktuelle GUI der Anwendung SIT-MapChecker

Gebäudekarte

Die Gebäudekarte visualisiert alle darin enthaltenen Elemente des aktuell ausgewählten Stockwerks. Sie dient ferner dazu, dass Elemente selektiert und gegebenenfalls damit verbundene Fehler angezeigt werden können. Zugleich ist dieser Prozess auch invertiert möglich, sodass, insofern ein gefundenes Problem an ein entsprechendes Objekt geknüpft ist, dieser Fehler innerhalb des Bereiches "Warnungen und Fehler" ausgewählt werden kann und das entsprechende Element in der Karte markiert wird. Überdies ist in diesem Bereich die Suchfunktion integriert.

Gebäudeinformationen (Building Information)

Der Bereich der Gebäudeinformationen beinhaltet zum einen die Angaben über die bisherige Abdeckung des gesamten Gebäudes sowie des ausgewählten Stockwerks mit entsprechenden OSM-Objekten. Zum anderen stellt er die Informationen über das Gebäude und alle dazugehörigen OSM-Objekte zur Verfügung. Darüber hinaus enthält er alle erreichten Achievements.

Warnungen und Fehler (Warnings & Errors)

In der rechten Spalte werden die verschiedenen Warnungen und Fehler anhand der Achievements ausgegeben, welche mit Hilfe eines Filters an- und abgewählt werden können.

Achievements zur Barrierefreiheit

Neben den eingangs erwähnten Anforderungen bezüglich der SIT-Konformität sowie der Vollständigkeit werden die Indoor-Daten auf die in Tabelle 2 enthaltene Barrierefreiheitsinformationen geprüft.

Analyse Tool für Simple Indoor Tagging

Tabelle 2: Übersicht über die aktuell implementierten Achievements bezüglich Barrierefreiheitsinformationen

Türen	Basic	Advanced
	Alle Türen des Gebäudes müssen den Tag <i>wheelchair</i> aufweisen, der folgende Werte besitzen kann: <i>yes</i> , <i>no</i> oder <i>limited</i> .	Wenn alle Türen eines Gebäudes korrekt mit dem <i>wheelchair</i> Tag versehen sind, müssen diese zusätzlich mit dem Tag <i>width</i> gekennzeichnet werden, welcher einen ganzzahligen Wert in Zentimeter besitzen sollte.
Rampen	Basic	Advanced
	Alle Rampenobjekte eines Gebäudes müssen mindestens den Tag <i>ramp</i> mit den Werten <i>yes</i> oder <i>no</i> besitzen.	Ist das Basic Achievement erreicht, müssen alle Rampenobjekte mit dem erweiterten Tag <i>ramp:wheelchair</i> mit den Werten <i>yes</i> oder <i>no</i> gekennzeichnet werden.
Fahrstühle	Basic	Advanced
	Jedes Fahrstuhlobjekt muss mit den Tags <i>wheelchair=yes/no</i> und <i>tactile_writing:braille:lg=yes/no</i> gekennzeichnet sein.	Alle Fahrstuhlobjekte müssen zusätzlich die Tags <i>handrail=yes/no</i> , <i>audio_announcement=yes/no</i> , <i>width</i> und <i>length</i> aufweisen. Die Werte für Länge und Breite des Fahrstuhls sollten wiederum ganzzahlige Angaben in Zentimetern sein.
Treppen	Basic	Advanced
	Alle Treppenobjekte eines Gebäudes benötigen die Tags <i>tactile_writing:braille:lg=yes/no</i> und <i>ramp:wheelchair=yes/no</i> .	Zusätzlich zu den Mindestanforderungen müssen die Tags <i>handrail:left:right:center=yes/no</i> , <i>tactile_paving=yes/no</i> , <i>conveying=yes/no</i> und <i>width</i> vorhanden sein.
Toiletten	Basic	
	Alle Toilettenräume sollten die Tags <i>wheelchair=yes/no</i> und <i>tactile_writing:braille:lg=yes/no</i> vorhanden sein.	
Taktile Informationen	Basic	
	Prozentuale Anzahl von SIT Objekten, die zusätzlich mit den Tags <i>tactile_paving</i> und <i>tactile_writing</i> gekennzeichnet sind	

Zusammenfassung

Das aktuell entwickelte Tool ist ein erster Schritt zur Unterstützung des Kartographierungsprozesses von Gebäuden im SIT-Format. Es gilt, dieses einerseits in weiteren Iterationen auf die Bedürfnisse der Mapper anzupassen. Andererseits müssen die Achievements bezüglich der Anforderungen diverser Nutzender erweitert werden.

Danksagung

Die hier entwickelte Softwarelösung wurde im Rahmen eines studentischen Praktikums an der Fakultät Informatik der TU Dresden innerhalb des Forschungsprojekts Accessible Maps [4] entwickelt. Daher möchten wir uns bei den Studierenden bedanken, die die Grundlage für die Anwendung geschaffen haben.

Finanziert durch das Bundesministerium für Arbeit und Soziales (BMAS) aus Mitteln des Ausgleichsfonds (Förderkennzeichen: 01KM151112).

Kontakt zum Autor:

Jan Schmalfuß-Schwarz

TU Dresden, Fakultät Informatik, Professur für Mensch-Computer Interaktion
Angestellter im Projekt Accessible Maps

Besucheranschrift:
Andreas-Pfitzmann-Bau, 1.0 OG, Raum 1062
Nöthnitzer Straße 46
01187 Dresden

Telefon: 0351 463-38495
eMail: jan.schmalfuss-schwarz@tu-dresden.de

Literatur

- [1] Julian Striegl, Claudia Loitsch, Jan Schmalfuss-Schwarz, Gerhard Weber. *Analysis of Indoor Maps Accounting the Needs of People with Impairments*. In: Computers Helping People with Special Needs, 17th International Conference, ICCHP 2020, Lecco, Italy
- [2] <https://github.com/PanierAvide/OsmlnEdit> (Stand: 01.05.2021)
- [3] https://wiki.openstreetmap.org/wiki/Simple_Indoor_Tagging (Stand: 01.05.2021)
- [4] <https://accessiblemaps.de>

FOSS in der Gewässerentwicklungsplanung – Erfahrungsbericht aus Praxis und Lehre

Die EU-WRRL fordert für alle natürlichen Oberflächengewässer den „guten ökologischen Zustand“. Es werden von verschiedensten Akteuren erhebliche Anstrengungen unternommen, um u.a. durch Renaturierungsplanwerke die Gewässerqualität zu verbessern. Am Beispiel eines zweisemestrigen Projektseminars in Kooperation mit einem Naturschutzverein werden die GIS-gestützten Analyseschritte einer Gewässerbewertung und -entwicklungsplanung (u.a. Drohnen-Photogrammetrie, Kartierung via App) präsentiert.

Die europäische Wasserrahmenrichtlinie fordert für alle natürlichen Oberflächengewässer den sogenannten „guten ökologischen Zustand“. Für die Bewertung werden unter anderem Fische und Wirbellose herangezogen. Aktuell erreicht nur ein Bruchteil der Gewässer die Umweltziele. Die Ursachen sind vielfältig, liegen aber meist in einer erhöhten Nährstoffbelastung und/oder einer tiefgreifenden Veränderung der Gewässerstruktur begründet. Von verschiedensten Akteuren werden erhebliche Anstrengungen unternommen, um u.a. durch umfangreiche Renaturierungskonzepte bzw. -planwerke die Gewässerqualität zu verbessern.

Damit befasste Absolvent:innen von Studiengängen mit ökologischen Schwerpunkten belegen daher neben fachlichen Vertiefungen (u.a. Ichthyologie, Entomologie bis allg. Gewässerökologie) in der Regel auch Kurse zur räumlichen Analyse und kartografischen Aufbereitung.

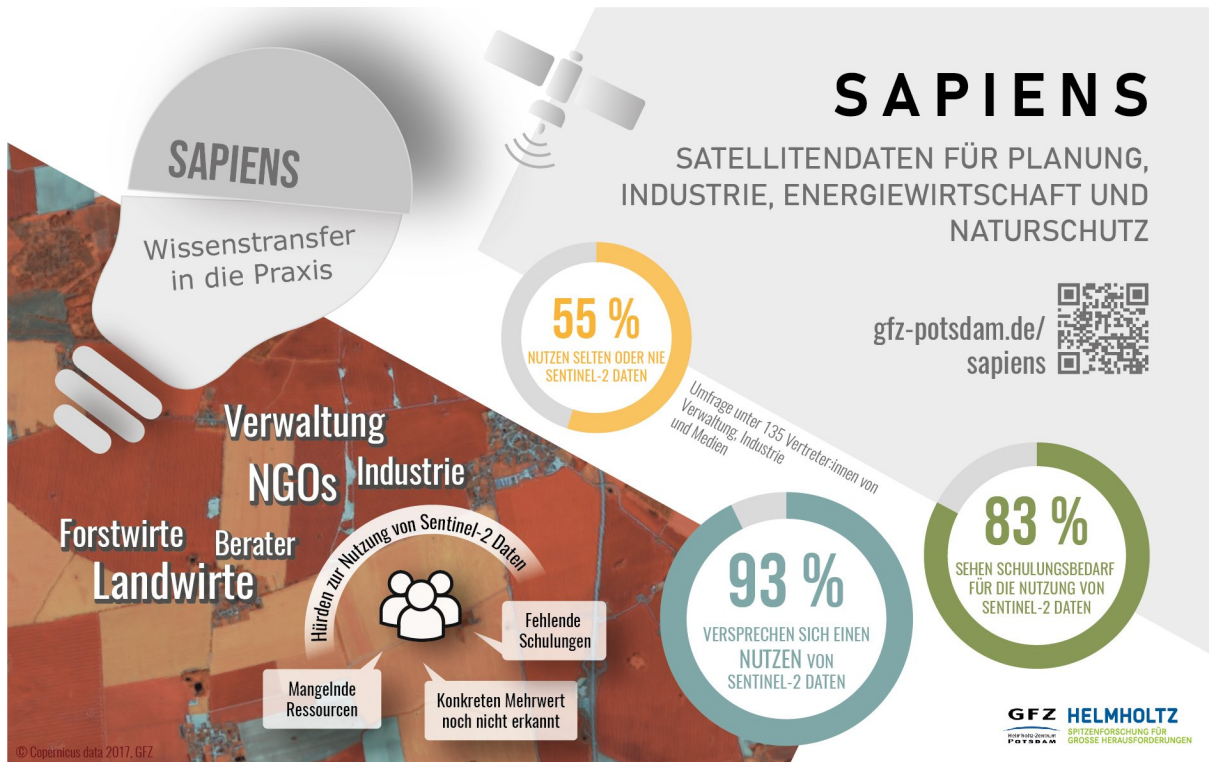
Am Beispiel eines zweisemestrigen Projektseminars in Kooperation mit einem lokalen Naturschutzverein werden die GIS-gestützten Analyseschritte einer Gewässerbewertung (u.a. Drohnen-Photogrammetrie, Tracking von Fischbestandserhebung, Kartierung via App) präsentiert.

Der Vortragende gibt zudem einen Erfahrungsbericht über die Erstellung von Gewässerentwicklungsplänen mit QGIS in Lehre und Ehrenamt und diskutiert weitere Publikationformen wie (partizipative) WebGIS-Anwendungen.

Jens Eligehausen

Wissenstransfer-Projekt SAPIENS: Satellitendaten für Planung, Industrie, Energiewirtschaft und Naturschutz

NORA MEYER ZU ERPEN



Dank der Copernicus-Mission als Teil des Satellitenprogramms der Europäischen Weltraumorganisation (ESA) sind seit 2015 räumlich hochauflösende, multispektrale Fernerkundungsdaten der Sentinel-2 Satelliten verfügbar. Diese kostenfreien Satellitendaten liefern hoch aktuelle und flächenhafte Umweltinformationen über Land, Wasser, Biomasse, Boden und vieles mehr. Diese Datengrundlage hat das Potenzial, Entscheidungsprozesse auf verschiedensten Ebenen und in unterschiedlichsten Organisationen zu unterstützen. Die Anwendungsfelder sind zahlreich: So können beispielsweise Landwirte:innen ihre Produktionsprozesse optimieren, NGOs aus der Ferne den Zustand von Regenwäldern überwachen oder Umweltbehörden schützenswerte Biotope monitoren und Zustandsänderungen schnell feststellen.

Der Schatz an frei verfügbaren Fernerkundungsdaten wächst zwar exponentiell an, aber die Nutzung ist bisher noch wenig in den Arbeitsalltag und in Entscheidungsprozesse integriert. Für viele unerfahrene Nutzer:innen ist die Hemmschwelle, Satellitendaten zu nutzen und in Arbeitsprozesse zu integrieren, hoch. Weder der Zugang zu den Daten noch ihre Nutzung sind selbsterklärend noch intuitiv. Tutorials oder Anleitungen sind in der Regel auf wissenschaftlichem Niveau und oft nur auf Englisch verfügbar.

Ähnlich verhält es sich mit digitalen Schulungsangeboten. Die Anzahl von digitalen Schulungsangeboten im Bereich Satellitenfernerkundung und Erdbeobachtung nimmt stetig zu, diese Entwicklung ist insbesondere durch die Covid19-Pandemie beflügelt worden. Dabei sind die Inhalte aber häufig von der Wissenschaft für die Wissenschaft aufbereitet, auf Englisch verfasst und durch Forschungsgegenstände, nicht aber zwingend Nutzerbedarfe, motiviert.

Wissenstransfer-Projekt SAPIENS: Satellitendaten für Planung, Industrie, Energiewirtschaft und Naturschutz

Um diese Lücke zwischen Nutzenden und Wissenschaft zu schließen, fördert die Helmholtz-Gemeinschaft das Wissenstransfer-Projekt SAPIENS (Satellitendaten für Planung, Industrie, Energiewirtschaft und Naturschutz). Im Rahmen des Projektes werden zielgruppenorientierte Weiterbildungsangebote entwickelt. Im engen Austausch mit Nutzer:innen werden Schulungsbedarfe ermittelt und das Wissen in Form von niederschwelligen digitalen Schulungsformaten verfügbar gemacht. Gemeinsam mit interessierten Partner:innen aus Wirtschaft, NGOs und öffentlicher Verwaltung entwickeln wir Pilotprojekte und passen die Inhalte, Lernziele und digitalen Lernformate an die jeweiligen Bedarfe an.

Neben der Vermittlung von Grundlagen stehen reale Anwendungsbeispiele im Fokus: so beispielsweise ein Modul zu Zeitreihenanalysen von Wäldern, um Aufforstungsprojekte gemeinnütziger Organisationen zu unterstützen. Darüber hinaus sind im Rahmen der Module auch praktische Übungen zu Themen wie regenerative Energien und / oder alternative Formen der Landwirtschaft (z.B. Agro-Forestry) geplant.

Zusätzlich entsteht eine Lern-Plattform, auf der diese Online Schulungen auch ohne festen Termin frei zugänglich sind und ein Austausch zwischen Schulungsteilnehmer:innen gefördert wird.

Bei Interesse an unserem SAPIENS-Projekt kontaktieren Sie mich gerne:

Nora Meyer zu Erpen
GFZ Deutsches GeoForschungsZentrum Potsdam
Telegrafenberg,
14473 Potsdam
0331-288 28887
Nora.meyerzuerpen@gfz-potsdam.de

Konfiguration eines WFS-T mit MapServer und TinyOWS

JÖRG THOMSEN

MapServer selbst kann keinen transaktionalen WFS bereit stellen. Mit dem Partner-Projekt TinyOWS ist das auf einfache Weise möglich. Dabei kann der TinyOWS auch über eine Map-Datei konfiguriert werden. Die Demo-Session zeigt wie das funktioniert. Darüber hinaus wird gezeigt wie TinyOWS als stand-alone-Anwendung über kurze XML-Dateien konfiguriert werden kann und wie über die Konfiguration des Webserver mehrere WFS parallel betrieben werden können.

Ein WFS-T mit XML-Konfiguration benötigt nur wenige Zeilen:

```
<tinyows online_resource="http://schulung.foss.academy/cgi-bin/tinyows"
  schema_dir="/usr/local/tinyows/schema/" check_schema="0">

  <pg host="localhost" user="xxxx" password="xxxx" dbname="mapserver" port="5432"/>
  <metadata name="TinyOWS Server" title="TinyOWS Server - Demo Service" />

  <layer retrievable="1" writable="1" ns_prefix="tows" ns_uri="http://www.tinyows.org"
    name="ne_10m_urban_areas" title="World Administrative Urban Areas" />
</tinyows>
```

Auch für eine TinyOWS-Konfiguration innerhalb einer Mapdatei sind nur wenige Einträge notwendig:

```
MAP
  NAME "TinyOWS"
  WEB
    METADATA
      "tinyows_schema_dir" "/usr/local/share/tinyows/schema/"
      "tinyows_onlineresource" "http://schulung.foss.academy/cgi-bin/tinyowsmapfile"
      "tinyows_check_schema" "0"
      ...
    END
  END
  LAYER
    ...
    METADATA
      'wfs_title' 'Urban Areas'
      'wfs_namespace_prefix' 'towsmap'
      'wfs_namespace_uri' 'http://www.mapserver.org/tinyows/'
      'wfs_srs' 'EPSG:4326'
      'tinyows_table' 'ne_10m_urban_areas'
      'tinyows_writable' '1'
      'tinyows_retrievable' '1'
    END
  END
END
```

Kontakt zum Autor:

Jörg Thomsen
WhereGroup GmbH
Bundesallee 23
10717 Berlin
joerg.thomsen@wheregroup.com

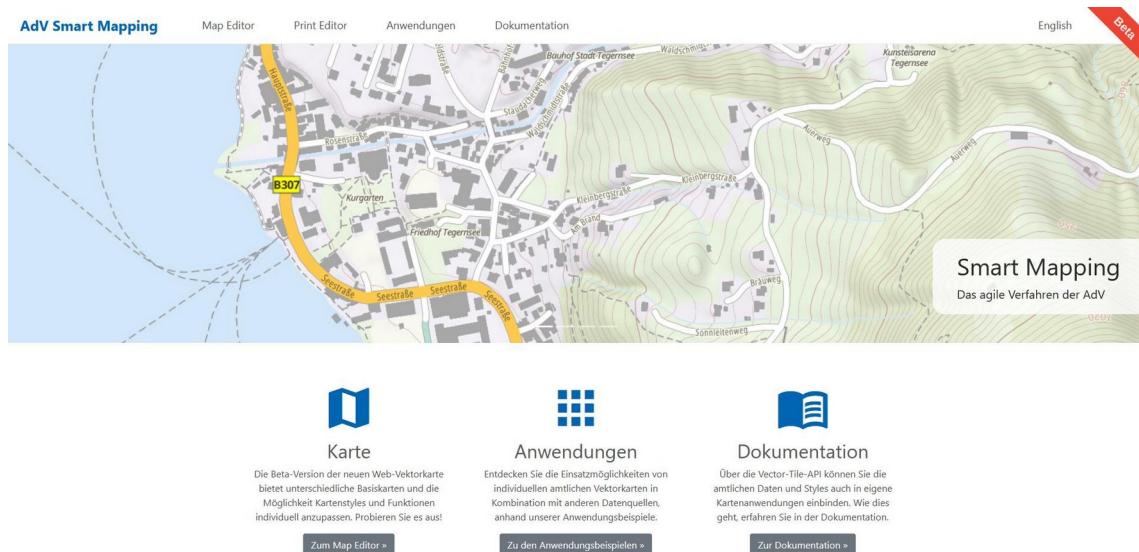
Quellen

[1] <https://mapserver.org/tinyows/>

[2] <https://github.com/MapServer/tinyows>

[3] https://www.fossgis-konferenz.de/2012/programm/attachments/353_fossgis2012_tinyows.pdf

Agile Entwicklung, Open Source und amtliche Daten



Wer heute nicht agil mit Open Source in der Cloud entwickelt ist irgendwie out. Hinter diesen Schlagwörtern verbergen sich auch wirklich tolle Konzepte und Technologien. Allerdings müssen wir erst lernen, sie auch wirklich sinnvoll einzusetzen. Die AG SmartMapping der AdV hat sich in fast zwei Jahren harter Arbeit eine wirklich gute Basis geschaffen. Dieser Vortrag berichtet von den Schwierigkeit aber auch Erfolgen mit agiler Entwicklung in einem eher statischen (und staatlichen) Umfeld.

Die AG SmartMapping der AdV wurde beauftragt, amtliche Daten mit neuen Technologien bereitzustellen. Nach einer anfänglichen Phase in der umfangreiche Konzeptpapiere entstanden sind, arbeitet seit Mitte 2019 eine Gruppe von 25 Entwicklern aus verschiedenen Bundesländern auf einer gemeinsamen Plattform an einem gemeinsamen Verfahren. Dabei ist ein bemerkenswertes Projekt entstanden, welches wirklich neue Wege geht, dabei den Kontakt zu den Geobasisdaten aber nicht verloren hat. Weitere Informationen sind auf der Projektwebseite <https://adv-smart.de> zu finden

Arnulf Christl

Erfassung und Visualisierung der Personenanzahl in Gebäuden unter Verwendung des Internets der Dinge

MARKUS KWAŚNICKI¹ & PASCAL NEIS²

¹Johannes Gutenberg-Universität Mainz, ² Hochschule Mainz - University of Applied Sciences

Die Mehrheit der Menschen trägt heutzutage ein Smartphone bei sich. Neben dem Mobiltelefon existieren noch weitere Wearables, wie Smartwatches, Bluetooth-Kopfhörer oder Tablets. Unabhängig davon, ob eines dieser Geräte mit einem WLAN verbunden ist oder nicht, sucht dieses alle paar Sekunden nach WLAN-Stationen in der Umgebung und versucht sich mit einem bekannten WLAN-Access Point zu verbinden. Durch die sich wiederholenden Verbindungsversuche hinterlassen die Geräte Informationen in ihrer Umgebung.

Unter der Grundannahme, dass die Menge an Wearables an einem Ort mit der Anzahl anwesender Personen korreliert, wurde eine Anwendung zur Erfassung und Visualisierung von Personen in Gebäuden und deren Räumen prototypisch implementiert. Bei der Erfassung der Personen greift der entwickelte Sensor auf die ständig versendeten Datenpakete der Geräte zurück, die als Indikator für die Präsenz herangezogen und ausgewertet werden. Bei der Entwicklung des Prototyps wurde stets darauf geachtet, dass dieser vollständig mit freier Software, unter Anwendung offener Formate sowie basierend auf standardisierten Schnittstellen, umgesetzt wird.

In dem Vortrag wird die Anwendung [1] mit ihren Komponenten vorgestellt:

- Entwickler Sensor (Raspberry Pi) zur Erfassung von Endgeräten im WLAN
- Datenhaltung und -abruf der erfassten Informationen mittels OGC SensorThings API
- Webbasierte Visualisierung (Leaflet) der erfassten Personen auf einer Karte (auf Gebäudegrundriss mit OpenStreetMap Daten)

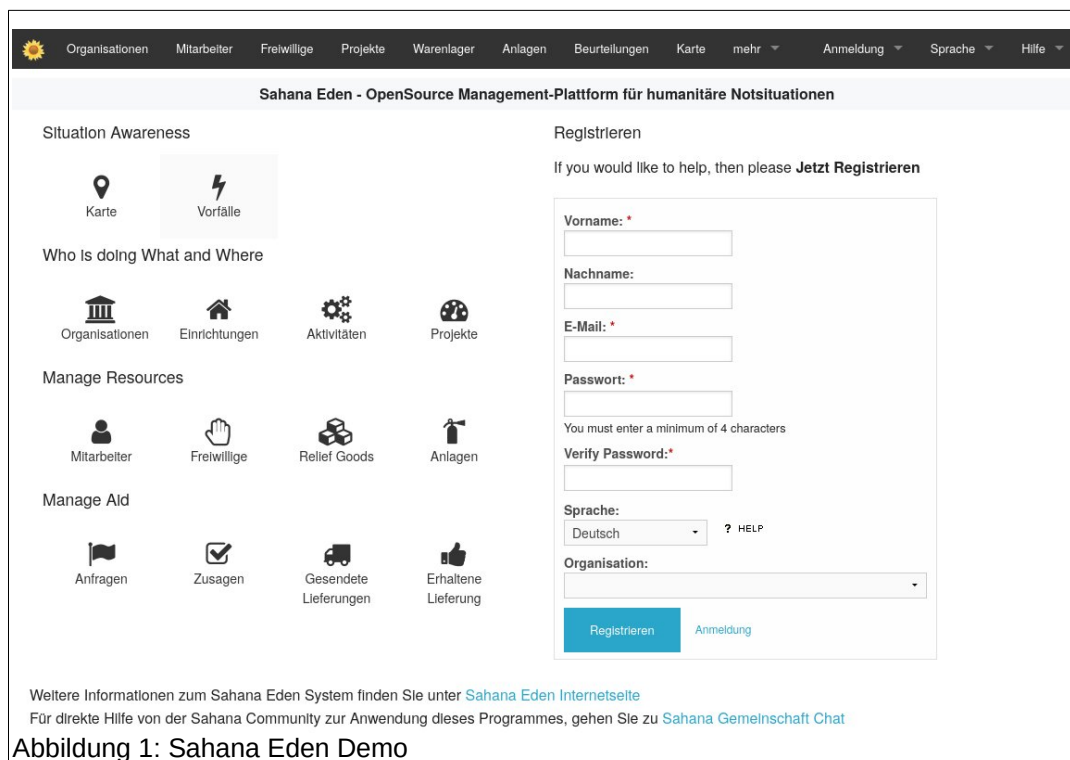
In einem Versuchsaufbau innerhalb der Hochschule Mainz wurde der entwickelte Prototyp mit seinen Bestandteilen über einen Zeitraum von 7 Tagen evaluiert. Die Ergebnisse werden ebenfalls im Vortrag gezeigt. Ursprünglich wurde der Prototyp für eine aktuelle Lagebewertung und den Einsatz in einem Evakuierungsszenario konzipiert. Durch die aktuelle Pandemie und die notwendigen Gegenmaßnahmen zeigten sich allerdings weitere Anwendungsfälle, wofür die Anwendung eingesetzt werden könnte. Auf diese wird im Ausblick des Vortrags eingegangen.

[1] Github: <https://github.com/mrkskwsnck/master-prototyp>

Keywords: Internet of Things (IoT), Open Source Software, Raspberry Pi, Open Geospatial Consortium (OGC), SensorThings API, Leaflet, OpenStreetMap, Smartphones, Wireless Local Area

Einsatz von FOSS bei der Bewältigung der COVID-19 Krise in RLP

ARMIN RETTERATH



Sahana Eden - OpenSource Management-Plattform für humanitäre Notsituationen

Situation Awareness

Registrieren

If you would like to help, then please **Jetzt Registrieren**

Vorname: *

Nachname:

E-Mail: *

Passwort: *

You must enter a minimum of 4 characters

Verify Password: *

Sprache: Deutsch ? HELP

Organisation:

Registrieren Anmeldung

Weitere Informationen zum Sahana Eden System finden Sie unter [Sahana Eden Internetseite](#)

Für direkte Hilfe von der Sahana Community zur Anwendung dieses Programmes, gehen Sie zu [Sahana Gemeinschaft Chat](#)

Abbildung 1: Sahana Eden Demo

Einleitung

Zu Beginn der COVID-19 Krise im März 2020, gab es in Rheinland-Pfalz zunächst die Anforderung einer kartographischen Aufbereitung von Infektionszahlen auf Ebene der PLZ-Gebiete. Die Grundlage für das Verfahren bildete die tägliche Meldung auf Basis einer Exceltabelle, die per Mail verschickt wurde. Um dieses Verfahren zu verbessern, wurde vom Autor der Einsatz der Freien Open Source Krisenmanagement Software Sahana Eden [1] vorgeschlagen. Die Software verfügt sowohl über Module für die Erfassung von Infektionsfällen, als auch über Möglichkeiten zur Kontaktnachverfolgung. Aufgrund der schon etablierten Meldewege bei den Kreisverwaltungen, wurde jedoch zunächst vom Einsatz für diese Anwendungsfälle abgesehen.

Die eigentliche Initiative für die Inbetriebnahme von Sahana war der dringende Bedarf der Zusammenführung der Personendaten aus verschiedenen Akquise-Kampagnen für Helfer. Diese sollten die Gesundheitsämter insbesondere bei der Kontaktnachverfolgung unterstützen.

Einsatz von FOSS bei der Bewältigung der COVID-19 Krise in RLP

Ein weiterer Anwendungsfall ergab sich, als im Januar 2021 nach einer einfachen Lösung gesucht wurde, die Kosten für die Schnelltests von Erziehern und Lehrern zu übernehmen. Bei beiden Anwendungsfällen werden auch Geodaten verwendet. Zum einen zur Erfassung der Wohnorte der Freiwilligen und der Adressen der Impfzentren sowie Gesundheitsämter, zum anderen zur kartographischen Darstellung der Lage der Teststellen (Praxen, Corona-Ambulanzen, ...) bei der Schnelltestvermittlung.

Die folgenden Phasen erläutern den Verlauf der Entwicklung des Einsatzes von FOSS-Komponenten im Zusammenhang mit der Bewältigung der COVID-19 Krise:

Phase 1 – Anforderung nach granularem Monitoring des Infektionsgeschehens

Nach dem im Anschluss an die FOSSGIS2020 der Autor und seine Kollegen aufgrund des bestätigten Corona-Falles direkt in Quarantäne gingen, beschäftigte er sich in seiner Freizeit mit Optimierungsmöglichkeiten für die bestehenden Meldeprozesse der Infektionszahlen. In Rheinland-Pfalz gab es ab Mitte März nur eine zentrale Webseite des Ministeriums für Soziales, Arbeit, Gesundheit und Demographie (MSAGD), auf der die Zahlen einmal täglich in Form einer HTML-Tabelle publiziert wurden. Um hieraus dynamisch Karten ableiten zu können, wurde ein einfacher Crawler programmiert, der die Inhalte der Webseite täglich parste und in eine CSV-Datei überführte. Diese wurden mit ihren jeweiligen Zeitstempeln in eine PostGIS importiert. Zur Visualisierung wurde ein Mapserver basierter WMS-Layer mit Time Dimension genutzt. Damit stand eine sehr simple Lösung zur Verfügung, mit der sich insbesondere die zeitlichen Abläufe sehr gut beobachten ließen, und die außerdem - aufgrund der interoperablen Schnittstellen - geeignet war in beliebige Prozesse integriert zu werden.

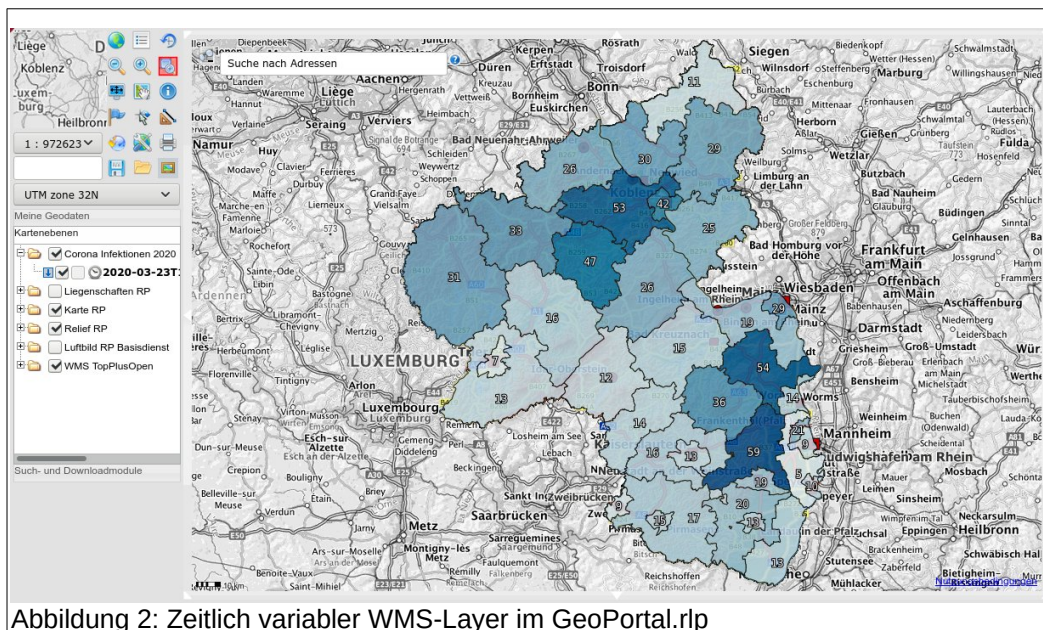


Abbildung 2: Zeitlich variabler WMS-Layer im GeoPortal.rlp

Am 27.03. fragte das MSAGD beim Autor an, ob es möglich wäre für die Staatskanzlei eine granulare Infektionskarte auf Basis von Postleitzahlbereichen zu erstellen. Bei einem Großteil der erfassten Infektionsfälle war auch die Postleitzahl vorhanden und man erhoffte sich, mit einer Kartendarstellung das lokale Infektionsgeschehen besser im Blick behalten zu können. Die Datenbasis für die Infektionszahlen war eine Exceltabelle und die benötigten Geometrien stammten aus OSM. Das Verfahren wurde analog zur Übersicht auf Landkreisebene auf Basis von WMS- und WFS-Schnittstellen aufgebaut und bis Ende September 2021 täglich aktualisiert.

Einsatz von FOSS bei der Bewältigung der COVID-19 Krise in RLP

Um eine nutzerfreundliche Präsentation zu bekommen, gab es neben den Kartendiensten und Schnittstellen auch noch einfache Leaflet-Clients [2] mit dem TimeDimension Modul [3].

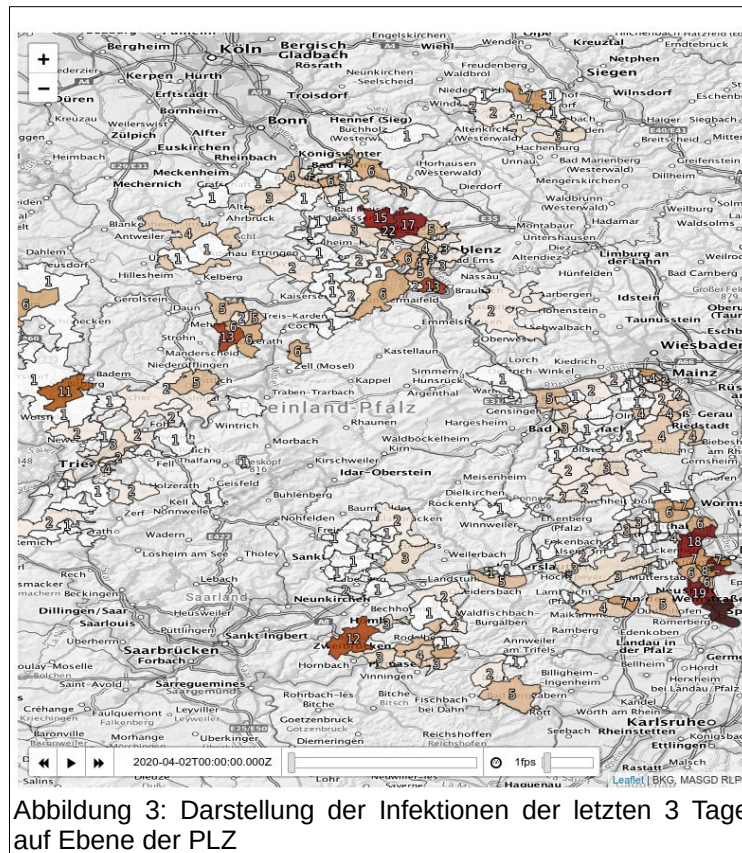


Abbildung 3: Darstellung der Infektionen der letzten 3 Tage auf Ebene der PLZ

Hinsichtlich der kartographischen Aufbereitung und Präsentation der Infektionszahlen konnte mit FOSS-GIS-Tools (PostGIS/Mapserver/Leaflet) eine sehr schnelle und einfache Lösung geschaffen werden, die zusätzlich noch die Möglichkeit bot direkt lagerelevante Daten über WMS/WFS in weitere Prozesse und Anwendungen einzubinden.

Phase 2 – Versuch der Optimierung der Prozesse durch Vorschlag der Einführung eines zentralen Krisenmanagementsystems

Hinsichtlich der realen Probleme, die im Kontext von Krisensituationen auftreten, ist eine landesweite, granulare Übersicht der Entwicklung der Infektionszahlen *nur ein kleiner* Baustein. Viel wichtiger ist die Digitalisierung und Vereinfachung der Prozesse zur Erfassung der aller relevanten Daten und die Koordinierung der benötigten Ressourcen. Sind diese Prozesse alle sauber umgesetzt, ergibt sich die Lageübersicht automatisch.

Um die für die Krisenbewältigung benötigten Prozesse zu digitalisieren und zu optimieren, hat der Autor die Verwendung der FOSS Sahana Eden vorgeschlagen und innerhalb von kürzester Zeit einen Prototypen aufgebaut anhand dessen sich die Möglichkeiten des Systems einfach verdeutlichen ließen.

Einsatz von FOSS bei der Bewältigung der COVID-19 Krise in RLP

Sahana Eden war schon von 2010 bis 2017 Bestandteil der *OSGeoLive* und ist eine webbasierte Krisenmanagementplattform, deren Entwicklung auf das Erdbeben im Indischen Ozean und den hierdurch hervorgerufenen Tsunami am 26.12.2004 zurückzuführen ist. Die Software wurde designed, um allgemeine Koordinationsprobleme zu lösen, die im Rahmen von humanitären Katastrophen auftreten z.B.:

- Personensuche,
- Management von Hilfe,
- Koordination von Freiwilligen,
- Tracking von Opfern,
- ...

Die Entwicklung wird von einer „non-profit“ Organisation mit Sitz in Californien koordiniert. Ziel der Organisation ist es *„Organisationen und Communities durch das Bereitstellen von Informationsmanagementsystemen in die Lage zu versetzen, besser auf Katastrophen vorbereitet zu sein und auf diese zu reagieren“*.

Notsituationen, bei denen Sahana Eden bisher u.a. zum Einsatz kam: Tsunami 2005 (Sri Lanka), Asian Quake 2005 (Pakistan), Yogyakarta Erdbeben 2006 (Indonesien), Erdbeben Peru (2007), Hurrican Myanmar (2008), Erdbeben Haiti (2010), Tsunami Japan (2011), Überflutungen Bosnien, Serbien und Kroatien (2014), Flüchtlingsmanagement DRK Baden-Württemberg (2016) [4] [5]

Wichtige Module:

- Dezentrale Organisationsregistry zur Verwaltung von Büros, Warenhäusern, ...
- Projekt Tracking – Wer macht Was, Wo und Wann? - z.B. Anfragen / Lieferungen,
- Human Resources – Verwaltung von Mitarbeitern und Freiwilligen sowie deren Fähigkeiten,
- Krankenhausmanagement – Meldesystem für Bettentypen, Personal, Zustand, ...
- Tracking von Krankheitsfällen (z.B. Ebola, COVID-19, ...)
- Lageübersicht auf interaktiver Karte (Integrationsmöglichkeit verschiedenster Datenquellen – local + remote)

Die Informationen zu den Möglichkeiten die der Einsatz einer solchen Software mit sich bringt, sowie der Link auf den Prototypen wurden Ende März 2020 an die verantwortlichen Institutionen weitergeleitet.

Phase 3 – Unterstützung bei der Einrichtung einer Sahana Eden Instanz zur Vermittlung von Freiwilligen an die Gesundheitsämter

Mitte April 2020 kam es zu der Anfrage des MSAGD, ob sich Sahana Eden auch eignen würde den Prozess der Vermittlung von Freiwilligen an die Gesundheitsämter abzubilden und ob das LVerGeo beim Aufbau und Betrieb der Anwendung Unterstützung leisten könne (Arbeitsname: „Helferdatenbank“).

Das Ministerium des Innern und das LVerGeo Rheinland-Pfalz sagten ihre Unterstützung in Form von Amtshilfe zu, und es wurde in kürzester Zeit ein Server im Rechenzentrum beschafft und für die notwendigen Anpassungsarbeiten an Sahana ein Dienstleistungsvertrag mit der Firma **Aidiq** [6] geschlossen.

Einsatz von FOSS bei der Bewältigung der COVID-19 Krise in RLP

Sahana Eden basiert im Kern auf dem Open Source Python Framework **web2py** [7] und nutzt Post-GIS als Backend. Die für das Deployment notwendigen Konfigurationen lassen sich Großteils über shell-Skripte automatisieren. Zur Erhöhung der Sicherheit wurde im nginx noch eine eigene WebApplicationFirewall aktiviert.

Ziel war es zunächst, allen Gesundheitsämtern einen Zugriff auf die gesamte Liste der Freiwilligen zu ermöglichen. Sie sollten sich die passenden Personen anhand ihrer Fähigkeiten, Wohnorte und weiterer Kriterien herausuchen und diese bei Bedarf eigenständig rekrutieren.

Abbildung 4: Suche nach Freiwilligen

Die Datengrundlage bestand aus Listen von Freiwilligen, die aus verschiedenen Akquise-Kampagnen stammten und dementsprechend heterogen vorlagen. Zusätzlich gab es Vorgaben des Datenschutzes, die eine Einteilung der Freiwilligen in verschiedene „Pools“ erforderlich machten. Bei den „managed Pools“ (Landesbedienstete) können die Gesundheitsämter die Personen nicht direkt rekrutieren, sondern fordern diese beim MSAGD an. Eine dort eingerichtete Arbeitsgruppe kümmert sich um die formellen Rahmenbedingungen für die Abordnung durch die bereitstellende Verwaltung.

Innerhalb von weniger als 2 Wochen konnte das Produktivsystem [8] freigeschaltet werden. Seit Anfang Mai 2020 können somit die Gesundheitsämter in Rheinland-Pfalz auf eine zentrale Datenbank von Freiwilligen zugreifen und diese bei Bedarf rekrutieren. Im Laufe des darauffolgenden halben Jahres wurden viele weitere Verbesserungen an den Prozessen durchgeführt. Und viele der geschaffenen Module sind in das Open Source Projekt zurückgefließen.

Phase 4 – Erweiterung der Plattform um die Vermittlung von Personal an die Impfzentren

Als Ende November 2020 in Rheinland-Pfalz 31 Impfzentren eingerichtet werden sollten, stellte sich ebenfalls die Frage, wie diese mit qualifiziertem Personal ausgestattet werden können. Hier bot sich

Einsatz von FOSS bei der Bewältigung der COVID-19 Krise in RLP

die Erweiterung der bestehenden „Helferdatenbank“ an. Das Datenmodell für die Freiwilligen wurde um zeitliche Verfügbarkeiten sowie räumliche Verwendungswünsche erweitert. Außerdem riefen die Landesapothekerkammer und Landesärztekammer dazu auf, sich für den Einsatz in den Impfzentren zu registrieren. Anfang Dezember explodierte der Bestand an Freiwilligen förmlich. Aktuell stehen mehr als 19.700 Freiwillige für den Einsatz in den Gesundheitsämtern und Impfzentren zur Verfügung. Seit Bestehen der Plattform wurden 1.780 Einsätze bestätigt.

Das System hat im vergangenen Jahr gezeigt, dass man mit minimalem Aufwand den Prozess zur Vermittlung von Freiwilligen digitalisieren kann. Als Nebeneffekt erhält man die aktuellen Standorte der Einrichtungen (Impfzentren und Gesundheitsämter) als Geodaten über verschiedene Schnittstellen.

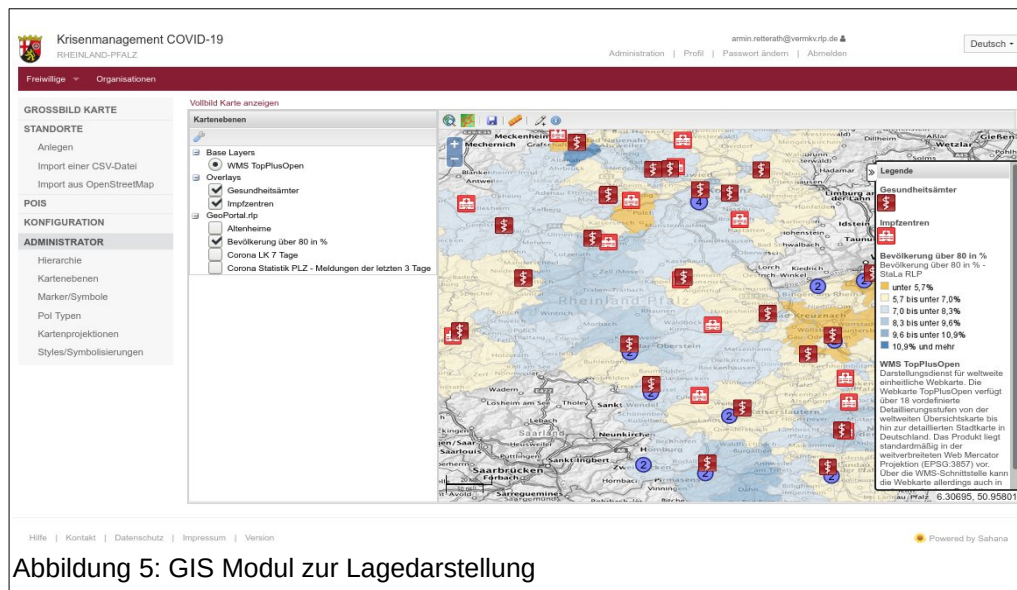


Abbildung 5: GIS Modul zur Lagedarstellung

Phase 5 – Unterstützung bei der Durchführung der COVID-19 Schnelltestkampagne des Landes Rheinland-Pfalz für Schulen und Kitas

Mitte Januar 2021 hatte die Politik entschieden, die Kosten für die Corona-Schnelltests von Lehrern und Erziehern zu übernehmen, um den sicheren Betrieb von Schulen und Kitas zu gewährleisten. Das mit der Durchführung Projektes betraute Landesamt für Soziales, Jugend und Versorgung (LSJV) fragte an, ob man die zentrale Sahana-Instanz des MSAG nicht auch nutzen könne, um diesen Prozess digital zu unterstützen. Nach Abstimmung mit dem MSAG wurde jedoch entschieden eine eigene Instanz [9] aufzusetzen.

Das System sollte dazu verwendet werden, den Lehrern und Erziehern „Gutscheine“ für kostenfreie Schnelltests auszustellen. Die Abrechnung der Tests basiert später auf der dokumentierten Registrierung der „Gutscheine“ durch die Leistungsträger.

Einsatz von FOSS bei der Bewältigung der COVID-19 Krise in RLP

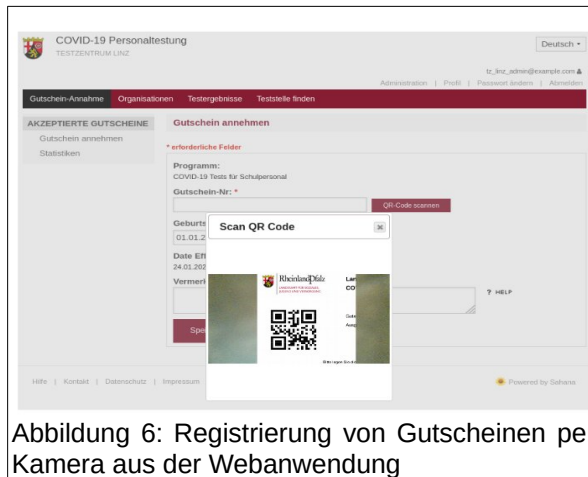


Abbildung 6: Registrierung von Gutscheinen per Kamera aus der Webanwendung

Die Entscheidung für die Abbildung der Gutscheine fiel zugunsten eines Verfahrens auf Basis von QR-Codes. Damit konnte man vermeiden, dass personenbezogene Daten im System abgelegt werden mussten. Die Schul- und Kita-Leitungen bekamen die Möglichkeit digitale Gutscheine auszustellen und die Leistungsträger (ca. 500 Einrichtungen) bekamen die Möglichkeit diese im System zu registrieren (z.B. über Scannen der QR-Codes). Zur Sicherheit, dass ein verlorener ausgedruckter Gutschein nicht unberechtigt verwendet werden konnte, wurde noch ein PIN eingeführt.



Abbildung 7: Beispielgutschein

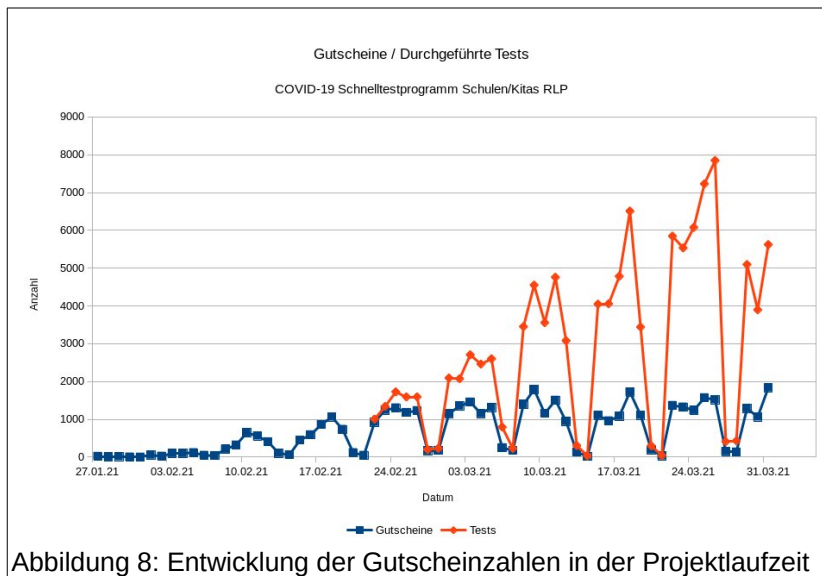
Die erste Herausforderung bestand darin, die Organisationen der möglichen Leistungsempfänger (Schulen, Kitas, Erziehungseinrichtungen, ...) automatisiert ins System zu übernehmen, für jede Organisation eine Administrator Account anzulegen und diese Daten dann den Leitungen zur Verfügung zu stellen (inkl. Dokumentation ;-)).

Ausserdem mussten sich die beteiligten Leistungsträger (DRK, Apotheken, Praxen, ...) selbstständig registrieren und durch das LSJV bestätigt werden.

Ende Januar 2021 konnte die Plattform „COVID-19 Testung“ freigeschaltet werden und es wurden automatisiert 5.000 E-Mails an die Leistungsempfänger verschickt.

Einsatz von FOSS bei der Bewältigung der COVID-19 Krise in RLP

Die Entwicklung der Zahl der durchgeführten Tests kann den folgenden Diagrammen entnommen wer-



den.

Nach der ersten Woche hat sich herausgestellt, dass es sinnvoller ist die Tests vor Ort in den Schulen und Kitas durchzuführen. Es wurde daher die Möglichkeit geschaffen „Sammelgutscheine“ auszustellen (daher orange Kurve ab 23.02.).

Neben der Abbildung des Prozesses wurde auch eine öffentliche Übersicht mit den Teststellen benötigt. Hierzu gibt es sowohl ein Modul in Sahana als auch WMS/WFS Interfaces die die Daten direkt aus der PostGIS lesen.



Beim Projekt „COVID-19 Schnelltests für Lehrer und Erzieher“ wurde bis zum 31.03.2021 insgesamt mehr als 124.000 Schnelltests von fast 400 verschiedenen Teststellen durchgeführt. Die Abrechnung,

Einsatz von FOSS bei der Bewältigung der COVID-19 Krise in RLP

deren Prozess ebenfalls mit Hilfe von Sahana Eden weitestgehend automatisiert wurde, umfasste ein Volumen von 4.8 Mio €.

Phase 6 - Unterstützung bei der Durchführung des Projekts „Schnelltests für Alle“

Nach der Aussage von Bundesgesundheitsminister Spahn im Februar 2021, dass der Bund die Kosten für das neue Programm „Schnelltests für Alle“ übernehmen wird, entfiel der Bedarf für ein Gutscheinsystem und auch für die Abrechnung, die in diesem Folgeprojekt von der Kassenärztlichen Vereinigung Rheinland-Pfalz (KV-RLP) durchgeführt wird.

Die Sahana Eden Instanz des LSJV wird aktuell (Ende April 2021) als Registrierungs- und Qualifizierungsplattform für die Teststellen verwendet, die an dem Projekt „Schnelltests für Alle“ teilnehmen. Organisationen, die eine Teststelle einrichten wollen, registrieren sich für das Programm und stehen dann auf einer „Warteliste“, die vom LSJV nach bestimmten Kriterien überprüft wird. Nach der erfolgreichen Bestätigung wird die Organisation in Sahana freigeschaltet und erhält einen Organisations-Admin Account.

* erforderliche Felder

Benutzerkonto

Vorname: *

Nachname: *

E-Mail: *

Passwort: *

Mindestens 8 Zeichen

Passwort-Kontrolle: *

Teststelle

Name: *

Bundesland: *

Auswahl Bundesland

Adresse:

PLZ:

[Auf Karte platzieren](#)

Telefon:

E-Mail:

Öffnungszeiten:

Programme: *

Wählen Sie die Programme an denen die Teststelle teilnimmt:

☐ COVID-19 Tests für alle

Kommentare:

Datenschutz

Einwilligung:

☐ Einwilligung in Vertragsbedingungen *

- Ich bestätige hiermit, die für die Testung geeigneten Räume selbst organisiert zu haben.

- Ich versichere, ein Hygienekonzept nach den geltenden Regeln etabliert zu haben und dieses bei der Registrierung vorzulegen.

- Ich verpflichte mich gemäß § 7 Abs. 1 Nr. 44a IfSG beim direkten oder indirekten Nachweis von Corona-

Abbildung 10: Registrierungsformular für Teststellen

Einsatz von FOSS bei der Bewältigung der COVID-19 Krise in RLP

Es wurde zusätzlich eine Schnittstelle entwickelt, mit der die KV-RLP online prüfen kann, ob eine Teststelle in Sahana registriert und für das Projekt freigegeben wurde. Die Freigabe dokumentiert die gem. Corona-Testverordnung – TestV vom 08.03.2021 notwendige Beauftragung durch den öffentlichen Gesundheitsdienst (ÖGD) und berechtigt die Organisation zur Abrechnung der Leistung mit der KV-RLP.

Teststellen, die von Hilfsorganisationen im Auftrag kommunaler Gebietskörperschaften betrieben werden, haben die Möglichkeit ihr jeweiliges Verbrauchsmaterial (z.B. FFP2 Masken) kostenfrei beim Land RLP (LSJV) zu bestellen. Hier wird das angepasste Warenwirtschaftsmodul von Sahana eingesetzt.

The screenshot shows the 'COVID-19 Testung' interface for RHEINLAND-PFALZ. The sidebar on the left contains navigation links: BESTELLUNGEN, VERSAND, STATISTIKEN, ARTIKEL, and WARENLAGER. The main area is titled 'Anfragen' and features a form for creating or viewing orders. Below the form is a table listing existing orders.

	Bestellung Nr.	Angefordert am	Liefern an	Details	Versandstatus	Bedarfsdeckung
Öffnen Lieferung registrieren	BANF-3-000011	27.04.2021 22:54	Testzentrum Linz	100 PoC Testkit	-	-
Öffnen Lieferung registrieren	BANF-3-000010	27.04.2021 14:36	Testzentrum Linz	8 Handdesinfektion	Vollständig	Vollständig
Öffnen Lieferung registrieren	BANF-3-000009	26.04.2021 22:49	Testzentrum Linz	10 Handdesinfektion	Vollständig	Vollständig
Öffnen Lieferung registrieren	BANF-3-000008	31.03.2021 10:36	Testzentrum Linz	1000 Schutzkittel, 200 Schutzmaske FFP2, 1000 PoC Testkit	Vollständig	Teil
Öffnen Lieferung registrieren	BANF-3-000007	18.03.2021 17:14	Testzentrum Linz	25 Handdesinfektion	Vollständig	Vollständig
Öffnen Lieferung registrieren	BANF-3-000006	17.03.2021 14:15	Testzentrum Linz	10 Flächendesinfektionsmittel, 200 PoC Testkit	Vollständig	Vollständig
Öffnen Lieferung registrieren	BANF-3-000005	15.03.2021 14:46	Testzentrum Linz	200 Schutzmaske FFP2, 100 Schutzkittel	Vollständig	Vollständig
Öffnen Lieferung registrieren	BANF-3-000004	15.03.2021 14:13	Testzentrum Linz	10 Schutzkittel	Vollständig	Vollständig

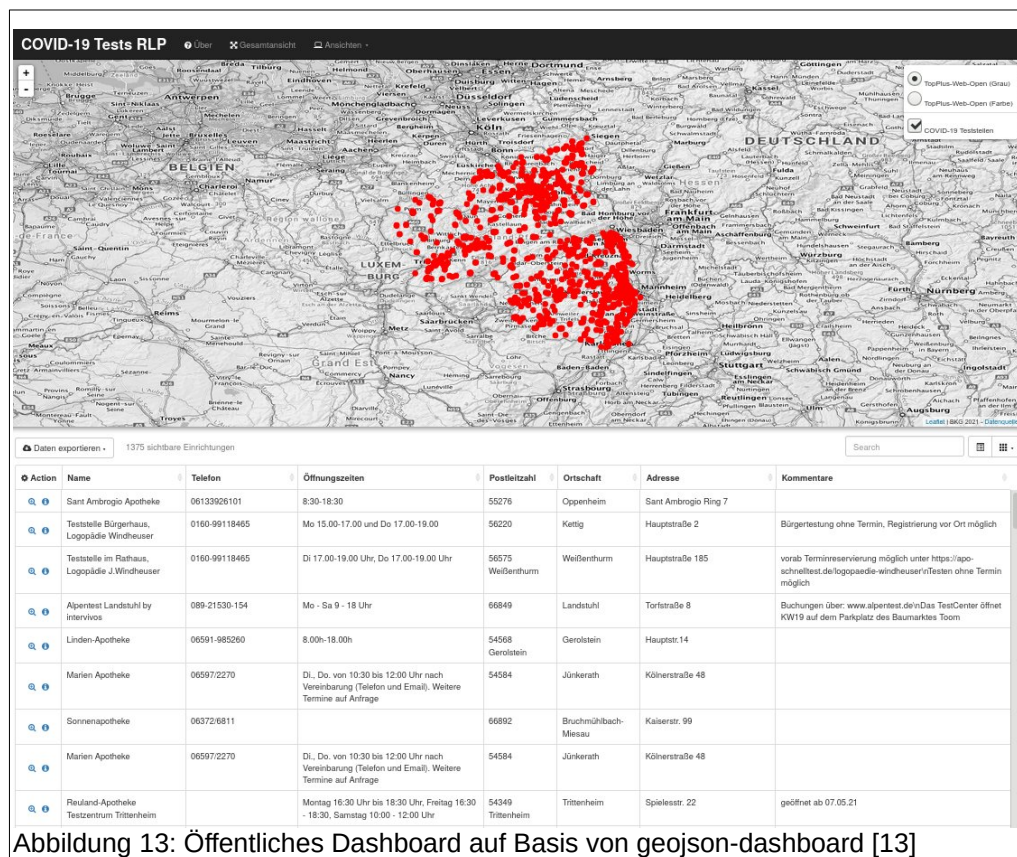
Abbildung 11: Bestellungen der kommunalen Teststellen beim Land (LSJV)

Die Liste aller „Teststellen“ ist öffentlich für alle Bürger zugänglich. Um eine Überlastung der zentralen Instanz zu vermeiden, wird die WFS-Schnittstelle verwendet. Diese ist im GeoPortal.rlp registriert und kann dort über den neue ISO19168-1:2020 (OGC API Features) [10] abgefragt werden.

Die Teststellen stehen als geojson zur Verfügung und werden gecached mit der kartenbasierten Auskunftskomponente bereitgestellt.

Auch hier kommt eine einfache Leaflet-basierte FOSSGIS Komponente zum Einsatz: geojson-dash-board [11].

Einsatz von FOSS bei der Bewältigung der COVID-19 Krise in RLP



Zusammenfassung

Der Einsatz von Freier Open Source Software aus dem FOSSGIS-Umfeld hat sich in Rheinland-Pfalz sehr bewährt. Erfolgsfaktoren sind dabei nicht nur die sehr niedrigen Betriebskosten, sondern insbesondere die hohe Qualität sowie die der hohe Grad an Interoperabilität, den die Komponenten mitbringen. Insbesondere das Sahana Eden Framework konnte hier immense Stärke und Flexibilität beweisen. Sahana bringt eine Fülle von Modulen mit, deren Einsatz sich schon in vielen Krisen bewährt hat. Da die Plattform nicht auch für die Erfassung der positiven Fälle sowie zur Kontaktverfolgung eingesetzt wurde, lässt sich im Hinblick auf die erreichten Ergebnisse bei den anderen Anwendungsfällen wohl verschmerzen ;-).

Der Autor möchte sich an dieser Stelle auch noch mal für die hervorragende Zusammenarbeit aller Beteiligten bedanken. Ein besonderer Dank geht an die Entwickler von AidIQ, die den größten Teil der Entwicklung von Sahana stemmen. Ohne ihre klaren und strukturierten Prozessanalysen, sowie konkreten Software-Umsetzungen, wären die Projekte nie erfolgreich verlaufen – thanx ;-).

Man kann nur hoffen, dass die öffentliche Verwaltung aus der Krise lernt und besonders im Bereich des Katastrophenschutzes mehr auf freie und offene Standards, sowie den Einsatz von FOSS setzt.

Kontakt zum Autor:

Armin Retterath
Zentrale Stelle GDI-RP
Von-Kuhl-Straße 49
56070 Koblenz
armin.retterath@vermkv.rlp.de

Quellen

- [1] <https://github.com/sahana/eden>
- [2] <https://github.com/Leaflet/Leaflet>
- [3] <https://github.com/socib/Leaflet.TimeDimension>
- [4] https://www.deutschlandfunk.de/open-source-software-fuer-die-fluechtlingserfassung.676.de.html?dram:article_id=342384
- [5] <https://rheinneckarblog.de/17/freiwilligenteam-macht-es-bamf-vor/93930.html>
- [6] <https://aidiq.com/>
- [7] <http://www.web2py.com/>
- [8] <https://covid-19-support.msagd.rlp.de>
- [9] <https://covid-19-support.lsjv.rlp.de>
- [10] <https://ogcapi.ogc.org/features/>
- [11] <https://github.com/fulcrumapp/geojson-dashboard>
- [12] https://www.geoportal.rlp.de/spatial-objects/563/collections/lsjv:covid19_common_test_facilities_rlp/items?f=html&limit=2000
- [13] <https://covid-19-support.lsjv.rlp.de/hilfe/covid-19-test-dashboard/>

Mit digitalen Naturschutzinformationen und OpenStreetMap ein nachhaltiges Verhalten in der Natur ermöglichen.

Mit digitalen Naturschutzinformationen und OpenStreetMap ein nachhaltiges Verhalten in der Natur ermöglichen.

DIGITIZE THE PLANET E.V. (SEBASTIAN SARX)



Outdoor-Aktivitäten und das Naturerlebnis liegen im Trend und immer mehr Menschen verbringen ihre Freizeit in der Natur. Verschiedene Online-Tourenportale und Navigationssysteme stellen digitale Informationen zu Routen und Erlebnismöglichkeiten zur Verfügung, zahlreiche User folgen diesen Vorschlägen. Die Daten sind von maßgeblicher Bedeutung für Informationsmedien und damit für das Nutzerverhalten. Dies gilt uneingeschränkt auch für Informationen, die Freizeitverhalten in sensiblen Landschaften und Schutzgebieten zugrunde liegen. Die primär genutzte Datenquelle von Plattformen und Navigationssystemen ist OpenStreetMap.

Dabei liegen den Routen, POIs und Flächen meist keine Informationen zu den Regelungen am Zielort und zur entsprechenden Aktivität vor. Zu Konflikten kommt es, wenn die Informationen zu Freizeitaktivitäten nicht mit bestehenden Regelungen, z. B. zum Naturschutz übereinstimmen. Ein Großteil der Nutzer plant seine Tour digital und nutzt digitale Navigation am Zielort. Ist der Gast einmal vor Ort mit entsprechenden Informationen zu Route und Aktivität ohne Informationen zu Regelungen und Verhalten, ist es zu spät. Dieses Problem gewinnt mit dem Zuwachs des User Generated Content dramatisch an Bedeutung: Zunehmend werden Tourenempfehlungen der Nutzer in den digitalen Medien weiterverbreitet. Diese folgen keinen Regeln, sondern bilden ab, was gefallen hat. Wenn keine Informationen zu Regelkonformität vorliegen, fehlt die Möglichkeit der Kontrolle und des Qualitätsmanagements seitens der User und Plattformen. Nochmals beschleunigt durch die Wirkungen der Corona-Pandemie (Zunahme der Outdoor-Aktivitäten im ländlichen Raum und v. a. in Schutzgebieten) melden die Schutzgebiete in Deutschland eine signifikante Zunahme von Störung der geschützten Gebiete durch Freizeitsuchende. Gerade im Bereich von sensiblen Naturräumen, oftmals auch eben beliebten Orten kann es zu großem Druck auf die Natur und infolge dessen zu Beeinträchtigungen und Konflikten kommen.

Für das zugrunde liegende Problem ist bisher niemand zuständig:

- Die Verwaltungen der Schutzgebiete sind für die Aufbereitung der Flächeninformationen in der Welt von Open Data wenig gerüstet, es gibt dafür bisher kaum Routinen und Anleitungen.
- Die Informationsdienste basieren auf vorhandenen Daten und sind nicht dazu in der Lage, Daten selbst zu generieren. Die bisher nur analog dokumentierten Regeln (z. B. Schutz-

Mit digitalen Naturschutzinformationen und OpenStreetMap ein nachhaltiges Verhalten in der Natur ermöglichen.

gebietsverordnungen und Gesetze) sind für diese Dienste nicht sichtbar und können damit nicht beachtet werden.

- Die Freizeitaktiven sind sich des Problems nicht bewusst: Sie vertrauen auf die Zuverlässigkeit der verfügbaren digitalen Informationen und folgen den empfohlenen Wegen, ohne mögliche Regelverletzungen (und damit Belastungen von Natur und Landschaft) zu vermuten.

Digitize the Planet möchte die Lücke mit offenen Daten füllen und dem Nutzer diese Informationen auf digitalen Kanälen bereitstellen. Der gemeinnützige Verein vertritt Akteure aus den Bereichen Outdoor-Sport, Naturschutz und Tourismus. Der Verein hat sich 5 Kernaufgaben gestellt, mit denen das Ziel einer offenen Datenbank mit Naturschutzinformationen erreicht werden soll.

Die fünf Kernaufgaben des Vereins

1. Regelwerke durchleuchten:

Welche Rahmenbedingungen werden durch EU und administrative Verwaltungseinheiten geschaffen?

2. Format schaffen:

Erarbeitung eines harmonisierten Abbildungsformats für die Regeln/Nutzungsbedingungen zur Freizeitnutzung von Natur- und naturnahen Räumen.

3. Daten schaffen:

Schaffung eines frei nutzbaren Datensatzes für die Regeln/Nutzungsbedingungen zur Freizeitnutzung von Natur- und naturnahen Räumen.

4. Daten verbreiten:

Daten frei nutzbar zur Verfügung stellen und dafür sorgen, dass sie auf den „richtigen“ Kanälen veröffentlicht werden.

5. Aufklären:

Information von Fachwelt und Publikumspressen über unseren Standard, welches Problem wir damit lösen, die verfügbaren Daten und die Möglichkeiten zur Nutzung.

In einem „Proof of Concept“ wurde das erste Datenkonzept mit Partnern aus Naturschutz, Verwaltung und Tourismus erstellt und die Arbeitsschritte konkretisiert sowie die weiteren Herausforderungen identifiziert. In der aktuellen Phase weitet der Verein die Versuchsgebiete auf Europa und die vielen verschiedenen Naturräume sowie Verwaltungsstrukturen aus. Dabei liegt der Fokus auf den Ansprüchen für die Nutzung auf Planungsplattformen und Navigationssystemen. Für diesen Aspekt ist OpenStreetMap ein wichtiger Kanal, mit dem Chancen und Grenzen verbunden sind. Daten von OSM sollen genutzt werden und gleichzeitig sollen Daten auf OSM veröffentlicht werden. Zukünftige Herausforderungen wurden definiert, die es im Austausch mit der OpenStreetMap Community zu lösen gilt:

- Welche Naturschutzinformationen können auf OSM eingetragen werden und bis zu welcher Tiefe können wir in Zukunft die Daten erweitern? Wo liegen die Möglichkeiten und Grenzen der OSM Datenbank?
- Wie können flächendeckend Daten auf OSM integriert werden? Wir wollen Stakeholdern aus Verwaltung und Naturschutz ermöglichen über eine Schnittstelle die Informationen effektiv und einfach einzutragen. Wichtig für eine weitreichende Nutzung ist, dass die Möglichkeit gegeben wird, ohne erweiterte digitale Kompetenzen sowie Infrastruktur im Bereich Geodaten oder OSM Informationen genau und ressourcenschonend einzutragen.
- Wie können wir die Community in OSM sensibilisieren und motivieren einen größeren Fokus auf die wichtigen Naturschutzinformationen zu legen? Wichtige Regeln und Gesetze zum Schutz aller im Bereich KFZ-Verkehr sind umfassend auf OSM zu finden. Können wir ähnliches für die Freizeitnutzung und den Schutz unserer Natur erreichen?

Mit digitalen Naturschutzinformationen und OpenStreetMap ein nachhaltiges Verhalten in der Natur ermöglichen.

Wir laden alle Interessierten und Experten aus der OSM Community ein, in einer weiteren Veranstaltung Lösungsansätze zu finden und zu konkretisieren.

Kontakt zum Autor:

Sebastian Sarx
Digitize the Planet e.V.
Kreuzbergstraße 30
10965 Berlin
sebastian.sarx@digitizetheplanet.org

[1] *Ramm, Ferderik; Topf, Jochen*: OpenStreetMap, Berlin, 2008.

OpenStreetMap in Nationalparken - Chancen, Folgen und Herausforderungen

Anne Seltmann, Julia Zink

Nationalparke sind großräumige Schutzgebiete mit dem Ziel, natürliche Prozesse, Dynamik, Tier- und Pflanzenarten zu schützen, gleichzeitig bieten sie Raum für Forschung, Bildung und Naturerlebnisse. Jedes Jahr besuchen über 50 Millionen Menschen die deutschen Nationalparke, die 0,6 % der Fläche Deutschlands ausmachen. Durch die zunehmende Digitalisierung im Freizeit- und Outdoorbereich sind digitale Karten und Apps zu einem wichtigen Tool für die Planung und Durchführung von Aktivitäten in der Natur – unabhängig vom Schutzstatus – geworden. So ist das Internet zu einer der wichtigsten Informationsquellen zur Vorbereitung von Wanderungen herangewachsen und auch unterwegs nutzt mehr als ein Drittel der Wanderer digitale Daten [1]. Zu den relevantesten digitalen Kartenanwendungen und Tourenportalen im Freizeitbereich zählen beispielsweise OsmAnd, komoot, Outdooractive, Bergfex, maps.me und AllTrails.

Die meisten solcher Anwendungen basieren auf der OpenStreetMap-Datenbank. Dabei werden die OSM-Daten nicht nur zur Darstellung von Karten genutzt, sondern fließen auch in die Berechnung von Routen in Tourenplanern und Navigationsapps ein. Nicht nur kostenfreie Anwendungen, sondern auch kommerzielle Anbieter (z.B. Garmin) greifen auf OSM-Daten zurück. Vor diesem Hintergrund beschäftigen sich immer mehr Schutzgebietsverwaltungen mit OpenStreetMap und sind daran interessiert, zu einer möglichst korrekten und vollständigen Datengrundlage beizutragen.

Eine detailgetreue Abbildung von Schutzgebieten und allen darin befindlichen Wegen (legalen und illegalen) auf OpenStreetMap bietet viele Vorteile. Die zahlreichen, meist OSM-basierten, digitalen Karten und Apps bieten hierbei die Chance, den BesucherInnen eine genaue Navigation zu ermöglichen. Für Schutzgebietsverwaltungen stellt die Bearbeitung der OSM-Datenbank eine Möglichkeit der digitalen Besucherlenkung und Kommunikationsform für Nutzungsregeln dar, da für Routing- und Navigationsfunktionen meist die in OSM vorhandenen Tags berücksichtigt werden.

Die Eintragung jeglicher Wege in Schutzgebieten birgt jedoch auch große Risiken: Einige Wege sind bestimmten Zielgruppen vorbehalten (z.B. reine Wanderwege, die nicht für RadfahrerInnen bestimmt sind), die Begehung mancher Wege ist nur zu bestimmten Zeiträumen erlaubt oder auch ganzjährig zum Schutz der Natur verboten. Allein durch deren Darstellung in verschiedenen Kartenanwendungen werden solche Wege einem breiten Publikum zugänglich gemacht, Nutzungseinschränkungen werden dabei nicht offensichtlich dargestellt. So werden viele Besucher dazu animiert diese zu nutzen, mögliche Einschränkungen oder Verbote vor Ort sind entweder nicht bekannt oder werden häufig ignoriert. Die Folgen dieser zunehmenden Nutzung sind vielfältig und reichen von Trailwidening über Trittschäden an der teils sensiblen Vegetation bis hin zur Zerschneidung der kleinflächigen, störungsfreien Areale in Nationalparken. Damit verbundene erhöhte Störungen wirken sich unter anderem negativ auf den Reproduktionserfolg einiger teils seltener und bedrohter Arten (z.B. Auerhuhn, Wanderfalke) aus.

Ein schwerwiegendes Problem aus Sicht von Schutzgebieten ergibt sich daraus, dass illegale Wege durch die On-the-ground-rule durchaus ihre Daseinsberechtigung in OSM haben, da sie im Gelände existieren und erkennbar sind. Durch deren Darstellung in Karten und Apps werden diese von einer Vielzahl von Menschen genutzt. Dadurch wird das Zuwachsen durch natürliche Sukzession verhindert, da der Boden ständig verdichtet und neu wachsende Pflanzen zertreten werden (bereits 200 Begehungen pro Jahr sind ausreichend, um einen Weg zu erhalten). Ein Beispiel für einen solchen illegalen "Trampelpfad" ist in Abbildung zu sehen. Es ist somit so gut wie unmöglich geworden, stillgelegte Wege der Natur zurückzugeben - ein selbsterhaltender Kreislauf ist entstanden. Die On-the-ground-ru-

OpenStreetMap in Nationalparks - Chancen, Folgen und Herausforderungen

le erschwert die Einhaltung naturschutzrechtlicher Regelungen und den Erhalt störungsfreier Areale zum Schutz bedrohter Tier- und Pflanzenarten.



Abbildung 1: Durch BesucherInnen entstandener Trampelpfad von über 1m Breite mit ausgeprägten Trittschäden und Bodenerosion. Foto: Anne Seltmann, CC-BY

Betretungsrechte und Regeln in Nationalparks können unter Umständen kompliziert sein, daher sind diese für BesucherInnen und Mapper im Gelände eventuell schwer erkennbar. Aufgrund der Größe der Gebiete und des ausgedehnten Wegenetzes (am Beispiel Nationalpark Bayerischer Wald: ca. 242,5 km² Fläche und über 1000 km Wege) ist es nicht möglich, jede Regel im Gelände an jedem Weg explizit zu beschildern. Der Nationalpark Sächsische Schweiz bringt, angeregt durch Gespräche mit der OSM-Community, an dauerhaft gesperrten Wegen Barrieren an, um die Sperrung deutlich zu kennzeichnen (vgl. Abbildung). Das ist allerdings ein sehr aufwändiges Unterfangen und an einigen Stellen aufgrund der örtlichen Gegebenheiten (z.B. Felsuntergrund) nicht möglich. Allein in den 25 km² Kernzone des Nationalparks Sächsische Schweiz gibt es 90 km illegale Wege, von denen durch diese Vorgehensweise in einem Jahr bisher lediglich 15 km als `disused:highway=*` bzw. `abandoned:highway=*` getaggt werden konnten.

Für uns ist es beim Mappen selbstverständlich, dass wir uns an die geltenden Regeln der OSM-Community halten. So werden Wege nicht gelöscht, sondern Lebenszyklus-Tags verwendet, wenn Wege durch das Wegekonzept zur Renaturierung bestimmt wurden. Trotz Einhalten dieser Regel gibt es jedoch immer wieder Fälle, bei denen NutzerInnen diese Arbeit wieder rückgängig machen (sowohl durch Vandalismus im Gelände als auch digital in OSM), uns persönlich angreifen und Gesprächsversuche abblocken.

Gemeinsam mit der Community - in Foren, bei OSM-Stammtischen, bei bilateralen Gesprächen mit Vertretern der Community und in Zusammenarbeit mit dem Ansprechpartner für Behörden bei OSM - konnten wir bereits einige Herausforderungen (z.B. das Darstellen unpassierbarer Wege und zielgerichtetes Rückgängigmachen unserer Änderungen durch einzelne Nutzer) bewältigen. Wir hoffen, durch unseren Beitrag Sensibilität für ein verantwortungsvolles Mappen in Schutzgebieten zu erzeugen, weiterhin den Austausch und die Zusammenarbeit zwischen der OSM-Community und staatli-

OpenStreetMap in Nationalparks - Chancen, Folgen und Herausforderungen

chen Schutzgebietsverwaltungen zu fördern und gemeinsam nach Lösungen zu suchen, wie sich OSM-Regeln und naturschutzrechtliche Regelungen gegenseitig ergänzen können.



Abbildung 2: Mitarbeiter der Nationalparkwacht Sächsische Schweiz baut Geländer, um die Sperrung des Weges im Gelände zu verdeutlichen. Foto: Marko Hänsel, CC-BY

Kontakt zu den Autorinnen:

Anne Seltmann
Staatsbetrieb Sachsenforst
Nationalparkverwaltung Sächsische Schweiz
An der Elbe 4
01814 Bad Schandau
+49 35022 900 612
anne.seltmann@smul.sachsen.de

Julia Zink
Nationalparkverwaltung Bayerischer Wald
Freyunger Straße 2
94481 Grafenau
+49 8552 9600 174
Julia.Zink@npv-bw.bayern.de

Literatur

[1] *BTE Tourismus- und Regionalberatung: Wandertourismus in Deutschland 2018. Ergebnisse einer Befragung zum digitalen Verhalten der Wanderer entlang der Customer Journey*, Berlin, 2018.

LiDAR langsam: Jetzt erst recht?

LiDAR langsam: Jetzt erst recht?

Ich erzähle kurz zum aktuellen Stand der Dinge in meinen Bestrebungen amtliche LiDAR-Daten bzw. das digitale Oberflächenmodell der Stadt Hamburg über das Hamburgische Transparenzgesetz zu befreien.

Ich erzähle kurz zum aktuellen Stand der Dinge in meinen Bestrebungen amtliche LiDAR-Daten bzw. das digitale Oberflächenmodell der Stadt Hamburg über das Hamburgische Transparenzgesetz zu befreien.

Stand Februar 2021 liegt der Fall dank Unterstützung durch den FOSSGIS e.V. beim Rechtsamt der Behörde für Stadtentwicklung und Wohnen. Die dortige Bearbeitung des Falls wurde wegen der aktuellen Belastungslage verzögert und für das Frühjahr 2021 versprochen.

Der dritte Teil der LiDAR-Trilogie nach Leider kein LiDAR? (2019) und Leider immer noch kein LiDAR?.

Johannes Kröger

Soziale Medien, dezentrale Netzwerke und OpenStreetMap

Im Rahmen des openEngiadina und DREAM arbeiten wir an einer Synthese von Sozialen Medien, dezentralen Netzwerken und OpenStreetMap. Wir berichten über die Anwendungszwecke für kleine politische Gemeinden in der Schweiz und lokale Organisationen die sich für gemeinwohlorientierte Lebensweisen einsetzen und präsentieren unsere technischen Ansätze (OpenStreetMap als Linked Data, ActivityPub Protokoll für dezentrale Soziale Netzwerke).



Das Internet schafft neue Möglichkeiten für partizipatives Kreieren und Kurieren von Wissen. Allerdings wird das Medium von grossen, kommerziellen Anbietern dominiert.

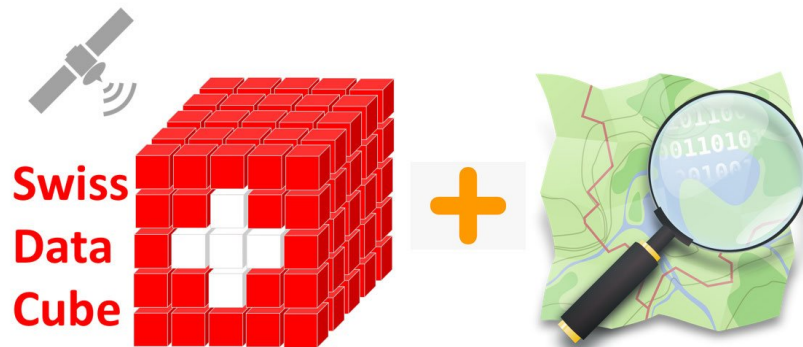
Kleine politische Gemeinden in der Schweiz und lokale Organisationen die sich für gemeinwohlorientierte Lebensweisen einsetzen sind vom Internet abgehängt. Entweder es muss zu Kosten der Datensicherheit und Datensouveränität auf kommerzielle Anbieter vertraut werden oder das zu erfassende Wissen passt schlichtweg nicht in das Schema von Google oder Facebook.

Im Rahmen der openEngiadina und DREAM Projekte erforschen und entwickeln wir Ideen und Technologien um einen nachhaltigen, sicheren und offenen Austausch über lokales Wissen zu ermöglichen. Kartographische Daten spielen dabei eine zentrale Rolle.

In diesem Vortrag möchten wir über die Projekte openEngiadina und DREAM berichten. Wir beschreiben die Anwendungen und unsere technischen Ansätze dafür, zum Beispiel die Verwendung von OpenStreetMap Daten als Linked Data, das Zusammenspiel mit dem ActivityPub Protokoll für dezentrale Soziale Netzwerke und die Anfragesprache Datalog.

pukkamustard

Ist Ground Truth nun unten oder oben? Vergleich von Satellitenbild- mit OpenStreetMap-Daten



Ground Truth bezeichnet direkt am Boden aufgenommene Informationen wodurch Fernerkundungsdaten präziser klassifiziert werden können. OpenStreetMap (OSM) liefert solche Ground-Truth-Daten. Können umgekehrt auch Satellitenbilder helfen, OSM-Daten zu verbessern? Ein Experiment.

In diesem Kurzbeitrag geht es um einen Vergleich von Satellitenbild-mit OpenStreetMap-Daten und zwar mit Hilfe von SwissDataCube und Datenprozessierung mittels Python (Jupyter Notebooks). SwissDataCube (SDC) nutzt die OpenDataCube-Open-Source-Technologie, um Zugang zu allen Landsat- und Sentinel-Daten zu bieten, welche die Schweiz abdecken. Das Ziel des Vergleichs-Experiments ist, am Beispiel von Siedlungsgrenzen die Möglichkeiten der Kombination beider Datenquellen auszuloten.

Ground Truth ("Bodenwirklichkeit", Feldvergleich) bezeichnet in der Fernerkundung und Kartografie direkt am Boden aufgenommene Informationen. Durch Ground-Truth-Daten ist es möglich, Satelliten- und Luftbilder präziser zu klassifizieren und das Ergebnis auf Korrektheit zu prüfen. OpenStreetMap (OSM) liefert solche Ground-Truth-Daten.

Auch umgekehrt kann ein solcher Vergleich Hinweise liefern, wo OSM-Daten - wie beispielsweise Gebäude und Siedlungsgebiete - fehlen! Eine aktuelle Herausforderung ist dabei das Auffinden und der technische Zugriff auf die Fernerkundungs- und OSM-Daten inklusive früherer Datenstände. Da sowohl für Sentinel- als auch für OSM historisierte Daten verfügbar sind, eröffnet dies weitere Analysemöglichkeiten.

Oiza Otaru

Nicola Jordan

20 Jahre PostGIS - dazu 20 hilfreiche Tipps zu PostGIS und Neuigkeiten rund um das Projekt

ASTRID EMDE (WHEREGROUP GMBH BONN)

PostGIS wird in diesem Jahr 20 Jahre alt. Am 31. Mai 2001 kam die PostGIS Version 0.1 heraus. Die Meldung zum initialen Release erfolgte auf diversen Kanälen, so auch auf der frisch eingerichteten PostGIS Yahoo! Groups Liste [1]. Die Nachricht über das neue Projekt verbreitete sich rasant und zog schon bald weite Kreise. Heute ist PostGIS in Kombination mit PostgreSQL die favorisierte Lösung zur Speicherung und Verarbeitung von Geodaten in der Datenbank. PostGIS ist performant, vielseitig, leistungsstark, weltweit verbreitet, wird von zahlreichen Projekten unterstützt und verfügt über diverse angegliederte Projekte [2]. Es gibt also viele Gründe PostGIS in 2021 zu feiern.



Wer PostGIS noch nicht kennt, sollte das Projekt unbedingt kennenlernen. Dies geht sehr einfach über das OSGeoLive-Projekt [3].

Die Entwicklung 2001 erfolgte durch Dave Blasby, Jeff Lounsbury und Paul Ramsey bei der Firma Refractions Research in Kanada. Schon einige Zeit bestand die Idee, Geodaten in der Datenbank vorzuhalten und so unterschiedliche Datenbestände leichter vergleichen zu können [4] [5].

Die Wahl fiel auf PostgreSQL, das schon damals leicht um „Custom Types“ erweitert werden konnte. Auch für die Nutzung der PostgreSQL Indices lagen Beispiele vor, die adaptiert werden konnten.

Schon die Version 0.1 orientierte sich an der Simple Feature Specification for SQL des damals noch OpenGIS Consortiums - heute Open Geospatial Consortium (OGC) [6].

Folgende Funktionalität lag bereits in der initialen Version 0.1 vor:

- Simple Features laut Simple Feature Specification for SQL
 - Point
 - LineString
 - Polygon mit Enklaven
 - MultiPoint
 - MultiLineString
 - MultiPolygon
 - GeometryCollection
- Geometrie als Well-Known Text
- Indizierung über R-Tree oder GiST
- Einfache räumliche Analysen
- PostgreSQL JDBC Erweiterung

Die aktuelle PostgreSQL Version war damals 7.1.x. PostgreSQL lief nur unter Linux. Für Windows wurde eine Cygwin-Umgebung benötigt. Der PostGIS Code lag im CVS (Concurrent Versions System) und auf Sourceforge erfolgte die Veröffentlichung.

20 Jahre PostGIS - dazu 20 hilfreiche Tipps zu PostGIS und Neuigkeiten rund um das Projekt

Das erste Jahr war voller Ereignisse. Das Projekt wurde viel beachtet. Auf der Mailingliste gab es rege Diskussion. Frank Warmedam vom GDAL/OGR-Projekt setzte nach kurzer Zeit eine Unterstützung für PostgreSQL/PostGIS in OGR um. MapServer der bisher Shape und OGR unterstützte, konnte schon im Juli 2001 Daten aus PostGIS über WMS visualisieren.

2002 wurde die JTS Topology Suite veröffentlicht [7]. Diese war in Java programmiert und verfügte über eine Vielzahl an Topology-Operatoren, die in Anlehnung an die OpenGIS Spezifikation umgesetzt wurden. Damit diese Funktionen in PostGIS genutzt werden konnten, musste die Bibliothek nach C++ portiert werden. Dies erfolgte durch einen Studenten der University of Victoria in 2003. Noch im November 2003 wurde dann die PostGIS Version 0.8 mit der ersten Version von GEOS veröffentlicht.

Im fernen Alaska wurde derweil Garry Sherman auf das PostGIS-Projekt aufmerksam und entwickelte ein Visualisierungswerkzeug für PostGIS. Dies veröffentlichte er im Juli 2002 mit dem Namen Quantum GIS [8]. Das heutige QGIS [9] ist heute ein unverzichtbares Werkzeug für den Umgang mit Geodaten und verfügt über zahlreiche Möglichkeiten im Zusammenhang mit PostGIS. Auch andere Desktop-GIS haben schon früh die Unterstützung für PostGIS eingebaut.

Im April 2005 kam die Version 1.0 heraus mit verbesserter Geometriespeicherung und einer signifikanten Performanzverbesserung.

Ab 2006 orientierte sich das Team an der ISO SQL/MM Spezifikation Teil 3. Die Version 1.2 beinhaltete erstmals den Datentyp Curves. Mit der Version 1.3 erhielten die Funktionen den Präfix ST_.

Einen großen Meilenstein stellte 2012 der Major-Release 2.0 dar [10]. Es gab viele Änderungen und alles wurde noch komfortabler. PostGIS verfügt nun über den Typmod Geometry, Rasterdaten werden integriert, KNN-Operatoren eingeführt, PolyhedralSurfaces unterstützt und vieles mehr [10].

2012 ist PostGIS nach einem Inkubationsprozess erfolgreich als OSGeo-Projekt aufgenommen worden und Teil einer globalen Community [11].

Nun in 2021 ist das Projekt bei der Version 3.1.1 angelangt [12]. Es verfügt zusätzlich über Topology-Unterstützung, MVT-Unterstützung, eine Erweiterung für Punktwolken, Integration von SFCGAL-Funktionen für 3D-Operationen, OGR Foreign Data Wrapper und vieles mehr.

PostGIS ist ein beeindruckendes Projekt und verfügt über ein starkes Team mit Paul Ramsey als Chair [13].

Herzlichen Glückwunsch zu 20 Jahren PostGIS!

Kontakt zur Autorin:

Astrid Emde
WhereGroup
Eifelstraße 7 53119 Bonn
+49 228 909038 22
astrid.emde@wheregroup.com

Literatur

- [1] PostGIS Release 0.1 Mail <https://lists.osgeo.org/pipermail/postgis-users/2001-May/000000.html>
- [2] PostGIS Feature und angegliederte Projekte <http://postgis.net/features/>
- [3] OSGeoLive <https://live.osgeo.org/en/index.html>
- [4] Refrations Research PostGIS Geschichte <http://refrations.net/products/postgis/history/>

20 Jahre PostGIS - dazu 20 hilfreiche Tipps zu PostGIS und Neuigkeiten rund um das Projekt

[5] Paul Ramsey 2019 Everthing about PostGIS https://docs.google.com/presentation/d/1a2ruZLRUQ-dH8AWv9Cz6XfPWrs-Vdz_vKQVRNMjqvc0/edit#slide=id.g4c1c926a55_0_252

Video: <https://www.youtube.com/watch?v=g4DgAVCmiDE>

[6] OpenGIS Simple Feature Specification for SQL 1999 <https://www.ogc.org/standards/sfs/>

[7] Martin Davis History of JTS and GEOS <https://lin-ear-th-inking.blogspot.com/2007/06/history-of-jts-and-geos.html>

[8] Founder of QGIS: Gary Sherman <https://www.xyht.com/spatial-itgis/godfather-of-qgis/>

[9] QGIS-Projekt <https://qgis.org/>

[10] Release 2.0 https://postgis.net/docs/release_notes.html#idm43572

[11] PostGIS OSGeo Projekt <https://www.osgeo.org/projects/postgis/>

[12] Release 3.1.1 https://postgis.net/docs/release_notes.html#idm41899

[13] PostGIS PSC https://postgis.net/docs/postgis_introduction.html#psc

GDI im Container

DIPL.-GEOGR. DAVID ARNDT

Bis vor wenigen Jahren wurde eine Virtualisierung vor allem durch virtuelle Maschinen (VMs) bereitgestellt. Dadurch ist eine Isolierung unterschiedlicher Prozesse auf einer Hardware möglich und somit eine Aufteilung der Systemressourcen der darunter liegenden Hardware. Die VMs beinhalten dabei allerdings ein komplettes Betriebssystem, welches auch dauerhaft gewartet werden muss. Immer populärer wird hingegen die Containervirtualisierung. In einem Container sind nur noch die Systembestandteile enthalten, die für den Betrieb des jeweiligen Service benötigt werden.

Virtuelle Maschinen und Container setzen auf unterschiedlichen Cloud-Service Modellen auf. Virtuelle Maschinen setzen auf einer Infrastructure-as-a-Service-Plattform auf, Container brauchen hingegen eine Laufzeitumgebung, die in einer Platform-as-a-Service-Umgebung laufen. Dabei ist die Grundidee eine leichtgewichtige Laufzeitumgebung und die Verteilung auf eine große Anzahl von Servern.

Das Referat Geoinformation und Raumbeobachtung des Regionalverbands Ruhr betreibt seit 2014 eine Serverinfrastruktur zur Bereitstellung von Geobasisdaten und Geofachdaten des Verbandes.

Durchschnittlich gab es bis Ende 2019 ca. 1,5 Mio. Kartenanfragen täglich, dies entspricht ca. 6 TB Datenvolumen im Monat. Dabei werden die Dienste sowohl hausintern als auch von unseren Mitgliedskommunen intensiv genutzt.

Seit Beginn der Corona-Pandemie hat sich die Nutzung der Dienste stark gesteigert, so dass derzeit ca. 3 Mio. Kartenanfragen pro Tag ankommen.

Durch stetig steigende Anforderungen hinsichtlich Verfügbarkeit, Geschwindigkeit und Datenmenge musste das Konzept zur Trennung unterschiedlicher Services über Hardware überdacht werden. Heutzutage ist eine größere Flexibilität bei der Bereitstellung von Diensten nötig.

Dementsprechend wurden alle Komponenten der GDI auf Microservices migriert. Die Isolation der einzelnen Anwendung durch Microservices stellt eine große Vereinfachung in der Entwicklung, Bereitstellung und Wartung dar. Alle Prozesse können über Mechanismen automatisiert werden und erlauben es so, schneller von der Entwicklung zur fertigen Anwendung zu kommen. Zudem sind Bibliotheksinkompatibilitäten zwischen verschiedenen Anwendungen ausgeschlossen, da die Anwendungen in voneinander getrennten Umgebungen laufen. Ein Microservice besteht aus jeweils einem Prozess, womit sich die Komplexität des Gesamtsystems verringert. Ein Austausch zwischen unterschiedlichen Komponenten findet über standardisierte Schnittstellen statt.

Als Basis für die Containerisierung beim Regionalverband Ruhr wird Docker mit Docker-Swarm genutzt. Docker-Swarm ist eine leichtgewichtige Möglichkeit eine Docker-Umgebung in einer Multihost

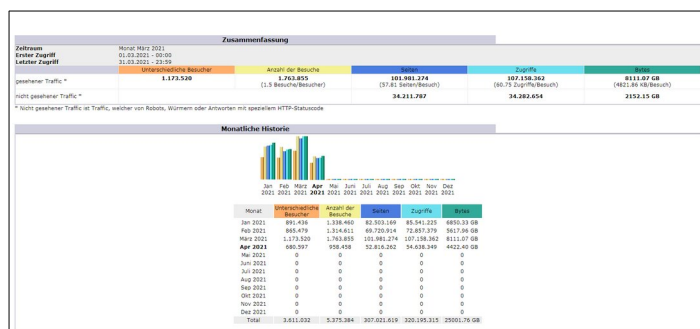


Abb. 1: Zugriffsstatistik

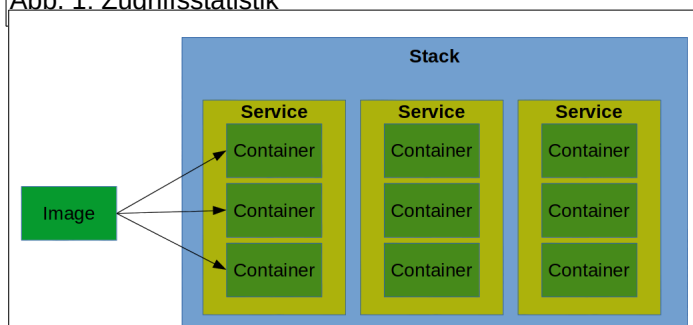


Abb. 2: Begriffs-Definition

GDI im Container

Umgebung laufen zu lassen und bietet alle benötigten Funktionalitäten wie z.B. virtuelle Netzwerke, Skalierung, Überwachung des Service-Status.

Um in der wachsenden Servicelandschaft einen Überblick zu behalten, können die unterschiedlichen Services in Docker-Swarm in Stacks gruppiert werden. Ein Stack bildet eine logische Einheit mit allen Komponenten für die Bereitstellung einer Funktionalität. In einem Stack befinden sich dann beliebig viele Services, die für die Bereitstellung einer Funktionalität nötig sind. Eine WebGIS-Komponente zum Beispiel benötigt einen Webserver, einen Datenbankserver und einen Service, der sich um die Backups kümmert.

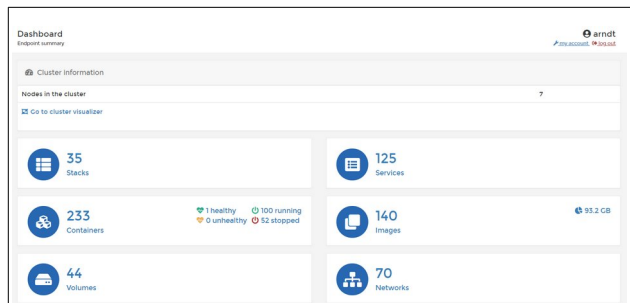


Abb. 3: Portainer Dashboard

Beschrieben wird ein Stack über die in Docker bekannten Compose-Files. Das sind Konfigurationsdateien in YAML-Syntax, die alle Komponenten eines Stacks beschreiben und somit auch zur Dokumentation eines Stacks dienen.

Für die Administration der Docker-Swarm-Umgebung wird auf verschiedene Tools gesetzt. Portainer ist eine grafische Oberfläche zum Administrieren von Stacks, Services, Containern und virtuellen Netzwerken. Integriert ist ein Nutzer/Rollenkonzept, so dass jeder Nutzer nur die Komponenten administrieren kann, für die eine Zuständigkeit besteht.

Da in einer Docker-Swarm Umgebung die Container auf einem beliebigen Host laufen können, ist ein dateibasiertes Logging nicht sinnvoll. Docker-Swarm erlaubt es das Logging auf einen zentralen Logserver zu leiten. Als Lösung wird auf Kibana/Elastic-Search gesetzt. Der Vorteil dieser Lösung ist, dass ein Request über die komplette Laufzeit an einer Stelle verfolgt werden kann.

Die für die Geodateninfrastruktur nötigen Images werden zum größten Teil selbst erstellt und in eine

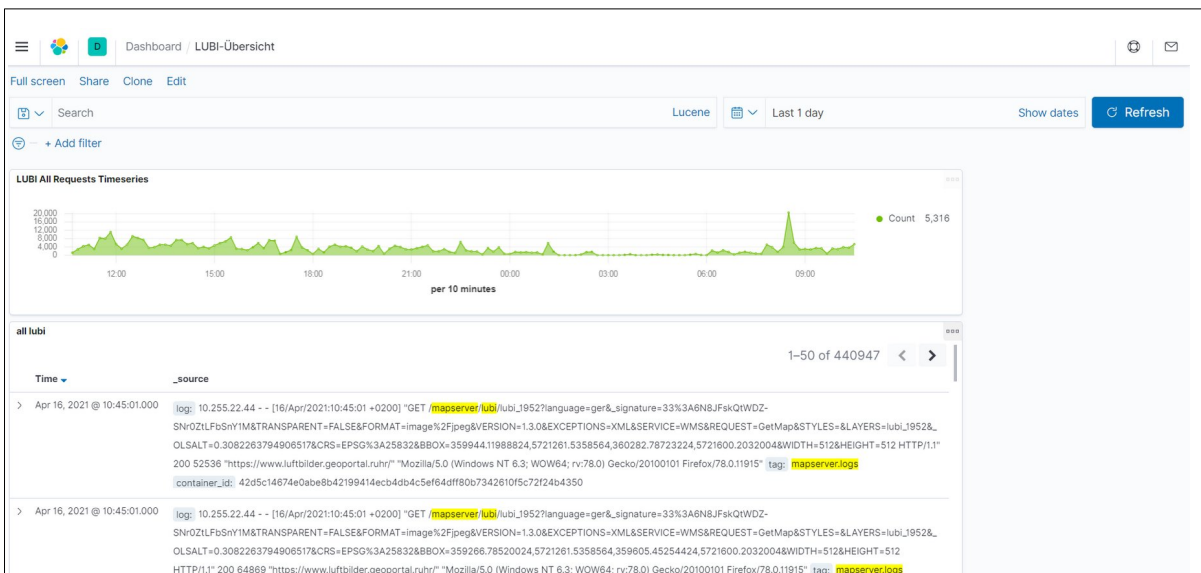


Abb. 4: Logging Dashboard

selbstgehostete Registry geladen. Die Images beruhen dabei auf der Basis des offiziellen Debian Stable Images in Docker Hub und werden dann um die nötigen Softwarekomponenten erweitert. Ein automatisierter Prozess kümmert sich um die regelmäßige Aktualisierung der Images auf das letzte Patchlevel.

GDI im Container

Die Dockerfiles und die Stackfiles landen im zentralen Projektmanagementsystem und werden über eine Versionsverwaltungssoftware gesichert. Die technische Dokumentation jedes Services und Stacks ist darüber sichergestellt und kann nachvollzogen werden.

Die selbstgehostete Registry dient zur Bereitstellung der Images auf allen Hardwareknoten.

Die Komplexität des Gesamtsystems ist in dieser Konfiguration überschaubar. Docker-Swarm ist eine einfach zu erlernende Orchestrierungslösung auf Microservice-Basis und eignet sich sehr gut für den Betrieb und die Entwicklung der Geodateninfrastruktur des Regionalverbandes Ruhr.

Benötigtes Know-How zur Umsetzung:

- Sehr gute Kenntnisse in der Administration von Linux Systemen
- Sehr gute Kenntnisse im Bereich der IT-Sicherheit
- Gute Kenntnisse in der Netzwerkadministration
- Erfahrung in der Prozess Automatisierung

Kontakt zum Autor:

Dipl.-Geogr. David Arndt
Regionalverband Ruhr
Referat Geoinformation und Raumbeobachtung – Team Geodaten-Technik
Kronprinzenstraße 6
45128 Essen
0201/2069-412
arndt@rvr.ruhr

MapLibre – Out of the (Map)box. Wenn Open Source plötzlich Closed Source ist

```
1  mapbox-gl-js v2.0
2
3  Mapbox Web SDK
4
5  Copyright (c) 2020, Mapbox
6
7  All rights reserved.
8
9  Mapbox gl-js version 2.0 or higher ("Mapbox Web SDK") must be used according to
10 the Mapbox Terms of Service. This license allows developers with a current active
11 Mapbox account to use and modify the Mapbox Web SDK. Developers may modify the
12 Mapbox Web SDK code so long as the modifications do not change or interfere with
13 marked portions of the code related to billing, accounting, and anonymized data
14 collection. The Mapbox Web SDK only sends anonymized usage data, which Mapbox uses
15 for fixing bugs and errors, accounting, and generating aggregated anonymized
16 statistics. This license terminates automatically if a user no longer has an
17 active Mapbox account.
```

Für viele unerwartet wurde Ende letzten Jahres die Open-Source-Lizenz des Web-Mapping Frameworks mapbox-gl-js in eine proprietäre Lizenz umgewandelt. Daraufhin wurde binnen weniger Stunden ein international breit getragener Community-Fork unter dem Namen MapLibre aus der Taufe gehoben. Dieser Talk beleuchtet die Hintergründe, zeigt, was MapLibre zu bieten hat, und wie es mit dem Projekt weitergeht.

Nur wenige Jahre brauchte Mapbox, um aus einem Leaflet-Klon und OpenStreetMap-Daten einen Konzern mit mehreren hundert Mitarbeitern zu schaffen. Die Mitarbeiter entwickelten rasant eine beeindruckende Fülle von Open-Source-Projekten, darunter die außerhalb der klassischen GIS-Branche beliebte Web-Mapping-Library mapbox-gl-js.

Trotz vereinzelter Warnzeichen kam es dann kurz vor Weihnachten 2020 für die meisten überraschend zu einem offenen Bruch mit der Open-Source-Strategie: Die am 8. Dezember veröffentlichte Version stand plötzlich unter einer Closed-Source-Lizenz. Keine 24 Stunden später hatte sich eine internationale Gruppe zusammengefunden, die in kürzester Zeit einen breit getragenen Community-Fork einleitete: MapLibre war geboren.

Im ersten Teil des Vortrags wird anhand von Anwendungsbeispielen gezeigt, was mit MapLibre möglich ist und wie es sich zu ähnlichen Open-Source-Libraries abgrenzt. Im zweiten Teil werden die Hintergründe der Mapbox-Entscheidung aus Open-Source-Sicht beleuchtet und aus den ersten Stunden des MapLibre-Projektes berichtet. Abschließend gibt es eine Einschätzung zum aktuellen Stand und den Perspektiven des Projekts.

Marcel Normann

Massenkonvertieren kartographischer Stildokumente mit GeoStyler-CLI

Kartenstile liegen nicht im gewünschten Format vor? Eine manuelle Konvertierung ist allein wegen der Menge der Stile nicht erschwinglich? Wie wäre es mit einem Blick auf die GeoStyler-CLI?

Das Problem: Kartenstile existieren in vielen verschiedenen (zum Teil nicht standardisierten) Formaten und erschweren die Weiterentwicklung hin zu modernen Technologien. Wo möglich, sollte deshalb auf Standards gesetzt werden. Wie kann jedoch die Fülle an Stilformationen gebändigt werden und was wenn der Datenbestand zu groß ist, um diesen manuell in ein standardisiertes Format zu überführen?

Die Lösung: GeoStyler-CLI. In diesem Talk wird live gezeigt, wie mit diesem Tool innerhalb weniger Minuten große Bestände von kartographischen Stildokumenten automatisiert konvertiert werden können.

Christian Mayer

Der MapQonverter - Austausch zwischen ArcMap und QGIS

PETER HEIDELBACH

Wer verschiedene Geoinformationssysteme nutzt, beispielsweise ArcMap und QGIS, weiß, dass der Austausch zwischen diesen beiden Systemen nur bedingt möglich ist. Da ArcMap ein binäres Dateiformat nutzt, lassen sich Projektdateien nicht ohne weiteres auslesen. Hier kommt der MapQonverter ins Spiel.

Der MapQonverter[1] ist eine auf Python basierende ArcMap-Toolbox, die es ermöglicht ein ArcMap-Projekt in einem für QGIS lesbaren Format (.qgs oder .qgz) zu exportieren. Unterstützt wird eine Vielzahl von Symbolisierungen und Layertypen. Eine vollständige Liste findet sich im Github-Repo. Zudem lässt sich ein Layout und verschiedene Data-Frames exportieren.

Außerdem ist es möglich ESRI-Style Dateien zu exportieren, um für ein Projekt einheitliche Symbole, Farbverläufe und Füllungen sowohl in ArcMap als auch in QGIS zu nutzen.

Nicht nur der Export von ArcMap nach QGIS ist möglich, auch der Import von QGIS Projekten in ArcMap kann vollzogen werden. So wird die Interoperabilität zwischen den Programmen noch weiter vergrößert. Diese Entwicklung befindet sich noch in einem Beta-Stadium.

Kontakt zum Autor:

Peter Heidelberg
WhereGroup GmbH
Niederlassung Freiburg, Schwimmbadstraße 2, 79100 Freiburg
peter.heidelberg@wheregroup.com

Link

[1] <https://github.com/WhereGroup/mapqonverter>

Erfahrungen mit QEP 179: Über Paketierung, Abhängigkeiten, ungenutztes technisches Potential sowie die Notwendigkeit von fundierten Programmierkenntnissen bei der Benutzung von GUI-Anwendungen

QGIS-Plugins werden primär in Python geschrieben. Sie sind dadurch Python-Module, jedoch keine Python-Pakete, und fallen dadurch – unnötigerweise – weit hinter die heutigen Möglichkeiten des Python-Ökosystems zurück. QGIS-Plugins könnten mehr, wenn man sie ließe, doch dies wirft einige technische Fragen auf. Meine Analysen und eine mögliche Antwort dokumentierte ich Mai 2020 in QEP 179. Dies ist ein Erfahrungsbericht über die bis heute andauernde Diskussion, die darauf folgte.

Python ist im Moment eine der wohl populärsten Programmiersprachen, die es gibt. Obwohl sie als dynamisch typisierte und interpretierte Sprache durchaus kontrovers diskutiert wird, ist ihre signifikante Rolle nicht mehr zu bestreiten. Das Python-Ökosystem bietet Möglichkeiten für faktisch jede technische und wissenschaftliche Disziplin. Python-Pakete sowie entsprechende Werkzeuge sind für das ganze Spektrum an Computern erhältlich, von Embedded-Systemen bis hin zu HPC-Clustern und diversen Hardware-Beschleunigern. Bemerkenswert ist, dass sich die meisten dieser Pakete relativ einfach bedienen lassen und für sich genommen, einmal installiert, relativ geringe Programmierkenntnisse erfordern. Gleichzeitig lässt sich QGIS durch in Python geschriebene Plugins ebenfalls relativ einfach erweitern. Es drängt sich eine eigentlich naheliegende Frage auf: Kann man QGIS mit alledem, was das heutige Python-Ökosystem anbietet, kombinieren? Die Antwort ist „im Prinzip ja“. Unglücklicherweise ist diese Form der Kombination zumeist äußerst komplex und nur technisch versierten Nutzern mit fundierten Kenntnissen über Software-Entwicklung und - Deployment vorbehalten. Die Weitergabe einer entsprechenden fertigen Lösung an dritte ist in den allermeisten Fällen kaum sinnvoll zu bewerkstelligen. Das Problem ist „einfach“ erklärt: QGIS-Plugins benutzen ihr eigenes, sehr eingeschränktes Paket-Format. Explizite Abhängigkeiten zu anderen „reinen“ Python-Paketen sind nicht vorgesehen, genauso wenig wie eine transparente Integration von relevanten Paket-Managern. Dies ließe sich ändern, wobei eine etwaige Änderung jedoch in jedem Fall sehr weitreichende technische Konsequenzen hätte. Die Diskussion eines etwaigen Lösungsansatzes mit den QGIS-Entwicklern hat gezeigt, dass überraschend viel grundlegender Aufklärungsbedarf über relevante Terminologie und betroffenen Technologien besteht. Der durchaus naheliegende Drang, eine möglichst einfache Lösung zu finden, ohne die jeweiligen Konsequenzen vollkommen zu durchdenken, erschwert den Prozess. Dieser Talk ist ein Erfahrungsbericht und -austausch: Über QEP 179 im Speziellen und Diskussionen in kommerziell orientierten Open-Source-Projekten im Allgemeinen.

Sebastian M. Ernst

Vorteile einer Datenhaltung in PostgreSQL/PostGIS, die ein Shapefile nicht bietet

Auch im Jahr 2021 erfreut sich das Shapefile als Dinosaurier unter den Geodatenformaten zwar einer abnehmenden aber immer noch weiten Verbreitung. Leider betrifft dessen Nutzung erfahrungsgemäß nicht nur dessen Funktion als Austauschformat, sondern auch die Nutzung als permanente Datenhaltung.

Im Rahmen einer Demosession möchte ich aufzeigen, wieso gerade für die Geo-Datenhaltung eine PostgreSQL/PostGIS-Datenbank die deutlich bessere Alternative zum Shapefile ist.

Auch im Jahr 2021 erfreut sich das Shapefile als Dinosaurier unter den Geodatenformaten zwar einer abnehmenden aber immer noch weiten Verbreitung.

Leider betrifft dessen Nutzung erfahrungsgemäß nicht nur dessen Funktion als Austauschformat, sondern auch die Nutzung als permanente Datenhaltung.

Im Rahmen einer Demosession würde ich gerne aufzeigen, wieso gerade für die Geo-Datenhaltung eine PostgreSQL/PostGIS-Datenbank die deutlich bessere Alternative ist zum Shapefile, welches es langsam verdient hat „in die Rente geschickt zu werden“.

Neben der Klärung der Frage, wie aufwändig das Bereitstellen einer hausinternen PostgreSQL/PostGIS Installation ist und wie ein minimales Setup incl. Backup mindestens aussehen sollte würde ich gerne die Vorteile demonstrieren hinsichtlich der Datenhaltung und dem Zusammenspiel mit QGIS.

Folgende Themen sollen beleuchtet werden:

mögliche Bausteine für konsistente Datenhaltung

- Primary-Keys/Foreign Keys
- Check-Constraints

Nutzung von Triggern und Rules bei Datenerfassung -/aktualisierung

Nutzung von Views als Abstraktionsebene

Genaue Festlegung des Geometrietyps (Bsp. Differenzierung zw. Single- und Multigeometrie)

Transparente Informationen zum Encoding und KBS

Speichern von Stilen und Standardstilen zu Layern in der Datenbank. Zentrale Pflege von gemeinsam genutzten Layerstilen und Herausforderung des Aktuell-Haltens der Layerstile in QGIS-Projekten, welche diese Layer enthalten

Triggern von Layer-Refresh und -Actions durch Notify-Signal aus der Datenbank

Ausblick: Möglichkeiten der Historisierung und Versionisierung

Thomas Baumann

Skalierbare Geographische Datenverarbeitung mit Open Source

Wenn man mit geographischen Daten arbeitet kommt man gelegentlich an die Grenzen der eigenen Hardware sobald man mit großen Datenmengen konfrontiert ist. Da kommt man nicht daran vorbei sich mit Big Data zu beschäftigen und die Elefanten im Raum zu konfrontieren. Speziell der blaue Elefant (PostgreSQL) und der gelbe Elefant (Hadoop) sind im Vordergrund, aber auch einige weitere Open Source Projekte können dafür hilfreich sein und werden hier vorgestellt und verglichen.

Dieser Vortrag soll einen Überblick verschaffen über die derzeitigen Open Source Projekte und Tools, die skalierbare Datenverarbeitung ermöglichen, wie zum Beispiel Punkt in Polygon Berechnungen oder Spatial Joins mit großen geographischen Datenmengen.

Die Projekte die verglichen werden, sind PostgreSQL mit PostGIS, Apache Spark mit GeoMesa und Hadoop, Presto, Dask mit GeoPandas und einige weitere mit geographischer Funktionalität. Der Vortrag stellt die einzelnen Projekte vor, vergleicht die Performanz und zeigt mögliche Anwendungsfälle.

Nikolai Janakiev

Aktuelles aus dem deegree-Projekt

Das OSGeo-Projekt deegree stellt seit 20 Jahren umfassende Referenzimplementierungen für OGC-Standards wie WFS, WMS und WMTS bereit. Wenn es um den Aufbau von Geodateninfrastrukturen wie INSPIRE geht, dann können Geodaten in unterschiedlichen Formaten mit den deegree Webservices publiziert werden.

Das Community-Projekt deegree hat gemeinsam mit den Anwendern in den letzten Jahren intensiv an einer verbesserten Unterstützung für die Bereitstellung von Geodaten im Rahmen der INSPIRE-Richtlinie gearbeitet. Jetzt steht mit der Umstellung auf Java 11 und umfangreichen Erweiterungen in der Konfiguration der Schritt auf die nächste Major-Version bevor.

Die Präsentation zeigt die wichtigsten Neuerungen in deegree, eine kurze Übersicht zum Stand der Übernahme des neuen Standards OGC API - Features in deegree und weiteren geplanten neuen Features.

Torsten Friebe

Mapbender 3.2 - besser denn je

OLAF KNOPP

Im Juli 2020 wurde mit dem Release 3.2 eine neue und stark überarbeitete Version von Mapbender veröffentlicht. Sie bildet den aktuellen Zweig der Mapbender-Entwicklung. Für die LTS Version 3.0.8.x gibt es seit dem zweiten Quartal 2021 keine Neuentwicklungen oder Bugfixes mehr.

Mit dem Release 3.2.4 wurde im März 2021 die aktuelle Version veröffentlicht, zur FOSSGIS 2021 ist ein weiteres Release geplant.

Integration von OpenLayers 6

Die neue Mapbender-Version wurde mit OpenLayers 6 ausgestattet, was einige Neuerungen mit sich bringt. So ermöglicht OpenLayers nun wieder ein stufenloses Zoomen und erfüllt damit einen der am häufigsten genannten User-Wünsche. Außerdem lässt sich die Karte jetzt drehen und unterstützt die Darstellung von Multi-Polygonen, -Linien und -Punkten.

Um den Umstieg von älteren Versionen auf das neue Release zu erleichtern, kann bei der Erstellung von Applikationen zwischen OpenLayers 2 und OpenLayers 6 gewählt werden.

Konfigurierbares responsives Verhalten von Elementen und Containern

Mapbender bietet in der neuen Version die Möglichkeit, eine WebGIS-Anwendung im Backend für den Aufruf über verschiedene Endgeräte zu konfigurieren. Inhaltselemente sowie der strukturelle Aufbau der Anwendung passen sich so der Bildschirmauflösung automatisch an.

Über die neue Funktion können Elemente mit sehr komplexen Inhalten bei der mobilen Ansicht ausgeblendet oder durch weniger komplexe Versionen ersetzt werden.

Zur einfachen Konfiguration dieses Verhaltens wurde die Administrationsoberfläche erweitert: Im Bereich „Layouts“ kann nun für jedes Element und für jeden Container (z. B. Header, Footer oder Sidepane) eingestellt werden, auf welchen Endgeräten es angezeigt werden soll. Statt wie bisher getrennte Applikationen für mobile Geräte beziehungsweise Desktop einzurichten und zu pflegen, können nun mit nur wenigen Klicks Anwendungen für das jeweilige Endgerät entsprechend angepasst werden.

Eine neue Funktion zur automatischen Erkennung der Bildschirmauflösung stellt außerdem sicher, dass die Karten auf allen Endgeräten – mobil oder Desktop – gestochen scharf dargestellt werden.

Wiederverwertbare Dienstinstanzen

Die auffälligste Neuerung in der Administration sind die „wiederverwertbaren Dienstinstanzen“. Bisher mussten Kartendienste für jede Anwendung neu konfiguriert werden, was häufig mit einem hohen zeitlichen Aufwand verbunden war. Nun können bereits konfigurierte Dienste – sogenannte Instanzen – in andere Anwendungen eingebunden werden, ohne sie dort neu anpassen zu müssen. Das zentrale Dienste-Repository kann so deutlich komfortabler und schneller verwaltet werden.

Share-Tools

Mit dem Release 3.2.x wurden der komplette Code refactored, Symfony upgegradet sowie diverse Bugs behoben und Aktualisierungen vorgenommen. Diesem „Aufräumen“ fielen einige Funktionen zum Opfer. Dazu gehören beispielsweise das Meeting Point- und das POI-Tool sowie das komplette WMC-Handling (was zugegebenermaßen nie WMC einsetzte). Anstelle dieser Werkzeuge wurden die Share-Tools entwickelt, die aus folgenden Elementen bestehen:

Mapbender 3.2 - besser denn je

- Persistente Kartenzustände: Der aktuelle Zustand der Karte (Maßstab, Mittelpunktkoordinate, Rotation und Projektion) wird mit Variablen vollständig in der URL abgebildet. Dies ist die Grundlage für das Teilen von Ansichten und den parametrisierten Aufruf von Mapbender-Applikationen.
- Share URL: Über einen Button in der Applikation wird die aktuelle URL in die Zwischenablage kopiert. Das Element fungiert so als Einstieg für die bekannte „Teilen-Funktion“ auf Mobilgeräten.
- View Manager: Das Element ermöglicht das Abspeichern des Applikationszustands in eine Datenbank. Gespeichert werden Projektion, aktueller Maßstab und Kartenmittelpunkt, Rotation und Zustände der Layer inklusive Opacity. Über ein Rechteckmanagement sind diese Ansichten für alle, nur den erstellenden User oder bestimmte Anwender und Gruppen aufrufbar.

Anwendungs-Wechsler

Über ein Pulldown-Menü kann nun aus einer Mapbender-Anwendung direkt in eine andere gewechselt werden. Kartenausschnitt, Rotation und Maßstab bleiben dabei erhalten. Unter Beachtung der Berechtigungen lässt sich konfigurieren, welche Applikationen für welche Anwender zur Auswahl stehen.

Neues Backend-Design

Zur Zeit beschäftigt sich das Mapbender-Team mit einem neuen Design-Konzept für Administration und Karten-Frontend. Ziel ist es, Mapbender optisch ansprechender und moderner zu gestalten und für die Nutzung auf mobilen Geräten zu optimieren. Grundsätzlich wird dabei dem Paradigma „Mobile First“ gefolgt.

Kontakt zum Autor:

Olaf Knopp
WhereGroup GmbH
Eifelstraße 7 | 53119 Bonn
+49 (0) 228 / 90 90 38 13
olaf.knopp@wheregroup.com

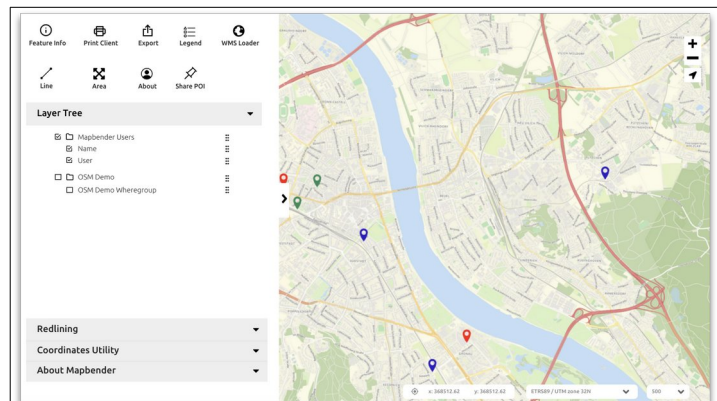


Abb. 1: Entwurf des neuen Mapbender-Frontend

GeoMapFish und QGIS Server

Mit der Kombination von GeoMapFish und QGIS Server ist die Erstellung und Verwaltung eines WebGIS einfacher geworden. Die vielseitigen Symbolisierungs- und Labelingoptionen von QGIS Desktop werden in GeoMapFish ohne weiteres übernommen. Mit dem Access Control QGIS Plugin ist die Rechtekontrolle über die Daten gewährleistet. In diesem Vortrag werden die Funktionen des neuesten Releases 2.6 von GeoMapFish im Zusammenspiel mit QGIS Desktop und QGIS Server gezeigt.



GeoMapFish ist eine Open Source Plattform für die Entwicklung von webbasierten geografischen Informationssystemen (WebGIS). Sie ist reich an Funktionalitäten, hochgradig anpassbar und basiert auf den neuesten Technologien und Standards in diesem Bereich. Sie bietet mehrere Schnittstellen: Desktop, Mobile, Administration und einige weitere für spezielle Zwecke (z.B. Anbindungen an spezifische Fachapplikationen) sowie eine API für die Integration von Karten in Anwendungen Dritter. Basierend auf den OGC-Standards (WMS, WFS) ermöglicht eine GeoMapFish-Anwendung die Übertragung von Geodaten in Form von Diensten für Desktop-Clients oder andere Web-Clients. Aktuell ist die Version 2.6 von GeoMapFish in Vorbereitung.

Als Backend-Kartenserver unterstützt GeoMapFish MapServer, QGIS Server und Geoserver. QGIS Server wurde zuerst mit Version 1.6 integriert. Ab GeoMapFish 2.2 wurden spezifischen QGIS Features benutzt und in letzter Zeit verwenden mehr und mehr GeoMapFish Projekte QGIS Server.

Mit QGIS Server wird die Konfiguration der Layers in QGIS Desktop definiert und in einem QGIS Projekt abgelegt. Die Symbologie wird dann direkt in GeoMapFish übernommen. Die Integration von Daten und die allfällige Änderungen der Konfiguration laufen dadurch reibungslos und profitieren von den vielen Vorteilen und Optionen von QGIS Desktop.

Das Access Control QGIS Plugin erlaubt eine feingranulare Rechtekontrolle über die Daten: gewisse Layer und Attribute können nur für gewisse Benutzer sichtbar sein. Die Konfiguration wird im QGIS Desktop eingestellt. Das Plugin erlaubt die Anbindung von externen Authentifizierungs- und Userverwaltungs-Systemen. Die Rechteverwaltung integriert sich kohärent auch mit anderen Karten-Quellen im Projekt.

Andrea Borghi

Cécile Vuilleumier

Ein Ausblick auf das kommende GRASS GIS 8

GRASS GIS wird in vielen verschiedenen Bereichen zur Geoprozessierung verwendet, z.B. als QGIS Toolbox oder für Big Data Verarbeitung mit hoch leistungsfähigen Computerclustern. In dieser Demo-Session werden die überarbeitete GRASS GIS-eigene graphische Benutzeroberfläche und Beispielanalysen vorgestellt.

GRASS GIS ist ein modular aufgebautes geographisches Analyse-System. Es besteht aus mehreren Hundert Kern-Modulen und einer beständig wachsenden Zahl von Erweiterungs-Modulen (GRASS addons). Diese Module sind in erster Linie eigenständige Programme, die in der Befehlszeile aufgerufen werden. Zusätzlich gibt es eine eigene graphische Benutzeroberfläche, die in dieser Demo-Session vorgestellt werden soll. Wir werden mit der Darstellung von Raster- und Vektorkarten beginnen und dazu die verschiedenen Möglichkeiten zum Styling vorstellen. Zwei verschiedene Rasterkarten können visuell mit dem mapswipe Werkzeug und analytisch mit einem Scatterplot verglichen werden. In der Kartenansicht steht auch 3D-Visualisierung zur Verfügung, mit der z.B. Orthophotos über ein DEM gelegt werden können. Mit GRASS GIS können auch Zeitreihen erstellt, verarbeitet und über die graphische Benutzeroberfläche untersucht werden. Im Ausblick zeigen wir, wie GRASS GIS mit anderer Software wie QGIS, R, usw. zusammenarbeitet.

Markus Neteler

Markus Metz

Vector-Tiles - Stand der Technik 2021

Mapbox Vector Tiles sind schon seit rund 5 Jahren in regem Gebrauch und immer noch kommt jedes Jahr mindestens eine neue Tile-Server Implementation dazu. Dieser Vortrag zeigt einen aktuellen Überblick über die verfügbaren Technologien und deren Einsatz.

Mapbox Vector Tiles sind schon seit rund 5 Jahren in regem Gebrauch und immer noch kommt jedes Jahr mindestens eine neue Tile-Server Implementation dazu. Dieser Vortrag zeigt einen aktuellen Überblick über die verfügbaren Technologien und deren Einsatz.

Pirmin Kalberer

swisstopo VectorTiles



swisstopo erweitert die Bundesgeodateninfrastruktur BGDI, um Vector Tiles für die Visualisierung von Karten und Geodaten im Web zu unterstützen und mit Open Street Map Daten für die weltweite Abdeckung zu kombinieren. Dafür wurde unter anderem eine neue Datenpipeline gebaut und eine neue Basiskarte, die Light

Base Map entwickelt. Der Vortrag bietet Ein- und Ausblicke auf aktuelle und geplante Arbeiten im Projekt.

swisstopo publiziert per 1.3.2021 einen frei verfügbaren VectorTile-Dienst (https://www.geo.admin.ch/de/geo-dienstleistungen/geodienste/darstellungsdienste-webmapping-webgis-anwendungen/vector_tiles_service.html).

Vector Tiles sind Pakete von Geodaten, die in vordefinierte, etwa quadratisch geformte "Kacheln" verpackt und über das Web übertragen werden. Sie können für die Bereitstellung von stilisierten Webkarten verwendet werden und kombinieren die Vorteile von vorgerenderten Rasterkartenkacheln mit Vektorkartendaten. Wie bei WMTS, werden die Daten vom Client als ein Satz von "Kacheln" angefordert, die quadratischen Landbereichen mit einer vordefinierten Grösse und Position entsprechen. Vektorkacheln enthalten Vektordarstellungen von Daten über eine Reihe von Skalen. Im Gegensatz zu Rasterkacheln,

- können sie sich an die Auflösung ihres Anzeigegerätes anpassen
- sind viel kleiner und daher schneller zu laden
- optimiert für mobile Lösungen

Darüber hinaus bieten Vektorkacheln die Möglichkeit, einen eigenen kartografischen Stil zu erstellen, ohne die zugrundeliegenden Daten hosten oder pflegen zu müssen.

swisstopo publizierte basierend auf dieser Technologie ein erstes Produkt: eine Leichte Basiskarte (<https://test.map.geo.admin.ch/>) (zum Zeitpunkt des Vortrags ev. bereits auf map.geo.admin.ch migriert). Diese ist im Gegensatz zu den traditionellen swisstopo Landeskarten inhaltlich und stylistisch leichter gehalten, um einer überlagernden Zusatzthematik genügend Raum zu geben und bietet in Kombination mit OSM-Daten weltweite Abdeckung.

Sebastian Denier

VectorTiles mithilfe von Tilemaker erstellen und nutzen

Überblick des Tilemakers-Projekt, sowie über den Einsatz und Prozessierung bis zur Einbindung der Vectortiles in einer Webkarte.

Das Tilemaker-Projekt bietet eine ressourcenarme Erstellung von VectorTiles aus OpenStreetMap-Daten.

Das Tilemaker-Projekt unterstützt das OpenMapTile-Daten-Schemata und ist somit kompatibel mit mehreren Kartenstilen für die Webanwendungen, die von der MapBox, MapLibre oder der OpenLayers-JavaScript-Bibliothek unterstützt werden, sowie in nativen mobilen SDKs für iOS und Android.

Im Gegensatz zu anderen Ansätzen kommt Tilemaker dabei ohne den Umweg über eine Postgres-Datenbank aus. Dadurch ist dieser Prozessierungsschritt deutlich schneller und lässt sich auch in Cloud-Umgebungen kostengünstig umsetzen. Die erzeugten Kacheln können in verschiedenen Viewern wie Leaflet, OpenLayers, QGIS oder ArcGIS, etc. geöffnet werden.

Dieser Vortrag bietet einen Überblick über den Einsatz und Prozessierung bis zur Einbindung von VectorTiles in einer Webkarte. Weitere Ansätze für das unabhängige Einbinden, sowie das Styling von Vektorkacheln wird ebenfalls vorgestellt.

Marcel Normann

Robert Klemm

FOSS hoch 5 - Praxisbericht eines Serversystems mit OpenSource-GIS

OLIVER SCHMIDT

Ausgangspunkt

Das Landesamt für Vermessung und Geobasisinformation Rheinland-Pfalz (LVermGeo RP) stellt eine Vielzahl seiner Geobasisdaten mittels Web Map Services (WMS), Web Feature Services (WFS) und Web Coverage Services (WCS) bereit. Zur Erstellung dieser Dienste werden verschiedene Quelldaten verwendet, die entweder in einem verwaltungsinternen oder einem bundeseinheitlichen Datenformat vorliegen.

Für die Bereitstellung der Webdienste wurde von Anfang an auf Open-Source-Software gesetzt. Es existiert eine Vielzahl an freien Betriebssystemen und Open-Source-Software im GIS-Bereich, die für eine Umsetzung infrage kommen. Jedoch hat die Praxis gezeigt, dass bestimmte Kombinationen von Softwareprodukten gut harmonisieren. Hierzu sind einige Punkte zu beachten, damit ein performantes und möglichst ausfallsicheres Gesamtsystem zur Verfügung steht. Nachfolgend wird das Serversystem des LVermGeo RP vorgestellt, das einen Großteil der vektordatenbasierten WMS und WFS bereitstellt. Dies schließt neben den AAA-Daten auch die INSPIRE-Daten mit ein, die durch Transformationen erstellt werden. Der Aufbau, Betrieb und die Weiterentwicklung eines solchen Systems werden mittels anschaulicher Beispiele aus der Praxis erläutert.

Hardware und Basistechnologie

Sämtliche Server des Systems (vgl. Abb. 1) werden als virtuelle Maschinen (VMs) betrieben. Dies hat den entscheidenden Vorteil, dass die Server problemlos verschoben werden könnten. Ebenfalls ist eine Erweiterung der Ressourcen einzelner Server unproblematisch, da nur eine virtuelle Aufrüstung stattfindet. Wenn Server skaliert werden müssen, können die jeweiligen VMs einfach auf dem gleichen Teilnetz im Rechenzentrum geklont werden.

Als Betriebssystem kommt ausschließlich Debian GNU/Linux 10 „Buster“ ohne grafische Oberfläche („headless“) zum Einsatz. Debian bietet zahlreiche stabile Programmpakete an, die meist einen aktuellen Stand haben. Hierdurch ergibt sich eine erhöhte Betriebssicherheit, da Inkompatibilitäten zwischen der Fachsoftware und dem Betriebssystem praktisch ausgeschlossen sind. Werden aktuellere Programmversionen benötigt als in dem Standard-Repository vorhanden sind, dann lohnt sich ein Blick in die Paketsuche von Debian ^[1]. Steht eine aktuellere und stabile Programmversion als in Debian „stable“ bereit, so wird diese in den Backports aufgeführt. Diese kann man oftmals bedenkenlos installieren, da die Backports auch hinsichtlich Kompatibilität von der Debian-Community geprüft werden.

Die Bemessung der Hardware-Ressourcen hängt von dem Umfang der Anwendung ab, jedoch sind die Minimalanforderungen für ein solches System nicht ausreichend. Daher muss schon beim Aufbau mit realistischen Ressourcen gerechnet werden. Die hier erwähnten Datenbankserver (db1/db2) wurden mit je 32 GB RAM und 4 CPU-Cores dimensioniert, die MapServer-Knoten (ap1/ap2) hingegen wurden mit 8 GB RAM und 4 CPU-Cores ausgestattet. Beide Prozessierungsserver (pn1/pn2) haben jeweils 8 CPU-Cores und 8 GB bzw. 64 GB RAM. Hingegen sind der Datenbankpooler (dbpool) und der Loadbalancer (balancer) mit 4 GB und 2 CPU-Cores deutlich reduzierter dimensioniert. Da auf diesen Systemen der Datenstrom nur durchgeleitet wird, sind hier keine großen Ansprüche an Hardware-Ressourcen nötig. Für die beiden deegree3-Knoten (ap3/ap4) wurde aufgrund der Java-Umgebung mit 16 GB RAM und 4 CPU-Cores eine etwas großzügigere Ausstattung gewählt. Bei steigenden Ansprüchen an die Systeme müssen diese entsprechend erweitert werden.

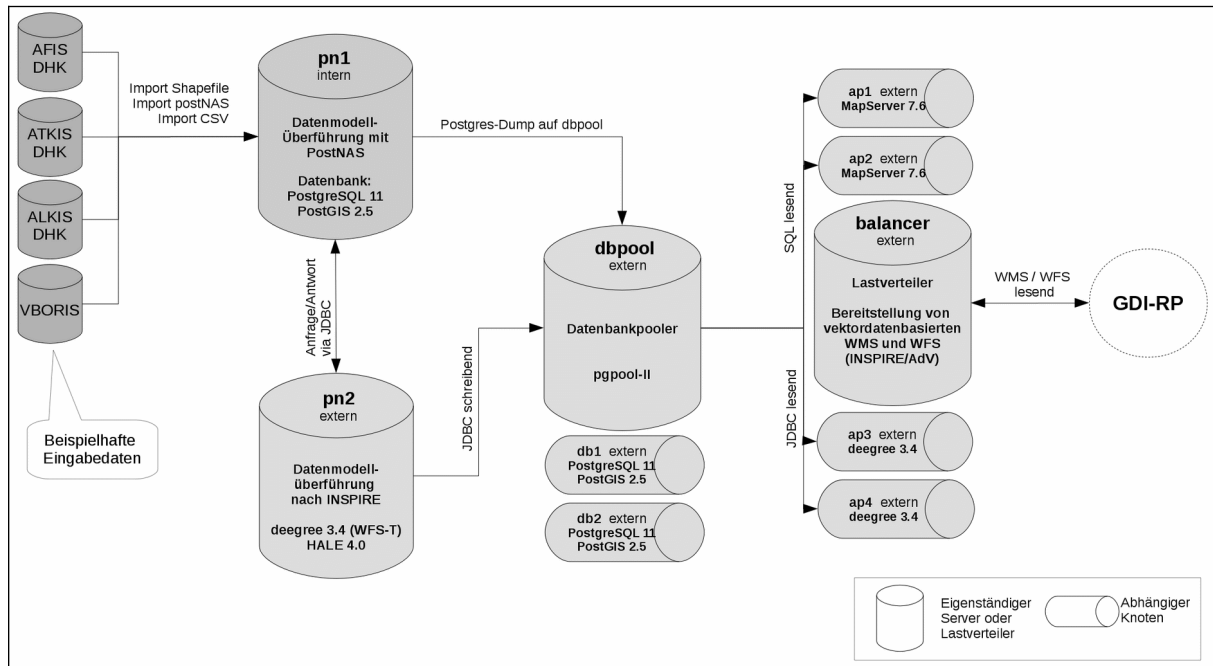


Abbildung 1: Serversystem zur Bereitstellung von vektorbasierten WMS und WFS

Praxistipp: Ergänzend zu der infrastrukturellen Ausfallsicherheit durch doppelte Applikations- und Datenbankserver kommt ein zeitliches „Wartungsfenster“, in dem Sicherheitsupdates des Betriebssystems und der Softwarekomponenten eingespielt werden. Es empfiehlt sich, dieses Wartungsfenster immer an einem bestimmten Tag in der Woche oder im Monat zur gleichen Zeit durchzuführen. Auch wenn die Debian-Updates zu über 95% fehlerfrei eingespielt werden können, kann eine Fehlfunktion nach einem großen Update (insbesondere Kernelversionen) nicht ausgeschlossen werden. Dies muss nicht zwingendermaßen gleich ein gravierendes Problem sein, jedoch kann eine solche Behebung Zeit in Anspruch nehmen. Daher ist es sinnvoll, von den Administratoren Snapshots des Zustandes der VM vor den Updates anlegen zu lassen. Diese können im Havariefall unkompliziert zurückgespielt werden und die Ursache für die Fehlfunktion in Ruhe ergründet werden. Ein möglicher Fehler für ein nicht mehr bootendes Debian kann übrigens in einem fehlerhaften GRUB-Update liegen. Zur Fehlerbeseitigung eines defekten GRUB gibt es diverse Anleitungen im Internet.

Noch weiterer Praxistipp: Bei Distributionsupdates wird dringend empfohlen, vorher einen Klon des zu updatenden Servers anzulegen und auf dieser Referenzmaschine das Update testweise durchzuspielen. Sollte es zu Problemen kommen, kann auch hier ohne ein stehendes Serversystem im Nacken nach der Fehlerquelle geforscht werden. Dieses Verfahren sollte man mit allen Servertypen (pn, ap, db, balancer, dbpool) durchgeführt werden.

The Big Five

Nachfolgend werden die Erfahrungen mit den fünf wichtigsten Softwarekomponenten des Serversystems erläutert.

Als Basis werden drei **PostgreSQL-Server mit der PostGIS-Erweiterung** verwendet, die sämtliche Vektordaten beinhalten. Hierzu finden Importe mittels diverser Techniken statt, die im Fortgang noch

dargelegt werden. Der Vorteil von PostgreSQL mit PostGIS liegt eindeutig in der Performanz von GIS-Operationen und der Skalierbarkeit. So konnte beim gezeigten System die Antwortzeit auf einen komplexeren Filterausdruck von über 20 Sekunden bei 8 GB RAM und normaler SAN-Anbindung auf unter 1,5 Sekunden bei 32 GB RAM und SAS-Festplatten gesenkt werden. Hierzu ist natürlich eine Anpassung der PostgreSQL-Konfiguration (meist in *postgresql.conf*) nötig, da die RAM-Erweiterung sonst keinen nennenswerten Effekt hat. Für diese Anpassungen können zwar sehr viele Parameter per Feinabstimmung angepasst werden, jedoch sind hier letztlich nur die folgenden Einstellungen von größerer Relevanz: `max_connections`, `shared_buffers`, `effective_cache_size`, `maintenance_work_mem`, `checkpoint_completion_target`, `wal_buffers`, `default_statistics_target`, `random_page_cost`, `effective_io_concurrency`, `work_mem`, `min_wal_size` und `max_wal_size`. Diese Parameter sind immer abhängig von der zugrundeliegenden Hardware und dem Anwendungsfall. Eine Datenbank als reiner Speicherort wird andere Einstellungen benötigen als eine Datenbank für Webanwendungen oder für komplexe Berechnungen. Ein guter Anhaltspunkt für die Grundkonfiguration dieser Parameter bietet das Tool pgTune^[2], für das es auch eine Webanwendung gibt^[3]. Hier werden die Szenarien und die relevanten Hardwarekomponenten abgefragt und eine entsprechende Musterkonfiguration aufgrund der eingegebenen Daten erstellt. Im vorgestellten System wird für die beiden Datenbankknoten db1 und db2 das Szenario „Web application“ genutzt, da die meisten Queries eher simple Anfragen sind. Für den internen Prozessierungsserver pn1 hat sich das Szenario „Online Transaction Processing“ als am besten herausgestellt.

Neben dem Import der Daten mit `shp2pgsql` kommt auf dem Server pn1 **GDAL/OGR mit PostNAS-Treiber** zum Einsatz. Zur Nutzung der ALKIS-Daten als Liegenschaftskarte bietet sich der norBIT ALKIS-Import^[4] inklusive der Postprozessierung an. Hierin sind zahlreiche Verarbeitungsskripte enthalten, die die ALKIS-Daten entsprechend verschneiden und aufbereiten. Zudem wurden SQL-Skripte selbst erstellt, die weitere Verarbeitungsschritte durchführen, um die ALKIS-Daten für AdV-konforme WFS oder interne Verfahren bereitzustellen. Ferner werden die AFIS- und ATKIS-Daten auf gleichem Wege importiert, wobei hier natürlich die Prozessierung der Daten wegfällt. Zu Beginn der Nutzung von PostNAS fiel auf, dass manche multiplen Attribute nur mit dem jeweils ersten Objekt importiert wurden. Nach zwei aufeinander aufbauenden Beauftragungen durch das LVerMGeo RP wurde zunächst das komplette ALKIS-Schema abgeleitet und dies im Nachgang zur Erweiterung des PostNAS-Importes verwendet. Somit können nun alle Attributinhalt über entsprechende Arrayfunktionen für die weitere Prozessierung verwendet werden. Eine weitere beauftragte Entwicklung berechnet die Flurstücksabschnitte mitsamt der jeweiligen öffentlich-rechtlichen und sonstigen Festsetzungen auf PostNAS-Basis. Die gesamte Verarbeitung der landesweiten ALKIS-Daten und die anschließende Postprozessierung dauern etwa fünf Tage, wodurch eine Monatsaktualität realisiert werden kann.

Ein Großteil der vektorbasierten WMS und WFS werden mittels **UMN MapServer** abgegeben. Hierfür werden zwei Applikationsserver (ap1, ap2) betrieben, auf denen der Apache Webserver die Anfragen an den MapServer weiterreicht. Der MapServer hat aufgrund seiner Architektur den Vorteil, dass viele Anfragen abgearbeitet werden können und der Fußabdruck im System für jede MapServer-Instanz sehr gering ist. Die Installation vom MapServer erfolgt auf Debian 10 am besten über das Backportrepository, wodurch die Version 7.6 ohne Kompilieren installiert werden kann. Grundsätzlich sollte man bei der Erstellung der Mapfiles sich an die Empfehlungen^[5] halten, die zu einer optimalen Performanz beitragen. Unbedingt befolgen sollte man die Empfehlung, den Pfad der Mapfiles nicht zu veröffentlichen. In unserem Fall wurde die Kaschierung über eine FCGI-Fassade implementiert. Eigentlich selbstverständlich sollte die Erstellung von Indizes auf die jeweiligen Geometriespalten sein, jedoch wird dies gerne mal vergessen, wodurch ein sehr schlechtes Antwortverhalten des MapServer zustande kommt. Zudem tragen Indizes auf den Spalten, die per WFS-Filterausdruck abgefragt werden sollen, zu einer schnelleren Antwort bei Filterung bei. Sind alle Voraussetzungen für eine gute Performanz erfüllt, werden auch komplexere WMS-Mapfiles in kurzer Zeit gerendert. Ein gutes Beispiel hierfür ist der WMS „Karte RP“, eine auf dem BasisDLM basierende und tagesaktuelle Darstellung zahlrei-

cher Informationen. Um auch bei kleinmaßstäbigen Darstellungen eine gute Antwortzeit zu erhalten, sind eine vorherige Generalisierung auf PostGIS-Basis und eine Reduzierung der darzustellenden Objekte sinnvoll. Generell bietet sich eine Vorprozessierung von den Schemata an, die vom MapServer als WMS oder WFS abgegeben werden sollen. Dabei sind Views oder materialisierte Tabellen möglich, wobei zahlreiche Verschneidungen bei Views zu längeren Antwortzeiten der Datenbank führen, im Gegensatz zu einer vorprozessierten Tabelle. Ein WFS mit vielen Objekten sollte pro GetFeature-Request mit dem Parameter "wfs_maxfeatures" auf eine angemessene Anzahl (z.B. 100.000) begrenzt werden, da sonst die Übertragung ggf. von einem serverseitigen Timeout beendet wird. Soll dennoch eine Zählung aller Objekte mittels `resultType=hits` im WFS ermöglicht werden, muss im Mapfile der Parameter `"wfs_maxfeatures_ignore_for_resulttype_hits"` auf "true" gesetzt werden. Hier gilt es jedoch zu beachten, dass die Anfrage des MapServer an die Datenbank nicht performant formuliert wird, da ein SELECT auf alle Spalten der Tabelle durchgeführt wird. Daher müssen die entsprechenden Timeouts des Servers (Webserver, Datenbank usw.) evtl. angepasst werden!

Für die Erstellung der INSPIRE-Datensätze wird **hale studio Open Source** ^[6] über die bash genutzt. Als Eingabedatensätze werden ALKIS und ATKIS (BasisDLM und DLM50) verwendet, die direkt aus der PostGIS-Datenbank auf dem internen Prozessierungsserver pn1 geladen werden. Auf dem externen Server pn2 werden dann die Transformationen nacheinander durchgeführt. Die INSPIRE-Alignments nutzen PostNAS als Quellschema, wodurch eine datenbankbasierte Vorhaltung der Quelldaten ermöglicht wird. Dieses Verfahren nutzt somit sämtliche Vorteile einer Datenbank (Indizes usw.) aus und gleichzeitig wird eine Redundanz an vorzuhaltenden Daten vermieden. Im Anschluss an die Transformation werden die erzeugten GML-Dateien durch hale studio erneut gelesen und per WFS-T an das lokal eingerichtete deegree3 weitergereicht. Um die umfangreichen Datensätze per WFS-T fehlerfrei übertragen zu können, musste einerseits der partitionierte Upload in hale studio aktiviert werden und andererseits eine Änderung im deegree3-Quellcode (s.u.) vorgenommen werden. Durch diesen zweistufigen Aufbau, der die Transformation und den Upload per WFS-T trennt, können etwaige Transformationsfehler bereinigt werden, ohne dass diese in den INSPIRE-Diensten sichtbar würden. Die gesamte Prozessierungskette wird mittels eigenentwickelter Shellskripte gesteuert und kann entweder in Teilen gestartet werden oder als Ganzes. Es werden alle Schritte – von der Anpassung der Alignments auf PostNAS, über die Transformation, die Erstellung der Datenbanken und deegree-FeatureStores, den WFS-T-Upload und die Inbetriebnahme der neuen Datenbanken – durch diese Orchestrierung ausgeführt. Dies ermöglicht eine zeitgesteuerte Ausführung per cronjobs der gesamten INSPIRE-Prozessierungskette.

Die INSPIRE-Dienste (View Services und Download Services) werden mittels **deegree3 Community Edition** realisiert. Bei INSPIRE-Themen mit übergeordneten „Netzwerken“, wie Transport Networks oder Hydrography Network, wird im ersten Teilpaket des Uploads ein Netzwerkelement übertragen, auf das die untergeordneten Elemente referenzieren. In den folgenden Teilen wird dieses übergeordnete Element nicht mehr inkludiert, da es sonst mehrfach in die Datenbank des deegree3 übertragen würde. Allerdings prüft deegree3, ob diese Referenzen lokal aufgelöst werden können. Da dies ab dem zweiten Teil des partitionierten Uploads nicht mehr der Fall ist, wurde für eine einfache Lösung dieses Problems die entsprechende Prüfung in der deegree3-Installation entfernt. Da hale studio bereits bei der Erzeugung der GML-Dateien eine Prüfung durchführt, kann auf die Prüfung durch deegree3 an dieser Stelle verzichtet werden. Zudem wird derzeit eine Entwicklung von deegree3 umgesetzt, die einerseits die Performanz beim Rendering der INSPIRE View Services (WMS) optimieren soll. Ebenfalls in Umsetzung befindet sich die Umstellung von Java 8 nach Java 11 (Open JDK). Beide Entwicklungen sind für die zukünftige Nutzung von deegree3 in diesem Serversystem unumgänglich.

The Little Five

Für das Serversystem mit seinen fünf wichtigsten Komponenten sind auch fünf weitere Softwarepakete unabdingbar, die aber nicht die zentrale Rolle einnehmen. Daher sind sie als „Helfer“ zu sehen, die für sich betrachtet natürlich keine kleinen Softwarepakete sind.

Sämtliche HTTP(S)-Anfragen und –Antworten werden nach außen über **Apache Webserver** abgewickelt. Für die genannten Anwendungsfälle ist die Konfiguration gut dokumentiert und nach einiger Einarbeitungszeit auch gut beherrschbar. In der Praxis hat es sich als vorteilhaft erwiesen, die wichtigsten Konfigurationen getrennt abzulegen und mittels `Include` zentral über die `apache2.conf` einzubinden. Bei WFS-Anfragen können Antworten von MapServer an Apache weitergereicht werden, die bei Ersetzungen in den Ergebnissen (z.B. URL des ap1 durch die öffentliche Internet-URL) eine bestimmte Größe überschreiten. Ein Beispiel hierfür wäre die Landesgrenze von Rheinland-Pfalz auf ALKIS-Basis, die viele Stützpunkte besitzt und somit eine große Antwortlänge aufweist. In der Apache-Konfiguration lässt sich die Antwortlänge bei aktiviertem `Substitute`-Modul wie folgt auf 30 MB erweitern: `<Location "/"> SubstituteMaxLineLength 30m </Location>` Es gilt zu beachten, dass solche Einstellungen bei der Verwendung eines Loadbalancers auch auf den jeweiligen Knoten durchgeführt werden müssen.

Als Webserver für die Java-Anwendung `deegree3` kommt der **Apache Tomcat** zum Einsatz. Da dieser zusammen mit `deegree3` in einem Docker-Container läuft, werden keine Anpassungen vorgenommen. Anfragen an `deegree3` werden durch den vorgeschalteten Apache Webserver an Tomcat weitergeleitet und über Port 80 wieder nach außen beantwortet. Somit ist von außen nicht erkennbar, dass ein Tomcat hier die Daten ausliefert.

Die Verwendung von **Docker** bietet sich im Umfeld von `deegree3` und `GeoHealthCheck` an, da es hierfür fertige Docker-Container gibt. In der Praxis hat es sich als positiv gezeigt, dass die jeweiligen Datenverzeichnisse im Docker als externes Volume eingebunden werden sollten. Somit können die durchgeführten Änderungen in der Konfiguration erhalten werden, wenn der Docker-Container durch eine neuere Version ersetzt wird. Der `deegree3`-Dockercontainer legt an zwei Stellen Logs ab: Einmal im Container selbst und außerhalb des Containers auf dem Host-System. Da es gerade durch Anfragen von außen des `deegree3` zu großen Logdateien kommen kann, wurde an beiden Stellen eine zeitlich begrenzte Log-Speicherung eingebaut. Logs außerhalb des Containers liegen unter `[...]/containers/[id]/`. Standardmäßig richtet Docker keinen Logrotate ein, sodass ein einziges Logfile beachtliche Größen erreichen kann. Um dies zu vermeiden, wurde unter `/etc/logrotate.d/docker-container` ein weiterer Logrotate-Auftrag hinterlegt. Dieser löscht nach 7 Tagen das jeweils älteste Log und komprimiert die Logs von vorgestern. Um die Logs innerhalb des Containers zu sehen, muss man sicher zunächst mit dem Befehl `docker exec -ti deegree3sfr bash` in die Shell vom Docker einloggen. Dort ist man automatisch als `root` unterwegs, also vorsichtig sein! Im Docker liegen sämtliche Logfiles des `deegree3` unter `/usr/local/tomcat/temp/`, die notfalls alle bedenkenlos gelöscht werden können. Ein Cronjob im Docker löscht täglich alle Dateien, die älter als 5 Tage sind, weswegen kein regelmäßiges manuelles Löschen nötig ist.

Zur Hochverfügbarkeit der PostgreSQL-Datenbanken kann **pgPool-II** eingesetzt werden. Dieses nützliche Tool besitzt neben einer automatischen Replizierung und Synchronisierung zwischen den Datenbankservern auch die Möglichkeit zur Einrichtung eines Failover-Clusters. Somit können aktive Datenbanktransfers im Fehlerfall auf einem Server fortgeführt werden und später auf den zwischenzeitlich ausgefallenen Server repliziert werden. Für das vorgestellte Serversystem müssen noch weitere Erfahrungen mit `pgPool-II` gesammelt werden, um umfassende Praxiserfahrungen zu erlangen.

Abschließend sollen natürlich sämtliche Geodatendienste unabhängig überwacht werden. Hierfür bietet sich **GeoHealthCheck** ^[7] an, das auf Python basiert und auch als Docker-Anwendung existiert. Aus systemtechnischen Gründen ist es empfehlenswert, dass `GeoHealthCheck` nicht auf einem der Server läuft, die überwacht werden sollen oder die Teil des Serversystems sind! Fällt der Loadbalancer aus, auf dem auch `GeoHealthCheck` installiert wäre, so würde

man hierüber keinerlei Informationen über die Nichtverfügbarkeit der Dienste bekommen. Daher ist ein unabhängiger Server, wenn möglich auch außerhalb des Rechenzentrums des Serversystems, hierfür vorzuziehen!

Orchestrierung

Ein Großteil der Steuerung übernehmen Shellskripte, die über cronjobs aufeinander abgestimmt ausgeführt werden. Eine übersichtliche Trennung der Importe und der Erstellung der Produkte lässt sich durch die Einrichtung von zwei Betriebssystem- und Datenbankbenutzern erreichen. Durch die Aufteilung in Prozessierungsserver (pn1, pn2), Datenbankserver (db1, db2) und Applikationsserver (ap1 bis ap4) lässt sich die individuelle Ausfallwahrscheinlichkeit zwar senken, jedoch steigt die Komplexität des Gesamtsystems erheblich an. Um dies beherrschbar zu halten, sollten die beiden Prozessierungsserver als „Master-Server“ für die Datenbankserver bzw. die Applikationsserver agieren. Von pn1 und pn2 werden somit auch die Importe in db1 und db2 kontrolliert und gestartet. Möchte man bei Änderungen an bestehenden Verfahren, Datenströmen oder Prozessierungsketten eine sinnvolle Nachvollziehbarkeit erreichen, so sollte eine Versionsverwaltung (z.B. GitLab) verwendet werden.

Ausblick und weitere Entwicklungen

Die zukünftige Einführung der GeoInfoDok 7.1 wird sich auf den PostNAS-Import mit norBIT ALKIS und die INSPIRE-Alignments für hale auswirken. Bereits abgeschlossen ist die Umsetzung der Kompatibilität des PostNAS-Importes für die GID 7.1. ^[8] Wünschenswert wäre für eine tagesaktuelle Bereitstellung von ALKIS-Daten, dass der norBIT ALKIS-Import inklusive der nachträglichen Prozessierung NBA-Differenzdaten einspielen könnte und nur die geänderten, gelöschten oder neuen Daten berücksichtigen würde. Wäre dies möglich, könnte die Laufzeit erheblich verkürzt werden. Weiterhin gibt es derzeit verschiedene Initiativen, die eine Verbesserung der Performanz des deegree3 zum Ziel haben. Hierzu zählen die Umsetzung von Java 8 nach Java 11 (OpenJDK), die Verbesserung des WMS-Rendings und die Nutzung von PROJ 8.0. Zudem gibt es unsererseits Überlegungen, einen AAA-konformen ALKIS-WFS auf Basis der GID 7.1 mit deegree3 zu realisieren. Diese Entwicklung würde für einige Nutzer sicherlich eine Vereinfachung bei der Datenakquise bedeuten. Zudem wird die Inbetriebnahme einer Web Application Firewall geplant, um eine zusätzliche IT-Sicherheitskomponente für das Serversystem nutzen zu können.

Kontakt zum Autor:

Oliver Schmidt
Landesamt für Vermessung und Geobasisinformation Rheinland-Pfalz
Postadresse
Telefon: 0261/492-346
eMail: oliver.schmidt@vermkv.rlp.de

Literatur und Verweise

- [1] <https://www.debian.org/distrib/packages>
- [2] <https://github.com/leopard/pgtune>
- [3] <https://pgtune.leopard.in.ua/>
- [4] <https://github.com/norBIT/alkisimport>

- [5] <https://mapserver.org/optimization/mapfile.html>
- [6] <https://www.wetransform.to/downloads/>
- [7] <https://github.com/geopython/GeoHealthCheck>
- [8] <https://github.com/norBIT/alkisimport/commit/b201f6db87c90e28f16dd46fdbb54ef69ab8af6b>

STAC - ein Ökosystem für raumzeitliche Daten



SpatioTemporal Asset Catalog (STAC) ist ein offener Standard um raumzeitliche Raster- und Vektordaten zu auszutauschen. Das primäre Ziel ist es raumzeitliche Daten einheitlich und einfach auffindbar zu machen. Das Ökosystem besteht bereits aus einer großen Anzahl an offener Software, die bereits auf die Nutzung großer Datenmengen in der Cloud optimiert ist.

SpatioTemporal Asset Catalog (STAC) ist ein offener und einfach erweiterbarer Standard um raumzeitliche Raster- und Vektordaten zu auszutauschen. Das primäre Ziel ist es raumzeitliche Daten einheitlich und einfach auffindbar zu machen. Dies soll insbesondere auch für fachfremde Nutzer über die für diesen Kreis bekannten Suchmaschinen wie Google und Bing möglich sein und proprietäre Portale und APIs ersetzen können. Dafür wurde eine interoperable Spezifikation und API geschaffen, die auch direkt für die Nutzung in der Cloud ausgelegt ist. Eine stetig wachsende Anzahl von offener Software erleichtert die Erstellung, Verwaltung, Veröffentlichung und Nutzung der Daten.

Matthias Mohr

Erfahrungen mit Online-Schulungen freier GIS-Software in Zeiten der Pandemie

JÖRG THOMSEN

Seit einem Jahr führe ich Pandemie-bedingt viele Schulungen online durch. Was sollte beachtet werden, wenn Schulungen mit hohem Praxisanteil online durchgeführt werden? Welche Technik wird eingesetzt? Wie ist können die Übungen effektiv betreut werden?



Seit der FOSSGIS 2020 mussten viele vor Ort-Veranstaltungen ausfallen, darunter auch zahlreiche Schulungen die ich für OSGeo-Software gebe (MapServer, Mapbender, QGIS, PostGIS, GeoServer, Geodaten-Infrastrukturen). Die Schulungen wurden in kurzer Zeit erfolgreich auf Online-Formate umgestellt.

Eine Umstellung oder gar Reduzierung der Inhalte, wie zunächst befürchtet, war nicht nötig. Im Gegenteil können teilweise sogar mehr Inhalte vermittelt werden. Die online-Schulungen finden nur halbtägig statt, aus 2 ganzen Tagen wurden 4 halbe Tage. Das eröffnet für die Teilnehmenden die Möglichkeit nachmittags das Gelernte zu vertiefen, neues auszuprobieren und die aufkommenden Fragen am nächsten Tag zu besprechen.

Für die Online-Formate sind in erster Linie technische und psychologische Hürden zu nehmen, die Schulungsunterlagen und Daten müssen im Vorfeld verteilt werden. Die Teilnehmenden müssen auf ihren Rechnern zusätzlich Software installieren, sei es die geschulte Software selbst oder für den Zugriff auf zentral bereit gestellte Schulungsserver. Und auch die Gruppendynamik ist online eine andere, als wenn alle zusammen in einem Raum sitzen und gemeinsame Kaffee- und Mittagspausen machen.

Kontakt zum Autor:

Jörg Thomsen
WhereGroup GmbH
Bundesallee 23
10717 Berlin
joerg.thomsen@wherogroup.com

<https://foss-academy.com/>

Site Reliability Engineering (SRE) in Geo-Diensten

ARNE SCHUBERT UND STEPHAN HERRITSCH

Site Reliability Engineering (SRE) ist eine von Google entwickelte Methode, um die Stabilität einer Software zu messen und aufrecht zu erhalten, die gleichzeitig ein hohes Maß an Innovation ermöglicht. SRE wird auch von Google selbst für alle in Produktion befindlichen Systeme angewendet.

Das wichtigste Prinzip von SRE ist, dass die Stabilität einer Software an erster Stelle steht. Stabilität definiert SRE aber in seinem zweiten Prinzip ganz klar an den expliziten, aber auch impliziten Erwartungen des Endnutzers. Also nicht das Entwicklerteam, oder das Management definieren Stabilität, sondern der Kunde! Jedoch sollte auch bewusst sein, dass die wenigsten Software Projekte eine Stabilität von 100% erreichen werden. In einem dritten Prinzip geht SRE auch davon aus, dass eine gut geschriebene Software typischerweise keine höhere Stabilität vorweisen wird als 99,9%. Es sollte sich sogar die Frage gestellt werden, ob in einem ohnehin stetig ändernden System, eine hundertprozentige Stabilität ein erstrebenswertes Ziel darstellt, da dies die Entwicklung neuer Features und damit Innovation stark beeinträchtigen kann.

Den Metriken im SRE wird zunächst eine "Critical User Journey (CUJ)" zu Grunde gelegt. In dieser Beschreibung sollten Nutzer, Service und Ziel des Nutzers genau spezifiziert sein. Dagegen können nun die "Service Level Indicators (SLI)" definiert werden. Die Indikatoren sollten stets in einem Verhältnis (also z.B. einer Prozentzahl) angegeben werden können in folgender Form: Anzahl der erfolgreichen Ereignisse geteilt durch die Anzahl der zutreffenden Ereignisse. Natürlich kann und soll hier ein gutes Monitoring seiner Services bereits eine Menge potentieller Indikatoren liefern, jedoch sind nicht alle gut geeignet kritische Ziele zu beschreiben. Ziele, welche in SRE "Service Level Objectives (SLO)" genannt werden, definieren eben genau welcher Wert für ein SLI nicht unterschritten werden darf, damit der Endnutzer die Stabilität der Software nicht in Frage stellt. Typischerweise werden darauf noch "Service Level Agreements (SLA)" vereinbart. Diese sichern den Kunden eine vertraglich Stabilität zu.

Übernehmen wir die Theorie in ein praktisches Beispiel einer GDI und fangen wir mit der CUJ an, gegen die wir dann alle weiteren Eckdaten entwickeln können. Wenn ein Nutzer über einen Browser auf unsere GDI kommt, erwartet dieser eine graphische Karte als Referenz, um leichter Punkte auf dieser verorten zu können. Diese Karte wird von einem Tile-Server bereitgestellt. Damit das Arbeiten mit der Karte flüssig erscheint sollte eine einzelne Tile der Karte innerhalb von 300 ms geladen werden. Unsere SLI ist dann das Verhältnis aus erfolgreichen Events, was in unserem Beispiel alle Requests gegen den Tile-Server sind, die innerhalb von z.B. 300 ms mit Status-Code 200 ausgeliefert werden konnten gegen validen Events, was die Nummer aller Requests gegen den Tile-Server entspricht. Da eine Karte aus vielen Tiles zusammengesetzt ist und das verzögerte Laden von einzelnen Tiles nicht so stark störend ist, definieren wir in unserer SLO das Ziel, dass 99% der Requests erfolgreich sein müssen.

Aus Entwicklersicht kommt nun das "Error Budget" noch als eine wichtige Größe hinzu. Diese ist die Gegenwahrscheinlichkeit zur SLO (also $1 - \text{SLO}$) und beschreibt, wie viel Spielraum für Fehler innerhalb eines Indikators noch existiert. Solange man die gesteckten Ziele erfüllt, können und sollen die Teams selbstorganisiert Features entwickeln. Wurde das Budget jedoch aufgebraucht müssen Konsequenzen folgen und das Team muss dediziert für mehr Stabilität sorgen. Wichtig ist hier auch, dass SRE nicht wirkungsvoll sein wird, wenn keine Konsequenzen aus einem verbrauchten Error Budget gezogen werden.

Site Reliability Engineering (SRE) in Geo-Diensten

Doch wie kann man nun das Error Budget verbrauchen, wenn es lediglich ein Verhältnis darstellt? Würde z.B. der Service ausfallen, wären 100% der Requests fehlerhaft und das Error Budget sofort um ein vielfaches überschritten. Aus diesem Grund wird das Error Budget in entsprechende Intervall-Äquivalente betrachtet. Nehmen wir an, dass pro Tag eine Millionen Requests gegen den Service getätigt werden. Dies würde für unser Beispiel bedeuten, dass an einem Tag 10.000 Requests außerhalb unserer gesteckten Grenzen sein dürften. Für eine Woche wären es entsprechend 70.000 Requests. Die Konsequenzen könnten nun so fixiert werden, dass sobald das Wochen-Budget aufgebraucht wurde, das Team die Feature-Entwicklung zurückstellen muss.

Dadurch ermöglicht SRE eine gute Balance zwischen Agilität für Entwicklung und Stabilität für Operations und stellt somit eine Form der Implementierung des DevOps-Gedanken in Entwicklungs-Teams dar. Darüber Hinaus fördert es die Selbstbestimmtheit der Teams.

Teams werden dazu angehalten Strategien zu entwickeln, die Auswirkungen von Störungen möglichst gering zu halten. Ein wichtiger Schlüssel ist dabei die Zeit zum Beheben, aber auch bis zum Entdecken eines Fehlers so gering wie möglich zu halten. Also die Kosten eines Fehlers in Bezug auf das Error Budgets so gering wie möglich zu halten. Eine weitere Strategie für die Entwicklerteams könnte es sein neue Features nur an einen ausgewählten Nutzerkreis oder nur auf bestimmten Systemen auszutesten, wodurch ein kleinerer Anteil von einer Störung betroffen wäre und somit diese auch nicht so viel Error Budget verbraucht.

Im Endeffekt ist SRE also eine kritische und ehrliche Betrachtung, was für eine Software Stabilität bedeutet, wie man diese Messen kann und den unterschiedlichen Stakeholdern der Software Freiheiten für Innovation einräumt, aber sich auch auf ein klares Mindestmaß an Stabilität, dass der Endnutzer nicht als störend empfindet, festlegt. Es ermutigt Teams die Kosten von Fehlern zu reduzieren, eigenverantwortlich mit Fehlern umzugehen, um seine Freiheiten zu behalten. Des Weiteren bewirkt es eine bessere Fehlerkultur im Unternehmen, hält dazu an möglichst viele Schritte zu automatisieren und schnellere Release Zyklen anzustreben.

Kontakt zum Autor:

Arne Schubert
a.schubert@yagajs.org
Stephan Herritsch
s.herritsch@yagajs.org

Weiterführende Links und Quellen:

Seite von Google über SRE: <https://sre.google/>

Google Book zu SRE: <https://sre.google/sre-book/table-of-contents/>

The Art of SLOs (Slides):

<https://static.googleusercontent.com/media/sre.google/de//static/pdf/art-of-slos-slides.pdf>

Online Kurs zu SRE: <https://www.coursera.org/learn/site-reliability-engineering-slos>

Webassembly: Wo wir sind und wo es hingehen könnte

Webassemblys Potential ist insbesondere für die Verarbeitung und Visualisierung von Geo-Daten noch nicht ausgeschöpft. In diesem Talk möchte ich Ihnen die aktuelle Adaption in FOSSGIS-Projekten, die technische Weiterentwicklung sowie mögliche Zukunftsszenarien von Webassembly aufzeigen.

Webassemblys Potential ist insbesondere für die Verarbeitung und Visualisierung von Geo-Daten noch nicht ausgeschöpft. In diesem Talk möchte ich Ihnen die aktuelle Adaption in FOSSGIS-Projekten, die technische Weiterentwicklung sowie mögliche Zukunftsszenarien von Webassembly aufzeigen.

Michael Schmuki

Neues aus dem OGC - Das Jahr 2021? Was lief schon? Was kommt noch?

Dieser Vortrag gibt ein update zu "heißen" Themen, die in 2021 bearbeitet wurden und was wir in der zweiten Jahreshälfte erwarten können.

Die OGC APIs sind in aller Munde in unserer Community. Z.B. war OGC API Features ein viel diskutiertes Thema im vergangenen Jahr und wir können uns auch auf Neuigkeiten in 2021 freuen. Anfang 2021 fand ein vielbeachteter gemeinsamer Hackathon der OSGeo, des OGC und der ASF (Apache Software Foundation) statt. Was waren die Themen und wie geht es weiter?

Dieser Vortrag gibt ein update zu den "heißen" Themen rund um OGC APIs und offenen Standards des OGC, die Zusammenarbeit der OSGeo und des OGC und wie wir gemeinsam offene Standards weiter entwickeln können.

Athina Trakas

Der neue OGC API Features Standard, wie ist der Stand und was ist mit Open Source Geospatial Software möglich

ASTRID EMDE

Nach einem Jahr möchte ich erneut einen Blick auf den neuen OGC API Features Standard werfen und dabei vor allem auf die vorliegenden Implementierungen auf der FOSSGIS-Welt eingehen.

Dieser Vortrag stellt die OGC API Familie vor (<https://ogcapi.org/>) und geht dabei vor allem auf den OGC API Features Standard ein.

Die neuen Standards der OGC API Familie verfolgen den RESTful-Ansatz.

OGC API Features dient zur Abfrage von Objekten über das Web und ist quasi der Nachfolger von OGC WFS. Die Umsetzung erfolgt in mehreren Phasen. Hier lohnt sich ein Blick auf den aktuellen Stand.

Unmittelbar mit der Definition der Standards wurden und werden bereits Referenzimplementierungen programmiert. Hierbei sind diverse Projekte aus dem Umfeld der OSGeo führend.

So gibt es bereits Implementierungen von OGC API Features Part 1 von GeoServer, QGIS Server, py-geoapi, pg_featureserv & mehr.

Was bietet der neue Standard und wie kann er verwendet werden?

Dies soll am Beispiel von Open-Source-Lösungen an Beispielen aufgezeigt werden.

Kontakt zur Autorin:

Astrid Emde
OSGeo - Open Source Geospatial Foundation
astrid_emde@osgeo.org

WhereGroup
Eifelstraße 7 53119 Bonn
+49 228 909038 22
astrid.emde@wheregroup.com

<https://ogcapi.org/>

<https://ogcapi.org/apiroadmap.html>

<https://www.ogc.org/resource/products/byspec/?specid=1022>

https://access.crunchydata.com/documentation/pg_featureserv/latest/

<https://pygeoapi.io/>

Der neue OGC API Features Standard, wie ist der Stand und was ist mit Open Source Geospatial Software möglich

EASYDAB (Earth System Data Branding) - eine neue Kennzeichnung von qualitativ hochwertigen Geodaten

EASYDAB ist ein derzeit entwickeltes Branding für offene, FAIRe und qualitätsgeprüfte Daten aus den Erdsystemwissenschaften, die mit DataCite DOIs in einem Repository veröffentlicht werden. Mit EASYDAB gekennzeichnete Daten sind disziplinspezifisch standardisiert und haben umfangreiche Metadateninformationen. Dies wird vom veröffentlichenden Repository mithilfe von Prüfungen sichergestellt und die zugehörigen Landing Pages werden mit dem EASYDAB-Logo gekennzeichnet.

In den Geowissenschaften gibt es eine Vielzahl von wissenschaftlichen Aktivitäten, die sich mit den zahlreichen wichtigen Aspekten eines guten Forschungsdaten Managements befassen. Was bisher jedoch nur wenig untersucht und behandelt wurde, sind die einfache Auffindbarkeit und Wiederverwendung von qualitätsgeprüften Forschungsdaten. Dieser Aspekt wird durch das Earth System Data Branding (EASYDAB, <https://www.easydab.de>) aufgegriffen.

EASYDAB ist ein derzeit entwickeltes Branding für offene, FAIRe (Wilkinson et al, 2016) und qualitätsgeprüfte Daten aus den Erdsystemwissenschaften, die mit DataCite DOIs in einem Repository veröffentlicht werden. Mit EASYDAB gekennzeichnete Daten sind disziplinspezifisch standardisiert und haben umfangreiche Metadateninformationen. Dies wird vom veröffentlichenden Repository mithilfe von Prüfungen sichergestellt und die zugehörigen Landing Pages werden mit dem EASYDAB-Logo gekennzeichnet. Dadurch können Nachnutzer der Daten einfacher qualitative hochwertige Daten finden und Repositorien zeigen, dass sie geprüfte Daten mit hoher Qualität veröffentlichen.

Der erste Standard, der die EASYDAB-Bedingungen erfüllt, ist der ATMODAT-Standard für Atmosphärenmodelldaten (Ganske et al., 2020). Er enthält konkrete Empfehlungen und Anforderungen in Bezug auf die Prüfung, Veröffentlichung und verbesserte FAIRness von qualitätsgesicherten Daten. Die Spezifikationen umfassen u. a. reichhaltige Metadaten mit kontrollierten Vokabularen, strukturierte Landing Pages, Dateiformate (netCDF) und die Struktur innerhalb der Dateien. Menschen- und maschinenlesbare Landing Pages sind ein weiteres Kernelement, genauso wie Checklisten für Datenproduzenten und Datenkuratoren.

Für den weiteren Ausbau von qualitätsgesicherten Datenpublikationen soll EASYDAB mit weiteren Standards verwendet werden, wobei die Geowissenschaften hier ein großes Potential darstellen. Das Ziel ist, mit EASYDAB die Wiederverwendung von FAIRen, offenen, geprüften und hochwertigen Daten zu verbessern.

Literatur:

Wilkinson et al., 2016: The FAIR Guiding Principles for scientific data management and stewardship. Sci Data 3, 160018. <https://doi.org/10.1038/sdata.2016.18>

Ganske et al., 2020: ATMODAT Standard (v3.0). https://doi.org/10.35095/WDCC/atmodat_standard_en_v3_0

Anette Ganske

Baremaps: Kontextuelles Mapping in unerforschten Gebieten

Baremaps ist ein Open-Source-Framework zur Erstellung kontextueller Karten: d. h. Karten, die sich an den Kontext des Benutzers anpassen. Baremaps erstellt kontextuelle Karten durch die Verarbeitung von Geodaten in großem Maßstab und durch die Veröffentlichung von Vektor-Kacheln mittels Cloud-Technologie.

Heute wird die Veröffentlichung kontextbezogener Karten den Teams von Softwareingenieuren und Datenwissenschaftlern überlassen. Infolgedessen betreiben nur wenige Unternehmen (z. B. Google, Mapbox, Esri, Klokant) Pipelines zur Veröffentlichung solcher Karten und adressieren dabei die Bedürfnisse des Massenmarktes (z. B. Suche, Transport, Spiel), zum Nachteil der Nischenanbieter (z. B. Tiefbauunternehmen, Katastervermessungsbüros). Baremaps geht dieses Problem der Zugänglichkeit an, indem es DevOps-Praktiken für den Bereich der Geomatik nutzt und es Ingenieuren ermöglicht, Datenverarbeitungspipelines auf ihrem Desktop-Computer zu prototypisieren und sie mühelos in der Cloud einzusetzen. Wir glauben, dass die weitgehende Verfügbarkeit der kontextuellen Kartenabstraktion entscheidend ist, um Innovationen in unerforschten Bereichen (z. B. Stadtplanung) zu fördern. Baremaps ist unter der sehr freizügigen Apache 2 Lizenz veröffentlicht.

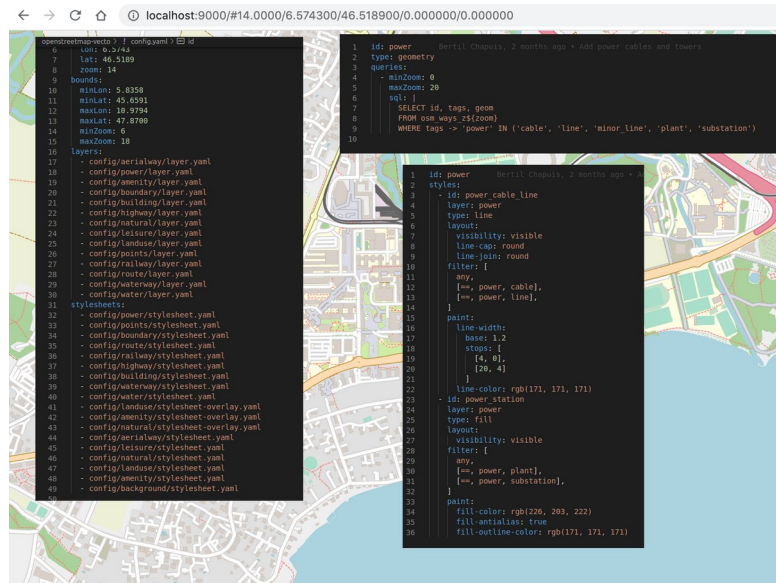
Tiling ist eine grundlegende Komponente des Web-Mappings, die es ermöglicht, nur die Daten herunterzuladen, die im Viewport des Benutzers angezeigt werden. Im Gegensatz zu Bitmaps werden Vektorkacheln auf dem Gerät des Nutzers gerendert, was es ermöglicht, dass sich Karten an den Kontext der Nutzer anpassen (z.B. Strassenkarten vs. öv-Karten) und sich dynamisch anpassen, wenn sich der Kontext ändert (z.B. beim Betreten eines Gebäudes). Edge Computing bringt die Daten physisch näher an den Benutzer heran, was die Latenzzeit reduziert und bei einer kontextbezogenen Karte die Benutzerfreundlichkeit verbessert.

Baremaps baut auf OpenStreetMap auf und verfügt über einen effizienten und parallelisierten Datenimporter von pbf nach postgis, der es erlaubt, benutzerdefinierte Transformationen (z.B. Geometrievereinfachungen) direkt beim Import der Daten in postgis hinzuzufügen.

Ein Webkartograph kann mit Baremaps sowohl seinen Stil als auch den Dateninhalt gleichzeitig entwickeln, indem er den eingebetteten Entwicklungsserver nutzt, um ein optimales Endergebnis zu erzielen.

Am Ende werden nur die Daten, die der Stil benötigt, verwendet und in die generierten Kacheln importiert. Die endgültigen Kacheln können vom Tiling-Prozessor nativ in Cloud-Umgebungen exportiert werden.

In dieser Demo zeigen wir, wie Baremaps für folgende Funktionen verwendet werden kann, um:



Baremaps: Kontextuelles Mapping in unerforschten Gebieten

- Dynamische Bearbeitung von Daten und Stil (Anpassung)
- Parallelisierter Import
- Parallelisierter Export
- Kacheln in der Cloud veröffentlichen (s3://)
- Kacheln in einem produktiven Setup zu verwenden

Wir zeigen dabei als Beispiel openstreetmap-vecto, das auf Baremaps basiert.

Github Link: <https://github.com/baremaps/baremaps>

Andrea Borghi

Geo Engine: Explorative Datenanalyse mit raum-zeitlicher Workflowverarbeitung

CHRISTIAN BEILSCHMIDT, JOHANNES DRÖNNER, MICHAEL MATTIG

Die Geo Engine ist eine Plattform, die die Integration und die effiziente Verarbeitung raum-zeitlicher Daten bündelt und neueste Visualisierungs- und Analysemethoden intuitiv erschließt. Dies ermöglicht es Forschungsgruppen und Unternehmen bisher ungenutzte Potenziale zu erschließen. Perspektivisch bieten wir die Plattform als cloud-basierten Dienst an und integrieren derzeit eine Deep-Learning-Anbindung.

Die Entwicklung basiert auf Forschungsergebnissen aus dem Bereich raum-zeitlicher Datenverarbeitung, die gebündelt im sog. VAT-System [1] zusammengefasst wurden, wobei VAT für Visualisierung, Analyse und Transformation steht. Es wird bereits in wissenschaftlichen Projekten eingesetzt, die sich auf Umwelt- und Biodiversitätsmonitoring konzentrieren, wo es native Zeitreihenverarbeitung, die Kombination von Raster- und Vektordaten und eine Benutzeroberfläche bietet, die verknüpfte Ansichten zwischen Karten, Tabellen und Plots bietet [2]. Darüber hinaus ermöglicht unsere Technologie die Erstellung maßgeschneiderter Apps, zum Beispiel für web-basierten Lernplattformen und Projektportale.

Der Vortrag gibt einen Überblick über das System und dessen Eigenschaften. Es wird auf das Prozessierungs-Backend eingegangen, dass eine Kachel-basierte (bei Raster-Daten) bzw. Chunk-basierte (bei Vektor-Daten) Prozessierung ermöglicht und dabei jeweils die Zeitsemantik der Daten berücksichtigt. Dies wird jeweils an praktischen Beispielszenarien illustriert. Zudem wird gezeigt, dass sich die Geo Engine auch als Portal-Software eignet, um Daten über standardisierte und offene Protokolle bereitzustellen. Zum Abschluss wird ein Ausblick über zukünftige Entwicklungsziele gegeben, die insbesondere die Integration von Deep Learning umfassen.

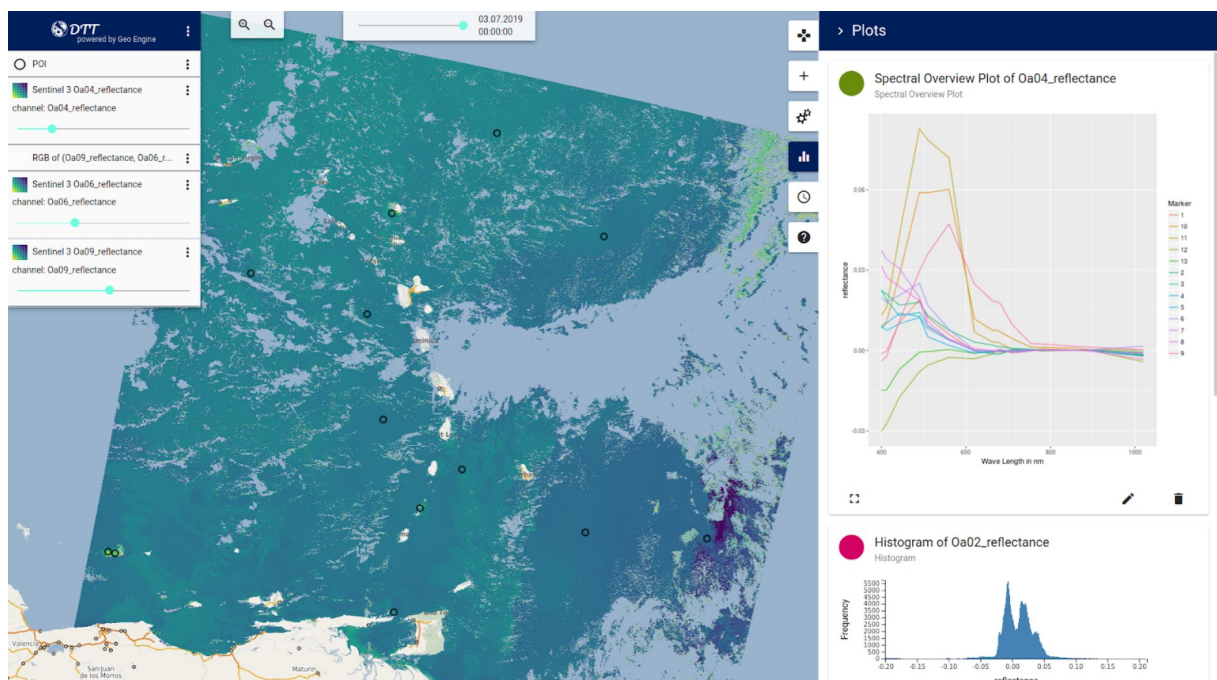


Abbildung 1: Beispielanwendung im User-Interface der Geo Engine, in der Raster- und Vektordaten gemeinsam prozessiert und auf einer Karte sowie über Plots visualisiert werden.

Kontakt zum Autor:

Geo Engine: Explorative Datenanalyse mit raum-zeitlicher Workflowverarbeitung

Dr. Christian Beilschmidt
Philipps-Universität Marburg
Hans-Meerwein-Straße 6, 35043 Marburg
christian.beilschmidt@geoengine.de

Literatur

- [1] Beilschmidt, C., Drönner, J., Mattig, M. et al.: VAT: A Scientific Toolbox for Interactive Geodata Exploration. Datenbank Spektrum 17, 233–243 (2017). <https://doi.org/10.1007/s13222-017-0266-5>
- [2] Geodaten explorieren und kombinieren – Der Elefanten-Use-Case. <https://www.geoengine.de/geodaten-explorieren-und-kombinieren-der-elefanten-use-case/> (29.04.21)

Wer hat an der Uhr gedreht? - Zeitbasierte Geodaten modellieren, abfragen und visualisieren

Der Vortrag gibt einen Überblick wie zeitbasierte Geodaten gespeichert, abgefragt und visualisiert werden können. Speziell wird dabei auf den Web Map Service (WMS TIME) und den Web Feature Service (WFS) eingegangen. Es wird beschrieben wie ein zeitbasierter Kartendienst mit GeoServer und MapServer erstellt werden kann. Die Darstellung wird anhand von QGIS und der auf OpenLayers basierenden Webanwendung Wegue demonstriert.

Ein Großteil aller Geodaten enthält neben der räumlichen Information auch eine zeitliche Komponente. Beispielsweise werden Luftbildern an bestimmten Tagen aufgenommen oder ein Datensatz von Gebäuden ist mit dem entsprechenden Baujahr versehen. Dieser Vortrag gibt einen Überblick welche gängigen Möglichkeiten es gibt zeitliche Information abzuspeichern, wie das konkret für Vektor und Rasterdaten umgesetzt wird und welche Schwierigkeiten und Herausforderung in der Umsetzung zu erwarten sind.

Für die zeitlichen Abfrage von Geodaten stellen die verschiedenen Dienste des OGC (Open Geospatial Consortium) Schnittstellen zu Verfügung. Web Map Services (WMS) können mittels des standardisierten WMS-TIME zeitbasiert abgefragt werden. Dieser lässt neben einzelnen Zeitpunkten auch Zeiträume zu. Beim Web Feature Service (WFS) werden für zeitliche Abfrage der FILTER-Parameter verwendet. Dieser akzeptiert die standard-konformen OGC-Filter.

Softwareseitig werden die zeitbasierten OGC-Dienste im FOSSGIS-Umfeld breit unterstützt. Als Kartenserver eignen sich GeoServer und MapServer. Auf dem Desktop bietet QGIS seit Version 3.14 umfangreiche WMS-Time Funktionen. In Web-Browsern können sowohl OpenLayers als auch Leaflet für zeitliche Abfragen genutzt werden.

Der Vortrag schließt mit einer Demonstration, wie im Projekt SAUBER zeitbasierte Dienste mittels GeoServer erstellt werden. Außerdem zeigen wir unseren auf Wegue (Vue.js und OpenLayers) basierenden Web-Client mit dem zeitliche Abfragen benutzerfreundlich durchgeführt werden können.

Jakob Miksch

SensorThings – Von der Sensordaten-Verarbeitung im Backend bis zur Visualisierung im Masterportal

DIRK ROHRMOSER

Einleitung

Sensordaten gewinnen seit Jahren in vielen Bereichen der Informationsverarbeitung zunehmend an Bedeutung. Gerade im Geoinformationswesen eröffnen sich durch die rasant steigende Anzahl an Sensoren und die logische Vernetzung der gewonnenen Daten viele neue und vielversprechende Nutzungsszenarien. Häufig werden in diesem Zusammenhang Buzzwords wie „Smart City“, „Big Data“ oder „Urban Twin“ genannt. Um diese Konzepte mit Inhalten zu füllen, sind smarte und innovative Lösungen in einem umfassenden Kontext erforderlich. Diese Lösungen dienen u. a. als strategische Bestandteile einer digital gestützten Stadtentwicklung, um den wachsenden Herausforderungen in hochverdichteten Räumen entgegen zu treten. Der vorliegende Beitrag stellt die Architektur zur Verarbeitung von Sensordaten und deren Nutzung in der Freien und Hansestadt Hamburg vor.

Architektur

Für die Entgegennahme und Bereitstellung von Sensordaten steht die Urban Data Platform Hamburg (HH_UDP) zur Verfügung. Hierbei handelt es sich um ein System, das vorhandene und künftige IT-Systeme miteinander verbindet und die integrierten Daten, je nach den Bedürfnissen der Nutzer, individuell konfigurierbar, auswertbar und in Echtzeit abrufbar macht [4].

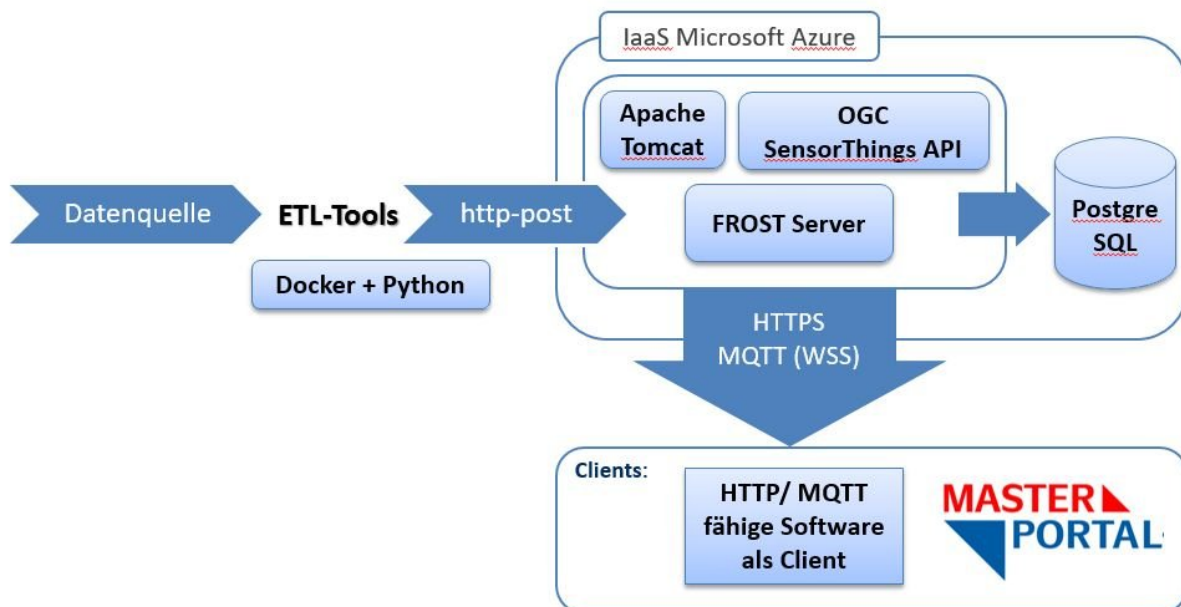


Abb. 1: Echtzeitdaten – Datenfluss

Der Datenfluss, von der Datenquelle bis zur Visualisierung im Client, ist in Abb. 1 dargestellt. Für die Datenintegration können unterschiedlichste Datenquellen angebunden werden. Hierbei werden die externen Daten über individuelle Schnittstellen mit marktüblichen ETL-Tools (Extract, Transform, Loading) bzw. selbst geschriebenen Python-Skripten entgegengenommen und über die standardisierte

SensorThings – Von der Sensordaten-Verarbeitung im Backend bis zur Visualisierung im Masterportal

Schnittstelle des FROST-Servers [2] in die HH_UDP eingespielt. Der FROST-Server, eine vom Fraunhofer IOSB entwickelte Open Source Implementierung der SensorThings API [7], ist eine zentrale Komponente der HH_UDP und dient u. a. als MQTT-Broker. MQTT steht für „Message Queuing Telemetry Transport“ und wird oft als Standardprotokoll für das Internet of Things (IoT) bezeichnet. In den unten beschriebenen Beispielen dient der Geodatenclient Masterportal [3] als Empfänger der Daten. Hierbei handelt es sich um eine webbasierte Open Source Anwendung, mit der die Sensordaten in Karten, Diagrammen und Tabellen visualisiert werden können. Das Masterportal lädt statische und historische Daten über eine REST-API, während dynamische Daten eventbasiert über MQTT, in Websocket verpackt, bezogen werden [1].

Anwendungsbeispiele

Über den oben skizzierten Weg sind z. B. die Sensordaten des Hamburger Radverkehrsnetzwerkes (HaRaZäN) in das Verkehrsportal der Behörde für Verkehr und Mobilitätswende eingebunden. Das Verkehrsportal steht im Internet für die Öffentlichkeit frei zur Verfügung [5]. Ein Screenshot der Anwendung ist in Abbildung 2 dargestellt.

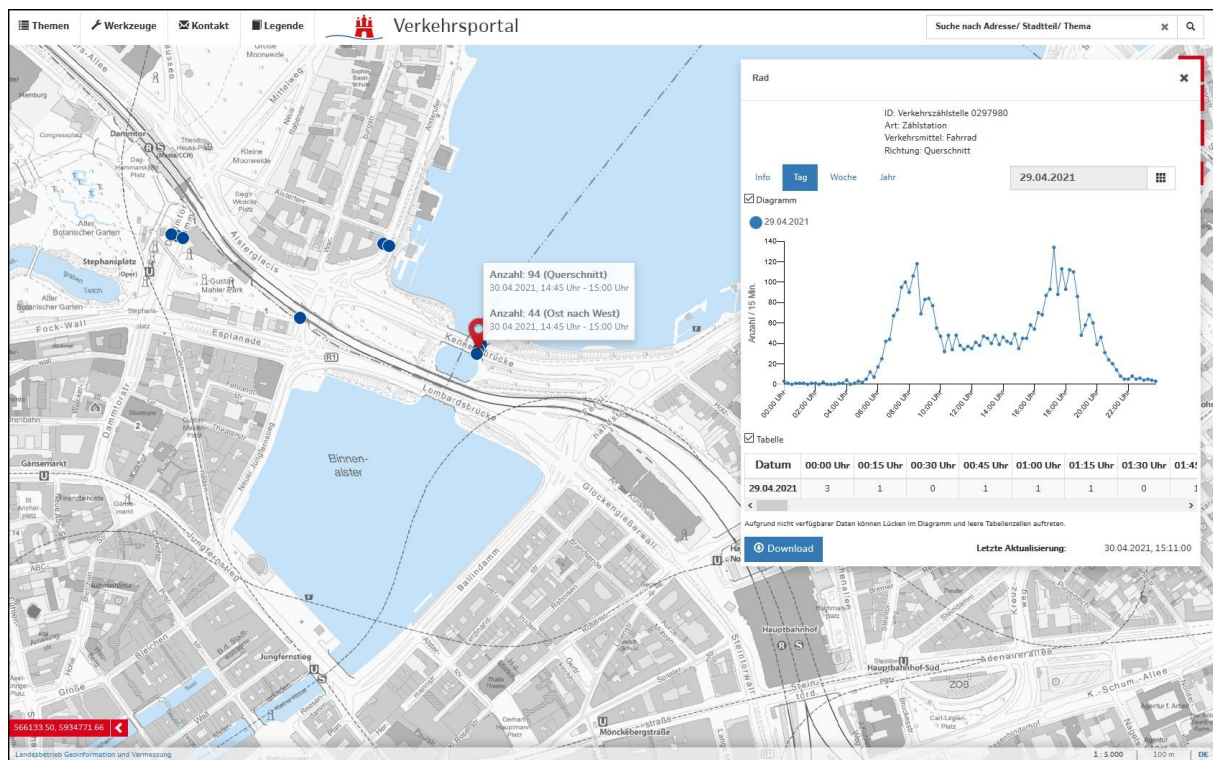


Abb. 2: Verkehrsportal der Behörde für Verkehr und Mobilitätswende, Hamburg

Bei einem Mouse Hover über eine Radverkehrsanzahlstelle in der Karte wird die Anzahl der dort detektierten Radfahrer bzw. Radfahrerinnen während des letzten 15 Minuten-Intervalls angezeigt. Darüber hinaus kann mit einem Klick auf die jeweilige Messstelle ein Infobereich aufgerufen werden, in dem neben den aktuellen Messwerten auch historische Daten in Form von Gangliniendiagrammen bzw. in Tabellenform dargestellt werden.

SensorThings – Von der Sensordaten-Verarbeitung im Backend bis zur Visualisierung im Masterportal

Neben der Visualisierung von aktuellen und historischen Sensordaten sind unzählige weitere Anwendungsfälle denkbar. Exemplarisch sei an dieser Stelle auf das Projekt Traffic Lights Forecast 2.0 verwiesen. Hier werden Daten der hamburgischen Lichtsignalanlagen zur Echtzeit verarbeitet, beispielsweise zur Berechnung optimaler Fahrgeschwindigkeiten, um damit den Verkehrsfluss zu verbessern [6].

Fazit

Für die Erfassung, Vernetzung, Prozessierung, Speicherung und Visualisierung von Sensordaten steht mit dem vorgestellten Ansatz eine Lösung bereit, die auf verschiedenen Open Source Komponenten basiert und sich bereits im produktiven Einsatz befindet. Sie leistet damit einen wertvollen Beitrag zur Umsetzung der Digitalstrategie der Freien und Hansestadt Hamburg.

Kontakt zum Autor:

Dipl.-Geogr. Dirk Rohrmoser
Freie und Hansestadt Hamburg
Landesbetrieb Geoinformation und Vermessung
Neuenfelder Str. 19
21109 Hamburg
Tel.: +49 (0)40 / 4 28 26 – 55 15
dirk.rohrmoser@gv.hamburg.de

Literatur und Webseiten

- [1] Fischer, M., P. Gras, S. Löwa u. S. Schuhart (2021): Urban Data Platform Hamburg: Integration von Echtzeit IoT-Daten mittels SensorThings API. In: DVW – Gesellschaft für Geodäsie, Geoinformation und Landmanagement e. V. (Hrsg.): zfv – Zeitschrift für Geodäsie, Geoinformation und Landmanagement, Heft 1/2021, 146. Jg., Augsburg, S. 47-56.
- [2] Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung IOSB (2021): FROST®-Server. www.iosb.fraunhofer.de/de/projekte-produkte/frostserver.html, abgerufen am 30.04.2021
- [3] Freie und Hansestadt Hamburg, Landesbetrieb Geoinformation und Vermessung (2021): Masterportal. www.masterportal.org, abgerufen am 30.04.2021
- [4] Freie und Hansestadt Hamburg, Landesbetrieb Geoinformation und Vermessung (2021): Urban Data Platform. www.urbandataplatform.hamburg, abgerufen am 30.04.2021
- [5] Freie und Hansestadt Hamburg, Behörde für Verkehr und Mobilitätswende (2021): Verkehrsstärken, Entwicklung und Echtzeitdaten des Radverkehrs. www.hamburg.de/bvm/verkehrsstaerken-rad/, abgerufen am 30.04.2021
- [6] Freie und Hansestadt Hamburg, Landesbetrieb Straßen, Brücken und Gewässer: Intelligent Transport Systems (ITS)-Projekte des LSBG. <https://lsbg.hamburg.de/its-projekte/>, abgerufen am 30.04.2021
- [7] Open Geospatial Consortium (OGC) (2021): OGC SensorThings API. www.ogc.org/standards/sensorthings, abgerufen am 30.04.2021

Nicht nur Kinderwagen-Routing mit OpenStreetMap

Was muss in OpenStreetMap noch gemappt werden, um zuverlässig Kinderwagen und Rollstuhlfahrer routen zu können? Nach "Barrierefreies Fußgängerrouting für Dortmund" von vor zwei Jahren

geht es diesmal um statische Auswertung, neue Aktivitäten außerhalb von DACH und die dauerhafte Lösung des Problems, auch implizit getaggte Bürgersteige auf realitätsnahen Weglinien zu mappen.

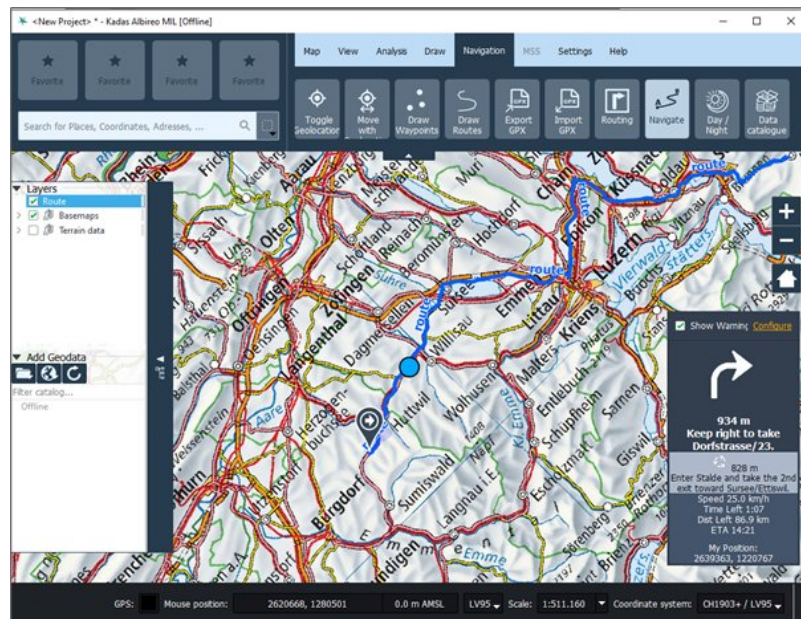
Konkurrenzlos weit vorne ist OpenStreetMap, wenn es um die in den Daten vorhandenen Fußwege geht. Grund genug, für einen Vater seit einem Jahr, herauszufinden, wie gut man mit dem Kinderwagen durchkommt und ob OpenStreetMap dafür zum Routen genug ist.

Tatsächlich schließt der Vortrag an den Vortrag "Barrierefreies Fußgängerrouting für Dortmund" von vor zwei Jahren an: Um Rollstuhlfahrer wie auch Kinderwagen-Schieber routen zu können, gibt es selbst in deutschen Städten noch viel zu mappen. Es geht dieses Mal zum einen um statische Auswertung, welche fehlenden Tags zum Routing noch fehlen. Zum zweiten gibt es nun auch anderswo Aktivitäten zu dem Thema. Nicht zuletzt soll auch der Frage nachgegangen werden, in welchem Umfang man aus den implizit getaggten Bürgersteigen ausreichend realitätsnahe Weglinien bekommt, um damit tatsächlich Routing betreiben zu können.

Dr. Roland Olbricht

KADAS Routing Plugin

Mit KADAS Albirero als Client und Valhalla als Routing Engine wurde eine performante Lösung für Anwendungen im Bereich Offline Routing geschaffen. KADAS Albirero basiert auf QGIS und wurde in diesem Open-Source Projekt mit einem Routingmodul erweitert. Valhalla wurde nicht als HTTP Server aufgesetzt, sondern für die Nutzung per Kommandozeile angepasst. Neben der Berechnung von Routen und der Navigation, wurde auch die Berechnung von Isochronen sowie Isodistanzen in KADAS ermöglicht.



Im laufenden Open-Source Projekt KADAS Routing wurde die Anwendung KADAS Albirero 2.0 mit einem Routing Plugin für den Offline Betrieb erweitert. KADAS ist eine Desktop-GIS Lösung für weniger erfahrene Nutzer, die auf QGIS basiert. Als Routing Engine wird Valhalla eingesetzt, das als lokaler Dienst vom Plugin abrufbar ist. Valhalla bietet den Vorteil unterschiedliche Routing-Profile mit demselben Graphen abzubilden, wobei weniger RAM benötigt wird. Letzteres ist ein klarer Vorteil bei der Offline-Nutzung.

Die Berechnung der Routen wird mit Hilfe von OpenStreetMap Daten oder mit alternativen Datensätzen durchgeführt. Hierfür wurde im Routing Modul ein Tool implementiert, dass es dem Nutzer ermöglicht über ein frei wählbares Portal das Datenpaket für ein Land oder eine Region herunterzuladen.

Bei der Berechnung einer Route kann der Nutzer von einer Liste von Profilen auswählen, die neben den Standard-Profilen (Fußgänger, Fahrrad und PKW) auch alternative Fahrzeugtypen beinhaltet. Die Route kann dann via Start- und Zielpunkt definiert werden, sowie über das Hinzufügen von weiteren Wegpunkten als Zwischenzielen. Nach der Eingabe der Parameter werden diese an Valhalla übergeben, das die berechnete Route zurückliefert. Die Route ist nun als Ebene im KADAS verfügbar und der Nutzer kann den Navigationsmodus starten.

Neben den Funktionalitäten für das Routing wurde die Berechnung von Isochronen und Isodistanzen implementiert. Hierbei liefert Valhalla nach Eingabe der Intervalle die berechneten Pufferzonen an KADAS zurück, die wieder als reguläre Ebene betrachtet und editiert werden können.

Weitere geplante Entwicklungen für den Client umfassen unter anderem die Navigation im Nachtmodus. Auf Seiten des Routing Engines wird derzeit die Implementierung des Chinese Postman Problem evaluiert, was eine signifikante Erweiterung im Open-Source Projekt Valhalla darstellen würde.

Andreas Jobst

Transitdaten und OpenStreetMap

In diesen Vortrag geht es um die Erstellung und Schätzung von Transitdaten im General Transit Feed Specification (GTFS) Format mithilfe von OpenStreetMap Daten. Weiters zeigen wir die Schwierigkeiten und Herausforderungen die auftauchen wenn man mit GTFS arbeitet, die verschiedenen Routing Engines die wir damit getestet haben und die Vollständigkeit von OpenStreetMap für diese Anwendung.

Für Routing von öffentlichen Verkehrsmittel arbeitet man in erster Linie mit General Transit Feed Specification (GTFS) Dateien, die man dann in verschiedenen Routing Engines verwenden kann. Es gibt mittlerweile viele offene Datenquellen wie zB.: Transitfeeds oder Transitland Operators. Dennoch gibt es für viele Regionen keine vollständige oder fehlende GTFS Abdeckung. Für diesen Fall gibt es die Möglichkeit auf OpenStreetMap zuzugreifen.

In diesen Vortrag geht es um die Erstellung und Schätzung von GTFS Dateien mithilfe von OpenStreetMap Daten. Weiters zeigen wir die Schwierigkeiten und Herausforderungen die auftauchen wenn man mit GTFS arbeitet, die verschiedenen Routing Engines die wir damit getestet haben und die Vollständigkeit von OpenStreetMap für diese Anwendung.

Nikolai Janakiev

Tivigi: Eine moderne Web-GIS-Komponentenbibliothek mit TypeScript, Vue.js und OpenLayers

SEBASTIAN BECHTOLD, METROPOLREGION RHEIN-NECKAR GMBH / GEO.NET.MRN E.V.

Was ist Tivigi?

Tivigi ist eine Sammlung von Vue.js-Komponenten und dazugehörigen Hilfsfunktionen mit Fokus auf die Entwicklung moderner Web-GIS-Anwendungen. Der Name "Tivigi" stammt von der englischen Aussprache des Akronyms "TVG", was wiederum für "TypeScript - Vue.js - GIS" steht. Der Name ist im wahrsten Sinne des Wortes Programm: Tivigi ist vollständig in TypeScript geschrieben und basiert neben Vue.js vor allem auf OpenLayers und einigen weiteren JavaScript-Bibliotheken.

Derzeit umfasst Tivigi ca. 70 Komponenten, die ein breites Feld von Web-GIS-Features wie Karten- und Legenden-Panels, Layer-Switcher, Kartenabfrage („Feature Info“) sowie verschiedene generische GUI-Elemente wie verschiebbare Fenster, Baummenüs und Aufklappbereiche implementieren. Durch die Möglichkeit, diese Komponenten größtenteils deklarativ, d.h. mittels der auf HTML aufbauenden Vue.js-Template-Sprache in Vue.js-Projekten zu verwenden, wird die Entwicklung moderner Web-GIS-Anwendungen drastisch vereinfacht und beschleunigt.

Tivigi wird seit Anfang 2019 von der Metropolregion Rhein-Neckar im Rahmen des Projekts "Metropolatlas Rhein-Neckar" ¹, dem offiziellen Geoportal der Metropolregion Rhein-Neckar, entwickelt. Der Metropolatlas dient als Referenz- und "Flaggschiff"-Projekt für Tivigi. Die Bibliothek kommt inzwischen jedoch auch in weiteren Projekten wie der Fördermittelkarte des Rhein-Neckar-Kreises ² und dem „Atlas der Innovation“ der Universität Speyer ³ zum Einsatz.

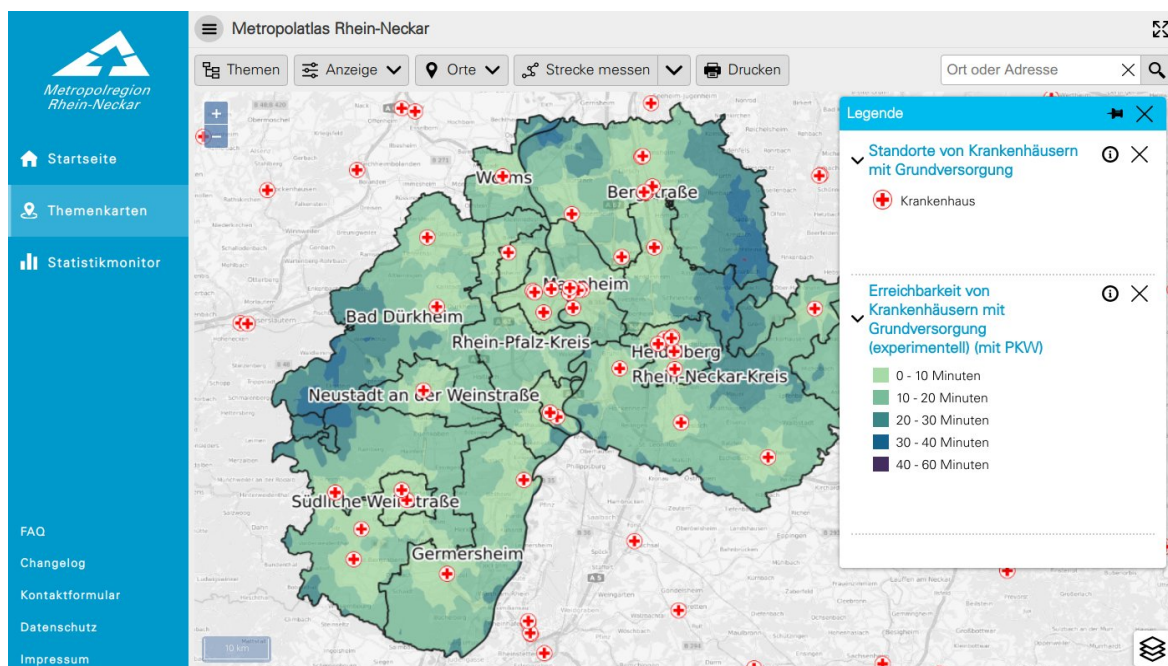


Abbildung 1: Der Metropolatlas Rhein-Neckar, die erste Tivigi-basierte Anwendung

1 <https://metropolatlas.digitale-mrn.de>

2 <https://deinfoerderprojekt.de>

3 <https://www.witi-innovation.de/atlas-der-innovation/>

Tivigi: Eine moderne Web-GIS-Komponentenbibliothek mit TypeScript, Vue.js und OpenLayers

Die folgenden Abschnitte stellen die grundlegenden Designkonzepte von Tivigi vor. Da bei diesem Thema erfahrungsgemäß leicht Missverständnisse auftreten, wird insbesondere auch erklärt, was Tivigi *nicht* ist.

Tivigi ist keine vollständige, konfigurierbare Anwendung

In der deutschsprachigen GIS-Community (und sicherlich auch außerhalb davon) existieren verschiedene Open-Source-Projekte, die Tivigi auf den ersten Blick ähneln und zum Teil auf denselben Technologien (Vue.js, OpenLayers etc.) aufbauen. Beispiele hierfür sind „Wegue“ von Meggsimum oder das Masterportal der Geowerkstatt Hamburg. Bei beiden handelt es sich um weitgehend vollständige Webanwendungen, die mehr oder weniger „out of the box“ verwendet und durch Bearbeitung von Konfigurationsdateien angepasst werden können.

Tivigi verfolgt einen anderen Ansatz. Tivigi ist *keine* fertige Anwendung (und auch kein „Framework“ im eigentlichen Sinne), sondern „lediglich“ eine Softwarebibliothek. Die Entwicklung von Anwendungen mit Tivigi erfordert mindestens Grundkenntnisse über Vue.js und Web-Front-End-Entwicklung allgemein. Im Gegenzug bietet Tivigi ein sehr hohes Maß an Flexibilität. Es bestehen keinerlei Einschränkungen oder Zwänge bezüglich GUI-Design oder der Verwendung bestimmter Komponenten. Auch im Hinblick auf die Softwarearchitektur „unter der Oberfläche“ gewährt Tivigi sehr viel Freiheit. Die Nutzung der Bibliothek ist an keinerlei globalen „Application State“ gekoppelt. Ein „Data Store“ wie Vuex oder ähnliches ist nicht Bestandteil des Konzepts.

Am besten kann man sich Tivigi als Erweiterung von HTML um zusätzliche Elemente für GIS-Anwendungen vorstellen. Tatsächlich können zumindest einfache Tivigi-Anwendungen vollständig deklarativ, also in der auf HTML aufbauenden Vue.js-Template-Sprache erstellt werden.

Tivigi hat kein „Kernsystem“ und keine „Plug-Ins“

Tivigi hat eine dezentrale Architektur. Alle Tivigi-Komponenten sind prinzipiell „gleichwertig“ und es gibt keine Grundstruktur, die in jeder Tivigi-Anwendung in einheitlicher Form vorhanden sein muss. Die einzige (zwangsläufige) Ausnahme von diesem Prinzip ist der Umstand, dass manche Komponenten ohne das gleichzeitige Vorhandensein anderer Komponenten, mit denen sie gemäß ihrer „Natur“ in enger Wechselwirkung stehen, keine sinnvolle Rolle erfüllen können. Beispielsweise können Komponenten zur Darstellung einer Kartenlegende oder zum Ein- und Ausblenden von Kartenlayern nur dann funktionieren, wenn gleichzeitig auch eine Kartenkomponente vorhanden ist, auf die sich die anderen Komponenten beziehen können.

In dieser Hinsicht kann z.B. die Kartenkomponente (und auch einige andere) als eine Art „Kernkomponente“ oder „Komponente höherer Ordnung“ verstanden werden. Jedoch ist auch diese Einordnung relativ, da eine Anwendung problemlos auch mehrere oder gar keine Kartenkomponenten enthalten kann. Die Nutzungsmöglichkeiten von Tivigi sind grundsätzlich auch nicht auf die Erstellung von Web-GIS-Anwendungen beschränkt. Generische, nicht-GIS-bezogene Teile von Tivigi (z.B. die verschieb- und skalierbaren Fenster) können im Prinzip in jeder Art von Vue.js-Anwendung sinnvoll verwendet werden.

Tivigi ist eine reine Front-End-Bibliothek

Zum gegenwärtigen Zeitpunkt enthält Tivigi keine serverseitigen Bestandteile. Die Bibliothek dient rein der Entwicklung von Web-Front-Ends nach dem „Single Page Application“-Prinzip. Dieses besagt, dass sämtliche Programmlogik in clientseitig (d.h. vom Browser) ausgeführtem JavaScript-Code implementiert ist. Das Front-End kann mit beliebiger serverseitiger Software über HTTP-Requests interagieren, mit denen Daten abgefragt und nachgeladen werden, typischerweise im JSON- oder XML-Format oder als Grafikdateien (z.B. für Kartendarstellungen via WMS).

Durch den Fokus auf GIS-Anwendungen sind WMS/WFS-Implementierungen wie GeoServer die typischsten serverseitigen „Partner“ von Tivigi. Auch ein HTTP-Proxy-Dienst zur Umgehung von CORS-Beschränkungen wird häufig eingesetzt. Es ist aber auch möglich, Tivigi-Anwendungen zu entwickeln,

Tivigi: Eine moderne Web-GIS-Komponentenbibliothek mit TypeScript, Vue.js und OpenLayers

die keinerlei serverseitige Komponenten benötigen oder ausschließlich Webdienste von Drittanbietern nutzen.

Tivigi setzt stark auf Modularisierung und Wiederverwendung von Komponenten

Eines der grundlegenden Designprinzipien von Tivigi ist, die Codebasis möglichst klein zu halten und zur Entwicklung neuer Komponenten bereits vorhandene Komponenten so weit wie sinnvoll möglich wiederzuverwenden. In der Softwareentwicklung ist dieses Prinzip allgemein bekannt als „DRY“ – „Don't repeat yourself“.

Vue.js unterstützt hierfür das Prinzip der Verschachtelung bzw. Komposition von Komponenten: Einfache, elementare Komponenten dienen als Bausteine zur Schaffung größerer, komplexerer Komponenten, welche wiederum als Bausteine für noch größere und komplexere Komponenten genutzt werden können. Aus technischer Sicht ist bei Vue.js auch eine komplette Anwendung auf oberster Ebene nichts anderes als eine Komponente, die sämtliche Unterkomponenten, aus denen die Programmoberfläche besteht, in sich vereint.

Dieser Ansatz bringt mehrere Vorteile mit sich:

- Der Entwicklungsaufwand für die Implementierung neuer Features wird reduziert.
- Fehlerkorrekturen in einer einzelnen Komponente wirkt sich an vielen Stellen in der gesamten Bibliothek oder Anwendung aus, nämlich überall dort, wo die Komponente eingebunden ist.
- Das „Look & Feel“ einer Anwendung wird vereinheitlicht. Teile der GUI, die den gleichen elementaren Zweck erfüllen (z.B. Baummenüs oder Aufklappbereiche) sehen gleich aus und verhalten sich gleich, da sie mit der selben Komponente realisiert wurden.

Tivigi setzt stark auf deklarative Programmierung

Vue.js-Komponenten und -Anwendungen funktionieren als Wechselspiel zwischen imperativem JavaScript-Code für alle nicht direkt Ausgabe-bezogenen Belange, und deklarativem „Template“-Code, der primär dazu gedacht ist, die Darstellung einer Komponente zu definieren. Die Vue.js-Template-Sprache ist in ihrer Grundstruktur XML-artig. Sie basiert auf der Sprache des zu generierenden Dokuments - im Folgenden „Zielsprache“ genannt (meist HTML, aber auch SVG und andere XML-basierte Sprachen sind möglich) - und fügt einige Vue.js-spezifische Elemente und Attribute, sogenannte *Direktiven*, hinzu.

Vue.js bietet eine gewisse Freiheit bei der Verteilung von Programmlogik auf den imperativen oder deklarativen Aspekt einer Komponente. Grundsätzlich geschieht jedoch die Zusammenführung von Komponenten zu größeren, übergeordneten Komponenten und zu kompletten Anwendungen auf der Template-Ebene, also deklarativ.

Vue.js-Komponenten werden in der Template-Sprache durch gleichnamige Elemente bzw. „Tags“ repräsentiert und können so mit anderen Vue.js-Komponenten und mit Elementen der Zielsprache (z.B. Eingabefelder, Buttons oder Tabellen) kombiniert werden. Grundsätzlich kann man sich Vue.js als Werkzeug zur Schaffung eigener „Pseudo-Elementtypen“ für XML-artige Sprachen vorstellen, die von der Vue.js-Engine in normale Elemente der Zielsprache übersetzt werden. Durch die Definition eigener Pseudo-Elementtypen können komplexe Dokumente für die Zielsprache einfacher und übersichtlicher erzeugt und deren Generierung durch mit den „Pseudo-Elementtypen“ verknüpfte Programmlogik gesteuert werden.

Ein Entwicklungsziel von Tivigi ist, möglichst viel Freiheit bei der Kombination verschiedener Komponenten rein auf der Template-Ebene zu ermöglichen. Dies hat einen ganz bestimmten Hintergrund: Vue.js ermöglicht es, Komponenten-Templates in einer laufenden Anwendung „on the fly“ auszutauschen. Es ist also möglich, Template-Code per HTTP-Request nachzuladen und einer Komponente zuzuweisen, ohne dass dafür die Anwendung neu kompiliert, „gebündelt“ und auf den Server geladen werden muss. Dies ist ein äußerst mächtiges Feature, denn es ermöglicht auf einfache Weise, Teile von Vue.js-Anwendungen in Form von Templates in einem Content Management System (CMS) zu pflegen und über eine Weboberfläche zu bearbeiten.

Tivigi: Eine moderne Web-GIS-Komponentenbibliothek mit TypeScript, Vue.js und OpenLayers

Die größte Stärke von JavaScript-„Single Page Applications“ (SPAs), also Webanwendungen, die vollständig im Browser laufen, sind sehr schnelle Reaktionszeiten der GUI bzw. generell die Möglichkeit zur Einbindung von Animationen und interaktiven Elementen wie z.B. verschiebbaren Fenstern, die nicht Bestandteil des HTML-Sprachstandards sind. Aufgrund der Funktionsweise der üblichen „Toolchains“ bzw. Build-Prozesse zur Erstellung von SPAs sind diese jedoch dafür in anderer Hinsicht meist recht statischer Natur:

In der Regel ist es mit den modernen „Toolchains“ zur Erstellung von SPAs nicht ohne weiteres möglich, die Programmoberfläche grundlegend zu verändern oder zu erweitern, ohne mit einem Werkzeug wie Webpack ein neues Deployment-Paket der Anwendung zu generieren („bundle“) und auf den Server zu laden. Hier haben „klassische“ Webanwendungen, die primär von serverseitiger Programmlogik getrieben werden, einen Vorteil: Sie erzeugen den clientseitigen Teil der Anwendung, also den HTML- und ggf. JavaScript-Code, bei jedem Aufruf neu und bieten damit einen vergleichsweise einfachen Mechanismus für „ad hoc“-Änderungen, z.B. auf Grundlage von Datenbankinhalten.

Die Möglichkeit, Vue.js-Komponenten-Templates zur Laufzeit nachzuladen und auszutauschen, vereint die beiden genannten größten Stärken von SPAs und servergetriebenen Webanwendungen: Weboberflächen können gleichzeitig beliebig komplex und interaktiv sein und dennoch per CMS o.ä. erstellt und bearbeitet werden. Da die Vue.js-Template-Sprache auf HTML bzw. XML aufbaut, ist dies zudem leicht erlernbar. Programmierkenntnisse in JavaScript/TypeScript sind kaum bis gar nicht erforderlich.

Zur maximalen Ausnutzung dieses Vorteils ist es jedoch erforderlich, möglichst viel Funktionalität und Gestaltungsfreiheit auf der Template-Ebene zu realisieren. Die „on the fly“-Manipulation des imperativen JavaScript-Codes von Komponenten ist in Vue.js nicht ohne weiteres möglich und würde zudem sowohl ein Sicherheitsrisiko darstellen als auch die CMS-basierte Erstellung von Oberflächen zusätzlich verkomplizieren.

Deshalb verfolgen wir bei der Entwicklung von Tivigi den Ansatz, die Template-Ebene so mächtig wie möglich zu gestalten. Dies hat weitreichende Auswirkungen auf die Designparadigmen und Funktionsweise von Komponenten und Anwendungen. Manches davon wirkt möglicherweise etwas unintuitiv und umständlich und entspricht nicht den „best practices“, die man verfolgen würde, wenn man der Template-basierten Entwicklung keinen derart hohen Stellenwert einräumen würde. Teilweise handelt es sich um eine „Zweckentfremdung“ des Template-Systems von Vue.js, indem es für Dinge genutzt wird, die nichts direkt mit der Generierung der Ausgabe zu tun haben. Mit diesem Ansatz ist es uns jedoch gelungen, viele wesentlichen Teile von Tivigi über rein deklarativen Code nutzbar zu machen, so dass einfache Tivigi-Anwendungen sogar vollständig in Vue.js-Template-Code realisiert werden können.

Ein Codebeispiel

Die folgende Abbildung zeigt ein Codebeispiel für eine einfache Tivigi-Anwendung. Es handelt sich um das fünfte Codespielspiel aus der README-Datei des Tivigi-Repositories auf GitHub (Stand: 29.04.2021), wobei einige nebensächliche Bestandteile entfernt wurden.

```
1  <Data name="showLayerTree" :value="false" />
2  <Data name="homeExtent" :value="[7.8,49.77,9.62,48.95]" />
3  <Data name="mapHomeTrigger" :value="false" />
4  <DataMap name="map" />
5  <DataJsonFetch name="layerDef" url="layerdefs.json" />
6  <DataJsonFetch name="layerTreeDef" url="layertree.json" />
7
8  <div style="padding:8px;display:flex;flex-direction: row;">
9    <Superbutton label="Layers" icon="tivigi/img/tree.svg"
10     v-model="local.showLayerTree" :set="true" :unset="false" />
11
12    <Superbutton label="Reset Map Extent" icon="tivigi/img/home.svg"
13     @click="local.mapHomeTrigger = !local.mapHomeTrigger" />
14
15    <LocationSearch :map="local.map" />
16  </div>
17
18  <Modal class="Card floating" :show.sync="local.showLayerTree"
19    style="z-index:1;max-height:70%;width:300px;left:4px;top:64px;overflow:auto;position:absolute;"
20    :allowOutsideInteraction="true">
21    <LayerTree :treeDef="local.layerTreeDef" :layerDef="local.layerDef" :map="local.map" />
22  </Modal>
23
24  <div style="display:flex;flex-direction: row;height:100%;">
25    <ActiveLayers :map="local.map" style="width:400px;padding:13px;" />
26
27    <div style="position:relative;flex:1;">
28      <MapPanel :map="local.map" :initialExtent="local.homeExtent" />
29      <MouseCoordinatesTool :map="local.map" style="position:absolute;right:6px;top:10px;" />
30    </div>
31
32    <FeatureInfoTool :show="true" :map="local.map" style="width:400px;padding:13px;" />
33  </div>
34
35  <SetMapExtentTool :map="local.map" :trigger="local.mapHomeTrigger" :extent="local.homeExtent" />
```

Abbildung 2: Vue.js-Template-Code für eine einfache, komplett deklarativ realisierte Tivigi-Anwendung

Das Beispiel zeigt, wie die Struktur der Anwendungsoberfläche durch Kombination normaler HTML-Elemente und -Attribute („<div>“, „style“ etc.) mit Vue.js- bzw. Tivigi-spezifischen Teilen („<MapPanel>“, „<Modal>“, „v-model“ etc.) aufgebaut wird.

Die Elemente am Anfang des Codebeispiels, deren Namen mit dem Präfix „Data“ beginnen, repräsentieren keine sichtbaren Elemente der Oberfläche. Stattdessen dienen sie der Bereitstellung von Daten und Funktionsobjekten, die von anderen Komponenten benötigt werden. Das „<DataMap>“-Element (Zeile 4) erzeugt beispielsweise ein OpenLayers-Map-Objekt. Es wird unter dem Variablennamen „map“ im Speicher hinterlegt und kann fortan von anderen Komponenten unter dem Namen „local.map“ referenziert werden. Dies geschieht z.B. in Zeile 28, wo das Kartenobjekt als Wert für das Attribut „map“ an die „<MapPanel>“-Komponenten übergeben wird. Die „<MapPanel>“-Komponente stellt ein OpenLayers-Map-Objekt auf dem Bildschirm dar. Es handelt sich im Wesentlichen um ein HTML-`<div>`-Element, das als „target“ des OpenLayers-Map-Objekts gesetzt wird. Das Attribut „map“ definiert, welches OpenLayers-Map-Objekt von der MapPanel-Komponente dargestellt werden soll. Der Doppelpunkt als Beginn eines Attributnamens in Vue.js-Templates gibt an, dass der Wert des Attributs

nicht, wie sonst in XML und HTML üblich, als Stringkonstante, sondern als JavaScript-Ausdruck (d.h. in diesem Fall als Variablenname) interpretiert werden soll. Die übrigen Tivigi-Komponenten in diesem Beispiel erfüllen folgende Aufgaben:

<Data> erzeugt eine Variable beliebigen Typs, die durch den als Attribut „*value*“ definierten JavaScript-Ausdruck initialisiert wird.

<DataJsonFetch> fordert per HTTP-Request JSON-Daten von der als Attribut „*url*“ definierten Web-Adresse an.

<Superbutton> erzeugt eine Schaltfläche mit im Vergleich zum Standard-HTML-Button-Element erweiterten Eigenschaften und Interaktionsmöglichkeiten. Beispielsweise kann ein *<Superbutton>* einen „gedrückt“-Status annehmen und sich dadurch optional wie eine Checkbox oder wie ein Radiobutton verhalten.

<LocationSearch> stellt ein Eingabefeld für eine Orts- und Adresssuche bereit, die im Hintergrund mit dem Nominatim-Dienst des OpenStreetMap-Projekts arbeitet. Durch Verknüpfung mit einem OpenLayers-Kartenobjekt (Attribut „*map*“) wird nach einer erfolgreichen Suchanfrage der Kartenausschnitt auf die zurückgelieferten Koordinaten zentriert.

<Modal> implementiert einen sogenannten modalen Dialog, also ein GUI-Element, das (im Standard-Verhalten) Interaktionen mit der übrigen Programmoberfläche blockiert, so lange es angezeigt wird. Das genaue Verhalten des *<Modal>*-Elements kann über Attribute konfiguriert werden. So kann z.B. eingestellt werden, dass ein *<Modal>*-Element bei Klick bzw. Touch-Interaktion außerhalb des Elements automatisch ausgeblendet wird statt diese Interaktionen zu ignorieren.

<ActiveLayers> ist eine kombinierte „Layer Switcher“- und Legenden-Komponente. Es listet alle momentan in die referenzierte OpenLayers-Karte eingebundenen Layer auf und zeigt (falls vorhanden) die dazugehörigen Legenden an. Zusätzlich können Buttons für verschiedene Layer-Aktionen wie Umschalten der Sichtbarkeit oder vollständiges Entfernen des Layers von der Karte eingeblendet werden. Dies ist über Attribute frei konfigurierbar.

<MouseCoordinatesTool> zeigt die aktuelle Position des Mauszeigers über der Karte in geographischen Koordinaten an.

<FeatureInfoTool> implementiert ein Kartenabfrage-Werkzeug mit Unterstützung für WMS-Layer (via „*REQUEST=GetFeatureInfo*“) und clientseitige Vektorlayer. Es wird über das Attribut „*map*“ mit einem OpenLayers-*Map*-Objekt verknüpft und zeigt bei Klick/Touch auf die Karte das Ergebnis der Abfrage in einem Popup-Fenster an.

<SetMapExtentTool> wird, ähnlich wie die „*Data*“-Komponenten, nicht in der Programmoberfläche dargestellt. Es setzt die über das Attribut „*map*“ referenzierte OpenLayers-Karte auf die über das Attribut „*extent*“ definierte Bounding Box, wenn sich der Wahrheitswert der über das Attribut „*trigger*“ referenzierten Bool-Variable ändert. Dies kann z.B. durch einen entsprechend programmierten Button ausgelöst werden. Es wurde keine Schaltfläche o.ä. direkt in die *<SetMapExtentTool>*-Komponente integriert, um maximale Flexibilität bei der Verwendung der Komponente zu ermöglichen. So kann der Zentrierungsvorgang auf unterschiedlichste Weise ausgelöst werden.

Wie eingangs bereits erwähnt, enthält Tivigi mehrere dutzend weitere Komponenten für unterschiedliche Zwecke wie Eingabe, Layout und Datenvisualisierung. Die Vorstellung aller dieser Elemente würde den Rahmen dieser Kurzeinführung sprengen. Die README-Datei des Tivigi-Repositories auf GitHub (siehe nächster Abschnitt) enthält weitere Informationen und ein Einsteiger-Tutorial, in dem die Erstellung einer einfachen Tivigi-Anwendung vollständig Schritt für Schritt erklärt wird.

Verfügbarkeit und Ausblick

Tivigi ist als Open Source unter der MIT License auf GitHub verfügbar ⁴. Zu beachten ist, dass die Qualität der verschiedenen Komponenten zurzeit noch stark unterschiedlich ist. Einige Teile sind bereits recht ausgereift, andere haben sehr experimentellen Charakter. Eine offizielle Versionierung und definierte Releases der Bibliothek gibt es noch nicht. Die Dokumentation ist bislang beschränkt auf die englischsprachige README-Datei mit einer ausführlicheren Beschreibung der Design- und Funktionsprinzipien von Tivigi sowie dem erwähnten Einsteigertutorial.

Tivigi wird von uns ständig weiterentwickelt. Derzeit liegt der Schwerpunkt auf der Entwicklung von neuen Konzepten und Komponenten zur Realisierung "Dashboard"-artiger Anwendungen. Dazu gehören v.a. verschiedene Typen von Diagrammen, Komponenten zur einfachen Implementierung responsiver Kachel-Layouts sowie verbesserte Möglichkeiten zur Definition komplexer Vektor-Layers-Styles für OpenLayers. Unsere regelmäßigen Änderungen und Ergänzungen an Tivigi fließen bisher nicht direkt in das öffentliche GitHub-Repository ein. Wir arbeiten auf einem separaten, nicht-öffentlichen Repository und übertragen die Änderungen in unregelmäßigen Abständen in größeren Blöcken in das öffentliche Repository. Da dies die kooperative Entwicklung mit externen Partnern natürlich erschwert, würden wir diese Vorgehensweise ggf. ändern, falls Interesse an Mitwirkung besteht.

Kontakt zum Autor:

Sebastian Bechtold
Metropolregion Rhein-Neckar GmbH
M1 4-5
68161 Mannheim
sebastian.bechtold@m-r-n.com

4 <https://github.com/metropolregion-rhein-neckar/tivigi>

Kommunales Geoportal zur dynamischen Bereitstellung INSPIRE-konformer Dienste - Beispiel einer interkommunalen Zusammenarbeit (IKZ)

OTTO DASSAU UND GÜNTER VAUPEL

Der Landkreis Marburg-Biedenkopf hat im Rahmen eines Projektes zur interkommunalen Zusammenarbeit (IKZ) einen kommunalen Geoservice (GDI Marburg-Biedenkopf) zur Erfüllung der INSPIRE Richtlinie für die Themen Bauleitplanung und Schulstandorte geplant und umgesetzt.

Die GDI Marburg-Biedenkopf besteht aus dem Landkreis Marburg-Biedenkopf, den Städten und Gemeinden Amöneburg, Angelburg, Bad Endbach, Biedenkopf, Breidenbach, Cölbe, Dautphetal, Ebsdorfergrund, Fronhausen, Gladenbach, Kirchhain, Lahntal, Lohra, Münchhausen, Neustadt, Rauschenberg, Stadtallendorf, Steffenberg, Weimar (Lahn), Wetter und Wohratal. Die entsprechende Kooperationsvereinbarung ist am 01.01.2017 in Kraft getreten. Die IKZ verfolgt folgende Ziele:

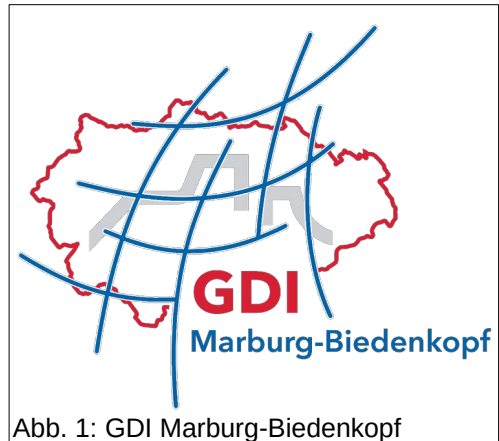


Abb. 1: GDI Marburg-Biedenkopf

- **INSPIRE-Konformität:** Erfüllung der gesetzlichen Verpflichtungen nach der Richtlinie 2007/2/EG des Europäischen Parlaments und des Rates vom 14. März 2007 zur Schaffung einer Geodateninfrastruktur in der Europäischen Gemeinschaft und dem hessischen Vermessungs- und Geoinformationsgesetz (HVGG).
- **Vereinfachte Datenverarbeitung** der Mitglieder durch gegenseitigen Zugriff auf Daten, Datenaustausch, Datenaufbereitung und gemeinsame Veröffentlichung von aktuellen kommunalen Geofachdaten;
- **Kostenreduktion** der Mitglieder insbesondere durch Synergien wie
 - Gemeinsame Nutzung und/oder Ausschreibung von Netzkomponenten, Hard- und Software
 - Vermeidung von Mehrfacherhebungen von Geofachdaten
 - Gemeinsame Bereitstellung einer digitalen Geodateninfrastruktur auf Landkreisebene
 - Optimierung kommunaler und interkommunaler Geschäftsprozesse
 - Verbesserung von Entscheidungsgrundlagen für politische Entscheidungsträger und Verwaltung
- **Förderung der regionalen Zusammenarbeit** und Bündelung vorhandener Kompetenzen

Die Umsetzung des Projekts wurde in die zwei Aufgabenbereiche „Digitale Aufbereitung der Bauleitpläne“ und „Aufbau und Betrieb eines kommunalen Geoservers zur Erfüllung der INSPIRE-Richtlinie“ unterteilt.

Digitale Aufbereitung der Bauleitpläne

Damit die INSPIRE-konforme Bereitstellung des Themas Bauleitplanung reibungslos stattfindet, müssen die relevanten Daten im Vorfeld durch die Kommune aufbereitet und in einem ZIP-Archiv abgelegt werden. Hierzu wurde ein Merkblatt mit einer Empfehlung/Hilfestellung für die Erstbeauftragung von Planerstellungsverfahren bzw. für die Digitalisierung von analogen Plänen entwickelt, damit die erstellten Bauleit- u. Landschaftspläne im notwendigen Format und der erforderlichen Qualität in die GDI Marburg-Biedenkopf integriert und dargestellt werden können. Die Arbeiten umfassen u.a.:

- Erstellung und georeferenzieren digitaler Pläne
- Digitalisierung der Geltungsbereiche aller Teilflächen (Objektbildung)
- Verknüpfung der Geltungsbereiche mit den Sachinformationen
- Ausstanzen und georeferenzierung der Rasterpläne zur Darstellung
- Aufbereitung zusätzlicher Dokumente (Begründungen, Umweltberichte, ...)

Aufbau und Betrieb der kommunalen Geoservices

Technisch basiert die Lösung vollständig auf Open Source Software installiert auf einem virtuellen Linux (Ubuntu) Server, der beim Landkreis Marburg-Biedenkopf gehostet wird (<https://geoservices.marburg-biedenkopf.de>).

Die installierten Hauptkomponenten sind ein PostgreSQL Datenbankserver (<https://www.postgresql.org>) mit PostGIS Erweiterung (<https://postgis.net>) und die GBD WebSuite (<https://gbd-websuite.de>) als WebGIS Plattform zur Geodatenverarbeitung. Beides war bereits im Vorfeld als Geoportal beim Landkreis Marburg-Biedenkopf im Einsatz (https://www.marburg-biedenkopf.de/dienste_und_leistungen/geoportal/geoportal.php).

Im Rahmen des IKZ Projektes wurde die GBD WebSuite um ein INSPIRE Modul erweitert. Ziel war es, Daten zu den Themen Bauleitplanung und Schulstandorte dynamisch, INSPIRE-konform als WMS und WFS zur Verfügung zu stellen. Als Besonderheit sollte die INSPIRE-konforme Bereitstellung der in PostgreSQL/PostGIS verwalteten Daten bei Anfrage in Echtzeit, dynamisch stattfinden. Dadurch gibt es keine Redundanz. Es wird immer der aktuelle Datenbestand verwendet.

Die INSPIRE-konforme Bereitstellung des Themas Schulstandorte als WMS und WFS Dienste findet automatisch auf Grundlage einer Konfiguration durch das INSPIRE Modul der GBD WebSuite statt.

Da die Daten der Bauleitplanung regelmäßig und eigenständig in den Kommunen aktualisiert und erweitert werden sollen, wurde hierfür eine Administrationsoberfläche entwickelt, damit die Kommunen ihre digital aufbereiteten Daten der Bauleitplanung sowie Metadaten selbst aktualisieren und pflegen können (siehe Abb. 2).

Kommunales Geoportal zur dynamischen Bereitstellung INSPIRE-konformer Dienste - Beispiel einer interkommunalen Zusammenarbeit (IKZ)

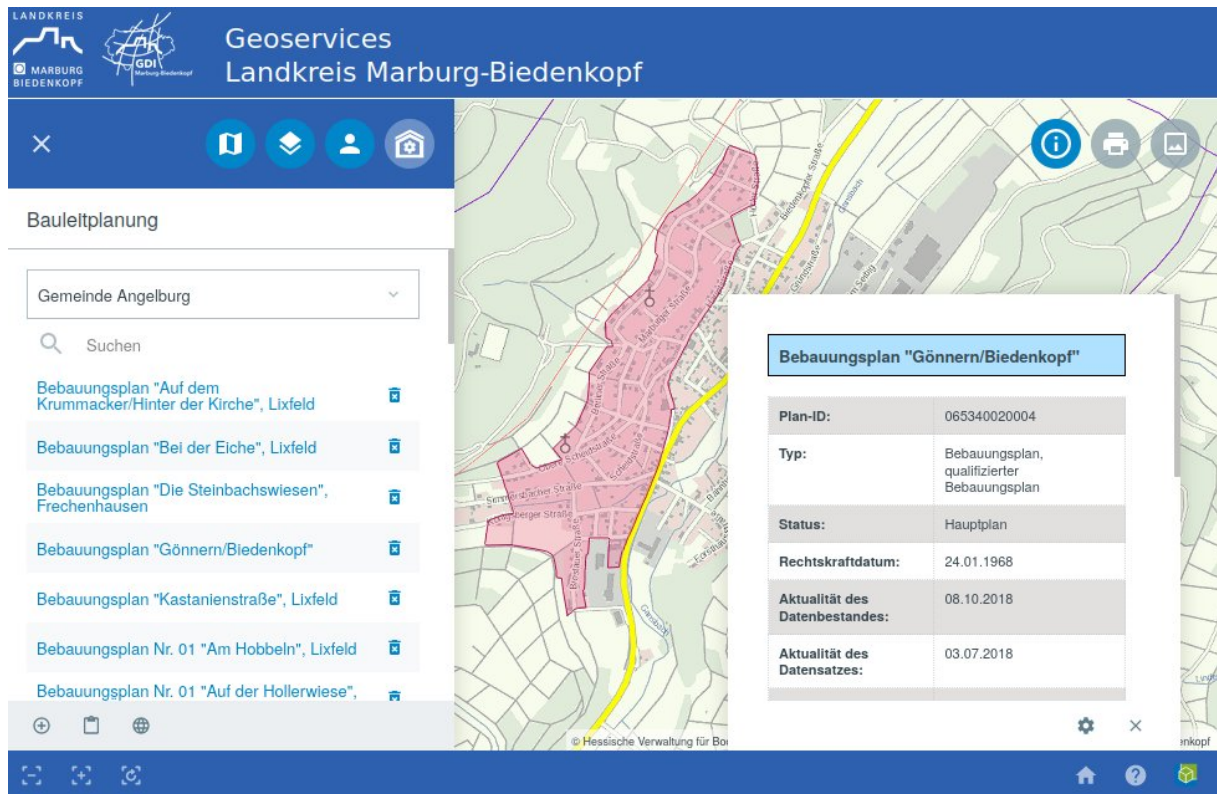


Abb. 2: Administrationsoberfläche Bauleitplanung Geoservice Portal Landkreis Marburg-Biedenkopf

Im Ergebnis wurde die GBD WebSuite um folgende Funktionalität erweitert:

- Aktualisierung von Daten der Bauleitplanung durch berechtigte Nutzer über eine Administrationsoberfläche
- Fortführung von Metadaten (einfügen, aktualisieren, entfernen) des CSW durch berechtigte Nutzer über die Administrationsoberfläche
- Bereitstellung eines INSPIRE konformen Web Catalogue Service (CSW)
- Dynamische Bereitstellung OGC- und INSPIRE-konformer WMS 1.1.0, 1.1.1 und 1.3.0 sowie WMTS 1.0.0 Dienste
- Dynamische Bereitstellung OGC- und INSPIRE-konformer WFS 2.0 Dienste
- Dynamische Bereitstellung INSPIRE-konformer WCS 1.0.0, 2.0.1 Dienste

Es wurden ca. 1700 Bauleitpläne der Kommunen digital aufbereitet und im Dezember 2020 im Geoserviceportal beim Landkreis Marburg-Biedenkopf sowie im Geoportal des Landes Hessen veröffentlicht.

Der gemeinsame Geodatenserver ist in Betrieb. Die Kommunen können auf das Geoserviceportal der GDI Marburg-Biedenkopf zugreifen und neue Bauleitpläne hochladen. Der Geodatenserver übermittelt die Daten direkt an das Geoportal Hessen und die INSPIRE Konformität ist erfüllt. Es werden entsprechende Dienste (WMS und WFS) bereitgestellt.

Im Ergebnis werden Daten der Bauleitplanung (Bebauungspläne, Flächennutzungspläne sowie Landschaftspläne) sowie Schulstandorte OGC- und INSPIRE-konform über die URL <https://geoservices.->

Kommunales Geoportal zur dynamischen Bereitstellung INSPIRE-konformer Dienste - Beispiel einer inter-kommunalen Zusammenarbeit (IKZ)

marburg-biedenkopf.de verfügbar gemacht und die INSPIRE-konformen Dienste in das Geoportal Hessen (<https://www.geoportal.hessen.de>) integriert.

Für die Zukunft ist die Entwicklung zur Bereitstellung weiterer INSPIRE Themen geplant.

Kontakt zu den Autoren:

Otto Dassau
Geoinformatikbüro Dassau GmbH
Rethelstraße 153, 40237 Düsseldorf
+49 (0)211 69937750
dassau@gbd-consult.de

Günter Vaupel
Landkreis Marburg-Biedenkopf
Hermann-Jacobsohn-Weg 1, D-35039 Marburg
+49 (0)6421 405-6904
VaupelG@marburg-biedenkopf.de

Links

- Geoportal Hessen (<https://www.geoportal.hessen.de>)
- Geoservice Portal Marburg-Biedenkopf (<https://geoservices.marburg-biedenkopf.de>)
- WebGIS Marburg-Biedenkopf (https://www.marburg-biedenkopf.de/dienste_und_leistungen/geoportal/geoportal.php)
- PostgreSQL (<https://www.postgresql.org>)
- PostGIS (<https://postgis.net>)
- GBD WebSuite (<https://gbd-websuite.de>)

Docker-basierte GDI in der freien Wildbahn

Im mFund Projekt SAUBER werden Geodaten zu Luftschadstoffen mittels KI-Methodik erstellt. Diese werden in einer zentralen Docker-basierten GDI durch standardisierte Schnittstellen veröffentlicht. Docker und verwandte Technologien erlauben eine hochgradig automatisierte Bereitstellung der GDI. Die SAUBER GDI und deren Aufbau mit Docker wird im Vortrag präsentiert. Eine solche Orchestrierung von GDIs bringt viele Vorteile, aber auch Herausforderungen, welche im Vortrag dargelegt werden.

Docker hat sich in den letzten Jahren als defacto Standard für Anwendungen mit Hilfe von Container-virtualisierung etabliert. Auch im Bereich der Geoinformatik, vor allem im Open Source Bereich,

werden essentielle Softwarekomponenten als Docker-Images bereitgestellt. Damit wird ein einfacher Distributionsweg inklusive Abhängigkeiten etabliert, sowie die Installation von offener und freier GIS-Software stark vereinfacht. Zum einen können Docker-basierte Architekturen für die Nutzung gemeinsamer, geteilter Entwicklungssetups extrem von Vorteil sein. Zum anderen bringt die Docker Nutzung Vorteile beim Aufsetzen sowie der Skalierung von Geodateninfrastrukturen in Cloud-basierten Systemen.

Im mFund Projekt SAUBER (Satellitenbasiertes System zur Anzeige, Prognose und Simulation von Luftschadstoffen für eine nachhaltige Stadt- und Regionalentwicklung) werden Geodaten, die Luftschadstoffe räumlich modellieren, mittels KI-Methodik erstellt. Diese Geodaten werden in einer zentralen Geodateninfrastruktur mittels standardisierter Schnittstellen veröffentlicht. Diese GDI ist komplett docker-basiert und mittels Docker-Compose / Docker-Swarm orchestriert. Dies erlaubt eine hochgradig automatisierte Bereitstellung der GDI. Somit kann die GDI sehr einfach entweder lokal zu Entwicklungszwecken aber auch auf (Cloud-) Servern aufgesetzt werden.

Der Vortrag gibt eine kurze Übersicht über die Basistechnologien Docker, Docker-Compose und Docker-Swarm. Außerdem wird die Methodik und der Aufbau der GDI aus dem SAUBER Projekt als Realweltbeispiel einer Docker-basierten GDI vorgestellt. Außerdem werden die Vorteile aber auch die Herausforderungen und Nachteile eines solchen Setups vorgestellt.

Christian Mayer

Tricks und Kniffe rund um den QGIS-Atlas

Ein einfacher QGIS-Atlas ist schnell erstellt.

Die präzise Steuerung von Blattschnitten, Seitenformaten, Drehungen, Beschriftungen, Sichtbarkeiten etc. ist nicht so ganz trivial. Die Demo-Session stellt einige Kniffe vor, mit denen sich die Möglichkeiten des Atlas umfassend nutzen lassen.

Ein einfacher QGIS-Atlas ist schnell erstellt.

Die präzise Steuerung von Blattschnitten, Seitenformaten, Drehungen, Beschriftungen, Sichtbarkeiten etc. ist nicht so ganz trivial. Die Demo-Session stellt einige Kniffe vor, mit denen sich die Möglichkeiten des Atlas umfassend nutzen lassen.

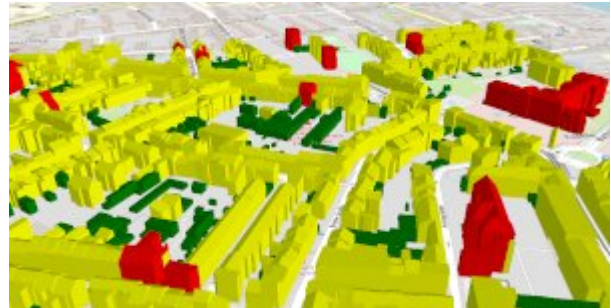
Der QGIS-Atlas ist ein unfassbar umfangreiches Werkzeug, um Kartenserien oder Objekt-Berichte zu erstellen. Die Demo-Session stellt einige Kniffe vor, mit denen sich Möglichkeiten des Atlas umfassend nutzen lassen,

- Blattschnitt-Layer oder Geometrieobjekte als Abdeckungslayer?
- Blattschnitte für beliebige Maßstäbe und Papierformate über eine Modellertool erstellen.
- Drehung von Blattschnitte, die Trassen oder Flüssen folgen.
- Seitenausrichtung und Format anhand der Geometrie und Ausrichtung von Atlasobjekten steuern.
- Beziehungen zu verknüpften Layern auswerten.
- Atlas-Variablen verwenden.
- Sichtbarkeit von Layern und Atlas-Objekten über regelbasierte Darstellung und Kartenthemen steuern
- etc.

Claas Leiner

3D-Geoapplikationen mit CesiumJS - Daten, Einsatzmöglichkeiten und Spezifikationen

Mit der Entwicklung von 3D-Applikationen mit geographischem Bezug, rücken verstärkt auch die Standards und Spezifikationen zur Bereitstellung von entsprechenden Daten in den Fokus. Der Vortrag beschäftigt sich mit dem aktuellen Entwicklungsstand der Bibliothek CesiumJS sowie den Standards und den Einsatzmöglichkeiten einzelner Features.



Nicht nur web-basierte Open-Source-3D-Applikationen mit geographischem Bezug entwickeln sich stetig weiter, auch die Entwicklung von Standards und Spezifikationen der Daten für die Darstellung von 3D-Daten im Web ist verstärkt in den Fokus gerückt. Für die Darstellung im Web kann auf eine Vielzahl von Bibliotheken zurückgegriffen werden (z.B. x3dom, o3d, threejs, BabylonJS, Open GEE). Eine weitere, und seit einigen Jahren kontinuierlich wachsende Bibliothek, ist CesiumJS. Diese wird zur Bearbeitung von geographischen Fragestellungen in zahlreichen Bereichen eingesetzt. Dazu zählen unter anderem der Immobilienmarkt, die Stadtplanung, der Sport oder die verschiedenen Umweltwissenschaften.

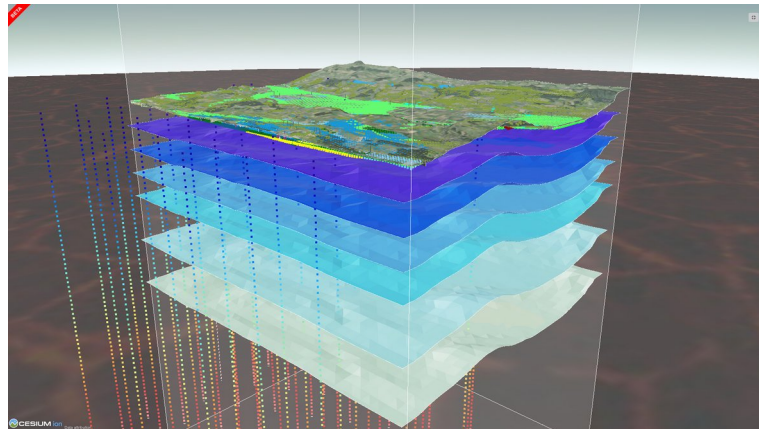
Wenn die Welt digital dargestellt wird, sollen dort auch entsprechende Daten platziert werden. Je nach Anwendungsbereich kann es sich dabei um eine relativ große Menge an Daten handeln, die idealerweise auch auf einfache Art und Weise auf die Karte gebracht werden sollen. Während es für die Darstellung von Raster- und Vektordaten im Web schon Standards gibt, sind für die Darstellung von Terrain, 3D-Modellen, Gebäuden und Punktwolken im Rahmen der Entwicklung von CesiumJS neue Standards entwickelt worden. Mit den 3D Tiles liegt nun auch ein OGC Community Standard vor.

In dem Vortrag wird der aktuelle Entwicklungsstand der Bibliothek und die Einsatzmöglichkeiten der Features und Daten von CesiumJS am Beispiel von Projekten kurz vorgestellt. Dabei wird auch ein Fokus auf die Anforderungen des Browsers gelegt. Neben der allgemeinen Verfügbarkeit und Bereitstellung von Daten sollen auch die Einsatzmöglichkeiten einzelner ausgewählter Features der Bibliothek vorgestellt und diskutiert werden.

Michael Holthausen

swissgeol.ch - Geologische Daten im Web

Geologische Daten fristen in der Regel ein “Mauerblümchen-Dasein”, da es sich um Fachdaten handelt, die nur wenigen Personen zugänglich sind und nur mittels Spezial-Software visualisiert und bearbeitet werden können. Das von swisstopo neu lancierte swissgeol.ch-Portal will dies ändern, indem die auf Open-Source Technologie basierende 3D Visualisierung die Daten niederschwellig und einfach im Internet zugänglich macht.



swissgeol.ch ist eine Webapplikation zur Visualisierung und Analyse von Geologischen Daten, welche seit 2020 als Beta-Version unter <https://swissgeol.ch> öffentlich zugänglich ist.

Ähnlich wie die swisstopo Kartenapplikation <https://map.geo.admin.ch> vor einigen Jahren für Geodaten im Allgemeinen, will swissgeol.ch die Geologischen Daten für diverse Analysen einfach verfügbar machen. Dabei setzt die swisstopo auf die 3D Visualisierung im Web, welche auf CesiumJS basiert und zahlreiche Experten-Tools bietet.

CesiumJS ist die weitverbreitetste Open Source 3D-Globe Library und wird weltweit breit eingesetzt. Dabei werden nicht nur globale Daten visualisiert, sondern auch sehr detaillierte Daten im lokalen Massstab wie z.B. in der 3D-Ansicht von map.geo.admin.ch.

Mit der Entwicklung von swissgeol.ch wurde erstmals eine Untergrund-Navigationsmöglichkeit in CesiumJS geschaffen, welche die Visualisierung von 3D-Objekten unterhalb des Terrains erlaubt. Neben dem Navigieren im Untergrund kann auch mittels Transparenz-Einstellung durch die Erdoberfläche gesehen werden, sowie die Ansicht aufgeschnitten werden.

Mit dem Einsatz von 3D-Tiles und einem präzisen Terrain werden die Daten für das Web optimiert ausgeliefert. Gleichzeitig wird der Download von Original-Daten gesamter Layer bzw. einzelner Objekte im Layer angeboten.

Im Moment befindet sich swissgeol.ch in der Beta-Phase, da laufend neue Funktionalitäten entwickelt und weitere Daten integriert werden.

Mittels einer Übersicht der Tools und Techniken, die in diesem Projekt verwendet werden sowie einem Showcase der verschiedenen Funktionalitäten und Daten zeigt diese Präsentation, dass die als komplex geltenden 3D-Visualisierungen schlank und performant auf dem Web für ein breites (Fach-)Publikum durch den Einsatz von Open-Source Software zugänglich gemacht werden können.

Elisabeth Leu

3D-Punktwolken und Usability

Insbesondere 3D-Punktwolken werden seit Jahrzehnten nicht nur im Bereich der Geodäsie zur Analyse herangezogen. Komplexe Formen stellen dabei eine Herausforderung an die Erfassung, Visualisierung und Weitergabe dar. Die hohe Auflösung geht meist mit einem hohen Datenvolumen einher, welches bei der Darstellung in Echtzeit performante Hardware benötigt. Mit unterschiedlichen Ansätzen versuchen Entwickler, auch im Open Source Bereich, einen Kompromiss zwischen Visualisierung und Usability zu finden

Arbeitsabläufe zur Erfassung von 3D-Daten durch Laserscanner oder die Erzeugung von 3D-Punktwolken mit photogrammetrischen Verfahren sind seit einigen Jahren etabliert. Die Erfassung komplexer Formen und Szenen erfordern eine hochauflösende Erfassung, welche durch Punktwolken mit mehreren Milliarden Punkten zu einem hohen Datenvolumen führen. Diese wiederum erfordern performante Hardware zur Verarbeitung und Visualisierung. Weitere Herausforderungen liegen in der Visualisierung für Endverbraucher, die Weitergabe der Daten sowie die Ausgabeformate zustande. Während triangulierte 3D-Modelle als Alternative herangezogen werden können, führen diese bei einer zu geringen Punktdichte zu wenig repräsentativen Modellen. Aktuell in Entwicklung befindliche Software aus dem Open-Source Bereich ist meist nicht an Nutzergruppen angepasst, bietet aber -vor allem durch Eigeninitiative- bereits Möglichkeiten und Ansätze, den Kompromiss zwischen Visualisierung und Weitergabe zu schaffen.

Im Rahmen des Vortrags sollen mit PotreeConverter und PotreeViewer zwei Anwendungen aus dem OpenSource Bereich vorgestellt werden, die in Kombination mit einem simplen FrontEnd das Potential bergen, 3D-Punktwolken zugänglicher zu machen. Durch Anpassungen am Viewer können außerdem unterschiedliche Nutzergruppen bedient werden (bspw. mobile Anwender, deren Web-Ansicht der Punktwolke so vorkonfiguriert ist, dass Punktwolken nur in Ausschnitten geladen werden, sodass Datenvolumen eingespart werden kann).

Güren Tan Dinga

Parkplatzzählung und Parkraumanalysen auf OSM-Basis

ALEX SEIDEL

Daten über die Anzahl und Verteilung von Kfz-Stellplätzen im Stadtraum stellen eine wertvolle Ressource für die Stadt- und Verkehrsplanung oder für Mobilitätsanwendungen dar. Im Zuge der Verkehrswende sind solche Daten auch für die Zivilgesellschaft interessant geworden, schließlich wird zunehmend politisch um die Neuverteilung des Straßenraums gestritten und die Reduzierung des ruhenden Verkehrs ist in diesem Zusammenhang als wichtiger Ansatzpunkt für mehr Flächengerechtigkeit identifiziert worden.

Vierorts gibt es jedoch noch gar kein systematisches Wissen, wo es wie viele Parkplätze gibt. In aufwendigen Studien müssen diese Daten bei Bedarf erst erfasst werden – und meist sind diese Daten anschließend nicht für die Öffentlichkeit zugänglich. Im Gegensatz dazu stellt OpenStreetMap eine optimale Umgebung dar, in der solche Daten frei zugänglich gesammelt und analysierbar gemacht werden können. Mit einer Parkraumanalyse für den Berliner Ortsteil Neukölln haben wir demonstriert, wie urbaner Parkraum systematisch auf OSM-Basis kartiert und mit QGIS und unter Einbezug weiterer offener Daten hochaufgelöst ausgewertet werden kann.

Einen ausführlichen Blick auf die Ergebnisse bietet die Neuköllner Parkraumkarte [1], in der die Parkplatzinformationen visualisiert werden, die Daten zum Download bereit stehen und bei Bedarf ausführlichere Infos zur Herangehensweise und Methodik zu finden sind [2].

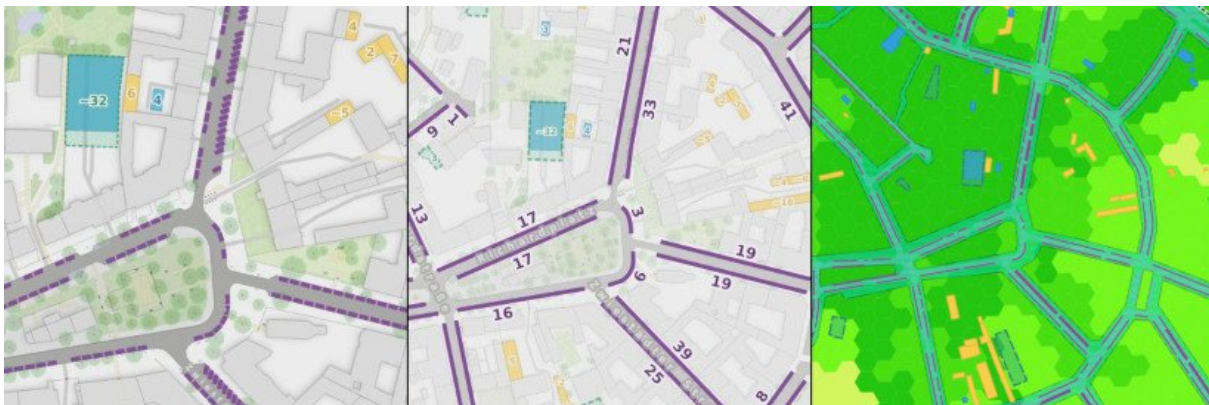


Abb. 1: Ausschnitt der Parkraumkarte, die auf verschiedenen Zoomstufen verschiedene Ergebnisse visualisiert: Von einzelnen Stellplätzen über straßenzugweise Stellplatzzahlen bis hin zur Stellplatzdichte und dem Flächenverbrauch durch geparkte Fahrzeuge.

Herangehensweise und Methodik

Für die Parkraumanalyse wurden zunächst systematisch alle straßenbegleitenden Parkplätze (Parkstreifen am Straßenrand) in OSM kartiert – ebenso wie andere relevante Objekte wie Grundstückseinfahrten und Informationen zu Gehwegübergängen (da dort nicht geparkt werden darf). Das zur Erfassung straßenbegleitenden Parkens etablierte OSM-Schema ist das parking:lane-Schema: Ansatz dieses Schemas ist es, einem Straßensegment jeweils Informationen zum Parken am linken und rechten Fahrbahnrand zuzuordnen (insbesondere Park- und Halteverbote oder Anordnung der geparkten Fahrzeuge). Anschließend wurden die OSM-Daten, teilautomatisiert durch Python-Skripte, in QGIS verarbeitet und dabei vor allem diese Arbeitsschritte durchlaufen:

Parkplatzzählung und Parkraumanalysen auf OSM-Basis

- Entsprechend der Fahrbahnbreite (entweder am Straßenobjekt hinterlegt oder aus seinen Attributen abgeschätzt) werden die räumlichen Verläufe der Parkstreifen (durch Versetzen) mit ihren jeweils spezifischen Eigenschaften abgeleitet.
- Bereiche, an denen ein Park- bzw. Halteverbot besteht oder die sich entsprechend ihrer baulichen Anlage nicht zum Parken eignen, können anschließend aus den Daten ausgeschlossen werden (z.B. an Kreuzungen und Fußgängerüberwegen, vor Einfahrten, an Bushaltestellen).
- Objekte innerhalb der Parkstreifen wurden vor der Auswertung systematisch kartiert und können so mit Sicherheitsabständen berücksichtigt werden (z.B. Straßenbäume, Laternen, Bordsteinstrukturen, Poller und andere Straßenmöbel).
- Das Ergebnis wurde anschließend auf Fehler überprüft und teils aufwendig manuell nachbearbeitet, um die Präzision der Ergebnisse zum Zweck der detaillierten Darstellung in einer Parkraumkarte zu erhöhen. Die allgemeine Aussagekraft der Ergebnisse ist (bei ausreichend guter Kartierung der wesentlichen Objekte) aber auch ohne eine solche manuelle Nachbearbeitung gegeben – sie ist daher grundsätzlich verzichtbar.
- Abschließend wurden Stellplatzkapazitäten für zusammenhängende Parkstreifensegmente berechnet (Quotient aus der Länge eines Segments und dem Abstand der dort geparkten Fahrzeuge, je nach Ausrichtung des Parkens).

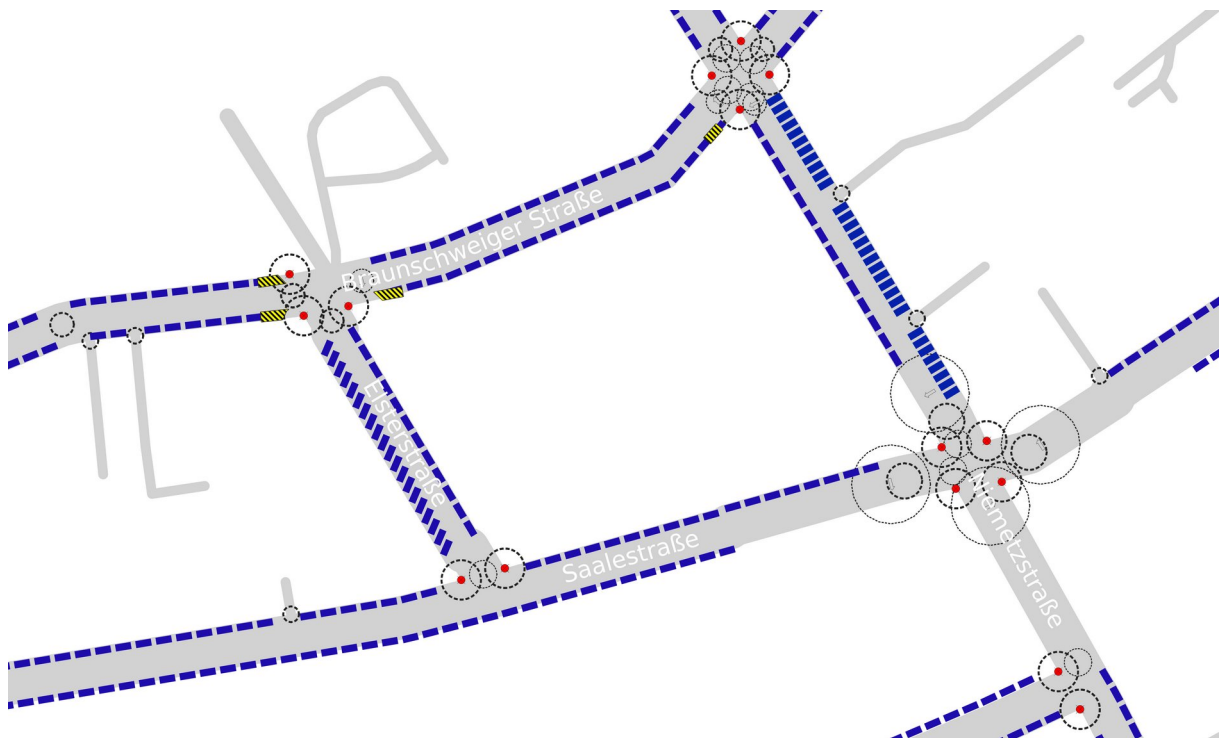


Abb. 2: Generierung von Parkstreifensegmenten und Park-/Halteverbotszonen (kleiner Ausschnitt des Testgebiets und provisorische Visualisierung während des Berechnungsprozesses).

Die Neuköllner Parkraumanalyse berücksichtigt darüber hinaus auch Informationen zu (meist privaten) Stellplätzen abseits des Straßenraums, die ebenso systematisch erhoben wurden. Dazu zählen allgemeine, meist ebenerdige Park- und Stellplätze, Tiefgaragen, Garagen und Carports sowie Parkhäuser. Da viele dieser Stellplätze nicht oder nur schwer aus dem öffentlichen Raum zugänglich sind, wurden für die Erhebung einige externe Geodatensätze einbezogen, die in Berlin in guter Qualität und frei verwendbar vorliegen (insbesondere Luftbilder und das Amtliche Liegenschaftskatasterinformationssystem ALKIS).

Parkplatzzählung und Parkraumanalysen auf OSM-Basis

Für das über 20 km² große Testgebiet lagen schließlich Parkrauminformationen für ein Straßennetz von etwa 170 km Länge vor sowie etwa 2.200 in OSM gemappte Grundstückseinfahrten und über 2.000 Stellplatzflächen abseits des Straßenraums (geschätzter Arbeitsaufwand bei ausreichendem Detailgrad der Erfassung: Etwa eine Stunde pro Straßenkilometer).

Zentrale Zahlen und Ergebnisse

Viele Ergebnisse und Zahlen des Projekts sind in der Neuköllner Parkraumkarte dargestellt. Für den Ortsteil Neukölln, ein großstädtisches Wohnquartier mit etwa 165.000 Einwohnenden und einer vergleichsweise niedrigen Kfz-Besitzquote, ergaben sich insgesamt über 27.300 Kfz-Stellplätze im öffentlichen Straßenraum. Dazu kommen noch einmal etwa 12.200 Stellplätze abseits des Straßenraums, die sich dauerhaft zum Parken eignen (sowie 8.100 nicht zum dauerhaften Parken geeignete Stellplätze wie Mitarbeiter- oder Kundenparkplätze und knapp 430 ungenutzte Stellplätze, beispielsweise in leerstehenden Tiefgaragen). Werden zwei Gewerbegebiete am Rande des Stadtteils außen vor gelassen – also nur die Wohnquartiere berücksichtigt – stehen zusammen 35.447 Stellplätze zum dauerhaften Parken zur Verfügung, denen 33.513 angemeldete Kraftfahrzeuge im gleichen Gebiet gegenüber stehen: Theoretisch stehen für jedes Kraftfahrzeug also 1,08 Stellplätze zur Verfügung.

Die Parkraumanalyse umfasst auch eine kleinräumige Berechnung von Stellplatzdichten, wofür ein eigenes Bevölkerungs- und Kfz-Datenmodell auf Grundlage von externen geografischen und demografischen Daten sowie Kfz-Meldedaten entwickelt wurde. Damit lassen sich die verfügbaren Stellplätze in einem kleineren Gebiet oder im Umkreis um einen Wohnort mit den tatsächlich dort zugelassenen Fahrzeugen ins Verhältnis setzen. Für die Parkraumanalyse wurden diese Stellplatzdichten für eine Nahdistanz von 350 Metern (also eine Entfernung, die innerhalb von 3 bis 4 Minuten fußläufig erreichbar ist) um einen Wohnort berechnet: Im Durchschnitt ergibt sich dabei für die Neuköllner Wohnquartiere eine Anzahl von 835 Stellplätzen (604 davon im öffentlichen Straßenraum) und eine Zahl von 759 zugelassenen Kraftfahrzeugen. Das Verhältnis schwankt jedoch stark zwischen verschiedenen Gebieten.

Im Zuge der Verkehrswende und Debatten über lebenswertere Städte stehen die am Straßenrand geparkten Fahrzeuge zunehmend im Fokus, da sie sich durch einen dauerhaften, vergleichsweise hohen Flächenverbrauch auszeichnen. Allein für die Neuköllner Wohnquartiere kann dieser Flächenverbrauch im öffentlichen Straßenraum auf eine Fläche von 327.000 m² beziffert werden, was 19 Prozent des öffentlichen Verkehrsraumes und 4,4 Prozent der Gesamtfläche des Gebiets entspricht. Parkplätze und Stellflächen abseits des Straßenraumes beanspruchen darüber hinaus noch einmal zusätzlich 171.000 m² (2,3 Prozent der Gesamtfläche des Gebiets). In die obligatorischen Fußballfelder umgerechnet bedeutet dies, dass 70 Fußballfelder allein in den Wohnquartieren Neuköllns für stehende Fahrzeuge in Anspruch genommen werden, 46 davon im öffentlichen Straßenraum.

Bewertung von Unsicherheitsfaktoren und Erkenntnisse für die Mapping-Praxis

Die Parkraumanalyse basiert auf einem interpolativen Datenmodell: Die Stellplätze werden nicht gezählt, sondern unter bestimmten Annahmen aus Straßeneigenschaften abgeleitet. Viele der zu Grunde liegenden Annahmen und Ergebnisse können in der tatsächlichen Realität überprüft, gezählt oder gemessen werden, um die Qualität der Ergebnisse zu überprüfen.

Die automatisiert erzeugten Parkstreifendaten wurden einer aufwendigen Nachbearbeitung unterzogen, allerdings wichen die Daten quantitativ nur um 0,6 Prozent voneinander ab. Mit einer Zählung von über 1.500 Stellplätzen wurden die interpolierten Daten mit der Realität verglichen. Dabei zeigte sich mit einer Abweichung von unter einem Prozent zwar eine insgesamt sehr hohe Übereinstimmung, allerdings gab es vereinzelte Straßenabschnitte mit hoher Abweichung: Insbesondere, wenn Schräg- oder Querparken angeordnet ist, addieren sich ungenau erhobene Daten schnell zu größeren Abweichungen.

Parkplatzzählung und Parkraumanalysen auf OSM-Basis

Darüber hinaus gibt es eine Reihe anderer Faktoren, die zu einer Über- oder Unterschätzung der realen Parkraumsituation im Vergleich zum Datenmodell führen können. So bildet das Datenmodell eine StVO-konforme Situation ab; in der Realität ist jedoch häufiges Falschparken zu beobachten, wofür in einem Beispielszenario ein Wert von mindestens 6 Prozent mehr „Stellplätzen“ ermittelt wurde. Andererseits fehlen in der zugrunde gelegten Meldestatistik insbesondere viele Dienst- und Mietwagen, wobei von einer Größenordnung von etwa 5 Prozent ausgegangen werden kann. Ein weiterer Faktor: Stellplatzangaben außerhalb des öffentlichen Straßenraums müssen als theoretische Stellplatzpotentiale verstanden werden und nicht als tatsächlich genutzte Kfz-Stellplätze, da ihre tatsächliche Auslastung meist unbekannt bleibt.

Die Parkraumanalyse hat gezeigt, dass mit einer genauen Datenerfassung eine äußerst präzise Wiedergabe der realen Parkraumsituation erreicht werden kann. Die sehr geringen Abweichungen zwischen der aus OSM erzeugten Interpolation und der Realität dürften vor allem auf den recht hohen Detailgrad der OSM-Daten im Testgebiet zurückzuführen sein. Während das dafür notwendige Micro-Mapping einigen OSM-Beitragenden fremd sein dürfte, zeigen sich in einem Projekt wie diesem, wie wertvoll detaillierte OSM-Daten sein können. Das „Zerstückeln“ von Straßen ist dabei aber nur selten nötig, da sich viele relevante Informationen aus anderen Objekten im Raum ableiten lassen, wenn diese gut erfasst sind.

Kontakt zum Autor:

Alex Seidel
OpenStreetMap-Beitragender aus Berlin-Neukölln
OSM-Account: Supaplex030
E-Mail: supaplex@riseup.net

Verweise

[1] Parkraumkarte Neukölln: Online verfügbar unter <https://supaplexosm.github.io/strassenraumkarte-neukoelln/?map=parkingmap>

[2] *Seidel, Alexander und OpenStreetMap-Beitragende*: Parkraumanalyse für den Berliner Ortsteil Neukölln – Methoden- und Ergebnisbericht, Berlin 2021. Online verfügbar unter <https://supaplexosm.github.io/strassenraumkarte-neukoelln/parkraumkarte/methodenbericht-2021-03.pdf>

Import detaillierter Daten zur barrierefreien Nutzung von Bahnhöfen und Haltestellen

Zur Auskunft der barrierefreien Nutzung des Bus- und Schienenpersonennahverkehrs werden 2021 umfangreiche Daten an Bahnhöfen und Haltestellen erhoben. Diese sollen in OpenStreetMap importiert und die Schritte dazu vorgestellt werden: Diskussion um Erweiterung von OSM-Tags und -Werten, Importantrag stellen und mit OSM-Community sprechen, Importteam bilden und den Import mit Qualitätswerkzeugen vorstellen.

Die öffentlichen Aufgabenträger für Bus- und Schienenpersonennahverkehr müssen bis zum 1.1.2022 u.a. die Auskunft geben können, welche Bahnhöfe und Bushaltestellen für Menschen mit Mobilitätseinschränkungen nutzbar sind und welche nicht. Dazu werden umfangreiche Daten vor Ort erhoben und sollen in OpenStreetMap importiert werden, damit anschliessend u.a. ein Routing für Menschen mit verschiedenen Einschränkungen bis zum Einstieg in die Verkehrsmittel ermöglicht wird.

Es werden die Schritte vorgestellt, die notwendig sind, um den Import durchzuführen: welche OSM-Tags sollten ergänzt oder OSM-Werte verfeinert werden, die Diskussion zum Tagging und zum Import mit der OSM-Community, Bildung eines Importteams, die Beschreibung des eigentlichen Imports sowie Qualitätssicherungsdienste. Die erhobenen Daten und umfangreiches Bildmaterial werden der Öffentlichkeit unter einer offenen Datenlizenz bereitgestellt.

Dietmar Seifert

Quellen- und Lizenzangabe(n) – Vorgaben der Open Database License (ODbL)

FALK ZSCHEILE

Einleitung

Wenn man sich mit Quellen- und Lizenzangaben bei der Nutzung digitaler Güter beschäftigt, so sind sich die meisten Menschen bewusst, dass es hier etwas zu beachten gibt. Gleichzeitig besteht jedoch häufig das Gefühl, sich hier mit einer Materie beschäftigen zu müssen, die zum lästigen Übel gehört, egal ob man etwas mit Dateninhalten machen will. Dies betrifft proprietäre Inhalte und Daten genauso, wie offene bzw. freie Inhalte und Daten. Dieser lästigen Pflicht entziehen sich Entwickler häufig durch eine einfache menschliche Heuristik: Sie schauen, wie machen es denn die anderen. Das ist, wie ein erster Blick auf online Karten, die aus OpenStreetMap-Daten hergestellt wurden, zeigt. Von entscheidender Bedeutung sind daneben auch Hilfestellungen, die vom Lizenzgeber in diesem Zusammenhang gegeben werden.

Im OpenStreetMap Projekt wird die Pflicht zur Quellen- und Lizenzangabe in verschiedenen Projekten sehr unterschiedlich gehandhabt, wie die nachfolgende Aufstellung (ohne Anspruch auf Vollständigkeit oder Repräsentativität) zeigt:

Auf der Seite von www.openstreetmap.org liest man beispielsweise das Folgende[10]:

OpenStreetMap-Mitwirkende Spenden. Nutzungsbedingungen der Website und API

Auf der deutschen Seite (www.openstreetmap.de/karte.html), die vom FOSSGIS e.V. als Local Chapter von OpenStreetMap in Deutschland betrieben wird, findet sich der folgende Hinweis[2]:

Karte hergestellt aus OpenStreetMap-Daten | Lizenz: Open Database License (ODbL)

Im Projekt ÖPNV-Karte (www.opnvkarte.de) findet sich der folgende Hinweis auf der Seite[7]:

Map data by OpenStreetMap contributors licenced under ODbL 1.0. This map is available under CC-BY-SA licence (by MeMoMaps).

Das Projekt OpenTopoMap (www.opentopomap.org) hat auf ihrer Seite den folgenden Hinweis angebracht[13]:

Kartendaten: OpenStreetMap-Mitwirkende, SRTM | Kartendarstellung OpenTopoMap (CC-BY-SA)

Die Wanderreitkarte von Nop (www.wanderreitkarte.de) enthält die folgenden Angaben[8]:

Nops Reit-& Wanderkarte Daten OSM DEM CIAT

Die Routinganwendung Open Route Service (maps.openrouteservice.org) verwendet auf seiner Seite wiederum andere Angaben[4]:

Leaflet | OpenStreetMap contributors

Die Vorgaben der Open Database License (ODbL) werden also durchaus vielseitig interpretiert bzw. angewendet.

Quellen- und Lizenzangabe(n) – Vorgaben der Open Database License (ODbL)

Welchen Funktionen einer Namens- Quellen- und Lizenzangabe zukommt und welche Umsetzung die ODbL fordert, dem soll im Weiteren nachgegangen werden.

Allgemein rechtliches zu Namens- und Quellenangaben

Welche Bedeutung beziehungsweise Funktion kommt Namens-, Quellen- und Lizenzangaben bei digitalen Erzeugnissen zu und warum sind diese von so großer Wichtigkeit? Diesen allgemeinen Fragestellungen soll zum besseren Verständnis zunächst nachgegangen werden.

Eigenschaften digitaler Güter

Eine der wichtigsten Eigenschaften digitaler Informationen, von denen sie sich von den greifbaren Gegenständen der Realwelt unterscheiden, ist, dass sie sich beliebig vervielfältigen, übertragen und speichern lassen. Anders als ein dinglicher Gegenstand können digitale Informationen ohne Qualitätsverlust dupliziert und weitergegeben werden. Den Besitzer eines dinglichen Gegenstandes, der häufig auch Eigentümer ist, erkennt man bei dinglichen Gütern an der tatsächlichen Herrschaft Gewalt über einen Gegenstand. In der dinglichen Welt knüpft das Recht an diesen ersten Anschein des Besitzes auch die Vermutung des Eigentums.

In der Welt der digitalen Güter fehlt es aufgrund der beliebigen Möglichkeit zur Vervielfältigung und Verbreitung an einem solchen Rechtsschein. Folgerichtig gewährt das Recht dem Besitzer einer digitalen Information im Rechtsverkehr nicht den Anschein der Eigentümerstellung bzw. Urheberschaft. Das Gegenteil ist der Fall. Befinden sich an einem digitalen Gute keine Informationen zum Ursprung oder zu den damit verbundenen Rechten, so ist das digitale Gut im Zweifel als geschützt anzusehen, § 31 Abs. 5 UrhG (sog. Zweckübertragungslehre). Die Konsequenz daraus ist, dass außer einer Kenntnisnahme keine weitere Nutzung oder Verwendung erlaubt ist. Das digitale Gut darf also weder vervielfältigt, verbreitet oder öffentlich zugänglich gemacht werden. Das Einbinden in fremde Webseiten mittels sog. Framing stellt jedoch keine in diesem Sinne erlaubnispflichtige Nutzung dar.

Beim Umgang mit digitalen Gütern ist die Namens-, Quellen- und Lizenzangabe damit eine wichtige Informationsquelle für einen potentiellen Nutzer, um sich über die rechtlichen Rahmenbedingungen zu informieren.

Namens- und Quellenangaben im Urheberrechtsgesetz

Dies ist der Punkt, an dem es sich lohnt, sich näher mit der Funktion der Namens- und Quellennennung im Urheberrechtsgesetz auseinanderzusetzen.

Die Namensnennung im deutschen Urheberrechtsgesetz hat ganz unterschiedliche Funktionen. Im Bereich der persönlich-geistigen Schöpfungen, also dem eigentlichen Urheberrecht, das sich mit der rechtlichen Behandlung kreativer Werke auseinandersetzt, ist die Namensnennung ein wichtiger Aspekt des Urheberpersönlichkeitsrechts, § 12 ff. UrhG. Aus ihm folgt das Recht des Urhebers auf Anerkennung seiner Urheberschaft und das Recht des Urhebers selber zu bestimmen, ob und in welcher Form er als Urheber eines Werkes benannt wird, § 13 UrhG. Dabei ist zu beachten, dass das Namensnennungsrecht, anders als die kommerzielle Verwertung geschützter Werke (Schranken des Urheberrechts), § 44a ff. UrhG, keinen gesetzlichen Beschränkungen unterliegt.^[1] Die Begrenzungen des Namensnennungsrechtes müssen daher vertraglich vereinbart werden und ergeben sich nicht unmittelbar aus dem Urheberrechtsgesetz.

Neben diesen für den Urheber als Persönlichkeit interessanten Aspekten treten aber auch wichtige Rechtsfolgen, die das Urheberrechtsgesetz an eine Namensnennung knüpft. Zu nennen ist hier zu allererst die Vermutung der Urheberschaft zugunsten des auf dem Werk genannten. Fehlt es an eine Urhebernennung auf dem Werk, so muss der Urheber, wenn er ein Recht aus seinem Urheberrecht

Quellen- und Lizenzangabe(n) – Vorgaben der Open Database License (ODbL)

geltend macht, zunächst beweisen, dass er der Urheber ist. Das Risiko des Nachweises trifft also den Urheber. Diese Nachweispflicht entfällt, wenn der Urheber bereits auf dem Werk genannt wurde. Nun ist es an demjenigen, der die Rechte bestreitet, zu beweisen, dass die genannte Person nicht der Urheber ist und diesem die geltend gemachten Rechte nicht zustehen. Das Risiko eines misslungenen Beweises trifft also den Prozessgegner des Urhebers. Die Juristen sprechen in diesem Zusammenhang von einer Beweislastumkehr, vgl. § 10 Abs. 1 UrhG.

Schließlich ist die Namensangabe, das Gesetz spricht in diesem Fall von Quellenangabe, notwendig, wenn man ein durch das Urheberrecht geschütztes Werk im Rahmen einer gesetzlichen Erlaubnis nutzt, vgl. § 63 Abs. 1 UrhG.

Namens- und Quellenangaben bei Datenbanken

Im vorangegangenen Abschnitt wurde zunächst auf die Vermutung der Urheberschaft im Zusammenhang mit einer persönlich-geistigen Schöpfung (Urheberrecht) behandelt. Bei der OpenStreetMap-Datenbank handelt es sich jedoch nicht um eine persönlich-geistige Schöpfung, sondern um eine Datenbank im Sinne des Datenbankherstellerechts, §§ 87a ff. UrhG. Bei dem Datenbankherstellerecht handelt es sich um einen Schutz für eine Datenbank, der aufgrund der erfolgten Investitionen rechtlicher Schutz gewährt wird. Da es sich hierbei also nicht um den klassischen Urheberrechtsschutz handelt, spricht man beim Datenbankherstellerecht von einem sogenannten verwandten Schutzrecht. Jedoch wird im Zusammenhang mit den rechtlichen Regelungen zum Datenbankherstellerecht auf die Vermutungsregel zu Gunsten des benannten Urhebers verwiesen, § 87a Abs. 2 UrhG i.V.m. § 10 Abs. 1 UrhG. Damit gilt die Beweislastumkehr auch zu Gunsten des Datenbankherstellers, wenn dieser im Zusammenhang mit der Datenbank genannt wird.

Namens- und Quellenangaben außerhalb des Urheberrechtsgesetzes

Neben den soeben skizzierten Funktionen der Namensnennung im deutschen Urheberrecht gibt es jedoch weitere Funktionen und Traditionen.

Der klassische Copyrightvermerk ([Jahr] [Rechteinhaber]), wie man ihn an vielen Stellen antrifft, ist beispielsweise ein Überbleibsel aus den Zeiten, als das Copyright US-amerikanischen Recht noch ein Registerrecht war. Heutzutage ist der Copyrightvermerk auch in den USA keine Schutzvoraussetzung mehr für ein Werk. Dem Copyrightvermerk kommt im US-amerikanischen Recht jedoch eine ähnliche Bedeutung zu, wie die Beweislastumkehr im deutschen Recht bei Benennung des Urhebers auf dem Werk. Der Copyrightvermerk bringt beweisrechtliche Vorteile und kann einen höheren Schadenersatz im Falle einer festgestellten Copyrightverletzung zur Folge haben[14].

Namens- und Quellenangaben bei Open Source Software und Open Data

Etwas anders gelagert ist die Funktion der Namensnennung im Rahmen der Open Source Software-entwicklung und bei Open Data. Die Nennung der beteiligten Softwareentwickler oder beitragenden Personen ist hier wichtig für die Reputation. Indem eine Person hier besonders hervortritt ergeben sich für diese Person Anerkennung in der Community und gegebenenfalls auch bessere berufliche Chancen, weil sie ihr Können unter Beweis gestellt hat. Daneben dient die Namensnennung hier aber auch als Anerkennung einer Leistung im Sinne eines Dankes an die beteiligten Personen. Wer der Community etwas kostenlos zur freien Nutzung zur Verfügung stellt, soll in diesem Zusammenhang zumindest dankend erwähnt werden.

Die Namens- bzw. Quellennennung ist Bestandteil aller freien Lizenzen in unterschiedlichen Ausprägungen, vergleiche zum Beispiel:

- [Ziff. 1 GPL 2.0](#)
- [Section 3 a. 1. A. cc-by-sa 4.0](#)

Quellen- und Lizenzangabe(n) – Vorgaben der Open Database License (ODbL)

- [Ziff. 4.3 ODbL](#)

Wird ein immaterielles Gut, dass unter einer freien Lizenz steht, weitergegeben, so darf bei dieser Weitergabe in der Regel die Namens- bzw. Quellenangabe nicht verloren gehen. Für alle nachnutzenden Personen soll erkennbar sein, wer bisher mitgewirkt hat. Bei großen Projekten wie OpenStreetMap, wo viele Personen sehr kleine Beiträge beisteuern, wird nicht auf die einzelnen Mitwirkenden in der Quellenangabe verwiesen, sondern diese gegebenenfalls insgesamt benannt. Die einzelnen Mitwirkenden sind bei OpenStreetMap jedoch aus der Datenbank ersichtlich.

Allgemeinlizenzrechtliches

In den bisherigen Ausführungen wurde zunächst lediglich auf die Namens- bzw. Quellenangabe im Zusammenhang mit einem digitalen Gut eingegangen. Die Lizenzangabe hat hierbei bisher keine Rolle gespielt. Die Pflicht zur Lizenzangabe ist eine Besonderheit der offenen und freien Lizenzen.

Lizenzangabe im klassischen Urheberrecht

Wie oben bereits dargestellt, geht das normale Urheberrecht von einem prinzipiellen und generellen Schutz aus. Ein wichtiges Ziel des Urheberrechtsgesetzes ist es, dem Rechteinhaber die umfängliche insbesondere auch kommerzielle Verwertung der ihm zustehenden Nutzungsrechte zu ermöglichen. Insoweit ist es folgerichtig, dass es ausschließlich Sache des Rechteinhabers ist, anderen mitzuteilen, ob und in welchem Umfang ein immaterielles Gut genutzt werden darf.

Wie oben bereits geschildert, führt der urheberrechtliche Grundsatz der Zweckübertragungslehre dazu, dass ein immaterielles Gut nicht weiterverwendet werden darf, wenn keine Nutzungsrechte (Lizenz) angegeben sind. Die Nutzungsrechte verbleiben beim Rechteinhaber. Ohne Nutzungsrechte dürfen Dritte aber ein Werk nicht nutzen.

Die Rechteeinräumung erfolgt hier in der Regel über individuelle vertragliche Regelungen, in denen Bezahlung und Umfang der Nutzungsrechteinräumung geklärt werden. Hierbei wird die Weiter- oder Unterlizenzierung regelmäßig ausgeschlossen. Insoweit besteht – anders als dies bei Namens- und Quellenangaben der Fall sein kann – kein Bedürfnis, die geltenden rechtlichen Regeln bekannt zu machen. Diese sind alleinige (vertrauliche) Sache der Vertragsparteien.

Lizenzangabe bei freien Lizenzen

Anders ist es im Bereich der freien und offenen Lizenzen. Bei offenen und freien Lizenzen, insbesondere jedoch bei den freien Lizenzen, soll die Weiternutzung gerade garantiert und gesichert sein. Dementsprechend verlangen freie Lizenzen von jedem Lizenznehmer, dass dieser Angaben zur Lizenz macht, sodass jeder weitere potentielle Nutzer darüber informiert ist, unter welchen rechtlichen Bedingungen die Weiternutzung erfolgen darf. Dieses sogenannte Copyleft oder Share Alike sorgt also dafür, dass die Freiheit des geschaffenen immateriellen Gutes erhalten bleibt, wie es Intention der Person war, die das Gut ursprünglich unter eine freie Lizenz gestellt hat.

Auslegung von Lizenztexten

Lizenzen sind ihrem Wesen nach nichts anderes als Verträge, die zwischen Lizenzgeber und Lizenznehmer geschlossen werden. Wie bei allen Verträgen sind solche Vertragsbedingungen auslegungsfähig und häufig auch auslegungsbedürftig. So gibt die OpenStreetMap Foundation zur Interpretation der Open Database License (ODbL) sogenannte Community Guidelines[3] und FAQs[12] heraus.

Für den Juristen ergeben sich bei der Auslegung zusätzliche Herausforderungen. Zum einen ist für die Lizenztextauslegung die Sprache des Lizenztextes maßgebend[5]. Für die ODbL wäre dies also die englische Sprache. Der internationale Kontext eines Lizenztextes bedeutet für die Auslegung eine zusätzliche unbekannte. Der gleiche Inhalt wird unter Umständen in unterschiedlichen Rechtskreisen

Quellen- und Lizenzangabe(n) – Vorgaben der Open Database License (ODbL)

verschieden interpretiert. Das Ganze wird zudem noch durch das in Deutschland maßgebliche Schutzlandprinzip überlagert. Das bedeutet, dass zur Auslegung des Lizenztextes zwar die englische Originalversion maßgebend ist, der Schutz jedoch auf Basis des im jeweiligen Land gültigen Urheberrechts zu beurteilen ist[6]. Für die ODbL bedeutet dies, dass zwar der englische Vertragstext verbindlich ist. Ob und wie ein unter ODbL lizenziertes Gut geschützt ist, beurteilt sich jedoch nach dem deutschen Urheberrechtsgesetz. Auf Details soll hier im Folgenden jedoch nicht eingegangen werden, sondern nur die Grundfunktionen der ODbL im Zusammenhang mit Namens-, Quellen- und Lizenzangaben in den Blick genommen werden.

Die Open Database License (ODbL) und die OpenStreetMap-Datenbank

Quellen- und Lizenzangabe nach der Open Database License

Bei der Open Database License handelt es sich um eine freie Datenbanklizenz. Eine Besonderheit dieser Lizenz ist es, dass sie zwischen dem Inhalt der eigentlichen Datenbank und einem daraus erzeugten Produkt unterscheidet. Die ODbL mit dem in ihr enthaltenen Share Alike gilt im Wesentlichen nur für den Datenbankinhalt und nicht für das daraus erzeugte Produkt. Das Produkt kann abweichend – auch proprietärer (!) – lizenziert werden[15]. Um jedoch den Nutzen eines Produktes die Möglichkeit zu geben, die freien Daten zu nutzen und ein vergleichbares oder besseres Produkt selbst zu erzeugen, verlangt die ODbL zumindest einen Hinweis auf die Herkunft der Daten und deren Lizenzierung unter der Open Database License. Nur so können Dritte erkennen, dass das Produkt in ihren Händen aus freien (OpenStreetMap-) Daten hergestellt wurde.

In Ziffer 4.3 ODbL[9] ist zur Quellen- und Lizenzangabe Folgendes vermerkt:

Creating and Using a Produced Work does not require the notice in Section 4.2. However, if you Publicly Use a Produced Work, You must include a notice associated with the Produced Work reasonably calculated to make [...] aware that Content was obtained from the Database, [...], and that it is available under this License.

Die Nutzung eines Produkts erfordert nicht den Hinweis aus Abschnitt 4.2. Wenn Sie ein Produkt jedoch öffentlich nutzen, müssen Sie einen Hinweis im Zusammenhang mit dem Produkt anbringen, der angemessen darauf abzielt, [...], darauf aufmerksam zu machen, dass der Inhalt aus dieser Datenbank, [...] entnommen wurde und dass er unter dieser Lizenz erhältlich ist.

In Ziffer 4.3 ODbL[9] wird zudem ein Beispiel genannt, wie eine Quellen- und Lizenzangabe aussehen könnte:

Example notice. The following text will satisfy notice under Section 4.3:

Contains information from DATABASE NAME, which is made available here under the Open Database License (ODbL).

Beispielhinweis. Der folgende Text erfüllt die Anforderung aus Abschnitt 4.3:

Enthält Informationen aus DATENBANKNAME und ist hier unter der Open Database License (ODbL) verfügbar.

Daraus würde sich für die Verwendung von OpenStreetMap Daten der folgende Lizenzhinweis ergeben:

Quellen- und Lizenzangabe(n) – Vorgaben der Open Database License (ODbL)

Map data by OpenStreetMap licensed under ODbL 1.0.

Quellen- und Lizenzangabe unter Berücksichtigung der Grundsätze bei Immaterialgütern

Mit der Angabe "Map data by OpenStreetMap licensed under ODbL 1.0." sind jedoch zunächst nur die formalen Anforderungen der ODbL, mit ihrer Pflicht zur Angabe der Lizenz und Nennung der Datenbank, für welche die Lizenz gilt, erfüllt. Es wird auf die freien Daten, die dem Produkt zugrunde liegen, aufmerksam gemacht. Wie oben bereits geschildert, trennt die ODbL jedoch zwischen Datenbank und dem daraus erzeugten Produkt. Das bedeutet, dass das Produkt seinerseits einer frei wählbaren Lizenzierung zugänglich ist. Erfolgt im Zusammenhang mit dem Produkt, also beispielsweise Kartenkacheln, keine Lizenzangabe zum Produkt, so bedeutet dies nach deutschem Urheberrecht, dass eine Vervielfältigung, Verbreitung und öffentliche Zugänglichmachung ausgeschlossen ist! Dies gilt nicht für die Einbindung von OpenStreetMap-Karten im Rahmen des sogenannten Framing. Beim Framing handelt es sich um eine ohne Erlaubnis des Rechteinhabers zulässige Nutzung des Internets.

Sollen also die Kartenkacheln ihrerseits frei genutzt werden können, so muss hierfür ebenfalls eine freie oder offene Lizenz angegeben werden. Andernfalls verbleiben die Rechte an dem erzeugten Produkt nach den Grundsätzen der Zweckübertragungslehre beim Hersteller des Produktes!

Um eine freie Weiternutzung beispielsweise unter der Namensnennung - Weitergabe unter der "Creative Commons Lizenz gleichen Bedingungen 4.0 International" (CC BY-SA 4.0) zu ermöglichen, so müsste die Angabe auf der Kartenseite wie folgt lauten:

Map data by OpenStreetMap licensed under ODbL 1.0. Map by [Kartenersteller] is available under CC-BY-SA 4.0.

Eine kürzere Version, die ebenfalls den Zweck noch erfüllen würde, wäre nach hier vertretener Auffassung:

Data by OpenStreetMap (ODbL), Map by [Kartenersteller] (CC-BY-SA 4.0).

Auch OpenStreetMap könnte noch als OSM abgekürzt werden. Bei der Verwendung von Abkürzungen sollten diese aber mit einem entsprechenden Link auf das Projekt oder die Lizenz versehen werden, damit auch Personen, die mit der Materie nicht vertraut sind, sich die Bedeutung erschließen können.

Empfehlungen der OpenStreetMap Foundation zur Quellen- und Lizenzangabe

Nimmt man nun die Quellen- und Lizenzangabe der OpenStreetMap Foundation unter www.openstreetmap.org in den Blick,

"OpenStreetMap-Mitwirkende Spenden. Nutzungsbedingungen der Website und API"

so fällt unmittelbar auf, dass diese nicht der von der Ziffer 4.3 ODbL beispielhaft vorgeschlagenen Form entspricht.

Mit dem Hinweis auf die Datenbank als Datenquelle ist mit "OpenStreetMap Contributors" mittelbar auch ein Hinweis auf die OpenStreetMap-Community verbunden. Rechtlich ist dies nicht geboten, aber andererseits auch nicht schädlich, sondern als Dankeschön an die vielen beitragenden Personen zu verstehen.

Die OpenStreetMap Foundation erläutert ihr Vorgehen bei der Namens- und Lizenzangabe in ihren FAQs. Dort heißt es^[12]:

How should I attribute you?

Quellen- und Lizenzangabe(n) – Vorgaben der Open Database License (ODbL)

Our requested attribution is “ OpenStreetMap contributors”. You must make it clear that the data is available under the Open Database Licence. This can be achieved by providing a “License” or “Terms” link which links to www.openstreetmap.org/copyright or www.opendatacommons.org/licenses/odbl.

[...]

You may optionally qualify the credit to explain what OSM content you are using. For example, if you have rendered OSM data to your own design, you may wish to use “Map data OpenStreetMap contributors”.

Wie soll ich auf OpenStreetMap verweisen?

Die von uns gewünschte Angabe lautet “ OpenStreetMap-Mitwirkende”. Sie müssen deutlich machen, dass die Daten unter der Open Database License verfügbar sind. Dies kann erfolgen, indem ein Link “Lizenz” oder “Nutzungsbedingungen” bereitgestellt wird, der auf www.openstreetmap.org/copyright oder www.opendatacommons.org/licenses/odbl.

[...]

Optional können Sie klarstellen, welche OSM-Inhalte Sie verwenden. Wenn Sie beispielsweise OSM-Daten in Ihrem eigenen Design gerendert haben, können Sie “Kartendaten OpenStreetMap Mitwirkende” verwenden.

Damit wird klar, dass die OpenStreetMap Foundation nicht dem beispielhaften Vorschlag von Ziffer 4.3 ODbL folgt, sondern von der Möglichkeit einer “Regelung im Zusammenhang mit dem Produkt” Gebrauch macht. Also auf eine Unterseite verlinkt.

Zudem fällt auf, dass die Formulierungen im Copyright von www.openstreetmap.org/copyright/[11] und in der FAQ[12] etwas voneinander abweichen. Auf www.openstreetmap.org/copyright/ werden zum Teil strengere Formulierungen gebraucht als in der FAQ.

Dies Umsetzung der OpenStreetMap Foundation ist insoweit ungünstig, als hier bereits ein Sonderfall der ODbL zur Anwendung kommt und die nicht durch die ODbL selbst empfohlene Herangehensweise. Die unter Umständen unreflektierte Übernahme durch Dritte, die nach einer Referenz suchen, kann eine Nebenfolge sein. Die ODbL zielt gerade darauf ab, darüber zu informieren, dass die Daten frei verwendet werden dürfen. Die Umsetzung der OSMF legt den Schwerpunkt dagegen darauf, dass es sich bei der Darstellung um Daten der OpenStreetMap-Community handelt.

Zudem fehlt auf www.openstreetmap.org der Hinweis auf die Lizenz für die dargestellten Kartenkacheln. Nach dem oben Gesagten bedeutet dies, dass diese Kartenkacheln, außer im Rahmen des Framing, durch Dritte nicht weiterverwendet werden dürften. Aus dem Kontext ergibt sich zwar unter Umständen, dass die OpenStreetMap Foundation mit einer freien Weiternutzung einverstanden ist, weil dies den Idealen entspricht, die sie vertritt. Es stellt sich jedoch die Frage, unter welchen Lizenzbedingungen dies gestattet wird. Auf www.openstreetmap.org/copyright wird nur darauf hingewiesen, dass die Dokumentation unter der Creative Commons Lizenz steht[11]:

Our documentation is licensed under the Creative Commons Attribution-ShareAlike 2.0 license (CC BY-SA 2.0).

Quellen- und Lizenzangabe(n) – Vorgaben der Open Database License (ODbL)

Unsere Dokumentation ist unter der "Creative-Commons"-Lizenz "Namensnennung- Weitergabe unter gleichen Bedingungen" 2.0 (CC BY-SA 2.0) verfügbar.

Die Kartendarstellung lässt sich aber nicht ohne weiteres als Dokumentation verstehen. Die OpenStreetMap Foundation sollte daher zumindest in ihren Copyright Hinweisen angeben, unter welcher Lizenz die erstellten Kacheln stehen. Dies zum einen, um den freien Charakter der gezeigten Kacheln zu dokumentieren, zum anderen aber auch, um anderen beispielhaft eine korrekte Anwendung der ODbL im Zusammenhang mit Daten und Datenprodukt zu zeigen. Auf die Möglichkeit einer abweichenden Lizenzierung von erstellten Kacheln weist die OpenStreetMap Foundation in ihrer FAQ ausdrücklich hin[12]:

You may optionally qualify the credit to explain what OSM content you are using. For example, if you have rendered OSM data to your own design, you may wish to use "Map data OpenStreetMap contributors".

Optional können Sie klarstellen, welche OSM-Inhalte Sie verwenden. Wenn Sie beispielsweise OSM-Daten in Ihrem eigenen Design gerendert haben, können Sie "Kartendaten OpenStreetMap Mitwirkende" verwenden.

Für Dritte, die nach einer Referenzimplementierung suchen, ist dieser Hinweis ohne Anwendung der OpenStreetMap Foundation in ihrem eigenen Kartenportal, kaum umsetzbar bzw. auf das eigene Projekt übertragbar. Zumal der entscheidende Hinweis auf die abweichende Lizenzierung der Kartendarstellung im Beispiel fehlt und zudem der Hinweis auf die Open Database License unterbleibt. Dritte werden so unter Umständen sogar in eine falsche Richtung geleitet, weil sie annehmen könnten, der Hinweis auf die Open Database License sei entbehrlich und nur der Hinweis auf das OpenStreetMap Projekt bzw. die Datenbank notwendig.

Im Ergebnis bleibt festzuhalten, dass andere Karten im OpenStreetMap-Projekt die Angabe der Quelle und Lizenzen schöner lösen. Zudem ist die Dokumentation der OpenStreetMap Foundation zur Anwendung der Open Database License verbesserungswürdig.

Kontakt zum Autor:

Falk Zscheile
c/o FOSSGIS e.V.
Bundesallee 23
10717 Berlin

Literatur

- [1] BGH. Urteil vom 16. Juni 1994. Az. I ZR 3/92, Recht des Architekten auf Namensnennung.
- [2] FOSSGIS e. V. www.openstreetmap.de/karte.html. 2021. url: <https://www.openstreetmap.de/karte.html> (besucht am 01. 05. 2021).
- [3] OpenStreetMap Foundation. Licence/Community Guidelines. 29. Aug. 2017. url: https://wiki.osmfoundation.org/wiki/Licence/Community_Guidelines (besucht am 01. 05. 2021).
- [4] HeiGIT gGmbH. maps.openrouteservice.org. 2021. url: <https://maps.openrouteservice.org> (besucht am 01. 05. 2021).

Quellen- und Lizenzangabe(n) – Vorgaben der Open Database License (ODbL)

- [5] Till Jaeger und Axel Metzger. Open Source Software. Rechtliche Rahmenbedingungen der Freien Software. 4. Aufl. München: C. H. Beck, 2016.
- [6] LG München I. Urteil vom 14. Mai 2012. Az. 21 O 14914/09.
- [7] Memomaps. [www.öpnvkarte.de](http://www.xn--pnvkarte-m4a.de). 2021. url: <http://www.xn--pnvkarte-m4a.de> (besucht am 01. 05. 2021).
- [8] Nop. www.wanderreitkarte.de. 2021. url: <https://www.wanderreitkarte.de/> (besucht am 01. 05. 2021).
- [9] Open Data Commons und Open Knowledge Foundation. Open Data Commons Open Database License (ODbL) v1.0. 2021. url: <https://opendatacommons.org/licenses/odbl/1-0/> (besucht am 01. 05. 2021).
- [10] OpenStreetMap Foundation. Copyright and License. 2021. url: <https://www.openstreetmap.org/copyright/en> (besucht am 01. 05. 2021).
- [11] OpenStreetMap Foundation. Licence/Licence and Legal FAQ. How should I attribute you? 29. Okt. 2021. url: https://wiki.osmfoundation.org/wiki/Licence/Licence_and_Legal_FAQ#How_should_I_attribute_you.3F (besucht am 01. 05. 2021).
- [12] OpenStreetMap Foundation. www.openstreetmap.org. 2021. url: www.openstreetmap.org (besucht am 01. 05. 2021).
- [13] OpentopoMap. www.opentopomap.org. 2021. url: <https://opentopomap.org/> (besucht am 01. 05. 2021).
- [14] Heimo Schack. Urheber- und Urhebervertragsrecht. 9. Aufl. Tübingen: Mohr Siebeck, 2019.
- [15] Falk Zscheile. „Kartenherstellung zwischen Lizenzen, Daten, Programmcode und Darstellung“. In: FOSSGIS Konferenz Freiburg im Breisgau 2020. Hrsg. von FOSSGIS e. V. Kirchzarten, 2020, S. 19–24. url: https://www.fossgis-konferenz.de/2020/data/FOSSGIS_Tagungsband_2020.pdf.

MapFish Print: Eine serverseitige Druck-Komponente

MapFish Print ist eine Druck-Komponente für Karten-Anwendungen, die auf dem Server läuft. Über ein Druck-Template wird ein Layout vorgegeben, das mit einem konkreten Request mit Inhalten gefüllt wird und auf dem Server als PDF oder Raster-Bild gerendert wird. Im Vortrag wird auf die Neuerungen der letzten Jahre und auf geplante zukünftige Erweiterung eingegangen.

MapFish Print ist eine in Java geschriebene Open-Source-Software (BSD-2 Lizenz) zum Drucken von Karten. Als klassische Druck-Komponente ist MapFish Print bereits in vielen Open-Source Web-Mapping Frameworks integriert, z.B. in GeoMapFish.

MapFish Print lässt sich in einem klassischen Servlet-Container (z.B. Tomcat) als WAR-File deployen oder steht als Docker-Image zur Verfügung und lässt sich so in Cloud-Umgebungen integrieren. MapFish Print steht auch als Bibliothek zur Verfügung und lässt sich dadurch in eigene Projekte integrieren.

MapFish Print unterstützt viele gängige Formate und Standards (z.B. WMS, WFS, WMTS, GeoJSON). Zudem bietet die Anwendung zahlreiche kartographische Gestaltungsmöglichkeiten wie Drehungen, Nordpfeile, Legenden oder Grids. Der Ausdruck wird über ein JasperReports-Template gestaltet und über eine YAML-Konfigurationsdatei definieren. Über einen JSON-Request wird dann vom Client der konkrete Druck-Inhalt (Layer, AOI, etc) übergeben.

Björn Höfling

GeoNetwork, nächste Generation

Geonetwork unterzieht sich derzeit einem größeren Refactoring, um die heutigen und zukünftigen Erwartungen an den führenden Open-Source-Metadatenkatalog für Geodaten zu erfüllen. Unser Ziel ist es, auf der Serverseite eine Microservice- und Cloud-Native-Architektur und auf der Clientseite eine neue UI von WebComponents mit einem starken Fokus auf Ergonomie und Index- und Finde-alles-Ansatz vorzuschlagen.

Im Mai 2020 traf sich die GeoNetwork-Community mehrfach (vor Ort und online), um über die nächste Generation von GeoNetwork nachzudenken. Nach einer erfolgreichen Präsentation vor mehr als 60 GeoNetwork-Enthusiasten während eines GeoNetwork-Workshops, begann die Entwickler-Community mit der Entwicklung einer brandneuen Architektur, die sich auf eine cloud-native Architektur konzentriert. Eine der wichtigsten Änderungen ist die Einführung einer neuen Indizierungs-Engine, nämlich Elastic Search, und der Beginn des Aufbaus von Microservices vor dieser Engine.

Auf der Client-Seite wird ein neues Frontend entwickelt, wobei der Fokus auf die Ergonomie der Lösung gelegt wird. Als Beispiel führen nun Widgets den Benutzer entlang der Prozesse, z.B. datafeeder. Auf der technischen Seite sind WebComponents und die neuesten Versionen von Angular und OpenLayers Teil der Lösung und spielen eine Schlüsselrolle in der Architektur.

In diesem Vortrag werden wir die neuesten Entwicklungen und die Roadmap für die nächste Version von GeoNetwork vorstellen, einschließlich der Unterstützung der neuen ogcapi-records Spezifikation.

Emmanuel Belo

GeoServer Microservices: der Weg von GeoServer in die Cloud

GeoServer ist für viele Szenarien zur Implementierung und Betrieb von Kartendiensten schnell einsetzbar, aber bislang für den Betrieb von hochverfügbaren Applikationen und/oder für den Betrieb in einer Cloud nicht optimal. In diesem Vortrag wird das Projekt GeoServer Microservices vorgestellt, welches diesen Schwachpunkt adressiert. Die Architektur von GeoServer Microservices wird erläutert, und dessen Einsatz im Projekt NexSIS, in dem die Notruf-Bearbeitung in Frankreich modernisiert wurde.

GeoServer ist eine Open-Source Software zur Implementierung und zum Betrieb von Kartendiensten. GeoServer ist eine bewährte und weit verbreitete Lösung. Für den Einsatz von GeoServer für Anwendungen, die eine sehr hohe Verfügbarkeit benötigen, wie zum Beispiel für Notruf-Bearbeitungssysteme, ist die klassische Architektur von GeoServer jedoch nicht optimal. Für solch ein System müssen mehrere Instanzen der Applikation betrieben werden, die dabei die gleiche Konfiguration teilen. Um dies zu erreichen, sind bislang relativ komplexe Setups erforderlich, die durch ihre Komplexität auch fehleranfällig sein können.

Im GeoServer Microservices Projekt kann nun GeoServer so eingesetzt werden, dass mehrere Instanzen auf einfachere Weise die gleiche Konfiguration verwenden, wodurch das Aufsetzen und der Betrieb von GeoServer als hochverfügbares System und/oder als Cloud-Dienst vereinfacht werden. Des Weiteren werden die verschiedenen Dienste von GeoServer durch unterschiedliche Microservices angeboten, um den Ressourceneinsatz optimieren zu können und dadurch eine bessere Skalierbarkeit zu erreichen.

In diesem Vortrag wird die Architektur von GeoServer Microservices erläutert und dessen Einsatz im Projekt NexSIS erklärt, in dem die GeoServer Microservices in einem Kubernetes-Cluster betrieben werden und somit hochverfügbare Kartendienste innerhalb der Notruf-Bearbeitung sicherstellen.

Wolfgang Kaltz

GBD WebSuite Demo-Session

Im Rahmen der Demo-Session wollen wir einen praktischen Einblick in die Installation der GBD WebSuite geben, welche seit 2017 durch das Geoinformatikbüro Dassau als Open Source WebGIS Lösung unter der Apache 2.0 License entwickelt wird.

Die GBD WebSuite bietet eine Vielzahl an Funktionalität und wird deutschlandweit vor allem als Geoportal-Lösung in komplexeren IT-Infrastrukturen genutzt. Hervorgehoben werden soll in dieser Demo-Session aber die einfache Bereitstellung von QGIS Projekten.

Im Rahmen der 60-minütigen Demo-Session wollen wir einen praktischen Einblick in die Installation der GBD WebSuite geben, welche seit 2017 durch das Geoinformatikbüro Dassau als Open Source WebGIS Lösung unter der Apache 2.0 License entwickelt wird.

Die GBD WebSuite bietet eine Vielzahl an Funktionalität und wird deutschlandweit vor allem als Geoportal-Lösung in komplexeren IT-Infrastrukturen genutzt. Hervorgehoben werden soll in dieser Demo-Session aber die einfache Bereitstellung von QGIS Projekten.

Inhalt der Demo-Session ist:

- Kurze Vorstellung der GBD WebSuite
- Installation der GBD WebSuite aus DockerHub
- Basiskonfiguration Server und Client
- Bereitstellung eines QGIS Projektes mit dem Plugin GBD WebSuite Manager

Weitere Links:

WebSeite: <https://gbd-websuite.de>

GitHub: <https://github.com/gbd-consult/gbd-websuite>

DockerHub: <https://hub.docker.com/r/gbdconsult/gws-server>

QGIS Plugin GBD WebSuite Manager: <https://github.com/gbd-consult/gbd-websuite-manager>

Geoinformatikbüro Dassau GmbH: <https://www.gbd-consult.de>

Otto Dassau

Geodaten auf Smartphones – ein drittes Paradigma nach Desktop- und Web-GIS?

CHRISTOPH JUNG

Einleitung

Geografische Informationssysteme (GIS) stellen eine Revolution in der Kartenherstellung dar, indem sie die Möglichkeiten zu räumlichen Analysen von Geodaten unterschiedlichster Quellen bieten und End- sowie Zwischenergebnisse der Analysen in Kartenform darstellen können (Kraak, 2004, S. 83). Die steigende und heute omnipräsente Verbreitung des Internets hatte auch auf GIS und Karten große Auswirkungen. So bieten Web-GIS heute den Zugriff auf interaktive Kartenanwendungen mit Funktionen klassischer Desktop-GIS. Ohne die hohen Lizenzkosten einer proprietären GIS-Applikation sowie durch die geringe Anzahl an Funktionen eines Web- gegenüber eines Desktop-GIS und der daraus resultierenden einfacheren Bedienung, erschlossen sich Geodaten einer größeren Anzahl an Nutzern (Lange & Plass, 2008, S. 177). Beide Systeme, Web- als auch Desktop-GIS, sind ursprünglich für Laptops und Desktop-PCs konzipiert und stellen vergleichbare Anforderungen an die Kartendarstellung. Seit Jahren ist jedoch das Smartphone unter den Internetnutzenden das am häufigsten verwendete Endgerät für den Konsum digitaler Inhalte (Statista, 2019). Diese Entwicklung hat auch auf Web-GIS Auswirkungen. Im Folgenden wird gezeigt, welche Besonderheiten und Herausforderungen bei Kartenanwendungen auf Smartphones beachtet werden sollten und wie sich eine Kartendarstellung in mobilen Applikationen gegenüber einem Web-GIS unterscheiden kann.

Web-GIS auf Smartphones

Bei der Entwicklung einer Web-GIS-Anwendung gilt es zu beachten, dass webbasierte Anwendungen auf unterschiedlichen Endgeräten, z.B. Desktop-PCs und Smartphones, aufgerufen werden können. Zwei populäre Technologien zur Unterstützung verschiedener Endgeräte sind das responsive Webdesign und explizite mobile Webanwendungen. In beiden Fällen müssen einige Besonderheiten von Smartphones gegenüber Desktop-PCs beachtet werden, u.a.:

- Größe und Form des Displays
- Eingabemöglichkeiten
- Nutzungsgewohnheiten

Um eine Kartenanwendung auch auf kleinen Displays nutzen zu können, müssen ggf. Anpassungen hinsichtlich des Karteninhaltes oder der Darstellungsmaßstäbe vorgenommen werden. Auch die Präsentation von Sachdaten zu einem Kartenobjekt sollte an den im Vergleich zu Desktop-PCs geringen Display-Platz angepasst werden.

Die Interaktion mit Kartenobjekten, z.B. Selektion, wird auf Desktop-PCs mittels Mauszeiger realisiert, auf Smartphones per Finger. Außerdem unterstützen Smartphones verschiedene Gesten als Eingabemöglichkeiten, sodass sich Interaktionen mit der digitalen Karte mit Nutzereingaben des Browsers oder des Betriebssystems überlagern können (z.B. Verschieben des Kartenausschnitts mit dem Zurückwischen im Browser).

Smartphones werden in anderen Situationen genutzt als Desktop-PCs, wodurch sich neue Anforderungen an die Lesbarkeit einer Karte ergeben. Navigations-Apps wie Google Maps und Apple Karten gehören zu den meist genutzten Karten-Apps auf Smartphones und beeinflussen Erfahrungen mit Geodaten und Erwartungen an interaktive Karten der Smartphone-Nutzenden.

Geodaten auf Smartphones – ein drittes Paradigma nach Desktop- und Web-GIS?

Vor allem durch die eingeschränkte Display-Größe und die Überlagerung von Eingabemöglichkeiten sind umfangreiche Web-GIS-Anwendungen auf Smartphones schlecht bis hin zu gar nicht bedienbar. In Kombination mit einem nicht für Smartphones angepassten Kartenbild sorgt dies für eine geringere Nutzung von Web-GIS-Anwendungen und deren Geodaten.

Vergleich zwischen Web-GIS und mobilen Karten-Apps

Kartenbasierte mobile Apps wie Google Maps und Apple Karten gehören heutzutage zu den Standardanwendungen auf Smartphones. Da mobile Apps im Gegensatz zu Web-GIS nicht in einem Browser, sondern als eigenständige Anwendungen laufen, gibt es keine Probleme mit der Überlagerung von Eingabemöglichkeiten und die Apps können mit anderen Anwendungen und Diensten auf Smartphones integriert werden (z.B. Automatisierung, Sprachassistenten).

Neben diesen technischen Unterschieden zwischen mobile Apps und Web-GIS haben sich bei Google Maps und Apple Karten auch Anpassungen an der Kartendarstellung und der Kartensteuerung herausgebildet, die sich auf die Benutzung auf Smartphones fokussieren. Zum einen weisen die beiden mobile Apps keine granulare Layer-Steuerung auf, wie man es aus Desktop- und Web-GIS kennt, sondern einige wenige Themen. Zum anderen ist die Interaktion mit Kartenobjekten auf die Benutzung des Smartphones mit dem Finger optimiert. So sind Linien und Polygone nur in Ausnahmefällen selektier- und abfragbar. Um dennoch Sachdaten abfragen zu können, werden zusätzliche punktförmige Marker (in Apples MapKit-Framework als Annotations bezeichnet) der Karte hinzugefügt. Diese sind von ihrer Größe her deutlich einfacher anzutippen als beispielsweise eine dünn gezeichnete Linie. Außerdem signalisieren Annotations Nutzern, welche keinerlei Vorkenntnisse und Erfahrungen im Bereich GIS haben, deutlicher eine Interaktionsmöglichkeit als sich harmonisch in das Kartenbild einfügende Polygone und Linien



Abbildung 1: Darstellung von Interaktionsmöglichkeiten mittels Annotations in Apple Karten

Native kartenbasierte Apps mit OGC-Services

Eine native, kartenbasierte mobile App mit Annotations als touch-optimiertes Kartenelement zur Abfrage von Sachdaten ist mit OGC-Diensten realisierbar. Dabei können Grundkarten, Polygone und Linien per WM(T)S, TMS oder WCS ohne die Nutzung des GetFeatureInfo-Requests eingebunden werden. Um dennoch Sachdaten zu Polygonen und Linien zur Verfügung zustellen, müssen repräsentative Punkte, z.B. Schwerpunkte, dieser Objekte im Vorfeld erzeugt und als WFS der Karte hinzugefügt werden.

Geodaten auf Smartphones – ein drittes Paradigma nach Desktop- und Web-GIS?

Eine weitere Möglichkeit wäre die Konfiguration eines WMS, der entsprechende Annotations schon enthält und ein `GeatFeatureInfo` auf die Punkt-Geometrien verarbeitet. Vorteil dieser Methode wäre eine bessere Laufzeit bei großen Mengen an Annotations. Jedoch würde sich die grafische Ausprägung und die Nutzererfahrung der WMS-Annotations tendenziell von denen der Plattform unterscheiden.

Die Einbindung der OGC-Services ist direkt per Maps SDK (Android) bzw. MapKit (iOS) realisierbar und erfordert keine Nutzung weiterer Software-Bibliotheken Dritter.

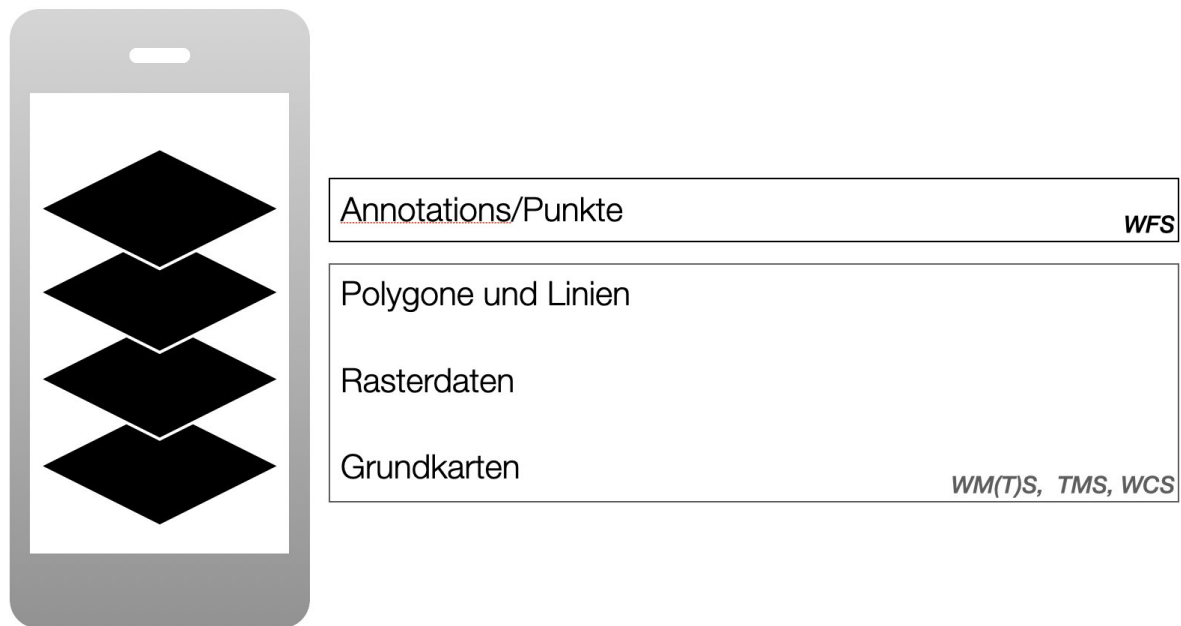


Abbildung 2: Kartenaufbau einer für Smartphones optimierten interaktiven Karte

Zusammenfassung

Bedingt durch die großen technischen Unterschiede aber auch die verschiedenen Nutzungsgewohnheiten zwischen Desktop-PCs und Smartphones ist eine Optimierung von Web-GIS auf beide Arten von Endgeräten sehr aufwändig, wobei zu meist die Web-GIS-Nutzung auf Smartphones leidet. Um Geodaten und interaktive Karten auch Smartphone-Nutzenden zur Verfügung zu stellen, bietet sich die Entwicklung einer speziell auf Smartphones fokussierten Kartendarstellung an. Dabei können bestehende OGC-Dienste genutzt werden, um Karten-Apps analog zu Google Maps und Apple Karten herzustellen, wobei die Plattformen Android und iOS schon entsprechende Frameworks mitbringen, um OGC-Dienste einzubinden.

Kontakt zum Autor:

Christoph Jung
jagodki.cj@gmail.com
<https://github.com/jagodki>

Literatur

De Lange, N. & Plass, C. (2008). WebGIS with Google maps. Digital Earth Summit on Geoinformatics, 176-181.

Geodaten auf Smartphones – ein drittes Paradigma nach Desktop- und Web-GIS?

Kraak, M. J. (2004). The role of the map in a Web-GIS environment. *Journal of Geographical Systems*, 6(2), 83-93.

Statista Research Department (2019). Internetnutzung nach Endgeräten in Deutschland 2018. Retrieved Apr 06, 2021 from <https://de.statista.com/statistik/daten/studie/912595/umfrage/internetnutzung-nach-endgeraeten-in-deutschland/>.

MrMap – GeoPortal.rlp Reloaded

ANDRE HOLL / JONAS KIEFER / ARMIN RETTERATH

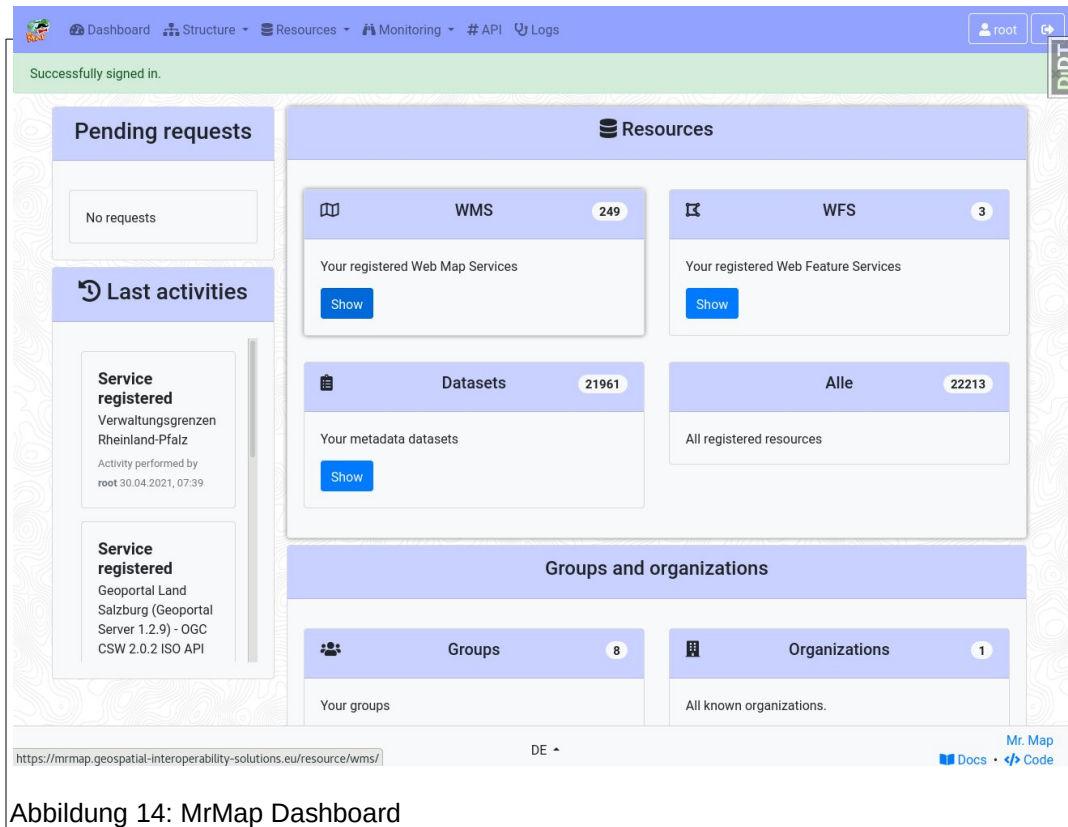


Abbildung 14: MrMap Dashboard

Einleitung

Die Entwicklung von MrMap [1] als Reengineering des GeoPortal.rlp Backends [2] läuft jetzt schon mehr als 2 Jahre. Auf der FOSSGIS 2020 wurde von Michel Peltriaux der erste Entwicklungsstand der auf Django basierenden Webservice-Registry vorgestellt [3]. Da Michel in die Umweltverwaltung des Landes Rheinland-Pfalz gewechselt ist, hat Mitte 2020 Jonas Kiefer (<https://github.com/jokiefer>) den Posten als Lead Developer des Projektes übernommen.

Im Laufe des letzten Jahres sind einige neue Features hinzugekommen, und es wurde intensiv daran gearbeitet Stabilität und Sicherheit in den Entwicklungsprozess zu bekommen.

Mr. Map verfolgt einen alternatives GDI-Architekturmodell, dass den Geodateninfrastrukturen der Län- der Hessen, Rheinland-Pfalz und Saarland zugrunde liegt. Der Fokus dieses Modells liegt weniger im Management von ISO-Metadaten, als in der integralen Verwaltung der für eine GDI-relevanten Entitä- ten und deren Beziehungen untereinander. Dabei spielt das zentrale Authentifizierungs- und Autorisie- rungsmodell des Security-Proxys eine wichtige Rolle. Es stellt die Basiskomponente dar, die von allen an der Infrastruktur beteiligten Organisationen verwendet wird.

Neue Funktionen

- Generisches Monitoringmodul für registrierte Ressourcen

MrMap – GeoPortal.rlp Reloaded

- Verbesserung der Möglichkeiten zum Harvesting von CSW-Ressourcen
- Ein Qualitätsmodul für die Integration von externen Testplattformen (ETF-API)
- Ablage hierarchischer Daten (bspw. WMS-Layer) als MPTT
- Major changes „under the hood“

Monitoring von OWS

Im GeoPortal.rlp gibt es schon seit dem Zeitpunkt der Freischaltung 2007 ein Monitoring Tool, dass alles 1.600 Webservices regelmässig prüft, Statistiken zu deren Verfügbarkeit erstellt und Anbieter sowie Abonnenten von Diensten bei Problemen benachrichtigt. Das neue MrMap Monitoring Tool erlaubt ebenfalls ein asynchrones Monitoring der registrierten Ressourcen (zunächst OWS) und einzelner Operationen. Es können dabei Grenzwerte für die Qualität vorgegeben werden.



The screenshot shows a table with columns: Title, Layers, Status, Health, Version, Data provider, ID, Published for, Created on, and Actions. The first row is highlighted with a tooltip that says: 'Everything is OK. Last check runs on 2021-04-28 11:26:22. Click on this icon to see details.' The tooltip points to a green square icon in the Health column.

Title	Layers	Status	Health	Version	Data provider	ID	Published for	Created on	Actions
Abfragegrenzen Betroffenheit 2012			100.0 %	1.1.1	LANDESAMT FÜR UMWELT RHEINLAND-PFALZ	Testgroup	—	15.04.2021 12:41	[Icons]
Abfragegrenzen Betroffenheit 2012	2			1.1.1	LANDESAMT FÜR UMWELT RHEINLAND-PFALZ	Testgroup	—	15.04.2021 12:12	[Icons]
Alken	10			1.1.1	LVermGeo im Auftrag von Alken	Testgroup	—	15.04.2021 12:42	[Icons]

Abbildung 15: Health Zustand von registrierten Diensten in MrMap

Metadata Catalogue Management

Die bisherige Nutzung von externen Katalogschnittstellen im GeoPortal.rlp beschränkte sich auf die Registrierung von CSW, die Live-Suche in den angeschlossenen Katalogen, sowie die Synchronisierung von CSW's mit angeschlossenen Open Data Portalen über die CKAN API.

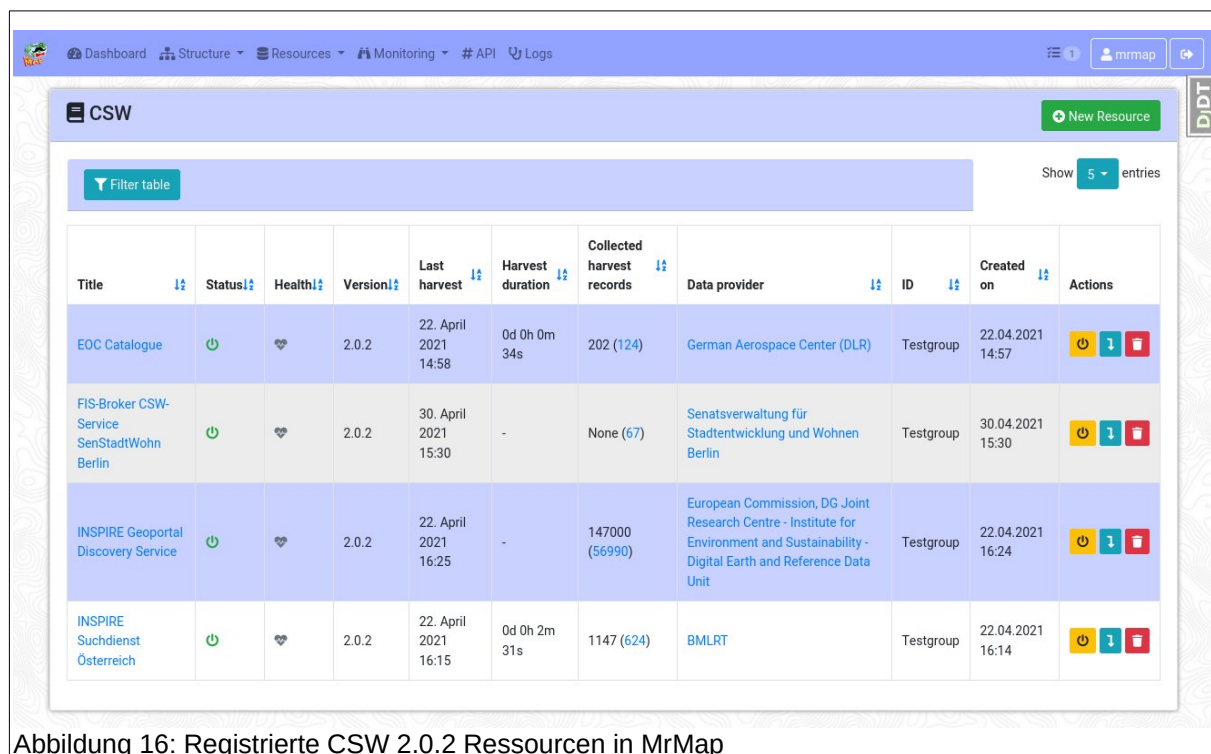
Da MrMap zusätzlich auch die Möglichkeit haben soll, beliebige registrierte Metadatenkataloge zu harvesten, wurden alle an den Geodatenkatalog.DE [4] angeschlossenen Katalogsoftwareprodukte getestet [5]. Hier stellte sich heraus, dass die verschiedenen Produkte die CSW-Spezifikation des OGC [6] *immer noch* unterschiedlich umsetzen.

Aufgetretene Probleme:

- Unterschiedliche Angabe der Operation URIs
- Unterschiedliche Interpretation von Parameter Namen
- Fehlende Rückgabe von „nextRecord“
- Missachten des Parameters „maxRecords“ beim GetRecords-Request

Nach langem *trial and error* ist es gelungen, alle in Deutschland eingesetzten Katalogsoftwareprodukte in MrMap registrieren und auch erfolgreich zu harvesten ;-).

MrMap – GeoPortal.rlp Reloaded



CSW

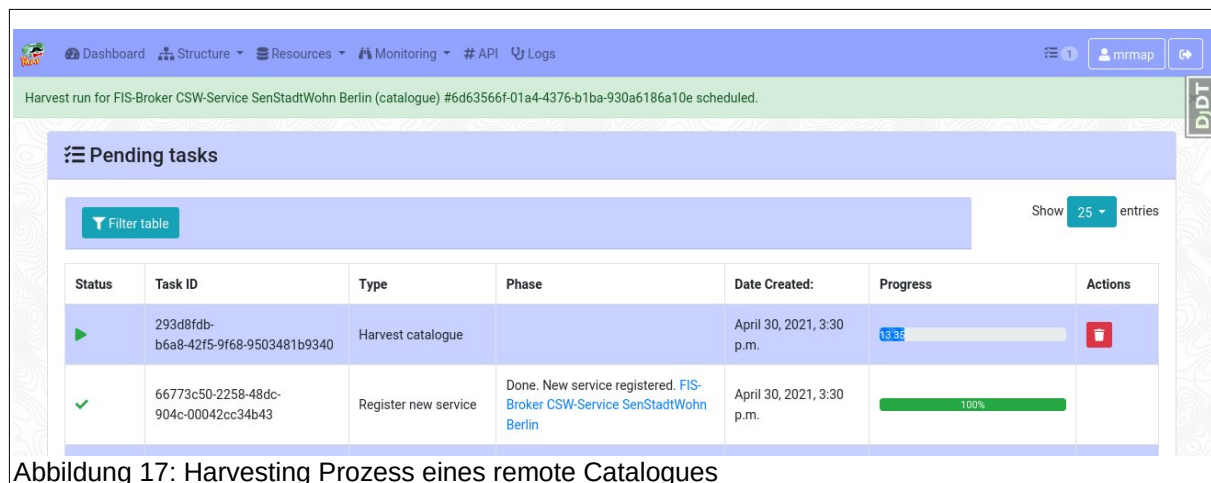
Filter table

Show 5 entries

Title	Status	Health	Version	Last harvest	Harvest duration	Collected harvest records	Data provider	ID	Created on	Actions
EOC Catalogue	🟢	🟢	2.0.2	22. April 2021 14:58	0d 0h 0m 34s	202 (124)	German Aerospace Center (DLR)	Testgroup	22.04.2021 14:57	🔌 📄 🗑️
FIS-Broker CSW-Service SenStadtWohn Berlin	🟢	🟢	2.0.2	30. April 2021 15:30	-	None (67)	Senatsverwaltung für Stadtentwicklung und Wohnen Berlin	Testgroup	30.04.2021 15:30	🔌 📄 🗑️
INSPIRE Geoportal Discovery Service	🟢	🟢	2.0.2	22. April 2021 16:25	-	147000 (56990)	European Commission, DG Joint Research Centre - Institute for Environment and Sustainability - Digital Earth and Reference Data Unit	Testgroup	22.04.2021 16:24	🔌 📄 🗑️
INSPIRE Suchdienst Österreich	🟢	🟢	2.0.2	22. April 2021 16:15	0d 0h 2m 31s	1147 (624)	BMLRT	Testgroup	22.04.2021 16:14	🔌 📄 🗑️

Abbildung 16: Registrierte CSW 2.0.2 Ressourcen in MrMap

Der Harvesting-Prozess erfolgt über gestaffelte GetRecords-Anfragen. Die Übersicht über den Zustand der Harvesting-Prozesse, die jeweils im Hintergrund laufen, erhält man im Menu „Pending tasks“.



Harvest run for FIS-Broker CSW-Service SenStadtWohn Berlin (catalogue) #6d63566f-01a4-4376-b1ba-930a6186a10e scheduled.

Pending tasks

Filter table

Show 25 entries

Status	Task ID	Type	Phase	Date Created:	Progress	Actions
▶	293d8fdb-b6a8-42f5-9f68-9503481b9340	Harvest catalogue		April 30, 2021, 3:30 p.m.	13.3%	🗑️
✓	66773c50-2258-48dc-904c-00042cc34b43	Register new service	Done. New service registered. FIS-Broker CSW-Service SenStadtWohn Berlin	April 30, 2021, 3:30 p.m.	100%	

Abbildung 17: Harvesting Prozess eines remote Catalogues

Integration der Validatoren von INSPIRE und der GDI-DE

MrMap – GeoPortal.rlp Reloaded

Die bisherige Überprüfung der Konformität von Diensten und Metadatenätzen im GeoPortal.rlp erfolgte über die API des INSPIRE Validators 2. Dieser wurde aber vor ca. 2 Jahren durch den neuen ETF-basierten Validator ersetzt.

Um für MrMap eine nachhaltige Möglichkeit zu schaffen, dass Daten- und Dienstebereitsteller die Konformität ihrer publizierten Informationen direkt testen können, wurde im Laufe des letzten Jahres ein Quality Modul durch die Firma Terrestris entwickelt [7].

Die Prüfung der Ressourcen erfolgt dabei über die REST API des ETF-Validators, der sowohl der GDI-DE-Testsuite [8], als auch dem INSPIRE-Validator [9] zugrunde liegt.

Es können beliebig viele ETF-Endpunkte angelegt werden. Im Frontend gibt es für jede Testmöglichkeit einen Eintrag in einer Dropdownliste.

Die Ergebnisberichte des Tests werden in Form von HTML-Dateien mit Zeitstempeln in der MrMap Datenbank vorgehalten.

Performante Ablage komplexer hierarchischer Strukturen

Im GeoPortal.rlp, dessen Informationsmodell noch auf Mapbender2.8 [10] basiert, erfolgt die hierarchische Ablage der registrierten WMS-Layer nach einem standard parent-child Verfahren. Andere hierarchisch zusammengesetzte Strukturen, wie zum Beispiel WebMapContext-Dokumente (WMC), werden in ihrer originären XML-Struktur verwaltet und nur einige Metadaten sind über das relationale Modell direkt verfügbar.

In MrMap sollen die Kartenkontexte nicht mehr in Form von WMC, sondern als eigene „mapContext“-Objekte verwaltet werden. Diese sind – wie auch WMS-Layer - hierarchisch strukturierbar. Die Beziehungen von registrierten Ressourcen und den Kontexten, die sich aus diesen zusammensetzen, sollen über Relationen abgebildet werden [11].

Da im aktuellen Portal schon fast 10.000 personalisierte WMC existieren und diese sehr häufig gelesen und ausgewertet werden müssen, wird in MrMap auf das für Lesegeschwindigkeit optimierte Modified Preorder Tree Traversal (MPTT) Verfahren gesetzt [12]. Für die Implementierung gibt es in Django ein eigenes package: <https://django-mptt.readthedocs.io/en/latest/> .

Die optimierte Lesegeschwindigkeit zeigt sich der Umsetzung der neuen Tree-Ansicht im WMS View. Da der bisher für Tests verwendete australische Firewatch-WMS mit fast 1.400 Layern in bis zu 5 Hierarchieebenen leider nicht mehr online ist, wird jetzt der Geoproxy aus Thüringen verwendet [13]. Dieser umfasst 840 Layer in mehreren Verschachtelungsebenen.



Abbildung 18: WMS des Geoproxys Thüringen registriert in MrMap

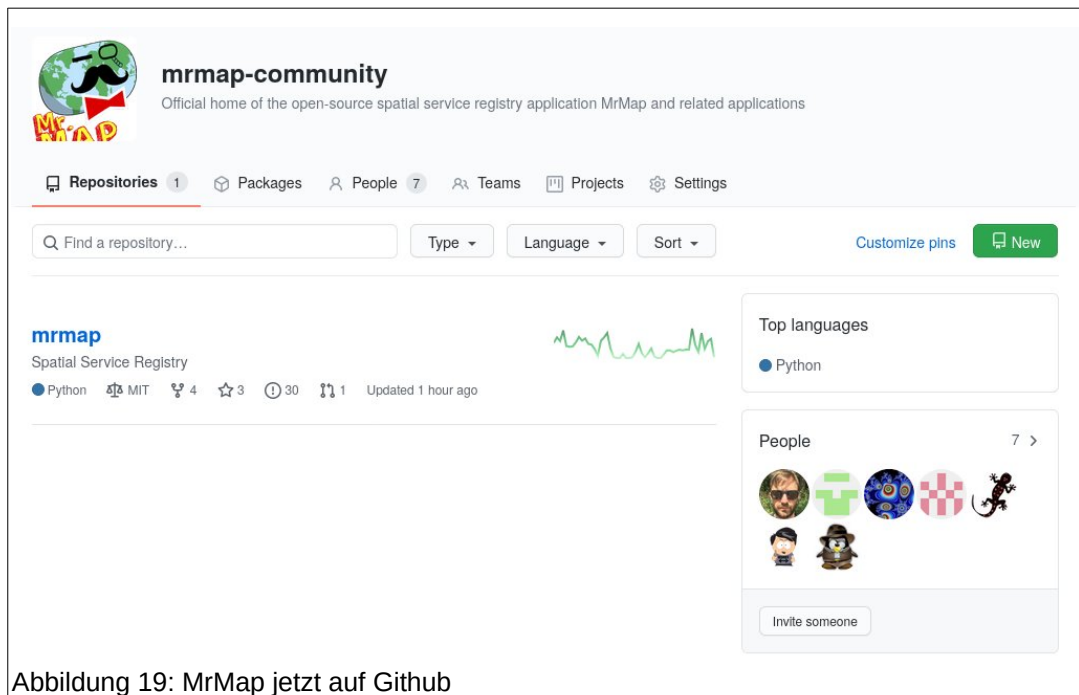
„Under the hood“

Umzug auf Github

Das Projekt wurde zunächst im gitea-Repo der OSSEO gestartet. Es zeigte sich jedoch schnell, dass die Möglichkeiten, die dieses GIT aktuell bereitstellt, für eine schnelle und sichere Entwicklung nicht ausreichen.

Das Projekt wurde daher auf Github umgezogen. In diesem Zusammenhang wurde das neue Dokumentationssystem angelegt, dass es ermöglicht die Dokumentation ebenfalls über Github.io zu hosten.

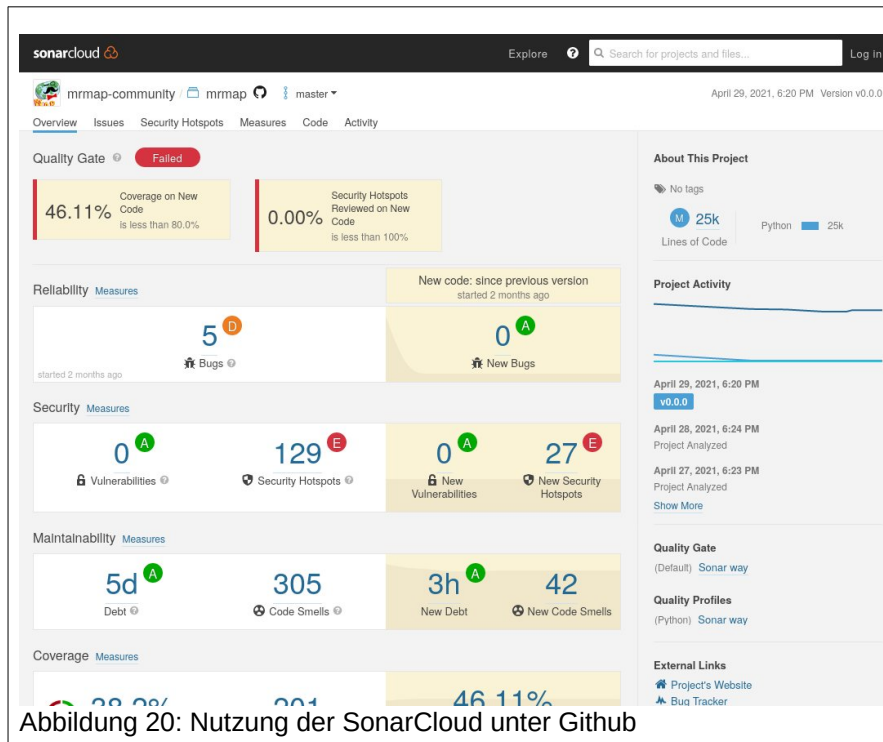
MrMap – GeoPortal.rlp Reloaded



Automatische Qualitätssicherung

Eine weitere wichtige Neuerung ist die Einführung des Quality Gates auf Basis der SonarCloud [14]. Für Open-Source Projekte ist die Nutzung dieses Services kostenfrei. Fehler, Sicherheitslücken und „Code smells“ werden automatisch identifiziert und dokumentiert. Jeder pull requests wird ebenfalls automatisch geprüft und muss vorgegebene Qualitätskriterien einhalten, bevor er gemerged werden kann [15].

MrMap – GeoPortal.rlp Reloaded



Dokumentation

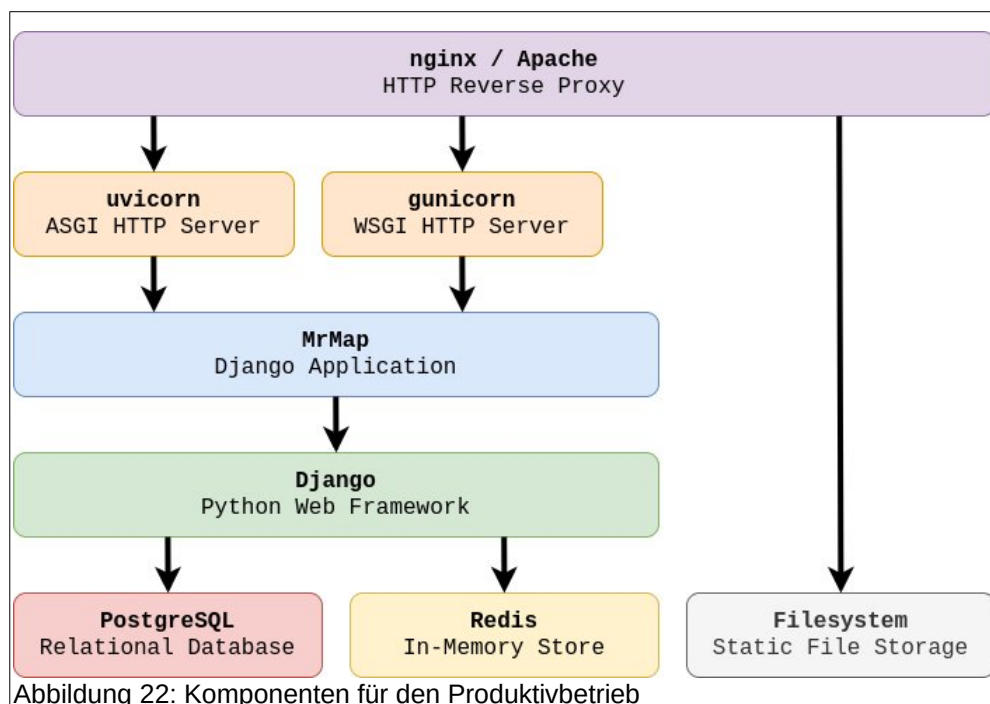
Die Doku des Projekts wurde ebenfalls komplett überarbeitet [16]. Es stehen jetzt sehr gute (englische) Installationsanleitungen für Produktiv- und Entwicklungsumgebungen zur Verfügung. Neu ist auch die Möglichkeit **docker-compose** zu nutzen [17].



Einführung von Websockets

MrMap – GeoPortal.rlp Reloaded

Viele Prozesse bei der Verwaltung von verteilten Webservices laufen i.d.R. über längere Zeiträume (Harvesting, Monitoring, Registrierung komplexer Webservices, ...). Aus Sicht der Infrastruktur macht es Sinn diese asynchron zu verwalten. Um dem Nutzer zu jeder Zeit die Möglichkeit zu geben den aktuellen Stand der von ihm initialisierten Prozesse im Blick zu behalten, wird oft AJAX verwendet. Der Zustand des Servers wird dabei regelmäßig vom Client „gepollt“. Diese Vorgehensweise ist aus architektonischer Sicht eher suboptimal. Das WebSocket Protokoll – RFC 6455 [18] erlaubt, im Gegensatz zum normalen HTTP Protokoll, bidirektionale Verbindungen und es lassen sich Anwendungen mit einem „Echtzeit“-Feeling entwickeln. Für Django gibt es das Django Channels Projekt [19], dass auf dem *Asynchronous Server Gateway Interface* (ASGI) aufsetzt. Die Einführung des neuen Protokolls hat direkte Auswirkungen auf die Architektur von Mr. Map. Neben dem WSGI kommt zusätzlich ein ASGI Server zum Einsatz.



Die Einführung von WebSockets erlaubt ein direktes Feedback des Nutzers über die laufenden und abgeschlossenen Tasks. Außerdem ist das System damit auch technologisch gut für die Integration von Echtzeitsensoren gerüstet.

Status	Task ID	Type	Phase	Date Created:	Progress	Actions
	e79971ea-5252-404a-b672-a964ea203849	Register new service	Parsing Stationierung Strassennetz	April 30, 2021, 4:04 p.m.	17.36%	

Abbildung 23: Registrierungsprozess eines komplexen WMS mit Daten-Service Kopplung

Class-based views

Views, im Kontext von Django, sind Schnittstellen über die der Nutzer Zugriff auf die Anwendung erhält. Dies kann beispielsweise die in Abbildung 148 gezeigte Baumansicht eines Dienstes sein. Das in MrMap verwendete Django Framework bietet verschiedene Möglichkeiten views zu erstellen. Bisher wurden im MrMap Projekt die aus Sicht des Frameworks primitivste Form der views genutzt, um die verschiedenen Schnittstellen zu definieren (Function-based views). Dies war ein architektonisch weitreichender Nachteil für die zukünftig zu wartende Quelltextbasis von MrMap. So sind Function-based views nur schwer zu erweitern und der Quelltext ist nur umständlich wiederverwendbar. Class-based views heben diese Nachteile auf. So können vordefinierte python-Klassen mit vorgefertigtem code-flow für spezielle Einsatzszenarien genutzt werden. Da Django für die meisten Standardfälle, beispielsweise Detailansichten, Listen, sowie das Erzeugen und Abändern von Objekten im Kontext der Datenbank, diese auf Klassen basierenden views schon bereitstellt, muss für Standardabläufe kein eigener Quelltext neu entwickelt werden. So wurden im vergangenen Jahr nahezu alle bisherigen views überarbeitet. Für eine genauere Gegenüberstellung von Function-based views und Class-based views wird auf [20] verwiesen.

Rechte-Rollensystem

Das bisherige Rechte-Rollensystem ermöglichte es Nutzern spezielle Rollen zuzuweisen. Rollen im Kontext von MrMap sind Gruppierungen von Nutzern und zugehörigen Berechtigungen. Objekte im Datenbankkontext sind bei diesen Rollen nicht berücksichtigt. Das hat den Nachteil, dass wenn Nutzer A die Berechtigung B hat, er dieses Recht für alle Objekte, für die die Berechtigung gilt, ausüben kann. Sprich ein Nutzer mit Löschberechtigung könnte alle Objekte des jeweiligen models löschen. Dies stellt eine erhebliche Sicherheitslücke dar. In Folge dessen wurde das Autorisierungskonzept auf *objektbasierte* Berechtigungen umgestellt. Als Basis dient die Erweiterung django-guardian [21]. Diese implementiert *objektbasierte* Berechtigungen und liefert bereits alle notwendigen Erweiterungen für die zuvor beschriebenen Class-based views mit. Dadurch wird die Erweiterung von views sehr einfach. Schwieriger jedoch ist die Überarbeitung der einzelnen Django models [22], sowie die Automatisierung der Rollen. Bei der Erzeugung von neuen Datenbankobjekten muss darauf geachtet werden, dass automatisiert alle notwendigen Berechtigungen für die jeweiligen Nutzergruppe angelegt werden.

Zusammenfassung

Beim MrMap Projekt hat sich im letzten Jahr einiges getan. Insbesondere im Hinblick auf Codestabilität und Qualität sind beachtliche Fortschritte erzielt worden. Der Umzug auf Github, die Integration automatischer CI-Tools und das einfache Erstellen einer Entwicklungsumgebung sollen die Weichen stellen und weitere Entwickler bzw. Nachnutzer ermutigen sich am Projekt zu beteiligen.

Kontakt zu den Autoren:

MrMap – GeoPortal.rlp Reloaded

Andre Holl
andre.holl@vermkv.rlp.de

Jonas Kiefer
jonas.kiefer@vermkv.rlp.de

Armin Retterath
armin.retterath@vermkv.rlp.de

jeweils
Zentrale Stelle GDI-RP
Von-Kuhl-Straße 49
56070 Koblenz

Quellen

- [1] <https://github.com/mrmap-community/mrmap>
- [2] <https://git.osgeo.org/gitea/GDI-RP/GeoPortal.rlp>
- [3] <https://www.foSSGIS-konferenz.de/2020/sessions/K7VNXP.php>
- [4] <https://gdk.gdi-de.org/geonetwork/>
- [5] https://gdk.gdi-de.org/gdk_harvesting/
- [6] https://portal.ogc.org/files/?artifact_id=20555
- [7] <https://github.com/mrmap-community/mrmap/pull/42>
- [8] <https://testsuite.gdi-de.org/>
- [9] <https://inspire.ec.europa.eu/validator/about/>
- [10] <https://git.osgeo.org/gitea/GDI-RP/Mapbender2.8>
- [11] <https://github.com/mrmap-community/mrmap/issues/17>
- [12] https://de.wikipedia.org/wiki/Nested_Sets
- [13] <http://www.geoproxy.geoportal-th.de/geoproxy/services?>
- [14] <https://sonarcloud.io/github>
- [15] https://sonarcloud.io/dashboard?id=mrmap-community_mrmap
- [16] <https://mrmap-community.github.io/mrmap/>
- [17] <https://mrmap-community.github.io/mrmap/development/docker.html>
- [18] <https://de.wikipedia.org/wiki/WebSocket>
- [19] <https://channels.readthedocs.io/en/stable/>
- [20] <https://medium.com/@ksarthak4ever/django-class-based-views-vs-function-based-view-e74b47b2e41b>
- [21] <https://github.com/django-guardian/django-guardian>
- [22] <https://docs.djangoproject.com/en/3.2/topics/db/models/>

SHOGun als Basis flexibler WebGIS-Lösungen — Oder: Weder "off-the-shelf" noch "fully handcrafted"

Das OpenSource Geo Framework SHOGun ist die Basis zahlreicher Webanwendungen, die insgesamt sehr unterschiedliche Anwendungsfälle betrachten.



Hierbei sind SHOGun-Applikationen weder ein Standardprodukt von der Stange noch 100%ige spezifische Individualentwicklungen. Im Vortrag wollen wir zeigen, welche Aspekte von SHOGun wo zwischen diesen Extremen "off-the-shelf" / "fully handcrafted" einzuordnen sind.

Das OpenSource Geo Framework SHOGun ist die Basis zahlreicher Webanwendungen, die insgesamt sehr unterschiedliche Anwendungsfälle betrachten. Hierzu zählen u.a. Anwendungen zum Gewässermanagement (Progemis), zur holistischen Ökosystemrestauration (HERMOSA) oder zur Verwaltung und Visualisierung von dreidimensionalen Geodaten. SHOGun ist in seiner Architektur dabei so ausgelegt, dass es sich zwischen den Extremen „off-the-shelf“ (Standardprodukt von der Stange) und „fully handcrafted“ (spezifische Individualentwicklung) bewegen kann.

Der Vortrag stellt zunächst die aktuelle Version von SHOGun vor und zeigt anschließend anhand von den oben aufgeführten Beispielen, wie mit SHOGun und einigen anderen Bibliotheken und Drittkomponenten Standardfunktionalität bereitgestellt werden kann, ohne dass kundenspezifische Anpassungen ausgeschlossen oder erschwert werden. Die neueste Version der Software SHOGun setzt nach wie vor auf extrem robuste Basisbibliotheken wie Spring und Hibernate und bietet ein gesundes Maß an Lösungsansätzen für immer wiederkehrende Fragestellungen. Gleichzeitig versucht die Software auch flexible und verlässliche Ableitungen von diesem Kern einfach zu ermöglichen, etwa um eigene Businesslogiken abbilden zu können. Hierzu stehen abstrakte Vaterklassen bereit, um gleich anzusprechende und abgesicherte (REST und GraphQL) Schnittstellen für die verwalteten Entitäten wie Layer oder Applikationen in der Basis aber auch in Projektableitungen ableiten zu können. SHOGun versucht sich in seinem Kern auf genau diese Anforderungen zu beschränken und zusätzliche Fragestellungen durch seine Microservice-Architektur einfach integrierbar zu machen. Exemplarisch sei hier etwa Keycloak für die Identitäts- und Zugriffsverwaltung genannt.

SHOGun umfasst hierbei im Grunde die reine serverseitige Logik und Programmbestandteile. Im Vortrag werden wir jedoch auch Client-Applikationen (sowie deren zugrunde liegende Bibliotheken) vorstellen. Wenn man diese Einzelbestandteile geschickt zusammenstellt, profitiert man zu einem gewissen Teil von den Vorteilen beider Welten (Standardprodukt vs. Eigenentwicklung). SHOGun ist seit vielen Jahren in Entwicklung, hat jedoch in den letzten Jahren massive Neuentwicklungen erfahren und auch das Verständnis des Frameworks hat sich geändert. Dieser Vortrag hofft, die Beweggründe für diese Neugestaltungen anschaulich zu erläutern.

URL: * <https://github.com/terrestris/shogun>

Marc Jansen

Geodatenverarbeitung mit Node-RED

SEBASTIAN BECHTOLD, GEO.NET.MRN E.V. / METROPOLREGION RHEIN-NECKAR GMBH

Was ist Node-RED?

Node-RED ist eine ursprünglich von IBM initiierte und seit 2016 unter der Apache License 2.0 als Open-Source-Projekt weiterentwickelte Softwareplattform für *visuelle Programmierung*. Unter visueller Programmierung versteht man die Erstellung von Programmen durch Zeichnen von Ablaufdiagrammen mittels einer grafischen Oberfläche statt wie sonst üblich durch Schreiben von textbasiertem Code. Ein mit visueller Programmierung erstelltes Ablaufdiagramm hat nicht (nur) dokumentarischen Charakter, sondern wird von einer Interpretersoftware direkt als Programm ausgeführt.

Vor- und Nachteile visueller Programmierung

Aufgrund ihrer hohen Anschaulichkeit und vergleichsweise einfachen Programmerstellung sind visuelle Programmierungsumgebungen insbesondere für Einsteiger:innen in die Softwareentwicklung gut geeignet. Doch auch für Fortgeschrittene bietet die Methode in manchen Anwendungsfällen große Vereinfachungen gegenüber textbasierter Programmierung. Dazu gehört vor allem, dass eine übersichtliche GUI zur Konfiguration, Bedienung und Statusanzeige der Programme bereits von der visuellen Programmierplattform bereitgestellt wird und nicht selbst entwickelt werden muss.

Nachteilig ist die im Vergleich zu klassischen Programmiersprachen stark beschränkte Ausdruckskraft bzw. die mit steigender Programmkomplexität zunehmende Unübersichtlichkeit und abnehmende Bearbeitungseffizienz. Visuelle Programmierung ist daher vor allem für die Erstellung von einfachen Programmen geeignet, mit denen bereits existierende, als „Bausteine“ bereitgestellte Komponenten für unterschiedliche Anwendungsfälle neu kombiniert werden.

Grundlegende Konzepte und Terminologie von Node-RED

In der Terminologie von Node-RED werden die „Grundbausteine“ eines visuellen Programms als *Nodes* bezeichnet. Ein komplettes Programm, also eine Verknüpfung mehrerer Nodes zu einem Prozess, wird *Flow* genannt.

Node-RED ist in JavaScript geschrieben und basiert auf der JavaScript-Laufzeitumgebung Node.js. Es besteht im Wesentlichen aus zwei Teilen: Der eine Teil ist ein Hintergrundprozess oder „Dienst“. Dieser läuft kontinuierlich auf einem Server und führt hinterlegte Flows zeitgesteuert (z.B. 1x pro Tag) oder auf explizite Anforderung durch eingehende HTTP-Requests aus. Der andere Teil ist eine web-basierte GUI, mit der Flows mittels „Drag & Drop“-Prinzip erstellt werden können. Die meisten Nodes verfügen zudem über einen Konfigurationsdialog, über den anwendungsfallspezifische Einstellungen vorgenommen werden können.

Geodatenverarbeitung mit Node-RED

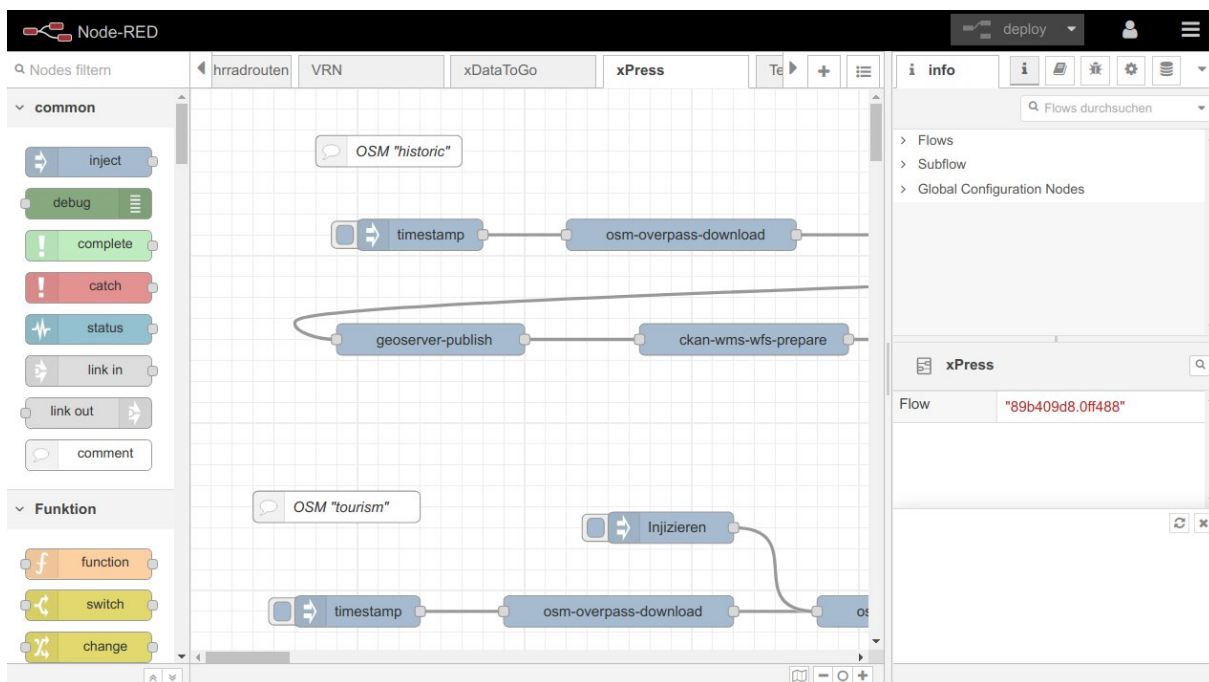


Abbildung 1: Die webbasierte Oberfläche von Node-RED

Um einen Node zu einem Flow hinzuzufügen, zieht man ihn aus einem „Palette“ genannten Bereich der GUI auf eine Zeichenfläche, die den größten Teil des Anwendungsfensters einnimmt und den Flow visualisiert. Die Verknüpfung mehrerer Nodes zu einem Flow erfolgt durch das Ziehen von Verbindungslinien zwischen je zwei Nodes. Hierfür besitzt die visuelle Darstellung der meisten Nodes je einen „Andockpunkt“ für eingehende Verbindungen auf der linken Seite und einen für ausgehende Verbindungen auf der rechten Seite („Eingang“ und „Ausgang“). Verbindungen werden per „Drag & Drop“ des Ausgangs eines Nodes auf den Eingang eines anderen Nodes hergestellt, bestehende Verbindungen werden als Linie visualisiert.

Verbindungen zwischen Nodes definieren den Datenfluss innerhalb eines Flows, von Node-Ausgängen zu Node-Eingängen. Ein Flow ist somit ein gerichteter Graph. Verzweigungen in parallele Teil-Flows, (Wieder-)zusammenführung von parallelen Teil-Flows und auch zyklische Konstruktionen sind möglich. Ein Datenobjekt, welches von einem Node an einen anderen über eine Verbindung übermittelt wird, wird als *Message* bezeichnet. Der eigentliche transportierte Inhalt einer Message wird *Payload* genannt. Zusätzlich zur Payload kann eine Message weitere Metadaten enthalten.

Node-RED basiert auf einem ereignisorientierten, asynchronen Programmiermodell, wie es für Node.js bzw. JavaScript allgemein typisch ist. Das Absenden einer Message von einem Node (i.d.R. mit dem Ergebnis des vom Node durchgeführten Arbeitsschritts als Payload) wird durch den Aufruf einer API-Funktion ausgelöst. Das Eintreffen der Nachricht am Zielnode ist als Ereignis modelliert, auf das mit im Programmcode des Ziel-Nodes implementierten Event-Handler-Funktionen reagiert werden kann. Der Ablauf eines Flows wird also realisiert durch das verkettete, abwechselnde Senden von Nachrichten-Ereignissen und Reaktionen von Nodes auf eintreffende Nachrichten-Ereignisse.

Bereits vorhandene Funktionalität

Node-RED enthält „ab Werk“ bereits eine große Auswahl an Nodes für vielfältige Aktionen wie HTTP-Requests, Parsen verschiedener Datenformate (XML, JSON, CSV etc.), Lesen und Schreiben von Dateien und allgemeine Programmflusskontrolle (Verzweigungen, Trigger, etc.). Weitere Nodes für viele verschiedene Zwecke können direkt über die Node-Red-Oberfläche in einem Online-Repository ge-

Geodatenverarbeitung mit Node-RED

sucht und mit wenigen Klicks installiert werden. Der Schwerpunkt liegt hierbei bislang auf dem „Internet of Things“ (IoT), also der Sammlung und Verarbeitung von Sensordaten und der Steuerung von Geräten. In diesem Anwendungsgebiet erfreut sich Node-RED großer Beliebtheit. Zahlreiche Funktionsblöcke für die Anbindung an verbreitete IoT-Lösungen wie MQTT oder InfluxDB sind entweder „out of the box“ oder als Erweiterungspakete verfügbar. In der Geoinformatik führt Node-RED hingegen bislang weitestgehend ein Schattendasein.

Motivation zur Entwicklung einer GIS-Erweiterung

In unserem Arbeitsalltag sind wir ständig mit der Situation konfrontiert, mit geringen Budgets und Personalgrößen vergleichsweise ambitionierte, umfangreiche Datenprojekte realisieren zu wollen. Dieser Umstand macht die größtmögliche Automatisierung von Datenmanagementprozessen zu einem unverzichtbaren Hilfsmittel. In der Vergangenheit entwickelten wir hierfür eine Vielzahl separater Python-Skripte, bei denen die Aspekte Modularität und Wiederverwendbarkeit in aller Regel auf die verwendeten Drittbibliotheken begrenzt blieb. Ein solcher Script-„Dschungel“ ist auf lange Sicht kaum pflegbar.

In einer kurzen Zwischenphase versuchten wir zunächst, die Idee eines vollständig modularen, komponentenbasierten Ansatzes mit einer selbstentwickelten konfigurationsgetriebenen „ETL-Pipeline“-Umgebung in Python zu realisieren. Es handelte sich dabei um einen Ansatz, der Node-RED bereits stark ähnelte, allerdings ohne grafische Oberfläche. Schließlich wurden wir jedoch auf Node-RED aufmerksam und beschlossen, unser Automatisierungsproblem durch Entwicklung einer eigenen GIS-Erweiterung dafür zu lösen.

Ein großer Vorteil dieses Ansatzes gegenüber einer Sammlung von Einzelskripten ist, dass mit wachsender Anzahl implementierter Nodes beim Auftreten eines neuen Anwendungsfalls immer weniger neuer Code geschrieben werden muss. Stattdessen kann auf bereits implementierte Funktionsblöcke zurückgegriffen werden. Diese müssen lediglich per visueller Programmierung miteinander verbunden und konfiguriert werden, um den jeweiligen Anwendungsfall zu erfüllen. Dies ist in der Regel in sehr kurzer Zeit und zudem ohne Kenntnisse in textbasierter Programmierung möglich.

Zudem erhöhen Fehlerkorrekturen und andere Verbesserungen an einem einzelnen Node die Qualität *aller* Flows, die diesen Node nutzen. Auf lange Sicht steigt somit das Verhältnis zwischen Nutzen und Arbeitsaufwand exponentiell: Je mehr Nodes implementiert werden, desto mehr Anwendungsfälle können schnell und einfach als Node-RED-Flows mittels reiner visueller Programmierung realisiert werden. Und je mehr Flows mit vorhandenen Nodes implementiert werden, desto mehr zahlen sich wiederum zukünftige Verbesserungen an diesen Nodes aus.

TypeScript als Grundlage

Node-RED-Erweiterungen basieren, wie auch das Hauptprogramm, auf JavaScript. Wir wollten unser Erweiterungspaket jedoch in TypeScript entwickeln. TypeScript ist eine Erweiterung bzw. Obermenge von JavaScript, die jedoch von gängigen JavaScript-Laufzeitumgebungen nicht direkt ausgeführt werden kann. Daher wird TypeScript-Code zunächst per Compiler in gewöhnlichen JavaScript-Code übersetzt.

Wir beschlossen, als Basis für unsere eigenen Nodes zunächst einen TypeScript-Wrapper für die Extension-API von Node-RED zu implementieren. Dieser Wrapper erfüllt drei Zwecke:

1. Er ermöglicht das Schreiben eigener Node-Klassen mit der "class"-Syntax von TypeScript bzw. ECMAScript 2015.
2. Er definiert ein typisiertes Interface für die wichtigsten Teile von Node-RED's Extension-API.

Geodatenverarbeitung mit Node-RED

3. Er enthält einige zusätzliche, von uns implementierte Funktionen, die von Node-RED nicht "out of the box" zur Verfügung gestellt werden.

Der größte Teil dieser Funktionalität ist in einer abstrakten TypeScript-Basisklasse gekapselt, die als Fundament für die Entwicklung eigener Nodes in TypeScript dient. Wir haben diese kleine Bibliothek "*Node-RED TypeScript Essentials*" getauft.

Grundkonzepte unserer GIS-Nodes für Node-RED

Ein wichtiges Ziel bei der Entwicklung unserer GIS-Erweiterungen für Node-RED ist größtmögliche Flexibilität. Teilfunktionen sollten so weit, wie es sinnvoll ist, auf separate Nodes verteilt und diese mit möglichst standardisierten Schnittstellen versehen werden, um möglichst viele unterschiedliche Kombinationen zu ermöglichen. Zudem erleichtert Konsistenz die Einarbeitung in die Nutzung der Node-Sammlung. Gleichzeitig sollte die Aufteilung jedoch auch nicht zu feingranular sein, um die Implementierung von Flows nicht unnötig zu verkomplizieren.

Die Weitergabe von Daten zwischen zwei Nodes in einem Node-RED-Flow erfolgt in Form von JavaScript-Objekten. U.a. deshalb haben wir uns für GeoJSON als Standard-Austauschformat zur Übermittlung von Geodaten zwischen unseren Nodes entschieden. Für Anwendungsfälle, in denen GeoJSON als Austauschformat ungeeignet ist, werden die Daten in geeigneten Alternativformaten (z.B. GeoPackage oder GeoTiff) als Dateien auf die Festplatte geschrieben und in Messages als Dateipfad-String referenziert.

Bisher implementierte Funktionen

Unsere Node-RED-GIS-Erweiterung befindet sich noch in einer sehr frühen Entwicklungsphase. Sie wird bisher nur vom GeoNet.MRN e.V. und von der Metropolregion Rhein-Neckar GmbH im Rahmen von Forschungsprojekten und für andere eigene Zwecke verwendet. Die Codequalität der verschiedenen Nodes sowie deren unmittelbare Einsetzbarkeit für Zwecke jenseits der Projekte, für die sie entwickelt wurden, schwankt stark. In den folgenden Abschnitten stellen wir eine Auswahl von Nodes vor, die nach unserer Einschätzung bislang am ehesten für die Nutzung durch interessierte Dritte geeignet sind.

Allgemeine GeoJSON-Operationen

Eine Sammlung von Nodes zur Manipulation von GeoJSON-Datensätzen bildet das Herzstück unserer Erweiterung. Die meisten dieser Nodes erwarten eine GeoJSON-FeatureCollection als Payload eingehender Messages und senden wiederum eine GeoJSON-FeatureCollection als Payload ausgehender Messages. Einige andere Nodes wandeln Daten aus anderen Formaten in GeoJSON um, oder transformieren umgekehrt GeoJSON in andere Formate. Die folgende Liste bietet eine (unvollständige) Übersicht über die wichtigsten Nodes dieser Kategorie:

geojson-merge: Vereinigt GeoJSON-FeatureCollections aus mehreren eingehenden Messages zu einer zusammenhängenden GeoJSON-FeatureCollection.

geojson-to-centroids: Berechnet die Centroid-Punkte aller Features bzw. Geometrien einer eingehenden GeoJSON-FeatureCollection und erzeugt eine neue FeatureCollection, in der die ursprünglichen Geometrien aller Features durch deren Centroid-Punkte ausgetauscht wurden. Die Attribute der Features bleiben erhalten. Die eingehende FeatureCollection kann sämtliche in GeoJSON definierten Geometrietypen enthalten, auch gemischt. Die resultierende FeatureCollection enthält ausschließlich „Point“-Geometrien.

geojson-to-gpkg: Konvertiert eine GeoJSON-FeatureCollection in eine GeoPackage-Datei. Dieser Node wird beispielsweise für die Publizierung von Geodaten mit GeoServer benötigt, da GeoServer GeoJSON als Dateiformat für Data Stores nicht unterstützt (mehr hierzu im Abschnitt über die GeoServer-Nodes). Eine einzelne GeoJSON-FeatureCollection kann Features mit unterschiedlichen Geometrietypen enthalten, während GeoPackage-Layer (in Konsistenz mit den meisten GIS-Datenformaten und -Anwendungen) nur je einen Geometrietyp pro Layer unterstützen. Daher wird in der GeoPackage-Datei je ein Layer für jeden in der FeatureCollection enthaltenen Geometrietyp erstellt und die Features werden nach ihrem jeweiligen Geometrietyp in den entsprechenden Layer „einsortiert“.

geojson-properties-filter: Entfernt aus den Features einer GeoJSON-FeatureCollection alle Properties, die nicht in einer „Whitelist“ enthalten sind, welche im Konfigurationsdialog des Nodes definiert werden kann. Dieser Node ist beispielsweise nützlich, um Datensätze zu „bereinigen“, die mittels der OSM Overpass-Nodes aus OpenStreetMap exportiert wurden und u.U. dutzende Attribute enthalten, von denen jedoch nur eine Auswahl benötigt wird.

Automatische WMS/WFS-Publizierung via GeoServer REST-API

Zwei weitere Nodes in unserem Erweiterungspaket ermöglichen die automatische Publizierung von Geodaten als WMS/WFS-Dienste via GeoServer. Hierfür wird die HTTP-REST-API von Geoserver genutzt.

Einer dieser Nodes ist ein sogenannter *Config-Node*. Er speichert alle Einstellungen, die zur Verbindung mit einer GeoServer-Instanz benötigt werden: Endpunkt-URL, Benutzername und Passwort. Diese Daten können über den Konfigurationsdialog des Config-Nodes eingegeben werden. Config-Nodes haben in Node-RED eine Sonderstellung. Sie sind nicht Bestandteil des Flow-Graphen und senden oder empfangen keine Messages. Statt dessen werden Config-Nodes über die Konfigurationsdialoge anderer Nodes erstellt und referenziert. Sie dienen der Auslagerung von Konfigurationsdaten in Untermodule, die überall dort, wo sie gebraucht werden, eingebunden werden können.

Ein typischer Anwendungsfall für Config-Nodes sind beispielsweise Angaben zu Datenbankverbindungen, die ggf. von mehreren Nodes benötigt werden, die alle mit derselben Datenbank interagieren sollen. Nach diesem Prinzip haben wir auch unseren GeoServer-Config-Node erstellt, wenngleich dieser bisher nur von einem einzigen „normalen“ Node verwendet wird.

Dieser Node ermöglicht das Hochladen von GeoPackage-Dateien auf eine GeoServer-Instanz und deren automatische Publizierung als GeoServer-Layer bzw. als WMS/WFS-Dienste. Die Funktionalität hierfür ist Teil der GeoServer-REST-API. Sie wird automatisch durch eine HTTP-Anfrage ausgelöst, mit der die zu publizierende GeoPackage-Datei an die GeoServer-Instanz übermittelt wird. Die GeoServer-REST-API unterstützt u.a. auch das Hochladen und Publizieren von Shapefile-Dateien sowie die Erstellung von z.B. PostGIS-basierten Layern durch Upload entsprechender Data-Store- bzw. Layer-Konfigurationsdateien im XML- oder JSON-Format. Diese Möglichkeiten werden von unserer Node-RED-Erweiterung bisher nicht unterstützt.

Datendownload von OpenStreetMap via Overpass-API

Unsere Erweiterung enthält zwei Nodes für die Arbeit mit OpenStreetMap-Daten: Ein Node lädt Daten von einem Overpass-API-Endpunkt herunter und schreibt sie im ursprünglichen, Overpass-spezifischen JSON-Format in eine Datei. Der zweite Node liest eine solche Datei und konvertiert die Daten nach GeoJSON.

Die Aufteilung der Funktionalität auf zwei separate Nodes dient der Flexibilität. So ist es z.B. möglich, zu Testzwecken das Ergebnis einer Overpass-Abfrage in eine Datei zu schreiben und für wiederholte Testläufe den Inhalt dieser Datei als Eingabe für den Overpass-zu-GeoJSON-Konvertierungsnode zu verwenden. Auf den mehrfachen Download der gleichen Daten kann so verzichtet werden, was so-

Geodatenverarbeitung mit Node-RED

wohl Zeit spart als auch das IP-Adressen-bezogene Downloadvolumen-Kontingent schont. Dieses ist bei vielen Overpass-Endpunkten zur Unterbindung übermäßiger Nutzung pro Zeiteinheit (z.B. pro Tag) begrenzt.

Der OSM-Download-Node ermöglicht über seinen Konfigurationsdialog die Einstellung einer Reihe von Parametern: Die URL des zu nutzenden Overpass-Endpunkts, die räumliche Begrenzung der Abfrage (Query Bounding Box), eine Liste von Attributfiltern bzw. „Tags“, welche an die Overpass Query Language angelehnt ist, sowie den Namen bzw. Pfad der zu schreibenden Datei.

Der Overpass-zu-GeoJSON-Konvertierungsnode besitzt keine anpassbaren Konfigurationsparameter. Er erwartet als Payload einer eingehenden Message den Pfad der zu lesenden Overpass-JSON-Datei, transformiert deren Inhalt in eine GeoJSON-FeatureClass und sendet diese als Payload der ausgehenden Message weiter.

Ein typischer Flow

Unser häufigster Anwendungsfall ist die Abfrage von Daten mit Raumbezug von Drittanbieter-APIs in unterschiedlichen Formaten und deren (Wieder-)Publizierung als statische GeoJSON-Dateien auf einem „gewöhnlichen“ Webserver oder als WMS/WFS-Dienste via GeoServer. Als Zwischenschritte kommen dabei in einigen Fällen zusätzliche Operationen wie die Vereinigung von Daten aus mehreren verschiedenen Quellen, räumliche Beschneidung oder Filterung nach bestimmten Attributwerten vor. Ein typischer Flow enthält daher in der Regel folgende Nodes (in der Reihenfolge ihrer Hintereinanderschaltung):

- Ein initialer Node zum manuellen oder automatischen (periodisch zeitgesteuerten) Start des Flows
- Ein Node zum Download von Daten von einem Webdienst über HTTP(S)
- Ein Node zur Konvertierung der heruntergeladenen Daten aus dem oft API-spezifischen Download-Format in eine GeoJSON-FeatureCollection
- Ggf. ein oder mehrere Nodes zur Manipulation der FeatureCollection (z.B. Filterung von Features nach Geometrie oder Attributen)
- Ein Node zur Konvertierung von GeoJSON nach GeoPackage für die Publizierung via GeoServer-API (hierfür unterstützt GeoServer nur GeoPackage und Shapefile).
- Ein Node zur Interaktion mit der GeoServer-REST-API zwecks Upload und Publizierung der GeoPackage-Datei als GeoServer-Layer

Die folgende Abbildung zeigt einen Screenshot eines Flows zur Abfrage von ÖPNV-Haltestellenstandorten aus OpenStreetMap (Overpass API), wie er in der Weboberfläche von Node-RED dargestellt wird.

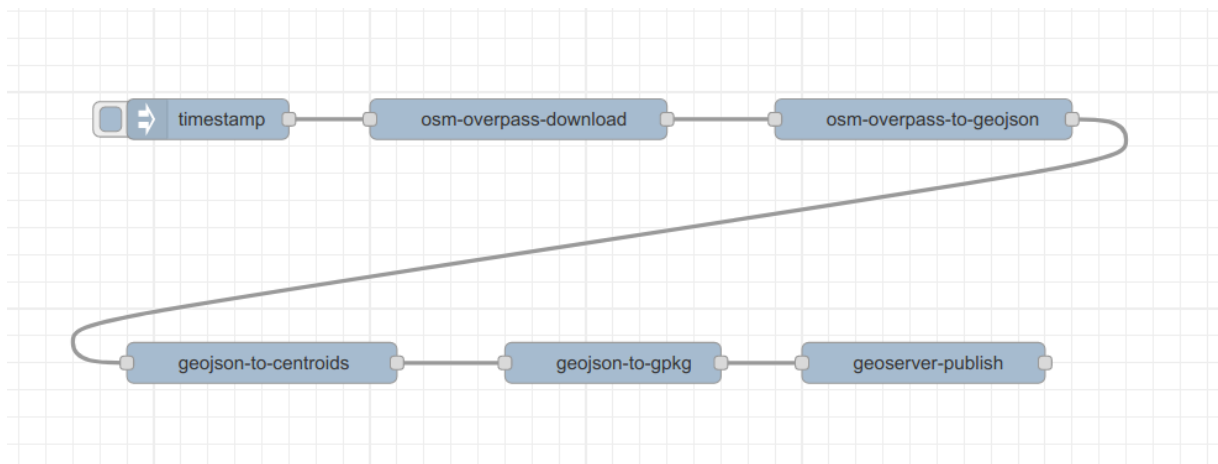


Abbildung 2: Ein typischer Node-RED-Flow mit Nodes aus unserer GIS-Erweiterung

Der Flow lädt Haltestellendaten von einem OSM-Overpass-Endpunkt herunter, konvertiert die Daten in eine GeoJSON-FeatureCollection und ersetzt alle enthaltenen Geometrien durch deren Centroid-Punkte. Die resultierende neue FeatureCollection enthält ausschließlich Point-Features. Diese wird als GeoPackage-Datei gespeichert, via GeoServer REST-API auf eine GeoServer-Instanz hochgeladen und automatisch als GeoServer-Layer (WMS/WFS-Dienste) veröffentlicht.

Verfügbarkeit, **Einschränkungen** und Ausblick

Unsere „Node-RED TypeScript Essentials“ - die Basis-Bibliothek, auf der unsere Node-RED-GIS-Erweiterung aufbaut - haben wir bereits vor einigen Monaten auf GitHub veröffentlicht ⁵.

Die GIS-Erweiterung selbst ist bisher nicht als öffentliches Repository verfügbar, da wir die Dokumentation und auch die bisherige Qualität der Implementierung für eine Veröffentlichung noch als unzureichend empfinden. Grundsätzlich stellen wir jedoch auch diese unter der MIT License als Open Source zur Verfügung. Der Code kann auf Anfrage gerne von uns bezogen werden.

Wir haben diese Node-RED-Erweiterung bisher rein nach eigenem, organisationsinternem Bedarf (u.a. für den Metropolatlas Rhein-Neckar) und als „proof of concept“ im Rahmen des Forschungsprojekts *xDataToGo – Experimentierfeld Digitaler Straßenraum* entwickelt. Bedienkomfort und Stabilität bzw. Codequalität sind bislang noch klar „ausbaufähig“. Insbesondere in Sachen Eingabeunterstützung (z.B. optimierte Bedienelemente), Eingabevalidierung und Rückmeldung von Fehlern und Problemen jeglicher Art besteht großes Verbesserungspotenzial. Eine „offizielle“ Betreuung des Projekts von unserer Seite (v.a. auch im Hinblick auf etwaige Community-Koordination, Entgegennahme von Codebeiträgen etc.) gibt es bisher nicht.

Wir freuen uns jedoch grundsätzlich über Interesse an dem Projekt sowie über eventuelle Beiträge. Auch Fragen zur Nutzung beantworten wir im Rahmen unserer zeitlichen Möglichkeiten gerne.

5 <https://github.com/geonet-mrn/node-red-typescript-essentials>

Über die Finanzierung dieser Arbeiten

Die in diesem Beitrag vorgestellten Forschungs- und Entwicklungsarbeiten wurden durchgeführt vom GeoNet.MRN e.V. in Kooperation mit der Metropolregion Rhein-Neckar GmbH, größtenteils im Rahmen des Verbundforschungsprojekts „xDataToGo – Experimentierfeld Digitaler Straßenraum“. Das Projekt xDataToGo wurde finanziert vom Bundesministerium für Verkehr und Digitale Infrastruktur (BMVI) als Teil der Forschungsförderinitiative *mFUND* (Modernitätsfonds). Der GeoNet.MRN e.V. und die Metropolregion Rhein-Neckar GmbH bedanken sich hiermit herzlich beim BMVI für die Unterstützung.



Abbildung 3: Das mFund-Logo

Kontakt zum Autor:

Sebastian Bechtold
GeoNet.MRN e.V.
Julius-Hatry-Straße 1
68163 Mannheim
sbechtold@geonet-mrn.de

GDAL - Konzept und Werkzeuge

GDAL ist nicht nur eine Programmbibliothek, die eine einheitliche Schnittstelle zu einer Vielzahl verschiedener Geodaten-Formate zur Verfügung stellt, sondern auch eine Sammlung verschiedener Werkzeuge zur Bearbeitung von Geodaten, die hier vorgestellt wird.

GDAL (Geospatial Data Abstraction Library) wird von über 100 Anwendungen als Übersetzungsbibliothek für eine Vielzahl von Raster- und Vektor-Formaten benutzt. Wir werden das GDAL Datenmodell für Raster- und Vektordaten anhand von Beispielen vorstellen, wobei das relativ neue COG Format (Cloud Optimized GeoTIFF) für cloud-basierte Datendienste besonders erwähnt werden soll.

Bei Geodaten spielt das Koordinatenreferenzsystem eine zentrale Rolle, deswegen werden wir auch auf die mit GDAL 3 eingeführten Änderungen eingehen, vor allem der Wechsel von WKT1 zu WKT2 zur Beschreibung von Koordinatenreferenzsystem. In diesem Zusammenhang wird auch die PROJ Bibliothek erwähnt, die (nicht nur) von GDAL zur Koordinatenkonversion und -transformation benutzt wird. Mit den neuen PROJ Versionen stehen mehr und genauere Methoden zur Koordinatentransformation zur Verfügung.

Die GDAL Werkzeuge `gdalinfo`, `gdal_translate`, `gdalwarp`, `ogrinfo`, `ogr2ogr` werden im Detail mit Beispielen erläutert.

Markus Metz

GIS mit der Kommandozeile

Dieser Vortrag gibt einen einsteigerfreundlichen Überblick über Kommandozeilen-Programme mit denen sich räumliche Aufgaben schnell und effizient lösen lassen. Diese sind sowohl auf Windows als auch auf Linux einsetzbar und lassen sich in wiederverwendbaren Skripten nutzen. Das Spektrum umfasst ältere, etablierte Tools, leichte auf Python oder JavaScript basierte Programme, Schnittstellen zu Desktop-Anwendungen und Werkzeuge um Web-APIs für Routing oder GeoCoding anzusprechen.

Geodaten können nicht nur mit einem Desktop-GIS oder einer Programmiersprache bearbeitet werden. Auch die Kommandozeile eignet sich dazu und bietet entscheidende Vorteile. So starten die entsprechenden Programme quasi sofort und die einmal erstellte Befehle lassen sich speichern und wiederverwenden. Außerdem entfällt eine aufwändige Benutzeroberfläche, so steht dem Geodaten-Prozessierung auf einem Server oder der „Cloud“ nichts im Wege. Zwar ist für Neueinsteiger/innen der Umgang in Terminal erst ungewohnt, jedoch lässt sich das schnell erlernen. Diese Investition zahlt sich auf alle Fälle aus, da sich damit eine ganz neue Welt an vorher unbekannten Möglichkeiten eröffnet.

Dieser Vortrag stellt ein breites Spektrum an etablierten Kommandozeilen-Werkzeugen vor, die sowohl auf Windows als auch auf Linux einsetzbar sind. Zum einen gibt es einen Überblick über Langzeit-Klassiker wie GDAL/OGR, Proj oder GEOS. Andererseits über eher leichtgewichtigen Tools liegen wie Fiona, Rasterio, mapshaper, topojson oder geojsonio. Es folgt ein Einblick in Desktopprogramme mit Schnittstelle zur Kommandozeile z.B. QGIS, GRASS oder SAGA GIS. Anschließend begeben wir uns auf ein Exkurs zum Ansprechen von Web-APIs für beispielsweise Routenplanung oder Geocoding. Zu guter Letzt gibt es einen Ausblick wie man Befehle in der Kommandozeile miteinander verketteten und in wiederverwendbare Skripte speichern kann, um regelmäßige Aufgaben zu automatisieren.

Jakob Miksch

Effizientes many-to-many ÖPNV-Routing mit gtfsrouter

Es gibt eine Fülle von FOSS für das One-to-One-GTFS-Routing, unter anderem opentripplanner. Effizientes Many-to-Many-Routing erfordert jedoch andere Algorithmen. Das gtfsrouter-Paket verfügt über einen neu entwickelten Algorithmus, der speziell für Many-to-many-Routing optimiert wurde und eine systemweite Abfrage von Routing-Statistiken für die großräumige Stadtplanung ermöglicht.



gtfsrouter ist ein FOSS-Package für das Routing mit "General Transport Feed Specification" (GTFS) Daten. Das Package ist in der Sprache R geschrieben, mit hocheffizienten C++-Algorithmen. Dieser Vortrag wird zwei der besonderen Aspekte des gtfsrouter-Pakets vorstellen. Der erste ist die Fähigkeit, Transfer-Tabellen zu konstruieren, die benötigt werden, um die erwarteten Zeiten für den Transfer zwischen den verschiedenen Haltestellen innerhalb eines Transportsystems zu spezifizieren. Viele Feeds verfügen nicht über diese Tabellen, so dass sie nicht für Routing-Abfragen verwendet werden können. Die Fähigkeit, Umsteigetabellen zu erstellen, ermöglicht es auch, Feeds miteinander zu verbinden, was für Städte wichtig ist, in denen verschiedene Verkehrsbetriebe unabhängige Feeds bereitstellen. Der Vortrag wird sich hauptsächlich auf einen zweiten Aspekt des gtfsrouter-Programms konzentrieren, nämlich eine hocheffiziente Many-to-Many-Routing-Engine. Many-to-many-Routing mit Fahrplandaten wie GTFS-Daten unterscheidet sich stark von konventionellem Routing durch statische Netzwerke wie Straßen, und es gibt nur sehr wenige FOSS-Lösungen für solche Abfragen, so dass viele Entwickler versuchen, effiziente Lösungen durch Parallelisierung von One-to-One-Abfragen mit opentripplanner zu finden. Solche Lösungen sind jedoch zu langsam für effiziente Lösungen im großen Maßstab, da sie zum Beispiel viele Stunden oder sogar Tage benötigen, um die Fahrzeiten zwischen allen Bahnhofspaaren in Berlin abzufragen. Der neue gtfsrouter-Algorithmus reduziert die Berechnungszeiten auf Minuten statt auf Stunden und bietet Städten die Möglichkeit, umfassende Übersichten über ihr gesamtes Verkehrssystem zu erstellen.

mark padgham

Barrierefreie Indoor-Karten

JAN SCHMALFUSS-SCHWARZ, CHRISTIN ENGEL, DR. CLAUDIA LOITSCH

Abstract

Innerhalb des Beitrags zum Thema „Barrierefreie Indoor-Karten“ wird auf das vielschichtige Gebiet der Gebäudekartendarstellung für Menschen mit Blindheit, Seh- sowie Mobilitätseinschränkungen eingegangen. Insbesondere sollen die Nützlichkeit und Relevanz frei verfügbarer Gebäudekartendaten dargestellt, der Bedarf von spezifischen Barrierefreiheitsinformationen erläutert sowie der aktuelle Stand der Technik für die barrierefreie Kartendarstellung aufgezeigt werden. Darüber hinaus werden Lösungsideen zur Erfassung der dafür benötigten Daten vorgestellt und diskutiert. Der Fokus liegt dabei auf der Annahme, dass Reisende sich bereits vorab über die örtlichen Gegebenheiten informieren möchten.

Die Relevanz dieses Themas ist vielschichtig. Digitale Barrierefreiheit wird durch die EU-Richtlinie über den barrierefreien Zugang zu Webangeboten zunehmend durchgesetzt. Für digitale Karten gibt es bisher jedoch weder Standards, wie diese umgesetzt werden kann, noch ausreichend Daten, welche die Barrierefreiheit unserer physischen Umgebung umfassend beschreiben. Letzteres trifft insbesondere für Gebäude zu, da es nur wenige georeferenzierte Innenkarten gibt (z. B. auf OpenStreet-Map), die für eine barrierefreie Darstellung von Gebäuden geeignet sind.

1. Einleitung

Mobilität spielt in unserem alltäglichen Leben eine zentrale Rolle. Dabei ist es essentiell, nicht nur die Wegfindung zwischen verschiedenen Orten und Gebäuden, sondern auch die Bewältigung von Wegen innerhalb eines Gebäudes zu betrachten. Darüber hinaus ist es für Menschen mit Beeinträchtigungen entscheidend, zu wissen, ob ein Gebäude oder Teile eines Gebäudes barrierefrei bzw. welche Barrieren evtl. vor Ort zu überwinden sind. Zusätzlich werden auch Informationen über spezielle Merkmale (z.B. taktile Beschriftungen) benötigt. Insbesondere für Menschen mit einer Einschränkung ist eine nahtlose und verlässliche (digitale) Assistenz für die selbstständige Mobilität unverzichtbar.

Da eine unabhängige Mobilität eine hohe soziale Relevanz hat und diese überdies spezifische Herausforderungen an Menschen mit Einschränkung stellt [1], ist es nötig, deren Bedürfnisse bei der Erfassung und Ausgabe von Indoor-Karten zu berücksichtigen. Der damit verbundene Aufwand zur Erzeugung solcher Karten ist derzeit hoch und zumeist manuell nur schwer realisierbar. Um dies in Zukunft zu verbessern, ist es notwendig, sowohl die Erfassung von Daten (insb. Daten über Gebäude) durch einfache und passgenaue Anwendungssoftware, aber auch die eigentliche Verwendung von Kartendaten durch diverse Nutzende mit ihren individuellen Bedürfnissen zu unterstützen. Dieser Beitrag soll das Thema weiter sensibilisieren und konkrete Lösungsansätze aufzeigen.

Im Folgenden gehen wir zunächst auf die speziellen Anforderungen von Menschen mit Einschränkungen an Gebäudeinformationen ein, beschreiben dann aktuelle Herausforderungen und präsentieren letztlich Lösungsansätze, welche wir im Rahmen des Forschungsprojektes AccessibleMaps¹ entwickeln.

1 <https://accessiblemaps.de>, Stand: 01.05.2021

2. Anforderungen von Menschen mit Einschränkungen an Gebäudeinformationen

Menschen mit Einschränkungen benötigen zur Findung ihres Weges innerhalb eines Gebäudes detaillierte Informationen bezüglich des Gebäudegrundrisses, der Orientierungsmerkmale, der Barrieren sowie der zeitlichen Zusammenhänge [2]. Diese unterschiedlichen Informationen werden dabei gleichsam für die Planung eines Weges und dessen spätere Bewältigung gebraucht. Die Diversität der nötigen Informationen ist gleichzeitig durch die Mannigfaltigkeit der verschiedenen Nutzendengruppen groß (vgl. Abbildung 1) [3]. So gibt es beispielsweise für Reisende mit Blindheit eine Vielzahl von Barrieren in Gebäuden, wobei eine unterlaufbare Treppe hier nur exemplarisch erwähnt sei. Ferner mangelt es aktuell an Lösungsansätzen, welche einerseits die hohe Informationsdichte bereitstellen und andererseits frei zugänglich sind. Zudem gibt es in diesem Bereich wenige Entwicklungen hinsichtlich barrierearmer Ausgabemedien. Um dieses Problemfeld in Zukunft in einem menschenzentrierten Prozess adressieren zu können, empfiehlt sich beispielsweise der Einsatz von Personas, welche in Form von fiktiven Personenbeschreibungen die verschiedenen Gruppen der Nutzenden repräsentieren sollen, um ein Verständnis für die unterschiedlichen Bedürfnisse und Anforderungen zu schaffen. Im Projekt AccessibleMaps haben wir sechs Personas² entwickelt, welche stellvertretend für die Zielgruppen Menschen mit Blindheit, Menschen mit Seheinschränkung und Menschen mit einer Mobilitätseinschränkung Herausforderungen, Orientierungsstrategien aber auch Hilfsmittel beschreiben.

	(eher) nicht wichtig			Teilweise wichtig			eher wichtig bis sehr wichtig		
	BL	SB	MOB	BL	SB	MOB	BL	SB	MOB
Adresse des Gebäudes	1.6	4.7	0.0	0.0	4.6	2.9	98.4	90.7	97.1
Raumname des Ziels	0.0	9.3	8.8	3.1	4.6	17.7	96.9	86.1	73.5
Lage aller barrierefreien Eingänge	51.6	39.5	14.7	23.4	23.3	2.9	23.4	34.9	79.4
Lage der Haupteingänge	3.1	4.7	8.8	12.5	14.0	11.8	84.4	81.4	79.4
Lage aller Ein- und Ausgänge	35.9	30.2	26.5	29.7	25.6	20.6	34.4	44.2	52.9
Gebäudeplan	34.4	30.2	41.2	20.3	25.6	17.7	43.8	44.2	38.2
Textuelle Beschreibung	12.5	34.9	53.0	20.3	16.3	20.6	64.1	48.8	26.5
Informationen zu Nachbargebäuden	25.0	30.2	64.7	39.1	41.9	20.6	35.9	27.9	11.8
Anzahl der Etagen	23.4	23.3	41.2	34.4	30.2	20.6	42.2	46.5	35.3
Lage der Treppen(-häuser)	7.8	18.6	50.0	21.9	27.9	5.9	70.3	53.5	41.2
Art der Treppen (z.B. Wendeltreppe)	43.8	51.2	58.8	23.4	25.6	5.9	32.8	23.3	35.3
Lage der Fahrstühle	26.6	32.6	2.9	32.8	14.0	2.9	40.6	53.5	94.1
Lage taktiler Beschreibungen	9.4	44.2	5.9	9.4	25.6	5.9	79.7	27.9	8.8
Lage einer taktilen Karte	21.9	48.8	8.8	15.6	20.9	5.9	59.4	27.9	5.9
Lage von Behindertenparkplätzen	64.1	72.1	5.9	20.3	14.0	8.8	14.1	9.3	79.4
Lage der Toiletten	12.5	23.3	0.0	23.4	14.0	20.6	64.1	62.8	79.4
Interessante Orte (z.B. Café)	10.9	20.9	11.8	32.8	34.9	20.6	54.7	44.2	64.7
Öffnungszeiten	4.7	4.7	2.9	9.4	16.3	5.9	85.9	79.1	91.2
Anderes	1.6	2.3	0.0	1.6	0.0	0.0	23.4	20.9	8.8
Temporäre Barrieren	1.6	11.6	0.0	14.1	16.3	5.9	78.1	69.8	91.2

Abbildung 1: Relevante Informationen für Menschen mit Blindheit (BL) und Seheinschränkungen (LV) für die Planung sowie Durchführung einer Reise [3]

3. Aktuelle Herausforderungen

Die bisher beschriebenen Problemstellungen lassen sich verkürzt in die Herausforderungen der Kategorien "Unzureichende Gebäude-Daten", "Umständliche Daten-Erfassung" sowie "Unzureichendes Wissen über spezifische Bedürfnisse der diversen Nutzendengruppen" untergliedern. Um diese zu adressieren, bedarf es eines möglichst offenen Standards zur Speicherung von Indoor-Karten und darin enthaltenen Barrierefreiheitsinformationen. Ein dafür geeignetes Format könnte das innerhalb von Open-

2 https://accessiblemaps.de/?page_id=453, Stand: 01.05.2021

Barrierefreie Indoor-Karten

StreetMap (OSM) beschriebene Simple Indoor Tagging³ (SIT) darstellen, da dieses zum einen durch die nicht hierarchische Struktur das Erfassen von Informationen erleichtert. Zum anderen gibt es einen freien Zugang zu den Daten, sodass unterschiedliche Interessengruppen eigene Möglichkeiten zur Ausgabe von Indoor-Karten umsetzen können. Eine Analyse der Verbreitung von SIT-konformen Karten in OSM zeigte aber ein aktuelles Defizit bezüglich der Anzahl der vollständig umgesetzten Karten auf. So wurden für eine Stichprobe die Städte Berlin, Rom, Wien und Paris hinsichtlich der enthaltenen Indoor-Karten in dem SIT-Format untersucht. Dabei ergab sich, dass nur wenige Gebäude mit den Tags *min_level* und *max_level* versehen sind, was der Minimalanforderung zur Identifizierung von repräsentierten Gebäuden im SIT-Format entspricht. Darüber hinaus sind nur ein kleiner Anteil von Gebäuden vollständig, mit Räumen, Korridoren, etc. in SIT erfasst (vgl. Tabelle 2). Ferner findet man aktuell häufig eine Vielzahl von SIT spezifischen Tags, die zugleich keinem Gebäude zuordenbar sind, welches die Mindestanforderungen zur Identifizierung von Gebäuden im SIT-Format erfüllt (vgl. Abbildung 2) [4]. Daraus lässt sich der Anspruch ableiten, die Anzahl der in SIT gemappten Indoor-Karten zu erhöhen und nach Möglichkeit diese mit Barrierefreiheitsinformationen anzureichern. Um diese Aufgabe zu bewältigen, ist es einerseits notwendig aufzuzeigen, welchen Mehrwert spezifische Indoor-Karten mit sich bringen und somit die Erfassung dafür benötigter Daten zu erhöhen. Andererseits bedarf es diverser Hilfswerkzeuge, um die Erfassung der benötigten Informationen zu erleichtern und einen Qualitätsstandard etablieren zu können.

Tabelle 2: Tabellarische Übersicht der Anzahl von Gebäuden, die den definierten SIT minimalen Anforderungen entsprechen, komplett in SIT erfasst sind sowie mit Barrierefreiheitsinformationen für Menschen mit Mobilitätseinschränkungen, Blindheit und Seheinschränkungen versehen sind (Stand: 10.04.2020)

Stadt	Berlin	Rom	Wien	Paris
<i>Gebäude mit minimal Anforderungen an SIT</i>	6	9	7	5
<i>Gebäude komplett mit SIT gemappt</i>	1	0	3	0
<i>Komplett nach SIT erfasste Gebäude mit Barrierefreiheitsinformationen für Mobilitätseingeschränkte Menschen</i>	1	0	3	0
<i>Komplett nach SIT erfasste Gebäude mit Barrierefreiheitsinformationen für Menschen mit Blindheit oder Seheinschränkungen</i>	0	0	0	0

3 https://wiki.openstreetmap.org/wiki/Simple_Indoor_Tagging, Stand: 01.05.2021

Barrierefreie Indoor-Karten

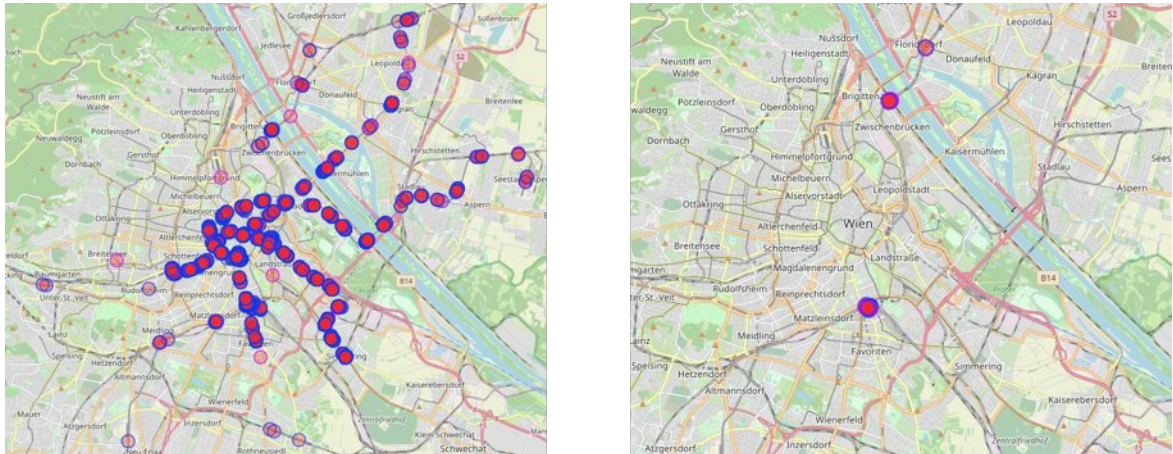


Abbildung 2: Vergleich aller SIT-Tags (links) mit allen in SIT komplett erfassten Gebäuden (rechts) in Wien, Quelle: <http://overpass-turbo.eu> (Stand: 06.05.2020)

4. Lösungsansätze

Wie bereits beschrieben, lässt sich die Problemstellung der "Barrierefreien Indoor-Karten" in zwei Themenfelder untergliedern (vgl. Abbildung 3). Dazu zählt die Erzeugung entsprechender Indoor-Karten sowie deren Darstellung, wobei innerhalb dieses Abschnitts auf beide Bereiche einzeln eingegangen werden soll.



Abbildung 3: Untergliederung der Problemstellung "Barrierefreier Indoor-Karten" in die Bereiche Erzeugung und Ausgabe

4.1 Erzeugung von Indoor-Karten, mit Barrierefreiheitsinformationen

Die Erzeugung entsprechender Indoor-Karten schließt neben der Erfassung noch nicht kartographierter Gebäude auch die Transformationen sowie Anreicherung bereits vorhandener Indoor-Karten mit ein. Diesbezüglich gilt es, eine Vielzahl differenter Ansätze für die gegebenen Problemstellungen zu entwickeln sowie für die Einhaltung spezifischer Eintragungen innerhalb der OSM-Community zu werben.

4.1.1 Transformation bestehender Karten in das neue Schema

Das Themengebiet der Transformation bereits vorhandener Indoor-Karten umfasst sowohl die OSM interne Transformation bestehender Indoor-Karten in andere Formate als auch die Umwandlung anderer Formate wie beispielsweise BIM (Building Information Modelling) in SIT. Dafür wurden innerhalb des Forschungsprojekts AccessibleMaps zwei prototypische Anwendungen geschaffen, die es den Nutzenden zum einen erlauben, Indoor-Karten des Formats Indoor-OSM in SIT zu konvertieren. Zum anderen wurde eine Datenstruktur entwickelt, welche architektonische Gebäudepläne vereinfacht und in das SIT-Format sowie in IndoorGML wandeln kann. Als Grundlage dafür dienen von der TU-Dresden bereitgestellte SVG-Pläne. Um die Verarbeitungspipeline des Tools nutzen zu können, bedarf es der Schaffung eines Parsers, welcher die Daten in die entsprechende interne Datenstruktur überführt. (vgl. Abbildung 4)

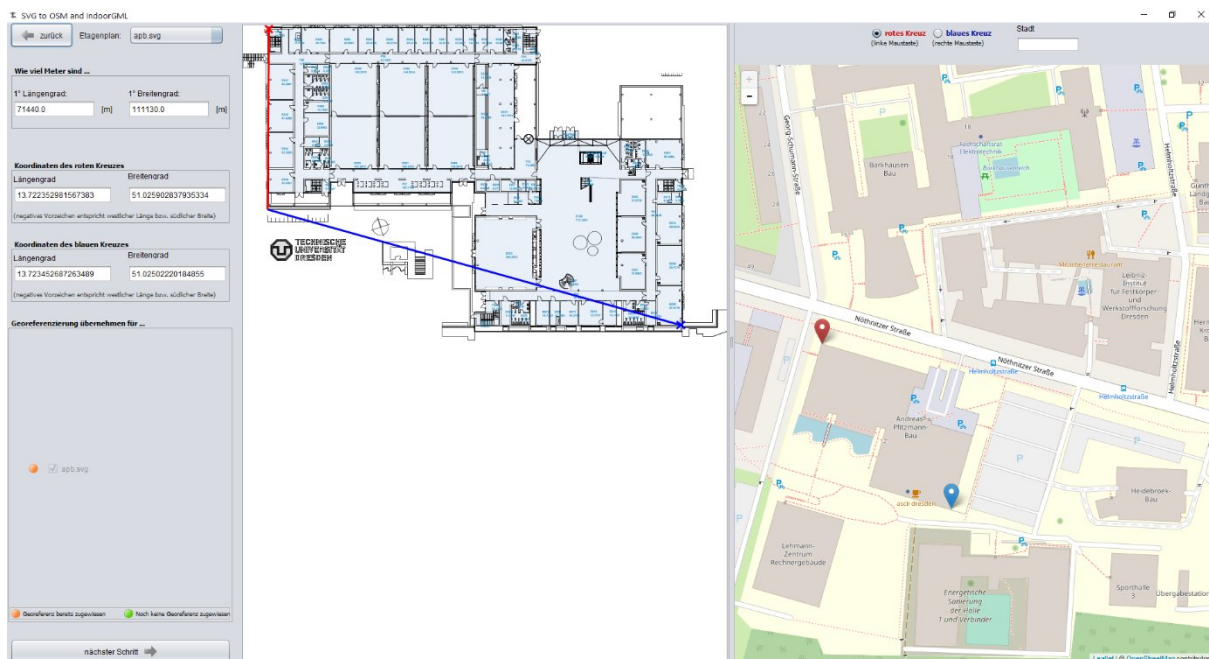


Abbildung 4: Prototypische Umsetzung zur Konvertierung architektonischer Gebäudepläne nach SIT

4.1.2 Analyse Tools zur Steigerung der Qualität der Indoor-Karten

Da nicht alle benötigten Informationen aus den verschiedenen Ausgangsformaten extrahiert werden können und darüber hinaus davon auszugehen ist, dass keine tiefgründige Expertise bezüglich der benötigten Barrierefreiheitsinformationen bei den meisten Erstellenden von Karten vorhanden ist, ist es nötig, Anwendungen zu entwickeln, welche den Kartenerzeugungsprozess unterstützen. Diese müssen die bereits vorhandenen Kartendaten analysieren und Fehler beziehungsweise Unvollständigkeiten aufzeigen. [5] (vgl. Abbildung 5)

Barrierefreie Indoor-Karten

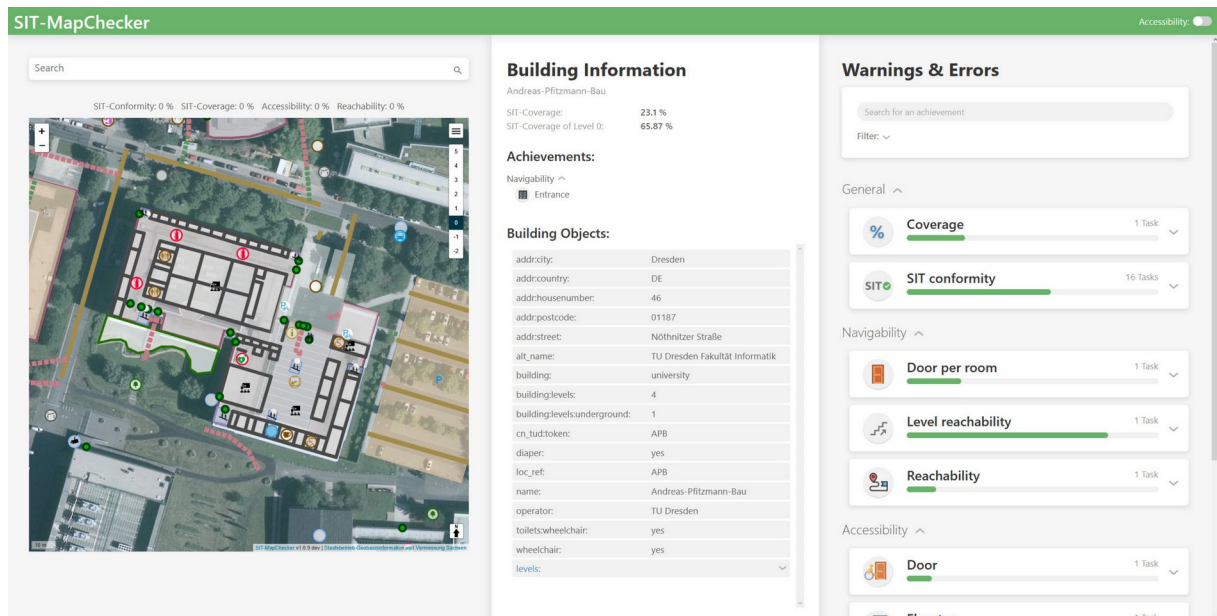


Abbildung 5: Prototypisches Analyse Tool zur Überprüfung auf SIT Konformität und auf Barrierefreiheitsinformationen

4.1.3 Unterstützte Crowdsourcing Ansätze zur Erzeugung von Indoor-Karten sowie deren Anreicherung

Der abschließende Bereich beleuchtet nochmals die grundlegende Generierung von Indoor-Karten-Daten. Da der Zugang zu anderen Formaten, welche bereits digital aufbereitete Gebäudepläne enthalten, nur in bestimmten Szenarien vorausgesetzt werden kann, bedarf es weiterer Hilfsmittel. Diese können einerseits auf der Analyse und Konvertierung von Fluchtplänen aufbauen (vgl. Abbildung 6). Andererseits können SLAM-Verfahren zur Erzeugung von Karten in Betracht gezogen werden.

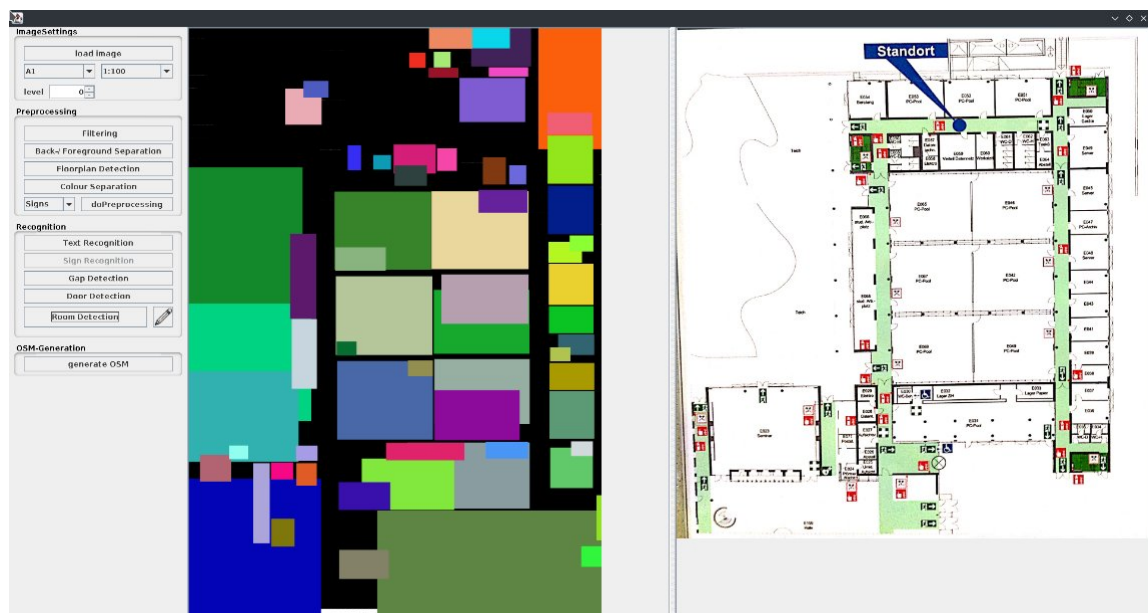


Abbildung 6: Prototyp zur Umwandlung von Fluchtplänen zu SIT

4.2 Automatische Erstellung von barrierearmen & individuellen Indoor-Karten-Darstellungen

Bei Kartenrepräsentationen handelt es sich um stark visuelle Konzepte, die die Lage von Objekten in Relation zueinander darstellen. Dabei kommen unterschiedliche Kartentypen wie Mobilitäts- oder Orientierungskarten zum Einsatz, die die Nutzenden bei der Orientierung und Navigation unterstützen sollen. Je nach avisiertem Zweck werden unterschiedlich detaillierte Informationen ausgegeben. Grafiken dieser Art stellen insbesondere für Menschen mit Blindheit oder Seheinschränkung eine Barriere dar. Nachfolgend werden verschiedene Ausgabeformate für Karten angerissen, die das Potential haben, Karten und deren Inhalte für Menschen mit Seheinschränkungen zugänglich wiederzugeben. Dabei können die verschiedenen Formate unterschiedliche Arten von Informationen vermitteln und somit für unterschiedliche Zwecke eingesetzt werden.

4.2.1 Visuell am Desktop-PC oder Smartphone

Digitale visuelle Karten können für Menschen mit unterschiedlichen Seheinschränkungen optimiert dargestellt werden. Aufgrund der Vielzahl an möglichen Einschränkungen des Sehsinns ist es nicht möglich, eine für alle Menschen optimierte Darstellung von Karten zu erzeugen. Vielmehr sollte es das Ziel sein, eine adaptierbare Form bereitzustellen, die gemäß der Bedürfnisse der einzelnen Nutzenden von diesen selbst angepasst werden kann. Für häufig auftretende Beeinträchtigungsformen können adaptierbare Nutzungsprofile zur Verfügung gestellt werden. Neben der Anwendung spezieller Farbpaletten und Hochkontrastdarstellungen, die für Menschen mit Farbseheinschränkungen konzipiert wurden, sollten auch Beschriftungen und Symbole skalierbar sein. Auch die Reduzierung der auf der Karte dargestellten Informationsdichte kann sich für Menschen mit Seheinschränkung positiv auf die Wahrnehmung auswirken. Hierfür können Filter dienen. Überlagerungen von Interaktionselementen auf der Karte sollten ebenso vermieden werden. Zu diesem Zweck kann eine Vollansicht der Karte unterstützend wirken, die es vor allem kleineren Geräten gestattet, die Karte möglichst groß darstellen zu können. Gerade bei der Arbeit mit kleinen Endgeräten kann es eine wertvolle Funktion für Menschen mit Seheinschränkung sein, Karten mittels einer Exportfunktion physisch ausdrucken zu können. Insgesamt gilt es hier im Sinne der Benutzungsfreundlichkeit, einen Trade-off zwischen wohldefinierten Voreinstellungen und anpassbaren Darstellungen für individuelle Bedarfe zu finden.

4.2.2 Taktile Drucke

Taktile Karten bestehen aus erhabenen, fühlbaren Elementen. Unterschiede werden hier mit Hilfe verschiedener Muster, Linienstile und Symbole dargestellt und zumeist in der Legende referenziert. Taktile Karten enthalten häufig Brailleschrift (sogenannte "Blindenschrift"), die die Inhalte der Karte erklären. Es gibt verschiedene Methoden, diese Art von Karten herzustellen. Schwellkopien, bei denen die Karte auf ein spezielles Papier gedruckt und anschließend mit einem Fuser erhitzt wird, sodass bedruckte Bereiche erhaben erscheinen sowie Prägedrucke, bei denen die Elemente in das Papier eingestanzt werden, sind zwei der gängigsten Methoden. Taktile Repräsentationen erfordern jedoch aufgrund der wesentlich geringeren Auflösung (etwa 10dpi bis 20dpi) die Einhaltung spezifischer Gestaltungsrichtlinien wie Minimalgrößen und Abstände für Elemente (z.B. [6] und [7]). Zur Gestaltung taktiler Karten ist daher ein wesentlich höherer Abstraktionsgrad und in der Folge eine stärkere Generalisierung notwendig, als es bei visuellen Karten der Fall ist. Es können im Vergleich nur wenige Informationen dargestellt werden. Dennoch ist die Nützlichkeit taktiler Materialien für das räumliche Verständnis, insbesondere im Bereich der Orientierung, bereits belegt. Die automatische Erzeugung taktiler Karten ist aufgrund des hohen Anspruchs an die Expertise der Erstellenden ein wichtiges Ziel für die Zukunft. Daneben ist es notwendig, auch den gesamten Erstellungsprozess taktiler Karten (inkl. der Auswahl des korrekten Kartenausschnitts sowie der angezeigten Elemente) auch Menschen mit Blindheit zu ermöglichen, um eine gleichberechtigte und unabhängige Teilhabe zu gewährleisten. In diesem Bereich sind noch weitere Entwicklungen und Forschungen, insbesondere für Indoor-Karten, notwendig.

4.2.3 Audio-Taktile Ausgabeformate

Brailleschrift, die häufig zur Erläuterung von Karteninhalten auf taktilen Materialien verwendet wird, stellt für viele Menschen mit Blindheit oder Seheinschränkung eine Barriere dar, da immer weniger Menschen das Lesen dieser Schrift beherrschen. Diesem Problem kann mit der Verwendung einer zusätzlichen Audio-Komponente begegnet werden. Diese kann ebenso dafür genutzt werden, weitere Informationen, z. B. zur Art von Türen oder Gefahrenstellen in Gebäuden, bereitzustellen. Dafür gibt es unterschiedliche technische Entwicklungen, um taktile Elemente mit einer Audioausgabe zu kombinieren, die interaktiv genutzt werden können. Unter anderem kommen hierfür Anwendungen in Frage, bei denen die taktile Grafik auf ein Tablet gelegt wird, um Eingaben durch die Nutzenden zu erkennen und auditiv per Detail-on-Demand (z. B. verbale Erläuterungen oder Geräusche) auszugeben. Ebenso können digitale Stifte oder Systeme, die die Finger der Nutzenden tracken, Anwendung finden. Die Audio-Komponente kann ebenso bei der Exploration der Karteninhalte unterstützen. Die Nützlichkeit der jeweiligen Systeme hängt insbesondere auch von dem Nutzungsszenario ab. So orientieren sich Personen mit Blindheit oder Seheinschränkung vorrangig auch über den Hörsinn, der für mobile Szenarien erhalten bleiben muss. Audio-Inhalte sollten in jedem Fall so gestaltet sein, dass sie die Informationen effektiv vermitteln und von den Nutzenden gesteuert werden können. Eine besondere Herausforderung bei der Entwicklung von audio-taktilen Karten stellt die Trennung zwischen Eingabe und Ausgabe dar, da das System im Stande sein muss zu unterscheiden, ob Nutzende bei Berührung die taktilen Elemente erkunden oder eine Eingabe tätigen wollen. Häufig wird deshalb die Doppel-Tab Geste als klassische Selektionsgeste gewählt.

4.2.4 Screenreader-kompatible Karten

Menschen mit Blindheit oder Seheinschränkung verwenden zur Ausgabe von auf dem Bildschirm präsentierten Informationen häufig ein Bildschirmvorleseprogramm (auch "Screenreader"). Ein Beispiel für einen kostenfreien und OpenSource-basierten Screenreader ist "NVDA", aber auch die meisten Betriebssysteme verfügen über ein Tool zum Vorlesen der Bildschirm Inhalte. Diese greifen zumeist auf die DOM-Struktur der Anwendungen zu und können so die Semantik der visuellen Bildschirm Inhalte, inklusive der Zustände von Buttons und Interaktionsobjekten (z. B. "aktiviert" oder "deaktiviert") erfassen und für Nutzende mit Blindheit auditiv oder auf einer Braillezeile wiedergeben. Mit Hilfe von Screenreadern können folglich auch Karten nicht-visuell bedient werden. Um Screenreader-Nutzenden den Zugriff auf Karteninhalte zu ermöglichen, werden wohldefinierte, semantisch ausgezeichnete SVG-Dateien zur Kartendarstellungen empfohlen. Die meisten Screenreader greifen dabei auf das "title" oder das "desc" Attribut von SVG-Elementen zu und können die entsprechenden Inhalte auslesen. Mit der Festlegung eines "tabindex" kann darüber hinaus die Reihenfolge, in der Screenreader-Nutzende Elemente repräsentiert bekommen, festgelegt werden. Mit "tabindex=-1" werden beispielsweise Screenreader angewiesen, das betreffende Element zu ignorieren. Dies kann genutzt werden, um reine Dekorationselemente ohne Inhaltskomponente zu kennzeichnen. Außerdem sollten Elemente einen aussagekräftigen Namen haben sowie zweckmäßig gruppiert werden. Darüber hinaus gilt es zu beachten, dass wiederkehrende Elemente nach Möglichkeit mittels "use" referenziert und somit wiederverwendet werden. Die Etablierung von wohlgeformten, screenreader-kompatiblen und semantisch aussagekräftigen SVG-Dateien zur Darstellung von Karten, kann durch eine Standardisierung vorangetrieben werden. Diese ist für die Zukunft anzustreben und hat das Potential, zugängliche Kartenformate für verschiedene Anwendungsfelder bereitzustellen. Dadurch würde auch die Erstellung und Verwendung von Karten-Parsern zur Weiterverarbeitung der Karten in SVG erheblich vereinfacht werden.

4.2.5 Mobile Interaktionskonzepte

Ein weiterer Aspekt ist die Nutzung von Karten während eines Gebäudebesuches. Dafür ist es nötig, mobile Interaktionskonzepte zu erforschen, die es den verschiedenen Nutzenden erlauben, während des Sichfortbewegens Informationen über ihre aktuelle Umgebung, ihren Weg oder andere Bereiche

Barrierefreie Indoor-Karten

des Gebäudes abzufragen. Die Interaktion muss dabei so gestaltet sein, dass sie keine neuen Barrieren schafft.

5. Zusammenfassung

Die Gestaltung barrierefreier Indoor-Karten bedingt einerseits die Schaffung möglichst einheitlicher und informationsreicher Karten-Daten sowie einer Vielzahl verschiedener Ausgabemöglichkeiten, um jeden einzelnen Nutzenden bestmöglich in der Orientierung und Navigation zu unterstützen. Die bisherigen Ansätze sind diesbezüglich noch unzureichend und müssen daher für eine inklusive Mobilität erweitert werden. Dafür ist es unabdingbar, Nutzende in den Forschungsprozess mit einzubeziehen.

6. Danksagung

Finanziert durch das Bundesministerium für Arbeit und Soziales (BMAS) aus Mitteln des Ausgleichsfonds (Förderkennzeichen: 01KM151112).

Kontakt zum Autor:

Jan Schmalfuß-Schwarz
TU Dresden, Fakultät Informatik, Professur für Mensch-Computer Interaktion
Angestellter im Projekt Accessible Maps

Besucheranschrift:

Andreas-Pfitzmann-Bau, 1.0 OG, Raum 1062
Nöthnitzer Straße 46
01187 Dresden

Telefon: 0351 463-38495

eMail: jan.schmalfuss-schwarz@tu-dresden.de

Literatur

- [1] Park, J., Chowdhury, S.: *Investigating the barriers in a typical journey by public transport users with disabilities*. Journal of Transport and Health 10 (May 2018), 361–368 (2018)
- [2] Froehlich, J.E., Brock, A.M., Caspi, A., Guerreiro, J., Hara, K., Kirkham, R., Schöning, J., Tannert, B.: *Grand challenges in accessible maps*. Interactions 26 (2), 78–81 (mar 2019)
- [3] Christin Engel, Karin Müller, Angela Constantinescu, Claudia Loitsch, Vanessa Petrausch, Gerhard Weber, Rainer Stiefelhagen. *Travelling more independently: A Requirements Analysis for Accessible Journeys to Unknown Buildings for People with Visual Impairments*. In: Computers Helping People with Special Needs, 17th International Conference, ICCHP 2020, Lecco, Italy
- [4] Julian Striegl, Claudia Loitsch, Jan Schmalfuss-Schwarz, Gerhard Weber. *Analysis of Indoor Maps Accounting the Needs of People with Impairments*. In: Computers Helping People with Special Needs, 17th International Conference, ICCHP 2020, Lecco, Italy
- [5] Jan Schmalfuß-Schwarz, Julian Striegl. *Analyse Tool für Simple Indoor Tagging*. In: Tagungsband, FOSSGIS 2021
- [6] Denise Prescher, Jens Bornschein (2016): *Richtlinien zur Umsetzung taktiler Grafiken*. URL: <https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa-196167>
- [7] Christin Engel (2018): *Kriterien zur Erstellung taktiler Diagramme und Diagrammbeschreibungen*. https://tu-dresden.de/ing/informatik/ai/mci/ressourcen/dateien/forschung/mosaik/Kriterien-fuer-taktile-Diagramme-und-Beschreibungen_v2_old.pdf?lang=de, Stand: 01.05.2021

Aufbau eines Geoportals mit QGIS Server und QWC2

Der QGIS Webclient 2 (QWC2) ist ein moderner Kartenclient, der auf die Publikation von Karten mit QGIS Server spezialisiert ist. Dank dem Einsatz von Micro-Services ist er sowohl für die Erstellung einfacher In-House Clients, als auch für umfangreiche Lösungen in Enterprise-Infrastrukturen geeignet.

Der QGIS Webclient 2 (QWC2) ist ein moderner Kartenclient, der auf die Publikation von Karten mit QGIS Server spezialisiert ist. Dank dem Einsatz von Micro-Services ist er sowohl für die Erstellung einfacher In-House Clients, als auch für umfangreiche Lösungen in Enterprise-Infrastrukturen geeignet.

Pirmin Kalberer

Neues von osm2pgsql

Das Programm osm2pgsql ist eines der Urgesteine der OpenStreetMap-Software-Welt. Seit 2006 wird es benutzt, um OSM-Daten in eine PostgreSQL/PostGIS-Datenbank zu importieren, um daraus dann zum Beispiel Karten zu rendern. In den letzten Jahren hat sich einiges getan bei osm2pgsql. Mit dem neuen Flex-Output ist osm2pgsql jetzt noch vielseitiger einsetzbar. Dadurch werden viele neue Anwendungen möglich.

Der Vortrag wird auf die Neuerungen bei osm2pgsql eingehen und zeigen, wie die neuen Features auf vielfältige Art genutzt werden können. Natürlich kann man sie zum Erzeugen von Bitmap- oder Vector-Tiles verwenden. Aber auch die Nutzung aus GIS-Programmen wie QGIS ist einfach möglich. Daneben können die Daten für geographische und statistische Analysen und für viele andere Dinge genutzt werden.

Jochen Topf

Die Nutzung öffentlicher Geodaten in OpenStreetMap am Beispiel Land Brandenburg

Seit dem Jahr 2020 gibt es die Möglichkeiten Daten der Landesvermessung und Geobasisinformation Brandenburg zu nutzen. Dieser kurze Talk zeigt den Weg dorthin und sollte als Vorlage für andere Behörden oder Datenanbieter genutzt werden.

Seit dem Jahr 2020 gibt es die Möglichkeiten Daten der Landesvermessung und Geobasisinformation Brandenburg zu nutzen. Dieser kurze Talk zeigt den Weg dorthin und sollte als Vorlage für andere Behörden oder Datenanbieter genutzt werden.

Es wird dargestellt, wie im der lokalen Community zusammengearbeitet wurde und wie eine Nutzung trotz inkompatibler Lizenz "Datenlizenz Deutschland 2.0 Namensnennung dl/de/by-2.0" in OpenStreetMap möglich wurde.

Christopher Lorenz

Ist die Zeit reif für permanente Identifikatoren in OpenStreetMap?

Ist die Zeit reif für permanente Identifikatoren in OpenStreetMap?

Für permanente bzw. persistente Identifikatoren in OpenStreetMap gibt es einige Vorschläge, doch gibt es weder einen akzeptierten Lösungsansatz noch verbreitete Implementationen dazu, abgesehen vom "Permanent ID"-Service von Overpass. Dieser Talk hofft, dass die Zeit reif ist und zeigt einen konkreten Lösungsansatz auf.

Die Diskussion um einen permanenten bzw. persistenten Identifikator für Objekte in OpenStreetMap (OSM) dauert bereits seit Jahren an. Es gibt eine Implementation, der "Permanent ID"-Service von Overpass, doch gibt es noch keine verbreitete Lösung für eine solche "perma_id" für OSM.

Inzwischen sind jedoch einige Vorschläge auf dem Tisch, die vielversprechend sind. Und es gibt immer mehr Projekte, die nach einer Lösung rufen - insbesondere Wikidata und Linked Data-Technologien - sowie eine wachsende Zahl von Webapplikationen mit eigenen Daten, die auf OSM-Objekte referenzieren.

Dieser Talk möchte einen konkreten Lösungsansatz aufzeigen und versucht die Realisierung eines perma_id-Services voranzutreiben in der Hoffnung, dass die Zeit langsam reif ist dafür!

Stefan Keller

Ein Lastenfahrrad-Routing und -Index: CargoRocket

Klassisches Fahrradrouting ist mittlerweile gut ausgereift. Durch das vermehrte Aufkommen von Lastenfahrrädern sind jedoch neue Herausforderungen hinzugekommen, vor allem in Städten. Es gibt viele Hindernisse, wie Poller oder Bordsteine und nicht geeignete, schmale und unebene Wege, die zu vermeiden sind. Auf Basis von OpenStreet-Map Daten haben wir ein Routing für Lastenfahrräder entwickelt und mit dem CargoBikeIndex bewerten wir die Lastenrad-Tauglichkeit einzelner Straßen.



OpenStreetMap enthält mittlerweile viele Attribute über den Zustand bzw. die Beschaffenheit von Straßen und Wegen, sowie zu Pollern und Bordsteinen. Diese Informationen haben wir verwendet um die Straßen in Baden-Württemberg auf ihre Lastenradfreundlichkeit zu bewerten. Da viele Teile der Infrastruktur noch nicht den Anforderungen von Lastenfahrrädern entsprechen, möchten wir in der bestehenden Infrastruktur Problemstellen identifizieren und durch Routing eine einfache Umfahrung ermöglichen. Hierfür haben unseren beiden freien Produkte entwickelt: mit R berechnen wir den CargoBikeIndex und auf Basis von der Routing Software Graphhopper das CargoRocket Routing. Alle Erkenntnisse und Entwicklungen sind offen und frei verfügbar. Zusätzlich haben wir uns damit beschäftigt, wie fehlende Daten gesammelt werden können und den Einbezug von bestehenden Apps wie StreetComplete erprobt.

Henri Chilla

Glühstrumpf, Galgen und Bischofsstab - Berliner Gasleuchten in OpenStreetMap

In Berlin gibt es noch ca. 25000 Gasleuchten, davon stehen 3300 unter Denkmalschutz.

Gemeinsam mit Gaslicht-Kultur e.V. kartieren wir die Berliner Gasleuchten in OpenStreetMap. Im Vortrag stellen wir Bauarten der Leuchten und Mastformen vor und erläutern dann das Tagging in OSM.

[1] Schließlich stellen wir diese Informationen in einer Karte dar.

[1] https://wiki.openstreetmap.org/wiki/Berlin/Gasleuchten_in_Berlin

In Berlin gibt es noch ca. 25000 Gasleuchten, davon stehen 3300 unter Denkmalschutz. [1]

Die ersten Gasleuchten wurden 1826 aufgestellt, die letzten um die Jahrtausendwende.

Gemeinsam mit Gaslicht-Kultur e.V., einem gemeinnützigen Verein, der sich in Berlin für den Erhalt der Gasbeleuchtung einsetzt [2], kartieren wir die Berliner Gasleuchten in OpenStreetMap. Dabei erfassen wir nicht nur die fünf Grundtypen der Berliner Gas-Straßenbeleuchtung, sondern auch weitere technisch und denkmalpflegerisch relevante Angaben wie Mastformen und die Anzahl der Leuchtkörper. [3]

Im Vortrag stellen wir Bauarten der Leuchten und Mastformen vor und erläutern dann das Tagging in OSM. Schließlich stellen wir diese Informationen in einer Karte dar. [4]

[1] <https://www.berlin.de/sen/uvk/verkehr/infrastruktur/oeffentliche-beleuchtung/gasbeleuchtung/>

[2] <http://www.gaslicht-kultur.de>

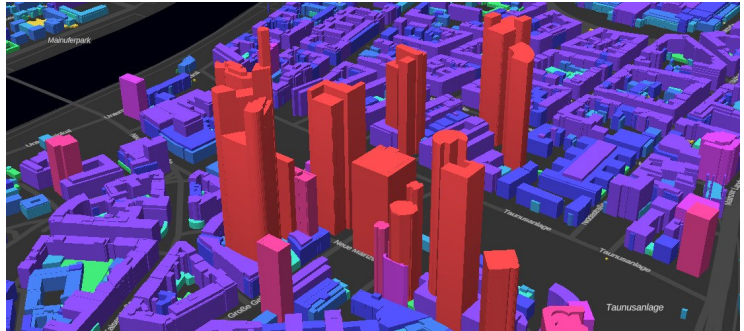
[3] https://wiki.openstreetmap.org/wiki/Berlin/Gasleuchten_in_Berlin

[4] <https://www.gasleuchtenkarte.de>

Oliver Rudzick

SmartMapping: Amtliche Daten als VectorTiles

Die AG SmartMapping der AdV entwickelt ein Verfahren zur Bereitstellung amtlicher Daten der Länder in aktuellen Formaten und Technologien. Der Vortrag stellt das Projekt vor und beschreibt in Kürze die technischen Aspekte vom Import der NAS- und NBA-Datenbestände in ein denormalisiertes (flaches) Datenmodell, Ableitung von Core-Tables und Erzeugung von VectorTiles.

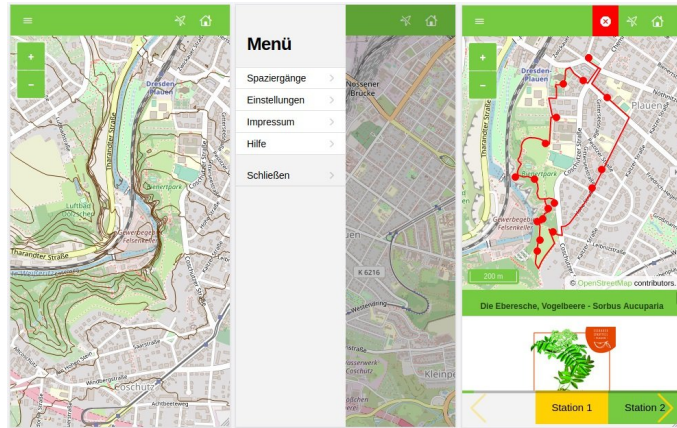


Smart Mapping ist eine gemeinsame Verfahrenslösung mit aktiven Entwicklern von Bund, Ländern und Kommunen. In dem neu entwickelten Verfahren werden amtliche Geobasisdaten aller deutschen Vermessungsverwaltungen in ein denormalisiertes, nutzerorientiertes Datenmodell zusammengeführt und mit weiteren Geodaten aus amtlicher und nicht-amtlicher Sicht angereichert. Das Verfahren nutzt im Kern Open Source Software und wird auf einer gemeinsamen Entwicklungs- und Kommunikations-Plattform entwickelt. Die Entwicklung erfolgt in mehrwöchigen Iterationen (Sprints) mit ausgewählten agilen Methoden wie sie aus Scrum und Kanban bekannt sind. Der zukünftige Betrieb ist für Cloud-Technologien vorbereitet und wird die Basis für neue AdV-Standard-Produkte sein. Die Betaversion ist über <https://adv-samrt.de> Live erreichbar und wird stetig erweitert.

Arnulf Christl

Interaktive Spaziergänge für die Erkundung von essbaren Pflanzen - Entwicklung einer OpenSource WebApp mit framework7

Die hier vorgestellte WebApp visualisiert mit einer interaktiven OpenLayers-Karte verschiedene „Spaziergänge“ in einem Dresdner Stadtteil. Die Spaziergänge sind darauf ausgerichtet, essbare Pflanzen zu erkunden. Der Vortrag präsentiert die Entwicklung der WebApp vom Auftrag über den Entwurf bis zur Implementierung. Tipps und Erfahrungen aus der praktischen Umsetzung geben Einblicke in das Projekt – und sollen auch Motivation für andere Initiativen sein, die Chancen interaktiver Karte zu nutzen.



Die Versorgung mit gesunden, regionalen und saisonalen Lebensmitteln gewinnt zunehmend an Bedeutung. Verbraucherinnen und Verbraucher gehen bewusster mit der Art und Weise ihrer Ernährung um, und hinterfragen zunehmend die auf Effizienz und niedrige Kosten getrimmten Prozesse der Lebensmittelproduktion. Im Rahmen dieser Entwicklung geht der Fokus auch auf die heimische Pflanzenvielfalt – vom Bio-Bauernhof bis hin zu essbaren Pflanzen, welche im öffentlichen Raum wachsen. Neben allgemein bekannten essbaren Sorten, wie zum Beispiel Apfelbäumen und verschiedene Kräutern, existieren viele weitere eher unbekannte essbare Pflanzen. Um ein Bewusstsein für die Möglichkeiten des Verzehrs von essbaren Pflanzen im urbanen Raum zu schaffen, entstand die Idee einer interaktiven Karte in Form einer WebApp. Das Untersuchungsgebiet für das Projekt der Initiative „essbarer Stadtteil Plauen“ (siehe auch <https://essbarer-stadtteil.de/>) bezieht sich hauptsächlich auf den Stadtteil Dresden-Plauen im Süden Dresdens.

Den Nutzerinnen und Nutzern soll mit dieser WebApp die Möglichkeit gegeben werden, auf Basis einer Kartenanwendung auf unterschiedliche, auf verschiedene Jahreszeiten und Zielgruppen hin optimierte Spaziergänge zuzugreifen. Von Station zu Station führt eine Route durch das Untersuchungsgebiet und erleichtert das Auffinden der essbaren Pflanzen. Zusätzlich werden Informationen zur Art und zum Verzehr geliefert, sowie eine Abbildung der Pflanze. Um einen Spaziergang abzurunden, können auch Aussichtspunkte oder weitere kulturhistorische Sehenswürdigkeiten als Station integriert werden. Mit dem Erstellen weiterer Spaziergänge durch das Projektteam kann die Wegeführung für verschiedene Zielgruppen angepasst werden. Der saisonale Aspekt wird durch Spaziergänge für bestimmte Jahreszeiten abgedeckt.

Nach einer Analyse der Anforderungen und Randbedingungen wurde ein Entwurf für die WebApp realisiert. So konnte ein erster Eindruck der zukünftigen App vermittelt und mit dem Projektteam diskutiert werden. Für die anschließende Implementierung wurden OpenSource-Technologien gewählt. Als Entwicklungsumgebung kam ein Ubuntu 18.04 LTS mit installierter Docker Engine zum Einsatz. Ein Docker Container wurde zur Implementierung mit dem Node Package Manager (NPM) genutzt, ein weiterer stellt NGINX als lokalen Webserver bereit. Die Wahl für das Front-End fiel auf framework7. Dieses Framework ist gut dokumentiert, bringt eine Vielzahl an vorgefertigten Komponenten mit und wird hauptsächlich in den bekannten Sprachen HTML, CSS und JavaScript geschrieben. Ergänzend wurden openlayers für die interaktive Karte und swiper.js für ein besseres Touch&Feel eines Spazierganges genutzt. Mit webpack konnte der Programmcode für die Auslieferung gebündelt werden. Inhalte

Interaktive Spaziergänge für die Erkundung von essbaren Pflanzen - Entwicklung einer OpenSource WebApp mit framework7

werden aus JSON-Dateien importiert, dazu werden abgelegte Fotos geladen. Eine Umsetzung mit einem Back-End, einer PostGIS-Datenbank und eventuell einer REST-Schnittstelle wurde zunächst angedacht, aber aus Zeit-Nutzen-Erwägungen zurückgestellt. Das Projekt geht im Frühjahr 2021 in die erste Testphase.

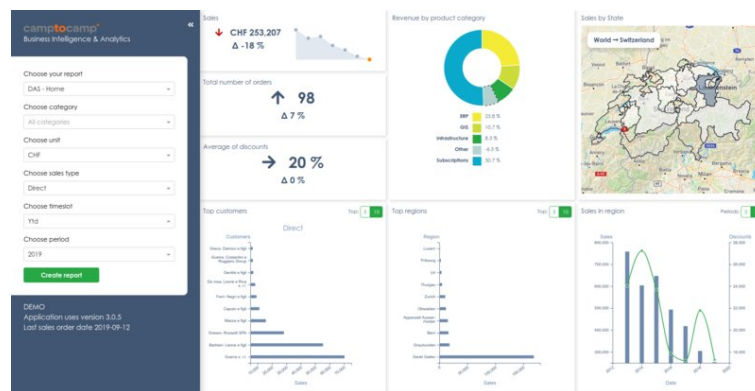
Im Vortrag werden die Erfahrungen beim Projektmanagement und bei der Implementierung der WebApp erläutert und veranschaulicht. Auch werden konkrete Fragen der Umsetzung thematisiert: Wie können sogenannte Router Components den Code übersichtlich halten und Page-Events, benötigte Daten sowie zusätzliche Methoden sinnvoll strukturiert werden? Wie kann das dynamische Nachladen von Inhalten realisiert werden? Welche Herausforderungen wurden gelöst und können den Einstieg für andere erleichtern? Gegebenenfalls können auch noch Erfahrungen aus der Testphase präsentiert werden. Zudem soll die Übertragbarkeit solch eines Projektes auf andere Fragestellungen und Use Cases diskutiert werden.

Theodor Rieche

Kartographierung von Businessdaten

Wie kann man Businessdaten Daten kartographieren und daraus Informationen über ein Unternehmen gewinnen? Wie stellt man solche Informationen verständlich und einfach dar? Wie kann man trotz komplexen Analysefunktionen eine gute Usability erreichen?

Diesen Fragen sind wir im Rahmen einer Kundenanforderung nachgegangen. Das Resultat ist ein generisches und flexibles geographisches BI-Tool, welches sich auf Open-Source Technologien stützt und erweitert.



Anhand einer Live-Anwendung wird ein Beispiel gezeigt, welches Business-Daten in einen geographischen Kontext stellt.

In der Anwendung werden die Business-Daten an geografische Punkte wie z.B Verkaufsstellen geknüpft. Damit können die vorhandenen Daten in geographische Regionen aggregiert und kartographisch dargestellt werden. Um die Nutzerfreundlichkeit zu unterstützen, werden nebst Karten auch Tabellen und Diagramme benutzt, um auf Einzelheiten der Aggregationen einzugehen.

Für die kartographischen Elemente wird OpenLayers verwendet. Für die Umsetzung der unterschiedlichen Diagramme wird d3 über die d3-helper Bibliothek verwendet. Die d3-helper Bibliothek ist eine Open-Source Bibliothek, welche aus der hier gezeigten Anwendung entstanden ist, da viele Diagramme in dem Tool die selbe Basis benutzen.

Die Präsentation zeigt anhand eines Beispiels wie kartografische und nicht-kartografische Data-Exploration kombiniert werden kann und die verwendeten Open Source Komponenten zu einem kompletten Business Intelligence Tool zusammenwachsen.

Marion Baumgartner

Gerber Benjamin

FORCE4Q Plugin - Ableitung wolkenfreier Bildkomposite aus frei verfügbaren Landsat und Sentinel-2 Satellitenbildern

Erdbeobachtungsdaten stellen eine immer wichtigere Grundlage für raumbezogene Analysen in Wirtschaft und Wissenschaft dar. Zugleich steigt die Verfügbarkeit von multispektralen Fernerkundungsdaten durch offene Datenzugangsrichtlinien der Erdbeobachtungsprogramme wie Copernicus (ESA) und Landsat (NASA/USGS) weltweit, flächendeckend und mit hoher zeitlicher Dichte.

Das vorgestellte FORCE4Q Plugin für QGIS erlaubt die qualitative Aufbereitung dieser Daten für weiterführende Analysen.

Erdbeobachtungsdaten stellen eine immer wichtigere Grundlage für raumbezogene Analysen in Wirtschaft und Wissenschaft dar. Gleichzeitig steigt ihre Verfügbarkeit infolge neuer Fernerkundungssatelliten und offener Datenzugangsrichtlinien stetig an. Erdbeobachtungsprogramme wie Copernicus (ESA) und Landsat (NASA/USGS) erfassen multispektrale Fernerkundungsdaten weltweit, flächendeckend und mit hoher zeitlicher Dichte.

Diese Entwicklung hat die Nachfrage nach freier und quelloffener Software zur Analyse großer und mehrdimensionaler Rasterdatenmengen gesteigert. Da die Anforderungen einer professionellen Fernerkundungsanalyse oft über die Möglichkeiten klassischer GIS-Software hinausgehen, entwickelt die Humboldt-Universität zu Berlin das FORCE4Q Plugin für QGIS.

FORCE4Q hilft Landsat und Sentinel-2 Daten in einen analysebereiten Datenwürfel (analysis-ready datacube) zu überführen. Dies beinhaltet das Zuschneiden der Daten in ein räumliches (global definiertes) Tiling-Scheme und ein qualitatives Screening zum Ausschluss unerwünschter Bildbereiche, wie Wolken, Wolkenschatten oder Schnee., Aerosol, Wasserdampf usw.. Die gescreenten Daten können direkt für weiterführende Analysen verwendet oder aber zeitlich weiter verdichtet werden. Das flexible zeitliche Binning erlaubt eine jährliche, quartalsweise, monatliche, wöchentliche oder tagesgenaue Aggregation der Daten. Neben den oft verwendeten wolkenfreien Mittelwert-Kompositen sind weitere statistische Funktionen wie Median, Minimum, Maximum, Standardabweichung oder Perzentile verfügbar. Auch können zusätzlich zu den originalen multispektralen Bändern (Blau, Grün, Rot, nahes Infrarot, ...) weitere Indizes (NDVI, EVI, Tasseled Cap usw.) direkt mit abgeleitet werden.

Die gescreenten und zeitlich verdichteten Daten sind nun direkt für weiterführende Zeitreihen-, Klassifikations- und Regressionsanalysen verwendbar.

Der Vortrag präsentiert den kompletten Ablauf, angefangen beim Download von Satellitendaten über den Copernicus Open Access Hub und das EarthExplorer Portal, über den Import ins TilingScheme, bis zur Ableitung analysebereiter Daten.

Projekt-Dokumentation: <https://force4q.readthedocs.io>

Andreas Rabe

Von Pixeln und Profilen: Die Nutzung von Spektralbibliotheken in QGIS mit der EnMAP-Box

Spektralbibliotheken beschreiben die „spektralen Fingerabdrücke“ von Oberflächen. Dieser Vortrag zeigt, wie sie sich mit der EnMAP-Box auch in QGIS visualisieren und erstellen lassen, und wie sie dort für die weitere Auswertung von Satellitenbildern genutzt werden können.

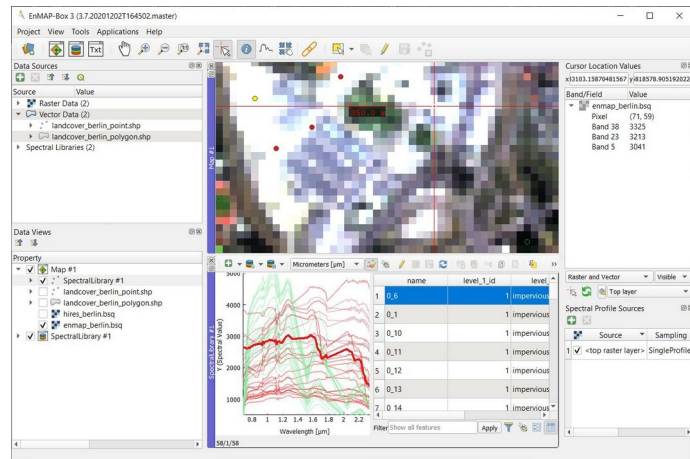
In der multi- und hyperspektralen Fernerkundung werden häufig Spektralbibliotheken verwendet, die Oberflächen durch „spektrale Fingerabdrücke“ charakterisieren. Solche Spektralprofile sind oft durch eine Vielzahl weiterer Informationen beschrieben, etwa dem Ort und die Uhrzeit ihrer Erfassung, Angaben über die Pflanzen- oder Mineralzusammensetzung der spektral vermessenen Oberfläche, sowie Informationen über den verwendeten Sensor. Spektralbibliotheken stammen oft aus Feld- und Labormessungen, aber auch Simulationen, Flug- und Satellitenbildern. Sie stellen eine wichtige Referenz für weitere Analysen dar, etwa der Klassifizierung von Satellitenbildern, der Entmischung von Landbedeckungsanteilen, oder der Kalibrierung biophysikalischer Modelle.

Mit der EnMAP-Box, einem QGIS-Plugin zur Visualisierung und Analyse der Bilder multi- und hyperspektraler Satelliten, wie etwa Landsat, Sentinel-2, PRISMA oder dem zukünftigen EnMAP-Satelliten, ist es nun möglich, Spektralbibliotheken auch in QGIS zu nutzen.

Nach einer kurzen Einführung in das Thema zeigt dieser Vortrag, wie Spektralbibliotheken aus unterschiedlichen Quellen in die EnMAP-Box importiert, dort um eigene Daten ergänzt und spektral und räumlich visualisiert werden können. Eine Besonderheit ist das flexible Datenmodell, welches es erlaubt, Spektralprofile unterschiedlicher Sensoren gemeinsam abzuspeichern. Abschließend wird gezeigt, wie mit einer Spektralbibliothek eine Sentinel-2 Szene klassifiziert werden kann und welche zukünftigen Weiterentwicklungen für die Spektraldatenverarbeitung in der EnMAP-Box angestrebt sind.

Der Vortrag richtet sich an ein breites Publikum und erfordert keine besonderen Kenntnisse der multi- und hyperspektralen Fernerkundung.

Benjamin Jakimow



Fortgeschrittene GIS-Analysen und Kartographie mit Geometrien-Ausdrücken in QGIS

JOHANNES KRÖGER

Das Werkzeug „Geometrie per Ausdruck“ (engl.: "Geometry from expression") ermöglicht es in QGIS auch ohne *allzu* tiefgehende Programmierkenntnisse komplexe Berechnungen umzusetzen. Es muss also nicht immer gleich eine Erweiterung, ein Umstieg auf die grafische Modellierung oder mühsame Handarbeit sein, wenn die mitgelieferten Werkzeuge nicht ausreichen.

Das Vorgehen bei der Verwendung von Ausdrücken ähnelt dabei einer sequenziellen, streng schrittweisen Ausrechnung: Berechne das, nimm das Ergebnis davon und tue damit dies, mit dem folgenden Ergebnis wiederum erzeuge jenes und so weiter. Der Ausdruck wird dabei von QGIS jeweils für die einzelnen Objekte des Layers individuell ausgewertet.

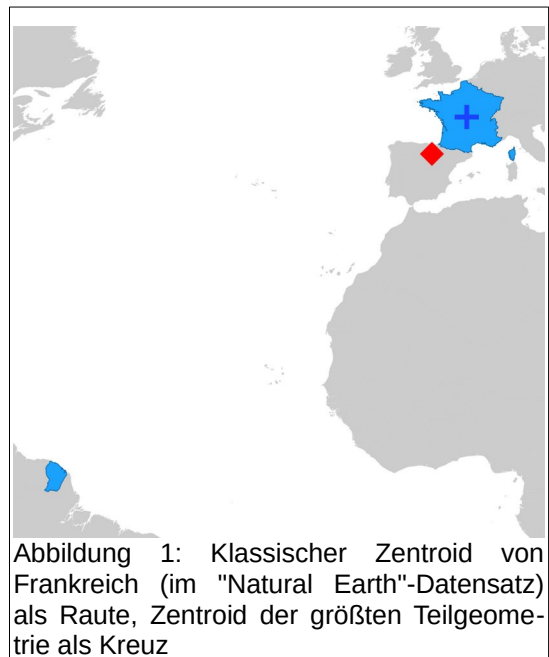
Die beiden folgenden Beispiele sollen als Ansatzpunkte dienen und hoffentlich Lust zum Ausprobieren machen.

Zentroid nur für den größten Teil einer mehrteiligen Geometrie berechnen

Liegen Geometrien als Multipolygone vor, also zum Beispiel die Gemeindefläche von Hamburg inklusive der schönen Nordseeinsel Neuwerk, so kann QGIS nur entweder den „globalen“ Zentroid der gesamten Geometrie oder die Zentroide einer jeden Teilgeometrie berechnen. Der globale Zentroid wird dabei gegebenenfalls durch, möglicherweise für den Anwendungsfall irrelevante, Teile der Geometrie beeinflusst. Am Beispiel von Frankreich im „Natural Earth“-Datensatz ist dies deutlich zu sehen (siehe Abbildung 1). Der Zentroid liegt hier aufgrund von Französisch-Guayana im Norden von Spanien.

Alternativ könnte man nur am Zentroid des flächenmäßig größten Teils der Geometrie interessiert sein. Und dafür kann ein mehrstufiger Ausdruck verwendet werden:

- Mit der `order_parts`-Funktion kann man die Teile einer Multigeometrie nach einem Kriterium sortieren, hier nach ihrer jeweiligen Größe (`area`-Funktion auf der jeweiligen `$geometry`) und in absteigender Reihenfolge, so dass die Ergebnisgeometrie der Funktion als erstes die größte Teilgeometrie enthält.
- Die `geometry_n`-Funktion gibt die n-te Teilgeometrie einer Multigeometrie aus.
- Den Zentroid einer Geometrie erhält man mit der `centroid`-Funktion.
- Die Variable `$geometry` „beinhaltet“ die Geometrie des jeweiligen Objekts.



Indem das Ergebnis der `order_parts`-Funktion in die `geometry_n`-Funktion gesteckt wird und dieser zusätzlich eine 1 („bitte die erste Teilgeometrie ausgeben“) übergeben wird, berechnet die `centroid`-Funktion am Ende den Zentroid der größten Teilgeometrie. Der fertige Ausdruck ist:

```
centroid(
```



```

geometry_n(
  order_parts(
    $geometry,
    orderby:='area($geometry) ',
    ascending:=False
  ),
  1
)

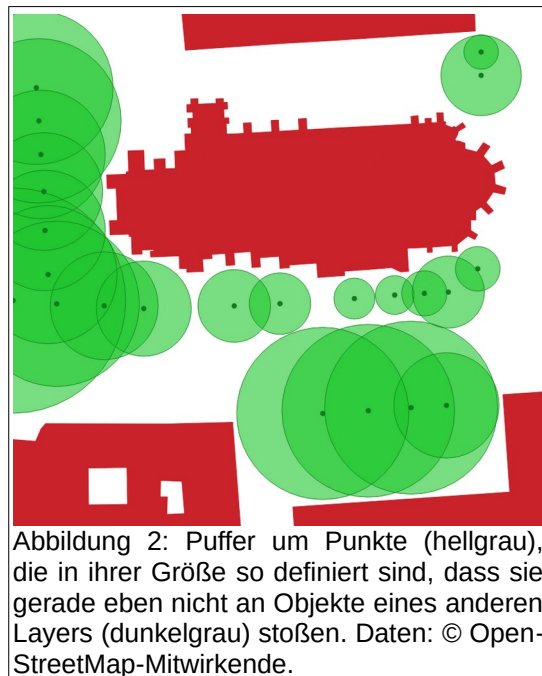
```

Zugegeben, als Neuling weiß man noch nicht, was für Funktionen es überhaupt gibt und wie diese zu verwenden sind, doch ist die Vorgehensweise der Funktionsverschachtelung von innen nach außen erst einmal verinnerlicht und das zu lösende Problem dementsprechend abstrakt durchdacht, so verlieren solche Ausdrücke schnell ihre Magie und werden zu Alltagswerkzeugen.

Puffergröße dynamisch über den Abstand zum nächsten Objekt anpassen

Das zweite Beispiel ist im Ergebnis beeindruckender, der Ausdruck dafür aber auch schon viel komplexer. Hier werden Puffer um Punkte jeweils so groß gemacht, wie sie maximal sein dürfen, ohne eine Geometrie auf einem anderen Layer zu berühren. Als Beispiel dienen Baumkronen neben Gebäuden.

- Um die Bäume sollen Puffer erzeugt werden, hierfür gibt es die `buffer`-Funktion. Diese nimmt mehrere Parameter an, erstens die zu puffernde Geometrie, zweitens den gewünschten Radius sowie optional als dritten Parameter die Anzahl der Segmente je Viertelkreis (damit kann man die „Eckigkeit“ des Puffers beeinflussen). Raus kommt eine entsprechende Fläche.
- Der Radius soll dabei abhängig vom Abstand zum jeweils nächstgelegenen Gebäude sein. Dazu müssen wir für den jeweiligen Baum die minimalen Abstände zu allen Gebäuden berechnen und davon den kleinsten Wert verwenden.



- Die minimale Distanz zwischen zwei Geometrien kann die `distance`-Funktion berechnen. Die Geometrien müssen dabei im selben, möglichst längentreuen Koordinatenreferenzsystem vorliegen.
- Eine der Möglichkeiten von einer Menge vieler Werte einen bestimmten auszuwählen bzw. zu berechnen ist eine Aggregation. QGIS bietet hierzu die `aggregate`-Funktion. Diese kann mit einem eigenen Ausdruck die Objekte eines anderen Layers verarbeiten und einen gewünschten Aggregatwert (z. B. Minimum) berechnen. Im Ausdruck der Aggregatfunktion zeigt die Variable `@parent` auf das jeweilige Objekt des anderen Layers; mit der `geometry`-Funktion kann dessen Geometrie abgefragt werden.
- Als Aggregat berechnen wir den minimalen Wert der Distanzen der Geometrie zu den Geometrien auf dem Gebäude-Layer. Dieser Wert wird noch um einen Meter verkleinert, damit die Baumkronen nicht an die Wände stoßen, und als Radius für den Puffer übergeben.

Fortgeschrittene GIS-Analysen und Kartographie mit Geometrien-Ausdrücken in QGIS

Als Ausdruck entsteht bei diesem Ansatz:

```
buffer (
    $geometry,
    distance:=aggregate (
        layer:='Gebäude',
        aggregate:='min',
        expression:=distance($geometry, geometry(@parent))
    ) - 1,          -- Radius auf 1 Meter *vor* Gebäude beschränken
    segments:=12     -- Rundheit des Puffers
)
```

Für die bessere Lesbarkeit sind Funktionsparameter in diesem Ausdruck mit ihren Namen beschrieben (parameter:=wert), zusätzlich wurden mit "--" Zeilenkommentare eingefügt.

Wie gesagt, dieses Beispiel ist schon eher etwas für Fortgeschrittene. Gerade für die ersten Schritte mit Aggregatfunktionen ist eine Websuche nach weiteren Beispielen oder das Nachfragen auf <https://gis.stackexchange.com> äußerst empfehlenswert. Viel Freude und Erfolg damit!

Kontakt zum Autor:

Johannes Kröger
WhereGroup GmbH
Hamburg
+ 49 (0) 228 / 90 90 38 - 0
johannes.kroeger@wherogroup.de

Webmaps effizient mit Wegue erstellen

Wegue (WebGIS mit OpenLayers und Vue.js) ist eine Open Source Webmapping-Lösung, die auf OpenLayers und Vue.js basiert. Wir zeigen wie man damit schnell und unkompliziert eigene WebGIS-Anwendungen erstellen kann. Es stehen eine Reihe von vorgefertigten Modulen wie z.B. GeoCoding oder Layer-Liste zur Verfügung, wodurch die eigene Anwendung individuell gestaltet werden können. Dies kann sowohl rein konfigurativ als auch programmatisch erfolgen.

Das Open Source Projekt Wegue (WebGIS mit OpenLayers und Vue.js) wurde seit 2017 schon in einer Vielzahl von Webmapping-Anwendungen als Basiskomponente eingesetzt. Es zeichnet sich durch einfache Konfiguration aus, kann aber auch nach eigenen Vorstellungen beliebig angepasst werden. In dieser Session zeigen wir den kompletten Arbeitsablauf von der Installation, dem Einbinden der Layer, dem Aktivieren von Zusatz-Modulen und wie man die fertige Anwendungen exportiert und publiziert.

Anhand von bestehenden Anwendungen demonstrieren wir wie verschiedene Wegue Module wie GeoCoding, Layer-Liste oder Messwerkzeug eingebunden werden können. Darüber hinaus stellen wir das Plugin QGIS2Wegue vor, mit dem Layer aus QGIS komfortabel in eine Wegue-Anwendung überführt werden können. Seit letztem Jahr hat Wegue eine eigene Website, auf der die Dokumentation übersichtlich dargestellt ist.

Neben der Vielzahl an vorgefertigten Einstellungsmöglichkeiten kann Wegue auch durch eigene Entwicklungen erweitert werden. Wegue baut auf der neuesten Version von OpenLayers auf und basiert auf dem Webframework Vue.js und der UI-Bibliothek Vuetify. Dadurch stehen alle Funktionen dieser etablierten Open Source Komponenten bereit, um Wegue Anwendungen bei Bedarf nach eigenen Wünschen maßzuschneidern zu können.

Jakob Miksch

In 60 Minuten zur eigenen Geo-App mit YAGA leaflet-ng2

ARNE SCHUBERT UND STEPHAN HERRITSCH

YAGA[1] (Yet Another Geo Application) bietet diverse Docker-Images zum Aufbau einer GDI im Backend und Frontend Bibliotheken zur Erstellung von Web-Karten-Anwendungen an. YAGA wird von einem Entwickler-Team, bestehend aus den drei Freunden Arne Schubert, Markus Strauß und Stephan Herritsch als Hobby in der Freizeit entwickelt und gepflegt. Erste Projekte wurden schon zu Schulzeiten angegangen. Ein Teil dieser Entwicklungen sind als OpenSource-Projekte veröffentlicht und für jedermann nutzbar. Ebenso ist jeder dazu eingeladen sich mit seinem Know-how an den OpenSource-Projekten zu beteiligen.

Eines dieser OpenSource-Projekte ist "leaflet-ng2"[2]. Mit leaflet-ng2 wurde eine Möglichkeit geschaffen das weit verbreitete Leaflet-Framework[3] in Angular.io[4] zu integrieren. Angular.io ist ein von Google entwickeltes Framework zur Erstellung von Single-Page-Webanwendungen (WebApps). Es wurde stark darauf geachtet, dass die typische Leaflet Namensgebung beibehalten bleibt, was wiederum dafür sorgt, dass Außenstehende, die bereits Erfahrungen mit Leaflet haben, sich schnell in YAGAs leaflet-ng2 einarbeiten können. Mit Hilfe von Ionic[5] können WebApps als nativ-hybride Apps im typischen "Look & Feel" der Endgeräte erstellt werden.

leaflet-ng2 ist dank TypeScript komplett typisiert, wodurch es nach die Datentypen statisch getestet ist und für IDEs (Integrated Develop Enviroment) Autovervollständigung und Syntaxprüfung mitliefert. Auf Grund von strengen Linting-Regeln ist der Quellcode leicht lesbar und einheitlich. Daher ist es auch für andere Entwickler leicht am Projekt mitzuwirken.

Es wurde bewusst auf Softwaretests, in Form von Unit-Tests, geachtet, um eine geringe Fehleranfälligkeit zu gewährleisten. Aber auch um eine schnelle Möglichkeit zu bieten, Fehler in Weiterentwicklungen zu lokalisieren. Zusätzlich wird auch die Abdeckung der Tests geprüft, die bei über 90% liegt.

Im Gegensatz zu anderen entwickelten Angular-Leaflet-Frameworks, bei denen eine Directive für Leaflet zum Einsatz kommt, wurde bei leaflet-ng2 granular für die einzelnen Leaflet Klassen Angular-Directives erstellt.

Alle Optionen der Leaflet-Klassen wurden mit Input-Decorators von Angular zur Verfügung gestellt, so dass ein nachträgliches Abändern zwar nicht im Funktionsumfang von Leaflet liegt, es jedoch in leaflet-ng2 möglich ist. Dort wo Leaflet eine Möglichkeit bietet Werte zu ändern, werden über zusätzliche Output-Decorators die veränderten Werte automatisch im eigenen Datenmodell synchron gehalten. Dieses so genannte Two-Way-Databinding hat den Vorteil, dass Leaflet in ein MVVC (Model View View-Controller) vollkommen eingebettet ist und damit Datenänderungen automatisch in der gesamten Anwendung übernommen werden und nicht mehrfach verwaltet werden müssen. Zusätzlich werden sämtliche Leaflet-Events über Output-Decorators an die Directives weitergeleitet.

Dieser Ansatz der Umsetzung bietet die Möglichkeit direkt im Template eine App weitgehend deskriptiv mit der bekannten Namensgebung von Leaflet zu erstellen, so dass für einfache Anwendungen hauptsächlich HTML-Kenntnisse von Nöten sind.

Wie mit YAGA leaflet-ng2 über ein Template eine Karte erstellt werden kann, wird im Folgenden exemplarisch dargestellt:

```
<yaga-map [lat]="48.5768558" [lng]="13.268283" [zoom]="10">
  <yaga-tile-layer [url]="http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png">
  </yaga-tile-layer>
```

```
<yaga-marker [lat]="50.73382" [lng]="7.102275">
  <yaga-popup>
    <h1>Rheinische Friedrich-Wilhelms-Universität Bonn</h1>
  </yaga-popup>
</yaga-marker>
</yaga-map>
```

In diesem Beispiel wurde im Template eine YAGA-Map-Component erstellt, in der die Universität Bonn als Position vorgegeben wurde, eine YAGA-TtileLayer-Directive mit einer Openstreetmap-Karte hinzugefügt, sowie eine YAGA-Marker-Directive mit einem YAGA-Popup. Nach diesem Prinzip lassen sich sämtliche leaflet-ng2 Directives zum YAGA-Map-Component hinzufügen. Abbildung 1 veranschaulicht die Struktur der einzelnen Directives, eine Auflistung mit ausführlichen Beschreibungen ist zudem in der Dokumentation zu finden.

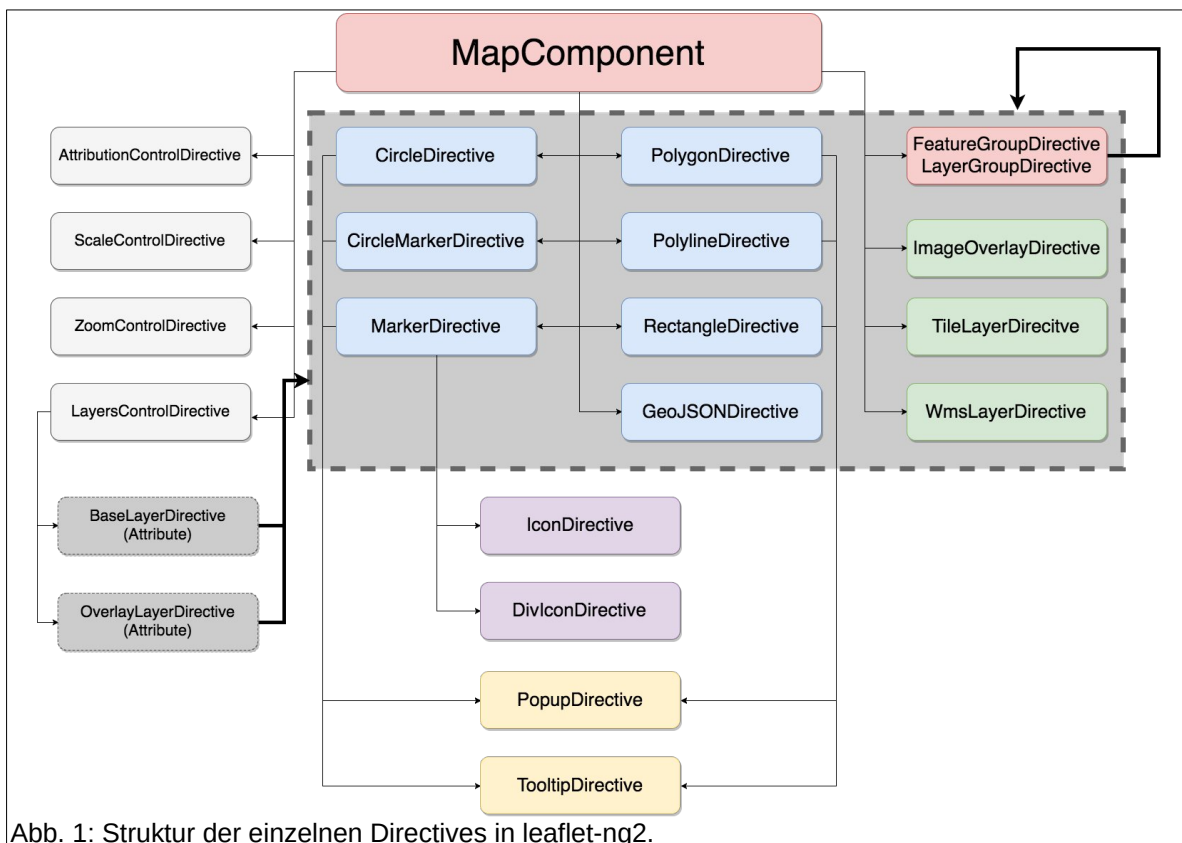


Abb. 1: Struktur der einzelnen Directives in leaflet-ng2.

YAGA leaflet-ng2 beschränkt sich ausschließlich auf den Funktionsumfang von Leaflet. Durch die modulare Struktur können problemlos mit weiteren Modulen eine Erweiterung des Funktionsumfangs realisiert werden. Ein Beispiel für eine Erweiterung ist leaflet-ng2-gps, bei dem auf die Gerätesensoren zurückgegriffen wird, um die aktuelle Position auf der Karte zu ermitteln.

Im Sommer 2019 wurde mit Geographie-Studierenden[6], die bisher keinerlei Erfahrungen in der Softwareentwicklung hatten (No-Code-Entwicklung), ein Experiment gewagt, ob diese innerhalb einer Woche eine eigene kleine Geo-App mit leaflet-ng2 erstellt bekommen. Bereits nach 3 Tagen war das Grundgerüst der Anwendung fertig, in den folgenden 2 Tagen wurde die Anwendung lediglich verfei-

In 60 Minuten zur eigenen Geo-App mit YAGA leaflet-ng2

nert. Herausgekommen ist ein Kneipenführer durch Bonn für Erstsemestler. Dieses Experiment zeigte, dass selbst Laien in kürzester Zeit in der Lage sind eine eigene kleine Geo-App ohne Vorkenntnisse zu entwickeln.

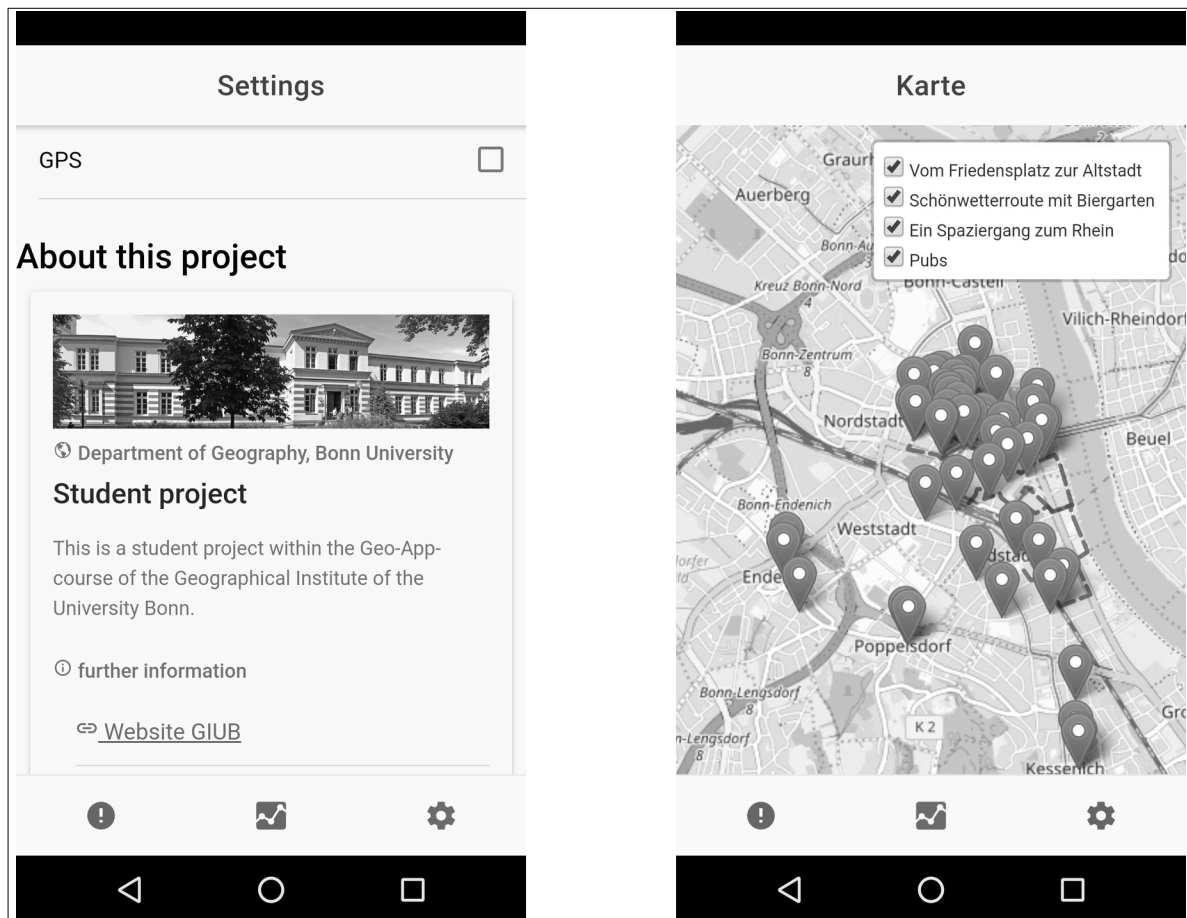


Abb. 2: Ergebnis des Studierenden-Experiments: Die Guide-App, Ein Kneipenführer durch Bonn erstellt von Studierenden der Geographie für Erstsemestler mit leaflet-ng2.

Anders als auf der FOSSGIS Konferenz 2019 in Dresden, bieten wir statt eines Workshops, auf Grund der derzeitigen Pandemiesituation eine Demo-Session an, in der wir der Community präsentieren wie man mit leaflet-ng2 in nur 60 Minuten eine eigene Geo-App erstellen kann. Die Vorführung wird im Anschluss auf GitHub[7] veröffentlicht, so dass die Teilnehmenden der diesjährigen Konferenz eine Blaupause für mögliche eigene Geo-Apps zur Verfügung gestellt bekommen.

Kontakt zu den Autoren:

Arne Schubert
a.schubert@yagajs.org
Stephan Herritsch
s.herritsch@yagajs.org

Literatur/Links:

[1] <https://yagajs.org/>

In 60 Minuten zur eigenen Geo-App mit YAGA leaflet-ng2

[2] <https://leafletjs.com/>

[3] <https://leafletjs.com/>

[4] <https://angular.io/>

[5] <https://ionicframework.com/>

[6] <https://basis.uni-bonn.de/qisserver/rds?state=verpublish&status=init&vmfile=no&publishid=168685&moduleCall=webInfo&publishConfFile=webInfo&publishSubDir=veranstaltung>

[7] <https://github.com/yagajs>

Geodaten im Einsatzleitsystem



Im Einsatzleitsystem von Schutz & Rettung Zürich werden verschiedene Daten verwendet, darunter vor allem diejenigen von OpenStreetMap (OSM). Von OSM werden hauptsächlich Geonamen genutzt sowie Routen, um den optimalen Weg der Rettungswagen zu berechnen. Dabei gibt es auch Probleme mit OSM-Daten, die angesprochen werden und die zum Teil mit Monitoring gelöst werden.

Das Einsatzleitsystem von Schutz & Rettung Zürich dient einerseits der schnellen Suche nach Geonamen und andererseits, um die nächstgelegenen Einsatzteams optimal ans Ziel zu führen. Am Zielobjekt ist zudem interessant, wo der optimale Zugang zum Eingang liegt. Die Daten von OpenStreetMap (OSM) leisten dabei einen unentbehrlichen Beitrag, nebst den Daten vom Eidg. Gebäude- und Wohnungsregister (GWR) und von swisstopo. Die OSM-Daten werden dabei periodisch heruntergeladen, aufbereitet und über einen Software-Dienstleister zur Verfügung gestellt. Um unerwünschte Edits in OSM frühzeitig zu erkennen und zu korrigieren werden einerseits bestimmte Tags verwendet und andererseits ist ein Monitoring Tool auf der Basis einer individuell erweiterten OSMCha-Webapplikation geplant. Ein kleines Team kümmert sich um das Monitoring und OSM-Editing mittels JOSM. Inzwischen wurde auch eine Kerngruppe Notfallorganisationen im Rahmen der Swiss OSM Association gegründet.

Michael Schmid

Wie Behörden und bundesnahe Betriebe OpenStreetMap nutzen können

OpenStreetMap (OSM) stösst bei immer mehr Behörden und bundesnahen Betrieben auf Interesse. In diesem Beitrag wird anhand ausgewählter konkreter Beispiele aus der Schweiz aufgezeigt, wie OSM genutzt wird. Zudem wird ein Überblick gegeben über vorhandene Software-Werkzeuge für das Monitoring, für die Qualitätssicherung, sowie für die Übernahme von OSM-Daten.

OpenStreetMap (OSM) stösst bei immer mehr Behörden und bundesnahen Betrieben auf Interesse. Die Nutzung geht von Geonamen, Gebäuden, POIs, Routingfähige Strassen bis zu Ersatzdaten für Basiskarten. Offensichtlich sind die OSM-Daten "fit-for-use". In diesem Beitrag wird anhand ausgewählter konkreter Beispiele aus der Schweiz aufgezeigt, wie OSM genutzt wird. Zudem wird ein Überblick gegeben über vorhandene Software-Werkzeuge für das Monitoring, für die Qualitätssicherung, sowie für die Übernahme von OSM-Daten. Die Vision ist "Public-OSM Partnership" (POP) - analog Public-Private Partnership - bei der Vertreter des Gemeinwesens partnerschaftlich zusammenarbeiten mit Bürgern und Vereinsvertretern des OSM zur Erfüllung einer öffentlichen Aufgabe, indem sie ihre Ressourcen und Tätigkeiten koordinieren und optimieren.

Stefan Keller

Open Data, Open Source – Open End?

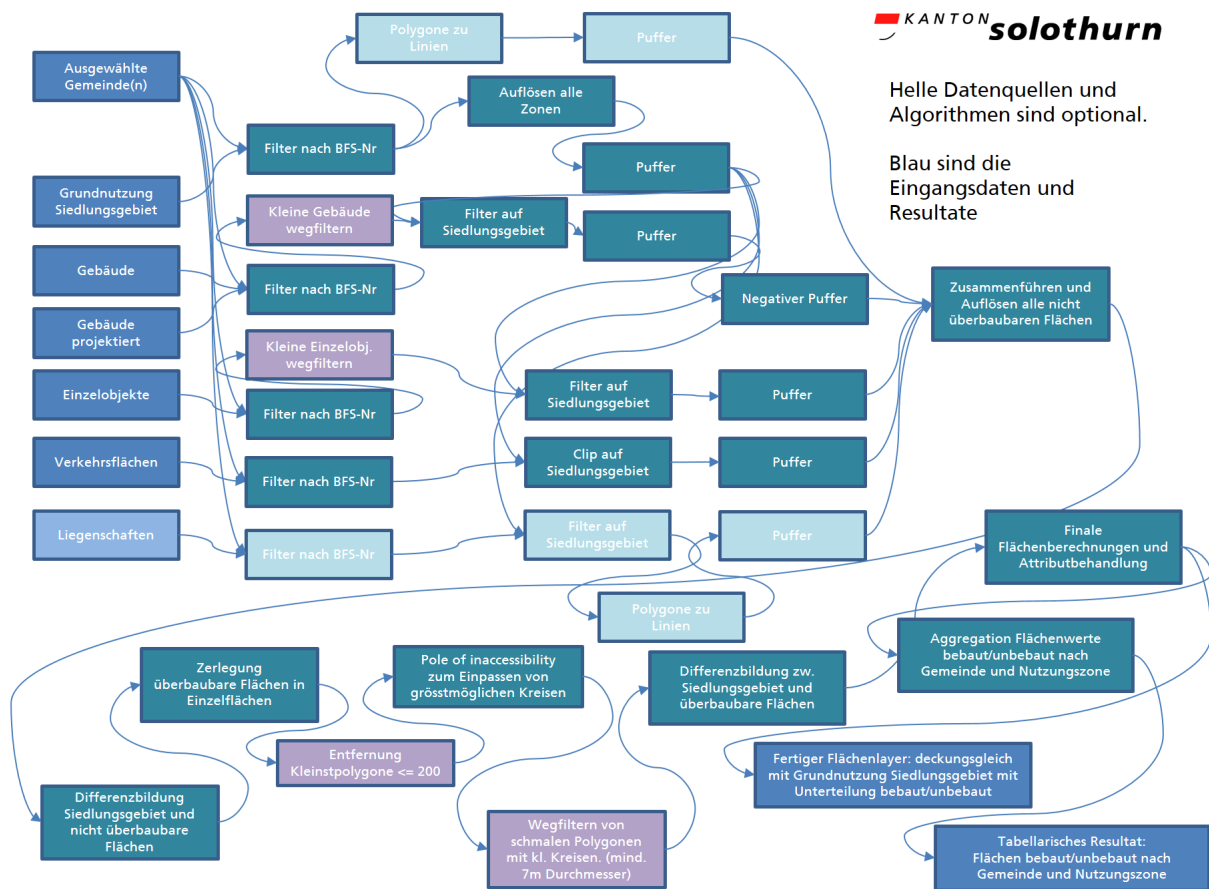
OSM-Daten und amtliche Geodaten ergänzen sich sehr gut. Das BKG berät öffentliche Einrichtungen zur Einsetzbarkeit nutzergenerierter Daten und bemüht sich, um Erhöhung ihres Stellenwertes. Geplant ist der Aufbau eines Qualitätsprüfverfahrens, das von Datenquellen unabhängig einsetzbar ist. Die Entwicklung neuer Technologien ist geplant, über die Potentiale und Risiken der Entwicklung und Benutzung der Open Source Software bei einer Behörde wird berichtet.

OSM-Daten und amtliche Geodaten verkörpern unterschiedliche Philosophien bezüglich der Datenerfassung. Sie ergänzen sich aber sehr gut, weshalb ein eindeutiger Trend zu erkennen ist: OSM-Daten werden bei immer mehr Bundesbehörden eingesetzt. Das BKG, der zentrale Geodatenbroker des Bundes, berät die Bundesbehörden und bemüht sich, den Stellenwert der OSM-Daten in der öffentlichen Verwaltung noch weiter zu erhöhen. Um dies zu erreichen, nimmt sich das BKG vor, neue Qualitätsprüfverfahren zu entwickeln und einzuführen, die über die für amtlichen Daten angewendeten Verfahren hinaus gehen. Im Vortrag wird dargestellt, was geplant ist und welche Perspektiven sich hiermit eröffnen.

Es wird auch über die Potentiale und Risiken der Entwicklung und Benutzung der Open Source Software bei einer Behörde berichtet.

Prof. Paul Becker

Statistik der Bauzonenreserven mit QGIS und PostGIS



Der Kanton Solothurn muss in regelmässigen Zeitabständen die Bauzonenreserven, aggregiert nach Gemeinden, kantonalen Nutzungszonen und dem Bebauungsstand (bebaut/unbebaut) ausweisen. Basierend auf Daten der Nutzungszonen und der Amtlichen Vermessung (Bodenbedeckung, Liegenschaften, Einzelobjekte) werden potentiell bebaubare Flächen gefunden, die gross genug für eine Bebauung sind, in einer Bauzone liegen und nicht zu schmal sind. Der Analyseablauf wurde mit QGIS und Postgis realisiert.

Im Analyseablauf werden Techniken eingesetzt, um von potentiell bebaubaren Flächen sehr schmale Bereiche ("Verbindungsstege" zwischen grösseren Flächen) oder spitze Ecken zu entfernen, die nicht bebaubar wären. Dies geschieht durch 2-maliges hin- und her Puffern nach Innen und Aussen (positive und negative Puffer-Werte). Die Puffer werden mit

Zu schmale Flächen werden mit der QGIS-Funktion "pole of inaccessibility", resp. Postgis-Funktion `ST_MaximumInscribedCircle()` (verfügbar ab Postgis 3.1/GEOS 3.9) aussortiert. Dieser Algorithmus berechnet eine Art "Polygonschwerpunkt" der garantiert innerhalb des Polygons liegt und den grösstmöglichen Kreis ins Polygon einpasst. Über den Radius/Durchmesser dieser Kreise können zu schmale Polygone aussortiert werden.

Statistik der Bauzonenreserven mit QGIS und PostGIS

Es werden 2 verschiedene Resultate geliefert: mit oder ohne Berücksichtigung der Liegenschaftsgrenzen als "Bebauungshindernis". Benachbarte Liegenschaften mit unterschiedlichen Besitzverhältnissen sind nicht immer so einfach zusammenzulegen und gelten daher in der 2. Variante als Hindernis.

Die Resultate der Bauzonenstatistik sind:

- Flächenlayer, aggregiert nach Gemeinde, Nutzungszone und Bebauungsstand
- Liegenschafts-layer mit Bebauungsstand (bebaut, unbebaut, teilweise bebaut)
- Statistiktabelle in verschiedenen Formaten

Im Vortrag wird der Analyseablauf gezeigt, und wie mit Geometrie-Unzulänglichkeiten (invalide Geometrien) umgegangen wird, welche entweder bereits in den Ausgangsdaten oder auch in den Zwischenresultaten auftreten.

In einer Pilotumsetzung mit QGIS Processing und dem grafischen Modellierungswerkzeug wurde gezeigt, dass die automatisierte Analyse ohne manuellen Eingriff für das Projekt brauchbare Resultate liefert. Für die definitive Realisierung wird der Analyse-Ablauf mit Postgis-SQL und Gretl (Gradle) umgesetzt, damit der Analyseablauf automatisiert, gesteuert über Jenkins, regelmässig ausgeführt werden kann. Gretl/Jenkins ist die Standard-Infrastruktur für automatisierte Datenkonvertierungen und Analysen im Kanton Solothurn. Die SQL-Abfragen werden derzeit umgesetzt. Nach Fertigstellung wird es interessant sein, zu vergleichen wie performant die beiden Lösungen, basierend auf QGIS und alternativ direkt in Postgis sind, und wie gut beide Varianten mit Geometriefehlern und -Problemen umgehen können.

Im angehängten Bild wird der vereinfachte Analyseablauf gezeigt, ohne Zwischenschritte für Geometriereparaturen oder das Management relevanter Attribute.

Andreas Neumann

Teilautomatische Erkennung von Innenentwicklungspotenzialen mit QGIS und Python auf Basis von amtlichen Geodaten

Die Suche nach Nachverdichtungspotenzialen (Baulücken) stellt im Rahmen einer nachhaltigen Stadt- und Raumplanung eine große Aufgabe dar. In diesem Vortrag soll eine Lösung auf Basis von QGIS und Python vorgestellt werden, die auf Basis von amtlichen Geodaten Nachverdichtungspotenziale automatisch auffinden kann und ihre theoretische Eignung als Wohnbauflächen bewerten kann.

Ein Ziel der Deutschen Nachhaltigkeitsstrategie ist es, dass Wachstum der Siedelungs- und Verkehrsfläche zu begrenzen und so die Flächenneuanspruchnahme insgesamt zu reduzieren. Ein wichtiger Handlungsbaustein in diesem Zusammenhang ist die Nachverdichtung im Innenbereich. Zur Suche und Erfassung von Nachverdichtungspotenzialen setzen viele Kommunen bereits GIS gestützte Baulückenkataster ein. Die Kataster werden allerdings oft händisch geführt und variieren stark hinsichtlich Qualität und Quantität.

Durch die zunehmende freie Bereitstellung von amtlichen Geodaten und der Weiterentwicklung von QGIS in den letzten Jahren ist es möglich Tools zu entwickeln, die Baulücken im Innenbereich automatisch erkennen und auf ihre grundsätzliche Eignung als Nachverdichtungspotenzial prüfen.

Im Rahmen einer Masterarbeit an der Universität Bonn ist ein solches Tool auf Basis von QGIS und Python entwickelt worden, das in der Lage ist auf Grundlage von amtlichen Geodaten (Alkis, Basis-DLM und LOD2-Gebäudemodellen) Nachverdichtungspotenziale automatisch zu erkennen und ihre Eignung als Wohnbaufläche zu prüfen. Beispielsweise wurde für die Ortslage „Köln“, also den linksrheinischen Bereich innerhalb des zweiten Ringes, ein theoretisches Nachverdichtungspotenzial von ca. 155 Hektar ermittelt.

In diesem Vortrag soll die Funktionsweise des Tools näher vorgestellt werden. An einem Beispiel aus dem Forschungsprojekt NACHWUCHS der Universität Bonn wird gezeigt, wie das Tool praktisch genutzt werden kann und wie die gefundenen Nachverdichtungspotenziale mit freien Geodaten beispielsweise aus OSM weiter untersucht und bewertet werden können.

Mirko Blinn

Der Weg zur modernen Basiskarte – Potenziale und Herausforderungen von Kartographie und Datengrundlage

Der Einsatz von Basiskarten zählt seit jeher zu den kartographischen Grundpfeilern einer jeden (Web)GIS Applikation. Sie bieten den Nutzern Orientierung und Wiedererkennungswert und bieten eine geeignete Basis für themenspezifische Fachdaten. Dieser Vortrag erläutert und diskutiert die Erstellung moderner Basiskarten unter Berücksichtigung von sowohl amtlichen- als auch OSM-Daten, dem Einsatz moderner kartographischer Trends sowie der Verwendung freier GIS Software.

Der Open-Source GIS Welt stehen mittlerweile eine Vielzahl an unterschiedlichen Tools zur Erstellung von Basiskarten zur Verfügung. Neben der Wahl der Kartenserver-Software, eng verknüpft mit Wahl zwischen Vektor- oder Rasterkacheln, muss ein geeignetes Styling-Format gefunden werden, das sämtliche Funktionen für den gewünschten Kartographie-Stil gerecht werden kann. Die Qualität der Karte ist zudem von der Datenbasis abhängig, beispielsweise des verfügbaren Straßen- und Wegenetzes.

Wenngleich Open Data zunehmend Bedeutung in öffentlichen Institutionen gewinnt und folglich mehr und mehr Geodatensätze zur Verfügung stehen, so reichen diese oftmals nicht als Basis für Gestaltung von Basiskarten aus. Hierbei kommen freie OpenStreetMap-Daten ins Spiel, die beispielsweise als Ergänzung dienen können. Schwierigkeiten sind hierbei unter anderen unterschiedliche Aktualität der Daten, Labeling oder im Allgemeinen die Topologie.

Diese und weitere Herausforderungen sollen in diesem Vortrag am Beispiel der neuen Basiskarten des Stadtmessungsamt Stuttgarts erläutert und diskutiert werden. Hintergrund ist die Migration von proprietärer zur quelloffener Software, in diesem Fall in erster Linie GeoServer, sowie der Nach- bzw. Neubau von Basiskarten in offenen Stilformaten. Konkret geht es um die Ausgestaltung von 60 Layern, vorliegend im GeoPackage Format, und das Zusammenstecken zu einer Basiskarte.

Hannes Blitza

Der Weg zur modernen Basiskarte – Potenziale und Herausforderungen von Kartographie und Datengrundlage

Geostyler Mapfile Parser



Der GeoStyler Mapfile Parser kann automatisch Darstellungsinformationen aus Mapfile Layer in andere Formate wie SLD oder QGIS Style übersetzen. Dies ermöglicht es zum Beispiel die Darstellung von MapServer Projekten in QGIS Projekte zu transferieren.

Es gibt eine grosse Vielfalt an Definitionen und Formaten graphischer Repräsentationen von räumlichen Informationen wie zum Beispiel QGIS Style File (QML), QGIS Layer Definition File (QLR) oder Styled Layer Descriptor (SLD) unter anderen. GeoStyler bietet ein intermediäres Format zur automatischen Übersetzung von Stilen zwischen unterschiedlichen Formaten.

In diesem Kontext wurde nun das GeoStyler Projekt um den Mapfile-Parser erweitert. Der GeoStyler Mapfile Parser wurde 2020 von Camptocamp als Case Study für das Bundesamt für Landestopographie (swisstopo) entwickelt. Aktuell ist es möglich, Mapfile Styles zu lesen und in andere verfügbare GeoStyler Formate zu übersetzen.

Die Präsentation wird eine kleine Einführung in das GeoStyler Framework geben sowie den aktuellen Stand des Mapfile Parsing genauer erläutern. Abgerundet wird der Vortrag mit der Diskussion der gemachten Erfahrungen und einem Ausblick auf mögliche weitere Entwicklungen.

Balthasar

Häufig gestellte Fragen

Was ist die FOSSGIS-Konferenz?

Die FOSSGIS-Konferenz ist im D-A-CH-Raum die führende Konferenz für Freie und Open Source Software für Geoinformationssysteme sowie für die Themen Open Data und OpenStreetMap.

Die FOSSGIS-Konferenz 2021 wird vom gemeinnützigen FOSSGIS e.V und der OpenStreetMap Community mit Unterstützung des Instituts für Software des OST Campus Rapperswil organisiert.

Ziel der jährlich stattfindenden Konferenz ist die Verbreitung von Freier und Open Source Software für Geoinformationssysteme. Hier treffen sich Anwender und Entwickler zum gemeinsamen Austausch über Anwendungs- und Arbeitsmöglichkeiten sowie neuste Entwicklungen in diesem Bereich. Es geht um Webmapping, Geodateninfrastrukturen, Geodatenmanagement, GeoProcessing und Desktop-GIS und um freie Geodaten, wie OpenStreetMap.

Die Konferenz wird vorwiegend durch ehrenamtliches Engagement getragen, wobei neben den Teilnehmergebühren, die Spenden- und vor allem die Sponsoren- und Ausstellerstandeinnahmen eine wichtige finanzielle Einnahmequelle darstellen, um die Kosten für die Konferenz zu decken und darüber hinaus Open Source Projekte zu fördern.

Was finden Sie in diesem Tagungsband?

Sie finden die Beiträge zum Programm der FOSSGIS 2021.

Einreichungen zum Tagungsband verfügen über die Angabe Verfasser:innen-Angabe unter dem Titel und die Kontaktdaten am Ende des Beitrags.

Die Beiträge, die einen Beschreibungstext enthalten und der Name der Verfasser:in unter dem Beitrag steht, sind am 05.05.2021 aus dem Vortragsverwaltungsprogramm Pretalx exportiert worden.

Unter welcher Lizenz stehen die Beiträge?

Die Konferenzbeiträge stehen unter der Lizenz CC BY 4.0 zur Verfügung. Der Konferenzbeitrag umfasst die Kurzbeschreibung des Beitrags, den Aufsatz im Konferenzband, die Vortragsfolien sowie die Videoaufzeichnung und Liveübertragung des Vortrags.

