

Anlass

ALKIS mit SpatialLite

GKG-ALKIS-Konverter - ein Skript-Konverter zur Aufbereitung von NAS-XML für QGIS auf Basis von *ogr2ogr* und *SpatialLite*

Dr.-Ing. Claas Leiner

Anlass

Daten aus dem Liegenschaftskataster in Deutschland werden von den Vermessungs- und Katasterverwaltungen der Länder über die *Normbasierte Austausch-Schnittstelle (NAS)*, einem XML-Format, ausgeliefert [1]. Der *PostNas*-Treiber von OGR liest NAS-XML-Dateien, so dass QGIS dieses Format direkt öffnet [2]. Allerdings ist ein direktes Laden ins QGIS bei großen Datenmengen wenig praktikabel, weil die XML-Dateien recht langsam gelesen werden. Zielführend ist in diesen Fällen eine Konvertierung über *ogr2ogr*. [3]

Anschließend sind in beiden Fällen umfangreiche SQL-Abfragen notwendig, um die vielfältigen Beziehungen zwischen Geometrien und Tabellen auszuwerten, welche eine sinnvolle Nutzung der Daten erst ermöglichen. Erst dann lassen sich Eigentümer von Flurstücken ermitteln, Adressen an Gebäude und Flurstücke anbinden, die Flächennutzung von Flurstücken bilanzieren oder die Gebäude eines Flurstücks auflisten. Die Beziehungen zwischen Flurstücken und Eigentümern werden über eine komplexe mehrstufige Relationskette abgebildet, welche die Einbeziehung von mindestens vier Tabellen erfordert.

AX_Flurstueck > (*AX_Buchungsblatt*) > *AX_Buchungsstelle* > *AX_Namensnummer* > *AX_Personen*

Die XML-Daten müssen notwendigerweise in eine SQL-fähige Datenbank überführt werden, um Sie im QGIS oder einem anderen GIS zu nutzen.

Die *PostNas-Suite* mit dem *NorGIS-ALKIS-Import* und dem dazugehörigen *QGIS-Plugin* bietet eine umfängliche Lösung auf Basis von *PostGis* an [4]. Die Anwendung von *NorGIS-ALKIS* erfordert zwingend, dass die Nutzer/innen dem Umgang mit einer serverbasierten Datenbank gewachsen sind. Zielgruppe sind entsprechend eher Nutzer großer Datenbestände und insbesondere die öffentliche Verwaltung.

Alternativ zu dieser bewährten Lösung wird hier der Ansatz einer kompakten und transportablen Umsetzung mit *SpatialLite* [4] vorgestellt, bei welchem sämtliche Daten und Verknüpfungen in einer *SQLite-Datei* vorliegen. Zusätzlich wird ein QGIS-Projekt mit einer einfachen aber gut lesbaren Symbolisierung und eine Eigentümerabfragemöglichkeit über 1:N Relationen erstellt. [5] Zielgruppe sind Nutzer/innen, die regelmäßig kleinere oder mittelgroße (bis etwas Landkreis-Größe) ALKIS-Datenbestände für ihre Planungsaufgaben verwenden und diese dateibasiert verwalten möchten. Weiterhin ermöglicht das Skript Tabellen mit sämtlichen Nutzern einzelner Flurstücke oder *Shapefiles* mit den vollständigen Eigentümerlisten jeden Flurstücks auszugeben.

Das Skript kommt auch mit größeren Datenmengen zu recht. So ist z.B. der ALKIS-Bestand von Berlin ohne Eigentümer in eine *SpatialLite*-Datenbank geschrieben worden.

Was macht der GKG-ALKIS-Konverter

Der GKG-ALKIS-Konverter konvertiert sämtliche NAS-XML-Dateien eines Verzeichnisses mit Hilfe von *ogr2ogr* in eine *SpatiaLite-Datenbank* und bereitet die Inhalte für eine sofortige Verwendung in QGIS auf. Die Anwender/innen müssen das Ursprungs-Bundesland angeben und können ein Ausgabe-Koordinatensystem frei wählen. Umfassen die Dateien des Verzeichnisses mehrere Layer eines Themas (z.B. *AX_flurstueck* für verschiedene Ortsteile), so werden sie in einer Tabelle zusammengefasst. Die Verarbeitung von Eigentümern und die Modellierung von Grundstücken kann aktiviert oder deaktiviert werden.

In der *SpatiaLite-Datenbank* sind anschließend sämtliche ALKIS-Layer und Tabellen, sowie zusätzlich die Layer bzw. Tabellen *Flächennutzung*, *Flurstueckvollinfo* und *Gebäudevollinfo*, *bs_nn_eigentuemmer* und *Grundstuecke* verfügbar.

- *Flächennutzung* enthält die zusammengefasste Flächennutzung mit detaillierter thematischer Klartext-Attributierung.
- *Gebäudevollinfo* enthält die Gebäudegeometrien mit Gebäudeadresse, Hausnummer und der Gebäudenutzung im Klartext sowie das Flurstückskennzeichen. Es sind nicht nur die Gebäude aus *ax_gebaeude*, sondern auch die Sondergebäude erfasst.
- Die Tabelle *bs_nn_eigentuemmer* stellt eine Liste sämtlicher Buchungsstellen mit den zugehörigen Eigentümern und ihrer Adresse zur Verfügung.
- *Flurstueckvollinfo* umfasst die Flurstücksgeometrien mit Flurstückskennzeichen und Adresse samt Hausnummer oder Flurbezeichnung sowie eine Auflistung sämtlicher Eigentümer. Außerdem werden die Flächennutzungen der Flurstücke mit Quadratmeterangabe als Attribut aufgelistet.

Für die drei Layer ist das QGIS-Projekt *alkis_ges.qgs* mit thematischer Symbolisierung, Flurstücksbeschriftung und Anbindung der Eigentümer vorbereitet. *Flurstueckvollinfo* und *bs_nn_eigentuemmer* sind im QGIS-Projekt über eine 1:N-Beziehung verknüpft, so dass im Objektformular sämtliche Eigentümer eines Flurstücks mit Ihren Adressen abfragbar sind.

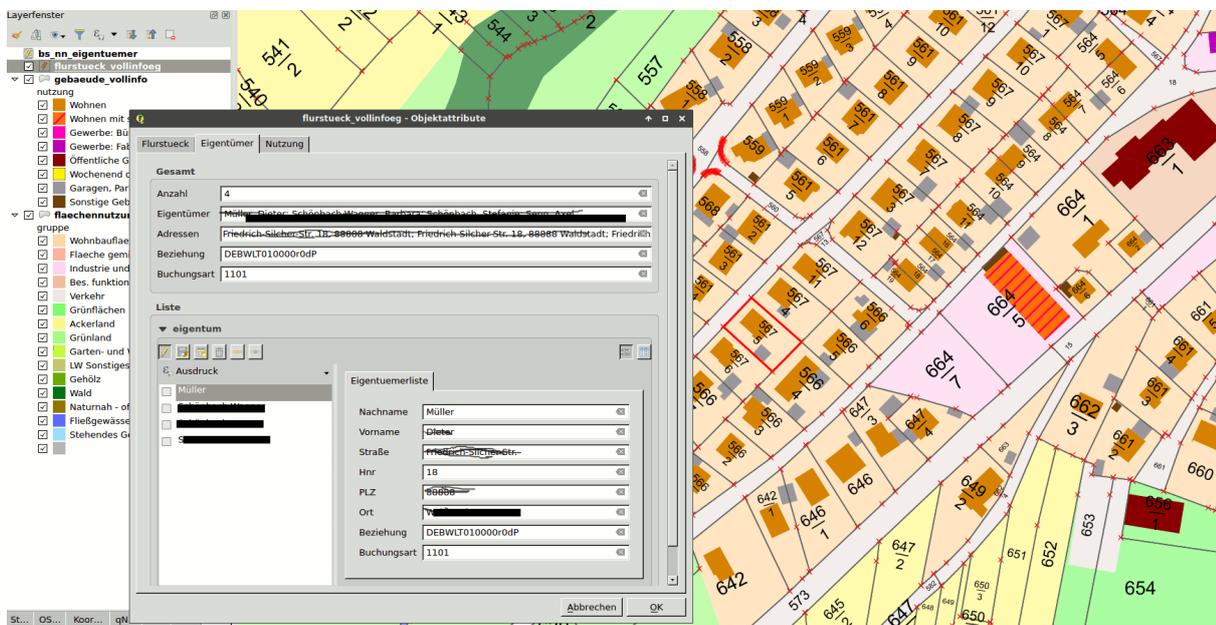


Abb. 1: Flurstücksabfrage und Symbolisierung

Arbeitsweise und Funktion

Der GKG-ALKIS-Konverter besteht im wesentlichen aus einem Batch-Skript für Windows (*alkis_conv.bat*), verschiedenen SQL-Skripten sowie einem Bash-Shell-Skript (*alkis_conv.sh*) für Linux-Betriebssysteme. Das Skript ruft *ogr2ogr* mit den, für das jeweilige Bundesland passenden Optionen auf, konvertiert sämtliche *NAS-Dateien* eines Verzeichnisses in eine *Spatialite-Datenbank* und startet verschiedene SQL-Skripte, um die thematischen und räumlichen Beziehungen zwischen den Daten abzufragen, welche für eine sinnvolle Attributierung und Präsentation der Daten notwendig sind. Außerdem kopiert es die passende Projektdatei in das Datenverzeichnis und startet QGIS mit diesem Projekt. Weiterhin werden Flurstücke, Gebäude sowie die Flächennutzung als *Shapefiles* und die Eigentümer mit Buchungsblattinformationen, Flurstücksbeziehungen etc. als Excel-Tabelle exportiert.

Um das Skript auszuführen ist ein 7z-File zu entpacken und das enthaltene Verzeichnis nach *C:\user\benutzername* bzw. das Shell-Skript und ein Verzeichnis nach */home/nutzername/bin* zu kopieren. Für Windows und Linux gibt es jeweils einen Desktop-Starter, so dass das die Nutzer selbst keine Shell öffnen müssen. QGIS mit *ogr2ogr* sowie *Spatialite* müssen auf dem Rechner vorhanden sein sein.

Beim Start prüft das Skript den Ort der QGIS-Installation und das Vorhandensein von *ogr2ogr* sowie *Spatialite* und weist gegebenenfalls auf eine erforderliche Installation hin.

Anschließend sind die Nutzer/innen gefordert ein paar Eingaben vornehmen:

- Pfad zu den XML-Dateien
- Sollen Eigentümer angebunden werden?
- In welches Koordinatenbezugssystem sollen die Daten geschrieben werden
- Aus welchem Land stammen die Daten?

Es müssen nur Länder angegeben werden, die nicht ETRS89 / UTM32N als Koordinatenbezugssystem verwenden, alle anderen fallen unter Sonstiges:

Anschließend ruft das Skript über eine *for-Schleife*, für jede XML-Datei im Verzeichnis, das zum jeweiligen Land passende *ogr2ogr*-Kommando mit der *-addfields* Option auf. Für Baden-Württemberg wird z.B. die exotische sechsstellige Gauß-Krüger-3-Definition berücksichtigt und die Daten in das vorher gewählte KBS umprojiziert:

```
ogr2ogr --config OGR_SQLITE_SYNCHRONOUS OFF --config OGR_SQLITE_CACHE 2048 -progress -f "SQLite" -dsco SPATIALITE=yes -lco SRID=$kbs -lco COMPRESS_GEOM=YES -nlt PROMOTE_TO_MULTI -skipfailures -addfields -gt unlimited -s_srs "+proj=tmerc +lat_0=0 +lon_0=12 +k=1 +x_0=500000 +y_0=0 +ellps=bessel +datum=potsdam +nadgrids=BETA2007.gsb +units=m +no_defs" -t_srs EPSG:$kbs $alkis.sqlite $xmls 2>> ogrammeldungen.txt
```

```
*****
Wenn Eigentüemerdaten vorhanden sind bitte ja eingeben ansonsten nein: ja
*****
Folgende zwei Optionen kosten ein wenig Zeit, im Zweifelsfall n für nein
Excelltabelle mit Buchungsblattinfos schreiben? j eingeben ansonsten n: j
*****
Wenn Grundstücke modelliert werden sollen j eingeben ansonsten n: j
*****
Die fertige SQLITE-Datei ist unter folgendem Pfad zu finden:
C:\Users\c\aa\Desktop\alks\heberle\alkis_ges.sqlite
*****
In welchem Koordinatensystem sollen die Daten ausgegeben werden? (Mit Ziffer wählen
Ohne Auswahl werden Die daten nach (ETRS89 / UTM 32) geschrieben.
[1] 31467 ist Gauß-Krüger 3: Tippen Sie 1 und RETURN
[2] 31468 ist Gauß-Krüger 4: Tippen Sie 2 und RETURN
[3] 25832 ist (ETRS89 / UTM 32): Tippen Sie 3 und RETURN
[4] 25833 ist (ETRS89 / UTM 33): Tippen Sie 4 und RETURN
Bitte eine Auswahl treffen: 3
KBS ist 25832
*****
Aus welchem Land stammen die Daten?
[1] Baden Württemberg: Tippen Sie 1 und RETURN
[2] Bayern: Tippen Sie 2 und RETURN
[3] BB, MeckP, Berlin Tippen Sie 3 und RETURN
[4] Sachsen: Tippen Sie 4 und RETURN
[5] Sonstige: Tippen Sie 5 und RETURN
Tippen Sie 1 (BaWü), 2 (Bay), 3 (BraB), 4 (Sach) oder 5 (Sonstige) und RETURN: 1
```

Abb. 2: Benutzerabfragen unter Windows

Arbeitsweise und Funktion

Sind sämtliche XML-Dateien in die SQLite-Datei importiert, arbeitet das Skript die verschiedenen SQL-Skripte ab. Dabei ruft es mit Hilfe des Programms *spatialite* entweder direkt SQL-Befehle auf oder (unter Windows) ruft die Befehle aus Dateien auf:

```
spatialite -silent $alkis.sqlite "SQL-Befehle;"
```

oder

```
spatialite -silent %alkis%.sqlite < "%HOMEDRIVE%\%HOMEPATH%\gkg\sqlskript.sql"
```

Da bei kleinen Datenbeständen häufig Tabellen oder einzelne Spalten fehlen, legt das Skript zunächst, soweit erforderlich, Dummy-Spalten- und Tabellen an, um einem Abbruch des Skriptes vorzubeugen.

Der *Spatialite-SQL-Dialekt* beinhaltet sämtliche notwendigen GIS-Funktionen um räumliche Abfragen und Verschneidungen umzusetzen.

```
1254 --Nutzung.mit.Flurstuecken.verschneiden
1255 create table flst_nutzung as
1256 select distinct
1257 casttomultipolygon(ST_intersection(a.geometry,b.geometry)) as geometry,
1258 round(area(casttomultipolygon(ST_intersection(a.geometry,b.geometry)))) as qm,
1259 a.gml_id as gml_id,
1260 b.nutzung as nutzung
1261 from ax_flurstueck as a inner join flaechnutzung as b on ST_intersects(a.geometry,b.geometry) = 1
1262 where qm > 0 and a.ROWID IN
1263 ..... (SELECT ROWID FROM SpatialIndex WHERE f_table_name='ax_flurstueck' AND search_frame=b.geometry)
1264 ;
1265 Select RecoverGeometryColumn('flst_nutzung','Geometry',$kbs,'MULTIPOLYGON');
1266 Select CreateSpatialIndex('flst_nutzung','Geometry');
1267 SELECT UpdateLayerStatistics('flst_nutzung');
```

Abb. 3: Verschneidung von Flurstücken und Nutzung zur Auswertung der Flächennutzung

```
--Die.anteilige.Flaechennutzung.in.die.Flurstuecke.schreiben
create table flurstueck_zwischeninfo as
select distinct
a.gml_id as gml_id,
a.Geometry as geometry,
a.lage as lage,
a.adresse as adresse,
a.hausnr as hausnr,
group_concat((b.nutzung || ':' || b.qm || '.qm'), ';' ) as nutzungen

from flurstueck_lage as a left join

-- (select
-- gml_id,
-- nutzung,
-- sum(qm) as qm
-- from
-- flst_nutzung
-- --Version.als.group-by.(gml_id||nutzung)
-- group by gml_id, nutzung)

as b on (a.gml_id = b.gml_id)
where qm > 0
group by a.gml_id;

Select RecoverGeometryColumn('flurstueck_zwischeninfo','Geometry',$kbs,'MULTIPOLYGON');
Select CreateSpatialIndex('flurstueck_zwischeninfo','Geometry');
SELECT UpdateLayerStatistics('flurstueck_zwischeninfo');
```

Abb. 4: Anteilige Nutzung in die Flurstücke schreiben

Im wesentlichen führt die SQL-Skripte folgende Schritte aus:

- Die Landnutzungsthemen werden zu einem Flächennutzungslayer zusammengeführt.
- AX_Gebäude und die Sondergebäudelayer (z.B. *ax_bauwerkoderanlagefuerindustrieundgewerbe* oder *ax_turm*) werden zusammengeführt und im Klartext attribuiert.

Arbeitsweise und Funktion

- Anbinden von Hausnummern und Adressen an Flurstücken und Gebäuden.
- Flurstückskennung als Attribut zu jedem Gebäude schreiben.
- Anteilige Flächennutzung jeden Flurstücks als Attribut des Flurstücks generieren.
- Sämtliche Eigentümer jeden Flurstücks mit Adresse als Attribut in den Flurstückslayer schreiben.
- Erzeugen einer Tabelle sämtlicher Eigentümer mit direkter Beziehung zu den Flurstücken und Adressen jeden Eigentümers als Attribut schreiben.
- Exportiert Gebäude, Flächennutzung und Flurstücke mit Eigentümern sowie Nutzungsanteilen als Shapefiles.

Am Ende ermittelt das Skript die räumliche Ausdehnung der Daten und startet QGIS mit der zum KBS passenden Projektdatei, so dass die ALKIS-Daten fertig symbolisiert zur Verfügung stehen.

```
*****  
Alle Prozesse sind durchgeführt das Programm kann nun beendet werden.  
*****  
Soll QGIS gestartet werden, um das Projekt zu öffnen? ja oder nein
```

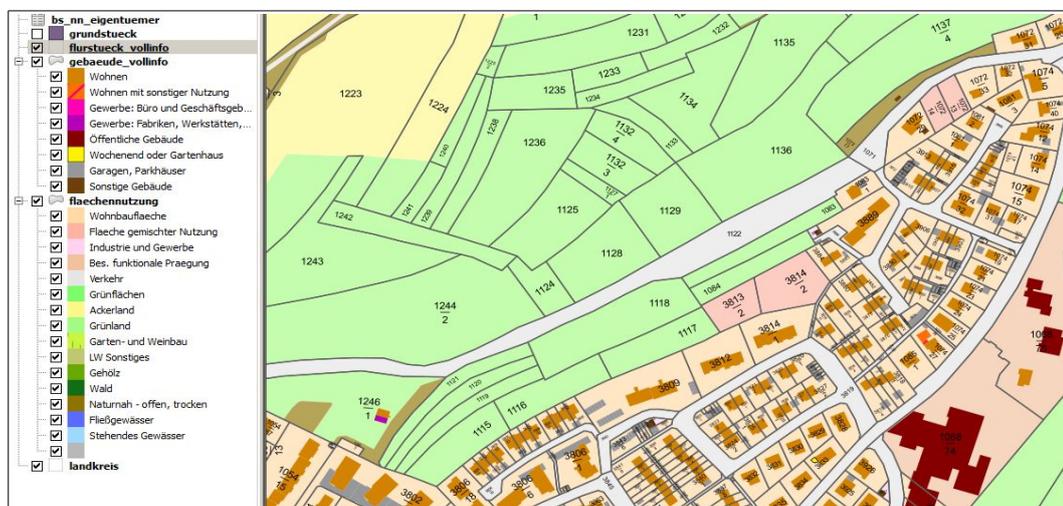


Abb. 5: ALKIS-Projekt im QGIS

Entwicklungsgeschichte

Das Skript umfasst inzwischen inklusive Kommentare mehr als 2000 Zeilen. Entstanden ist es aus verschiedenen Einzel-SQL-Skripten, die zu verschiedenen Anlässen erstellt wurden.

Ich biete seit 2014 Dienstleistungen rund um QGIS an und wurde in 2014 darum gebeten, Eigentümer mit Flurstücken zu verbinden. ALKIS war für mich Anlass, intensiver in SQL einzusteigen. Erstes Vorhaben war eine kurzes SQL-Skript, welches eine vollständige Eigentümerliste in eine Attributspalte des Flurstückslayers schrieb. Allerdings bleiben anschließend ca. 5 % der Flurstücke ohne Eigentümer. Was daran lag, dass die Relationskette:

AX_Flurstueck > AX_Buchungsstelle > AX_Namensnummer > AX_Personen

die Beziehung zwischen Eigentümern und Flurstücken nicht vollständig beschreibt.

Entwicklungsgeschichte

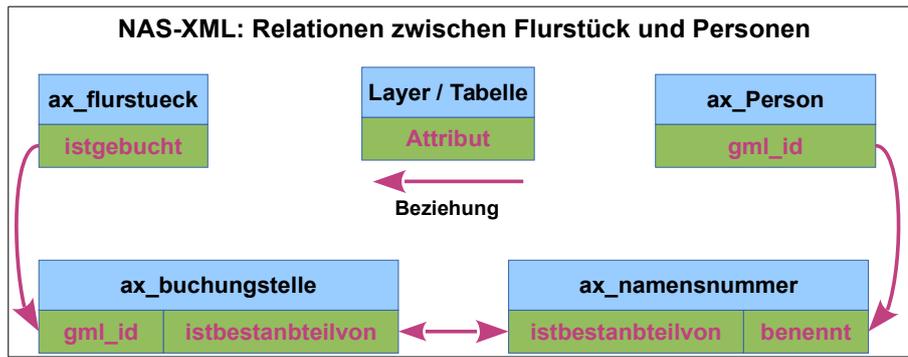


Abb. 6: Grundlegende Beziehung Personen und Flurstück

Die Buchungsstellen der Eigentümergemeinschaften (Buchungsart 1101) verweisen auf Flurstücke, verfügen für sich jedoch über keine Beziehung zu einzelnen Eigentümern. Abb. 6 beschreibt die Beziehungen zwischen Eigentümern und Flurstücken ohne Berücksichtigung von Eigentümergemeinschaften. In Abbildung 7 ist zu erkennen, dass zusätzlich die *Buchungsstellen* der Eigentümergemeinschaften über eine Relation aus der Tabelle *ALKIS-Beziehungen* mit sich selbst verbunden werden müssen, um von der *Buchungsstelle* der Gemeinschaften zu den *Buchungsstellen* der einzelnen Eigentümer zu gelangen.

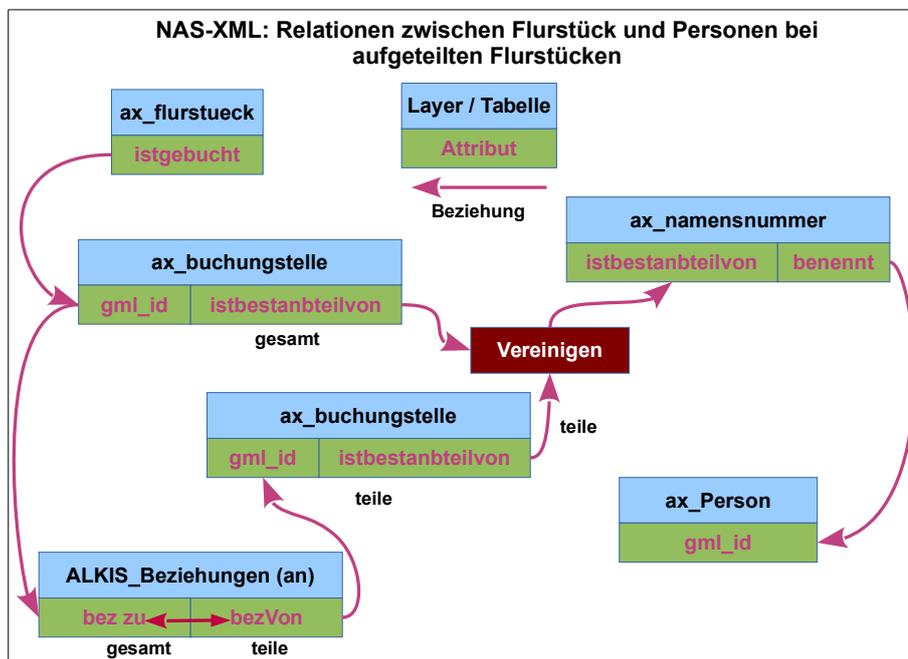


Abb. 7 : Anbindung unter Berücksichtigung von Eigentümergemeinschaften

Nachdem dieses Schema im SQL-Code umgesetzt wurde, ließen sich die Eigentumsverhältnisse sämtlicher Flurstücke abbilden. Einige Zeit später, trat ein Planungsbüro mit dem Wunsch an mich heran, sämtliche Nutzungstabellen (*AX_Landwirtschaft*, *AX_Wald*, *AX_Wohnbauflaeche* etc.) in einen einzigen Flächennutzungslayer zusammenzufassen und die Nutzungen als Klartext-Attribut einzutragen.

Entwicklungsgeschichte

Außerdem sollten die Gebäude mit ihrer Klartextnutzung und den Adressen sowie Hausnummern in einen Layer geschrieben werden. Nachdem dieses Skript fertiggestellt war, mussten die Kunden, um ihre ALKIS-Daten in die gewünschte Form zu bekommen, zunächst die *OSGeo4w-Shell* öffnen, über das Kommando *cd* ins gewünschte Verzeichnis wechseln und den von mir zusammengestellten *ogr2ogr-Befehl* für jede einzelne XML-Datei durchlaufen lassen. Anschließend konnten Sie die fertige SQLite-Datei mit dem *SpatiaLiteGui* öffnen und das SQL-Skript aufrufen. In einem abschließenden Schritt waren die Daten ins QGIS zu laden und mit einer QML-Datei zu symbolisieren.

Das war doch etwas aufwändig. Außerdem bedurfte es für Daten aus anderen Bundesländern zusätzliche angepasste *ogr2ogr*-Aufrufe und Varianten im SQL-Code. Auch brachen die Skripte häufig ab, weil z.B. einzelne Spalten bei kleinen Datensätzen nicht vorhanden waren.

Vor diesem Hintergrund, wollte ich die vorhandenen SQL-Skripte und die *ogr2ogr*-Kommandos zu einem Skript zusammenfassen, welches für jedes Land geeignet ist und auch bei kleinen Datensätzen nicht wegen fehlender Spalten abbricht.

Anforderungen an das Skript und Umsetzung

- *ogr2ogr* und SQL-Skripte sollen nacheinander aufgerufen werden, ohne das die Anwender weitere Programme öffnen müssen.
- Landesspezifische Besonderheiten sind zu berücksichtigen.
- Es müssen ganze Verzeichnisse von NAS-XML-Dateien angesprochen werden können und nicht nur einzelne Dateien.
- Fehlende Tabelle oder Spalten dürfen nicht zum Skript-Abbruch führen.
- Das Werkzeug soll auf jedem Windows-Rechner bei unterschiedlichen QGIS-Installationen laufen und den Ort der QGIS-Installation finden.
- Das Werkzeug soll über einen Desktop-Starter geöffnet werden und keine Kenntnisse über Kommandozeilen-Befehle erfordern.
- Eine Umsetzung in unterschiedliche Koordinatensysteme muss möglich sein.
- QGIS soll automatisch mit einem fertig symbolisierten Projekt starten.
- Das Skript muss die Anwender fragen, ob eine Anbindung von Eigentümern erforderlich ist

Für mich hieß es, in das Schreiben von Shell-Skripten unter Linux (bash) und Windows (cmd) einzusteigen und dort insbesondere Eingabemöglichkeiten für die Auswahl der landesspezifischen *ogr2ogr*-Befehle etc. umzusetzen. Als unerwartet langwierig erwies sich unter Windows, das Skript so zu gestalten, dass es mit den verschiedenen möglichen Pfaden der QGIS-Installation zurecht kam.

Nach dem alles lief, hatte ich die Möglichkeit schnell auf weitere Kundenwünsche zu reagieren und z.B. den ALKIS-Konverter um ein Skript zu ergänzen, welches die adressenbezogene Einwohnerstatistik mit Flurstücken und Gebäuden verbindet und auch eine Aggregation zu Baublöcken erlaubt.

Was fehlt?

Was fehlt?

Es gibt noch einige Probleme und viele Inhalte, die bisher nicht ausgewertet werden. Stellvertretend seien hier genannt:

- Insbesondere gibt es noch keine ALKIS-Stile gemäß Signaturenkatalog, sondern nur eine selbst gestrickte Symbolisierung und Flurstücksbeschriftung.
- Die Auswertung der anteiligen Flächennutzung kann bei einzelnen Flurstücken unkorrekt sein, da die SQL-Verschneidung im *Spatialite* über Geometriefehler stolpert. Sie erkennen das, wenn entweder keine anteilige Nutzung berechnet ist, oder die Summe der Nutzungen nicht der Gesamtfläche entspricht. Über Hinweise zur Lösung des Problems bin ich dankbar.
- Viele Attribute, die für die Gebäude in einigen Ländern vorhanden sind (Stockwerke, Zustand etc.) werden aktuell noch nicht ausgewertet.
- Die Erb-Pachverhältnisse der Flurstücke werden aktuell nicht ausgewertet.
- Es fehlt auch noch eine ansprechende grafische Benutzeroberfläche
- Oder eine Umsetzung als QGIS-Plugin.

Wie ist es zu nutzen und welche Pläne gibt es.

Der Autor Claas Leier stellt das Werkzeug allen Interessierten für die private Nutzung kostenfrei zur Verfügung. Für Mitglieder des QGIS.DE-Anwendervereins und des FOSSGIS e.V. ist auch die kommerzielle oder behördeninterne Nutzung uneingeschränkt und kostenfrei möglich.

Sonstigen behördlichen und kommerziellen Nutzern wird das Tool bis 2019 nur gegen Honorar zur Verfügung gestellt. Spätestens ab dem 31. Dezember 2019 möchte ich das Werkzeug als freie Software veröffentlichen.

Kontakt zum Autor:

Dr.-Ing. Claas leiner
GKG-Kassel
Wilhelmshöher Allee 304 E, 34131 Kassel
0561/56013445
claas.leiner@gkg-kassel.de

Weblinks

[1] <http://www.adv-online.de/AdV-Produkte/Liegenschaftskataster/ALKIS/>

[2] <http://trac.wherogroup.com/PostNAS>

[3] <http://www.gdal.org/ogr2ogr.html>

[3] <https://www.norbit.de/68/>

[4] <https://www.gaia-gis.it/fossil/libspatialite/home>

[5] <http://www.gkg-kassel.de/>