

FOSSGIS-Konferenz 2013

Im- und Export von dxf-Daten mit OGR

Datenmigration ...

ist wie der Umzug aus einer 5-Raumwohnung

in eine 5-Raumwohnung

oder

in ein 1-Raumapartment

gegebenenfalls muss die Einrichtung
angepasst werden

Datenmodelle für Geodaten sind schlichte Relationen (1:1)

Geometrie:

Simple Feature (Punkt, Linie, Fläche)

+

Beschreibung:

tabellarische Struktur (z.B. See, Wald, Name, Biographie, numerische und alphanumerische Beschreibung)

gid	geom	land	stadt	strasse	hs_nr
1	Geometrie	Land	Stadt	Strasse	Nr.

1:n Relationen werden im GIS abgebildet (Ausnahmen: NAS, Interlis u.a.)

dxg ist ebenfalls ein schlichtes Datenmodell

allerdings ist der Einsatzbereich größer
(Entwicklung, Steuerung und Fertigung)

und

eine historisch bedingte enge Bindung an ein
Layout

(jedes Objekt hat obligatorisch einen Style)

es gibt keinen Inhalt ohne Information zur Darstellung

Das dxf-Format ist ein ASCII-Format und kann mit jedem Texteditor manipuliert werden.

Die Struktur einer dxf-Datei hat die folgende Gliederung:

SECTION: HEADER (*dxg-Version, Encoding, Autor, ...*)

SECTION: CLASSES

SECTION: TABLES (*Linientypen, Schrifttypen, Layer, ...*)

SECTION: BLOCKS (*Symboldefinitionen*)

SECTION: ENTITIES (*Geometrieobjekte*)

SECTION: OBJECTS

SECTION: ACDSDATA (*ab AC1027, Release 2013/...*)

dxg schreiben bedeutet
die Informationen werden auf die
verschiedenen 'SECTION' verteilt

und

dxg lesen verluft umgekehrt
die Informationen werden aus den
'SECTION' zusammengebaut

Fallbeispiel: Gebäude mit Hausnummer und ...

In einem PostgreSQL/PostGIS
Datenbankmanagementsystem ist eine solche Information
tabellarisch in Geometrie und Attribute aufgeteilt.

Um diese Information in ein dxf-Format zu übersetzen,
werden mehrere Objekte (ENTITY) in der SECTION
ENTITIES erzeugt:

HATCH	Schraffur (optional)
LWPOLYLINE	Umring Gebäude
MTEXT	Hausnummer
MTEXT	ggf. weitere Attribute

Zu jedem ENTITY wird die geometrische Ausprägung, der Layer, die Farbe und ggf. ein Verweis auf den Textstil definiert.

Textstil und Layer werden zudem in der SECTION TABLES definiert.

Im dxf-Format ist eine Beziehung zwischen Geometrie und Attributen nicht vorgesehen.
(Ausnahme: Block mit Attributen)

Die Beziehung Gebäude und Hausnummer existiert somit nicht mehr.

Der gleiche Vorgang in die gegenläufige Richtung (Gebäude und Hausnummer in ein PostgreSQL/PostGIS DBMS):

Aus den dxf-Daten geht keinerlei Beziehung zwischen Umring und dem Text hervor. Auf die Schraffur wird verzichtet, der Umring wird als Polygon und der Text als Punkt mit Attribut Text übersetzt.

Sofern die Datenstruktur eindeutig ist kann in einem weiteren Prozessierungsschritt eine Verbindung zwischen dem Polygon und dem Punkt wieder hergestellt werden.

Mit der OGR-Bibliothek können Daten in jeder dxf-Version erstellt werden.

Einschränkung:

Seit der Version AC1021 (2007) erwartet AutoCAD eine dxf-Datei mit Unicodezeichensatz.

Hierbei ist völlig egal welcher Zeichensatz in der SECTION HEADER angegeben wird.

Um eine dxf-Datei zu erstellen verwendet OGR eine Headerdatei (Template).

Dieses Template ist eine vollständige dxf-Datei. Jedoch ohne Definition in der SECTION ENTITIES.

Dieses Template kann gezielt angesteuert werden und somit kann auf ein dem Datenmodell angepasstes Template verwiesen werden.

Hierin sollten bereits die verwendeten Blocks (Symbole) und Schrift- bzw. Textstile enthalten sein.

Layer müssen nicht vorab definiert sein. Vorzugsweise sollte dieses Template bereits mit dem Zielsystem erstellt werden.

Um Daten aus einem PostgreSQL/PostGIS DBMS in ein dxf-Format zu exportieren sind diese räumlich und fachlich zunächst zusammen zu stellen.

Hierzu wird eine fachliche Sicht (View) auf die Daten erstellt.

```

CREATE OR REPLACE VIEW ogr_dxf AS
SELECT ogc_fid AS gid,
       'strassennamen'::text AS layer,
       'LABEL(t:" || straname|| "',s:2.5g,a:' ||drehung||',p:4)'::text AS ogr_style, --Text
       NULL::text AS blockname,
       NULL::text AS blockscale,
       NULL::int AS blockangle,
       the_geom
FROM my_table WHERE the_geom IS NOT NULL
UNION
SELECT gid,
       'gebaeude_schraffur'::text AS layer,
       'BRUSH(fc:#0000FF,id:ogr-brush-0)'::text AS ogr_style, --Schraffur Polygon
       NULL::text AS blockname,
       NULL::text AS blockscale,
       NULL::int AS blockangle,
       the_geom
FROM schema.my_table WHERE the_geom IS NOT NULL
UNION
SELECT gid,
       'gebaeude'::text AS layer,
       'PEN(c:#141414,w:0.25g)'::text AS ogr_style, --Umring Polygon
       NULL::text AS blockname,
       NULL::text AS blockscale,
       NULL::int AS blockangle,
       ST_ExteriorRing((ST_Dump(the_geom)).geom) AS the_geom
FROM schema.my_table WHERE the_geom IS NOT NULL
UNION

```

```

SELECT gid,
       'hausnummer'::text AS layer,
       'LABEL(t:" || hs_nr || "',s:2.5g)'::text AS ogr_style, --Text
       NULL::text AS blockname,
       NULL::text AS blockscale,
       NULL::int AS blockangle,
       ST_Centroid((ST_Dump(the_geom)).geom) AS the_geom
FROM schema.my_table WHERE the_geom IS NOT NULL
UNION
SELECT ogc_fid AS gid,
       'symbol_block'::text AS layer,
       NULL::text AS ogr_style, --Symbol
       'Baum'::text AS blockname,
       '(10,10,10)'::text AS blockscale,
       10::int AS blockangle, --auch numeric
       the_geom
FROM my_table WHERE ST_GeometryType(the_geom) = 'ST_Point'
UNION
SELECT ogc_fid AS gid,
       'linie'::text AS layer,
       'PEN(c:#0000FF,w:2g)'::text AS ogr_style, --Linie
       NULL::text AS blockname,
       NULL::text AS blockscale,
       NULL::int AS blockangle,
       the_geom
FROM my_table WHERE ST_GeometryType(the_geom) = 'ST_LineString'
;

```


Steuerung OGR

```
ogr2ogr -sql "(SELECT * FROM ogr_dxf)"  
-f "dxf" ... \ogr_dxf.dxf  
-dsco header=... \header_2013.dxf  
optional -clipsrc 2558500 5676000 2561000 5678500  
"PG:dbname=my_db host=localhost user=ich password=xxx  
schemas=public tables=public.ogr_dxf"
```

Quellen:

http://www.gdal.org/ogr/ogr_feature_style.html

<http://gis.uster.ch/dokumentation/datenkonvertierung/ogr>

<http://postgis.net/docs/manual-2.0/>

Vielen Dank für ihre Aufmerksamkeit