




Der Weg zur modellgetriebenen GDI

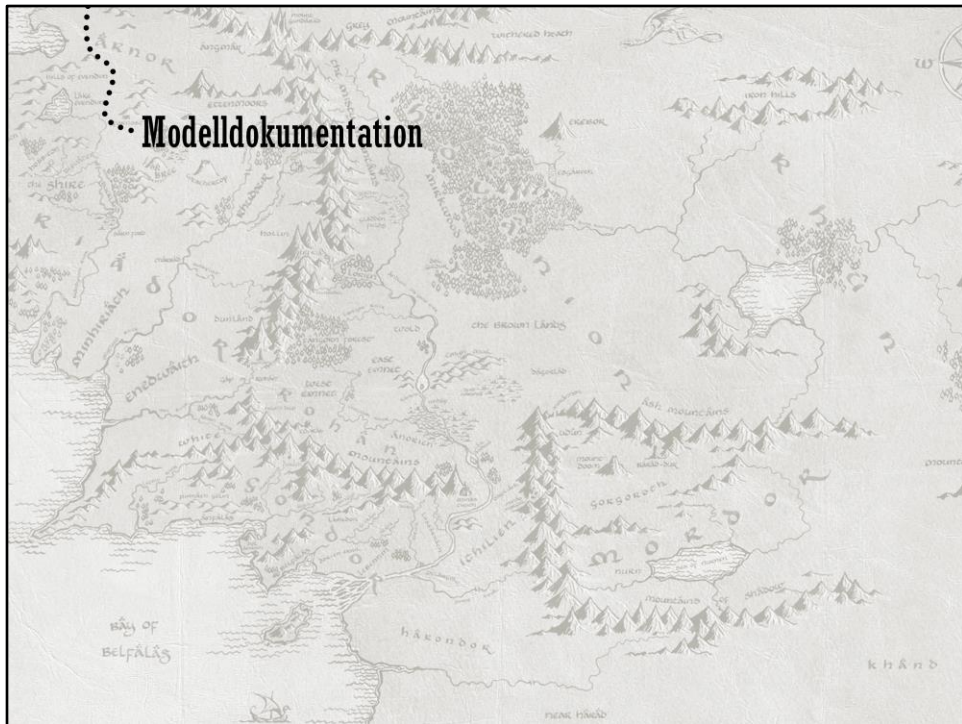
Spirgarten 2017

Bau und Umwelt | Raumentwicklung und Geoinformation

1

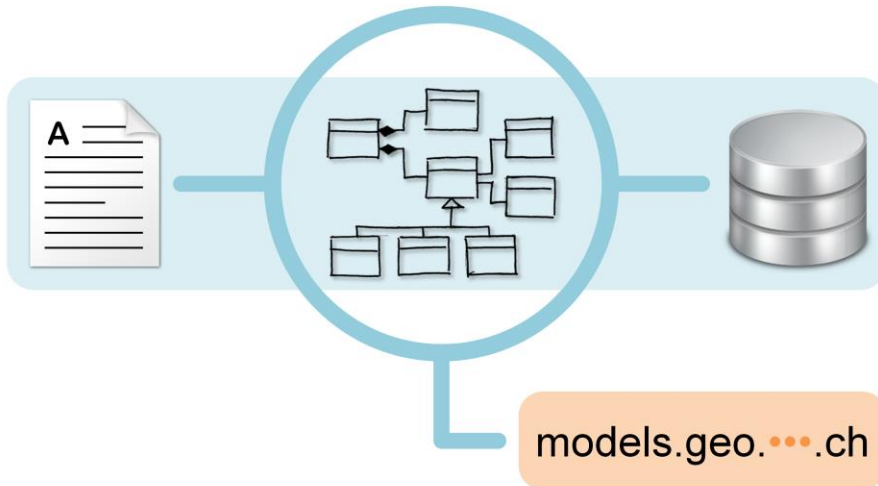


2



MODELLDOKUMENTATION

Inhaltsbeschreibung & Strukturdefinition



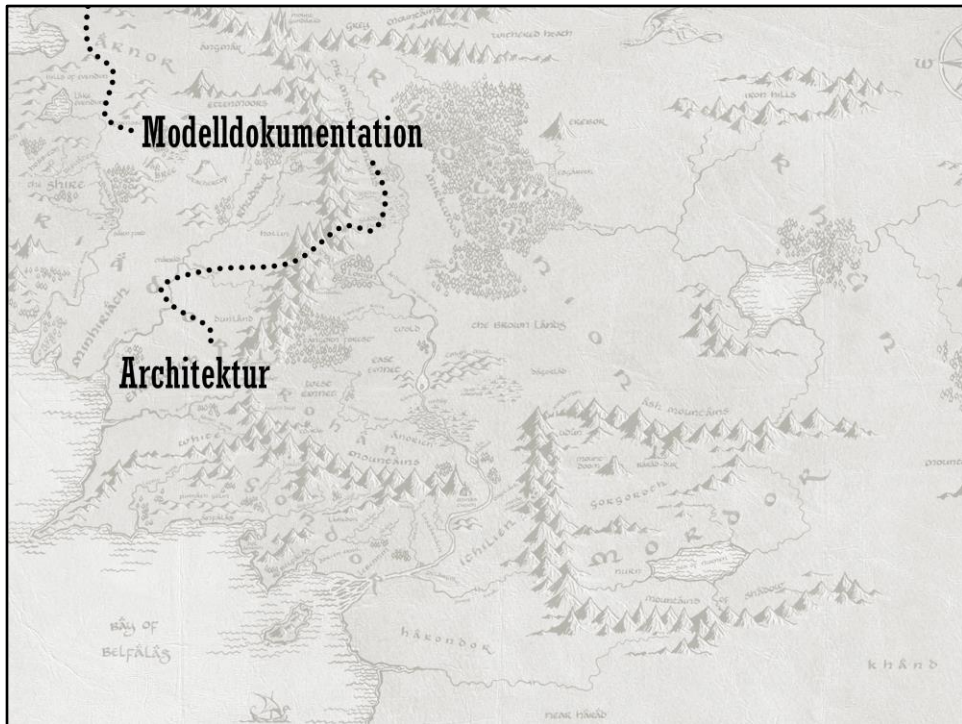
4

Im Rahmen der Umsetzung des EG GeoIG und im Zusammenhang mit dem modellbasierten Ansatz ist offenkundig, dass Datenmodelle von essentieller Bedeutung für den Austausch und die Nutzung von Geo(basis)daten sind.

Eine Datenmodelldokumentation besteht im Wesentlichen aus zwei Teilen: der Inhaltsbeschreibung (Semantik) und der Strukturdefinition. Die Inhaltsbeschreibung muss den kompletten Inhalt des Datenmodells vollständig beschreiben und zwar in einer Form, die eine fachfremde Person versteht. Die Überführung in das konzeptionelle Datenmodell erfolgt mittels Objektkatalog, wo ein erster Strukturierungsschritt stattfindet. In Tabellenform werden alle Objekte (Klassen) mit ihren Eigenschaften (Attributen, Attributtypen) und ggf. gegenseitigen Beziehungen (Assoziationen) aufgelistet.

Das konzeptionelle Datenmodell überführt die Semantik in eine formale Darstellung, üblicherweise als UML-Klassendiagramme. Mit dem UML/INTERLIS-Editor können so konzeptionelle Datenmodell entwickelt und geprüft werden. Aus dem UML-Modell kann der formale Text in Form von INTERLIS CSL exportiert werden. Dieser formale Text wird genutzt, um Datenmodelle in produktiven Systemen zu implementieren. INTERLIS-Datenmodelle werden in Datenmodellablagen (Model Repositories) online publiziert, um sie so allgemein zugänglich zu machen (insbesondere auch für Software-Werkzeuge!).

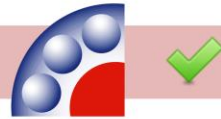
Zusammen mit den Transferdaten und den zugehörigen Metadaten ist die Modelldokumentation ein wichtiger Bestandteil von Datenlieferungen.



ARCHITEKTUR

GDI-Architektur

modellkonform



modelläquivalent



modellnah



6

modellkonform: Ein Datensatz ist modellkonform, wenn die Daten gemäss definierten und dokumentierten Kodierungsregeln mit dem für diese Daten geltenden konzeptionellen Datenmodell übereinstimmen. Kodierungsregeln können grundsätzlich für beliebige Datenformate definiert werden [KKGEO/GKG (2016): Handlungsanweisungen für die modellkonforme Bereitstellung von Geodaten mittels Download-Diensten gemäss GeoIG].

⇒ INTERLIS-Transferdaten gemäss eCH-0031 oder eCH-0118 sind modellkonform.

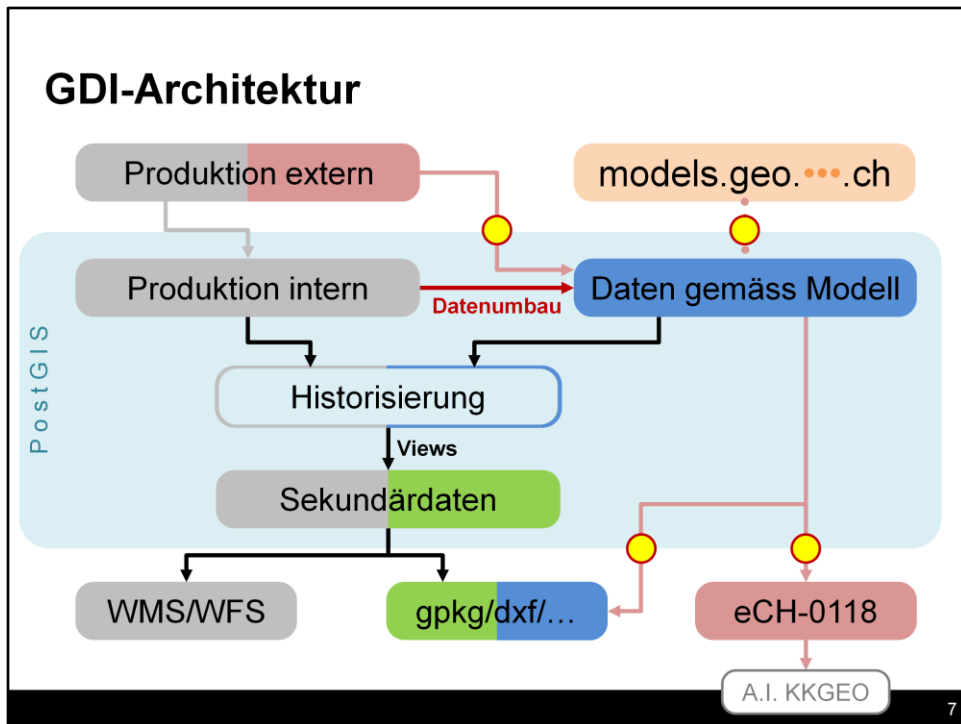
modelläquivalent: Daten, die in einem relationalen Datenbanksystem gespeichert sind, entsprechen einem internen logischen Schema und nicht einem konzeptionellen Datenmodell. Wenn dieses logische Schema jedoch nach definierten Abbildungsregeln mit einem generischen Schnittstellenwerkzeug erzeugt wurde, dann können Daten aus diesem Schema in einen modellkonformen Transferdatensatz exportiert werden.

⇒ Daten in einem von ili2pg erzeugten Datenbankschema sind modelläquivalent.

modellnah: Aus modelläquivalenten Daten können – beispielsweise durch die Definition von Sichten – abgeleitete Daten erzeugt werden, die für den nutzerorientierten Gebrauch optimiert sind. Die Objektstruktur und die Attributnamen/-werte orientieren sich so eng wie möglich am entsprechenden konzeptionellen Datenmodell. Solche Daten können als OGC Webservices publiziert werden oder auch als «nutzerorientierte Datenprodukte» in verschiedenen Formaten exportiert und abgegeben werden.

Beispiel: Aus den Daten der amtlichen Vermessung wird ein Datensatz mit den Grundstücksflächen abgeleitet, der gleichzeitig alle zugehörigen Sachinformationen wie die Grundstücksnummer oder den eidgenössischen Grundstücks-Identifikator (EGRID) enthält. Dieser Datensatz wird als OGC WMS-Layer publiziert sowie im Format GeoPackage (GPKG) exportiert und abgegeben. Solche Daten sind nicht modellkonform, aber aus Nutzersicht einfach nutzbar.

⇒ Daten, die aus modelläquivalenten Daten hergeleitet sind und eine für den praktischen Gebrauch optimierte Objektstruktur aufweisen, sich aber so nah wie möglich an den Definitionen des konzeptionellen Datenmodells orientieren, heissen modellnah.



Aus der **externen Datenproduktion** werden entweder modellkonforme (rot) oder beliebige Daten(-strukturen/-formate) geliefert und in die PostGIS-DB integriert.

Datenmodelle stehen in der Datenmodellablage (Model Repository) models.geo.gl.ch zur Verfügung.

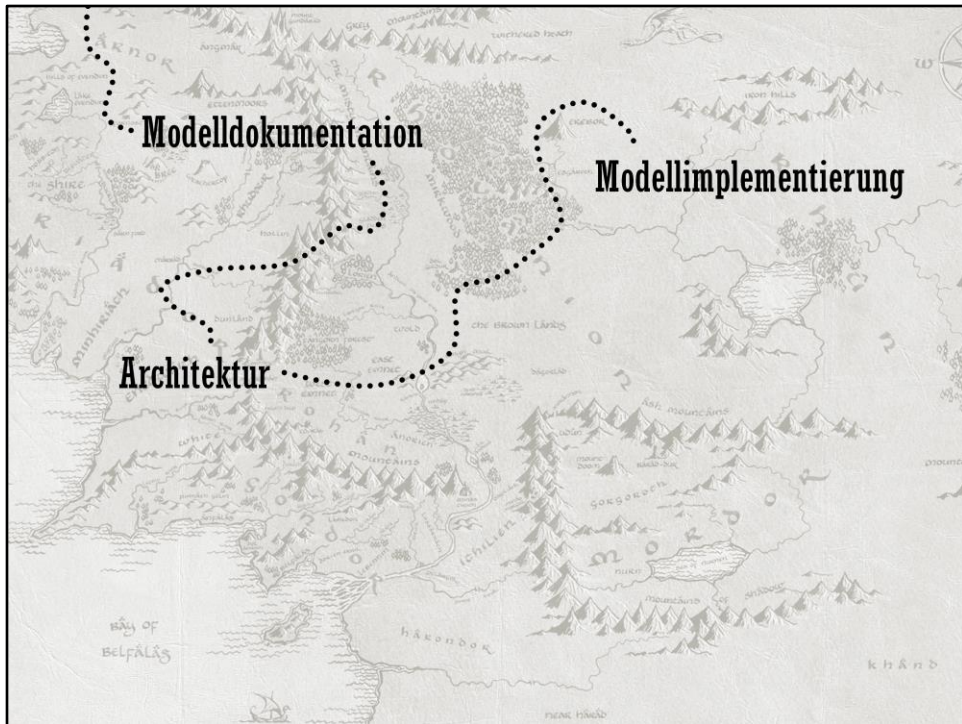
Durch einen **Datenumbau** (semantische Transformation) werden interne Produktionsdaten in die modelläquivalente Schemastruktur überführt (blau). Die Integration modellkonformer Transferdaten erfolgt ebenso in diese Schemastruktur.

Die produktiven Daten werden automatisch **historisiert** und es werden mittels Datenbanksichten (Views) **Sekundärdaten zur Publikation** im kantonalen Geoportal abgeleitet. Um eine optimierte und möglichst breite Nutzung zu gewährleisten, werden aus den modelläquivalenten Daten modellnahe Layer/Produkte abgeleitet (grün).

Daraus entstehen die Nutzerdaten im Geoportal: OGC Webservices und Download-Daten. Die Download-Daten sind nach Möglichkeit modellnah zu erstellen.

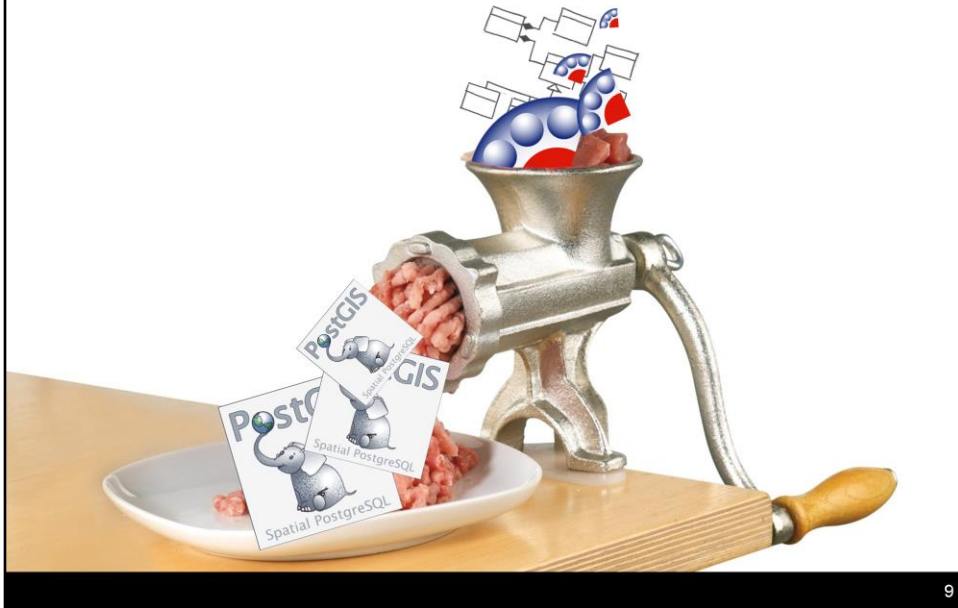
Aus den modelläquivalenten Daten (blau) können mittels definierter Regeln **modellkonforme Transferdaten** (rot) **exportiert**, geprüft und weiter verwendet werden. Daraus lassen sich zusätzlich modelläquivalente Datenprodukte ableiten, beispielsweise GeoPackage. Die modellkonformen Transferdaten werden insbesondere durch die Aggregationsinfrastruktur (A.I.) der KKGeo genutzt, um schweizweit harmonisierte Geobasisdaten bereitzustellen.

Für den Schemaimport, den Datenimport sowie den Datenexport werden **exakt definierte Abbildungsregeln** und entsprechende **generische Schnittstellenwerkzeuge** benötigt.



MODELLIMPLEMENTIERUNG

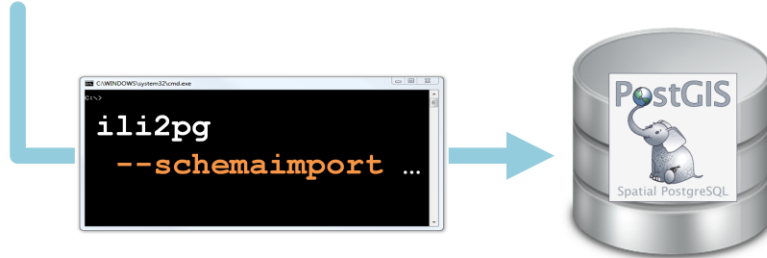
Modelle in die Datenbank!



Die konzeptionellen Datenmodelle müssen in eine Datenbank-Schemastruktur übersetzt werden. Dabei ist es zentral, dass es definierte Umsetzungsregeln gibt, wie die einzelnen Datenmodellelemente auf Datenbank-Schemaelemente abgebildet werden.

Modelle in die Datenbank!

models.geo....ch



10

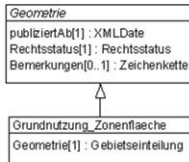
Mit dem generischen Schnittstellenprogramm **ili2pg** (OpenSource) wird dies bewerkstelligt. Das Werkzeug kann manuell über die Kommandozeile bedient werden oder natürlich auch leicht in automatisierte Prozesse eingebunden werden.

Als Beispiel folgt der Befehl, wie das Nutzungsplanungsmodell im Kanton Glarus mit ili2pg in PostGIS umgesetzt wurde (Kommandozeile unter Windows 7):

```
java -jar C:\PATH_TO\ili2pg.jar
--schemaimport
--dbhost MY_SERVER --dbusr MY_USER --dbpwd MY_PWD
--dbdatabase glarus
--defaultSrsAuth EPSG --defaultSrsCode 2056
--importTid
--nameByTopic
--createEnumTabs
--smart1Inheritance
--createGeomIdx
--createFk
--createUnique
--dbschema m_nutzungsplanung
--models GL_Nutzungsplanung_V1
```

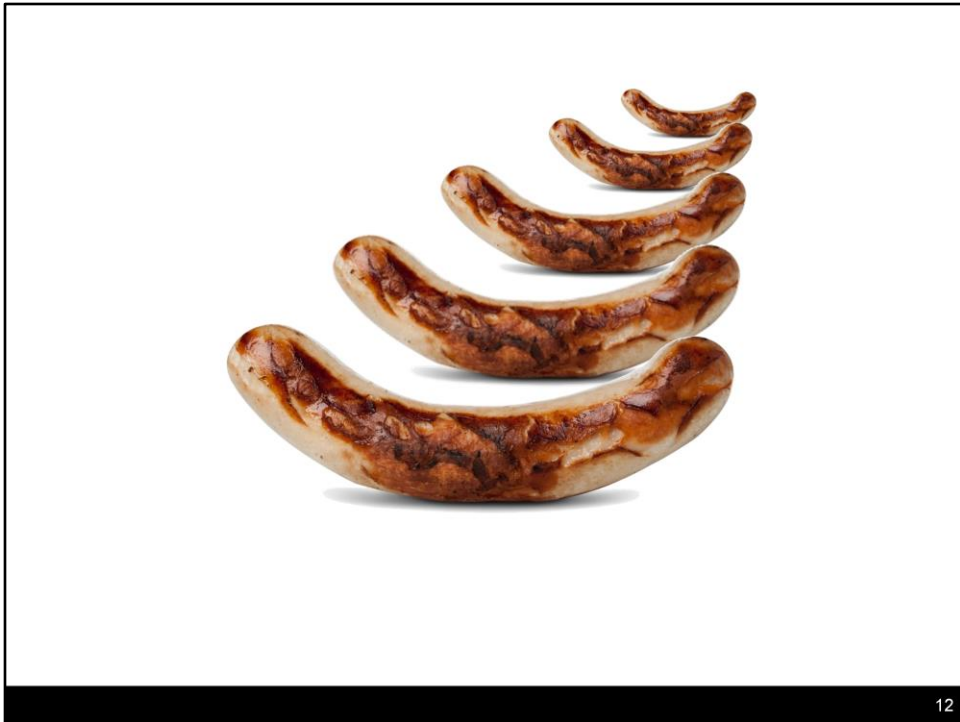
<http://www.eisenhutinformatik.ch/interlis/ili2pg/>

Modelle in die Datenbank!



```

CREATE TABLE m_nutzungsplanung.geobasisdaten_grundnutzung_zonenflaeche
(
  t_id bigint NOT NULL DEFAULT nextval('m_nutzungsplanung.t_id12db_seq'::regclass),
  t_type character varying(60) NOT NULL,
  t_ili_tid uuid DEFAULT uuid_generate_v4(),
  publiziertab date NOT NULL, -- @iliname publiziertAb
  rechtsstatus character varying(255) NOT NULL, -- @iliname Rechtsstatus
  bemerkungen text, -- @iliname Bemerkungen
  typ bigint NOT NULL,
  geometrie geometry(CurvePolygon,2056), -- @iliname Geometrie
  CONSTRAINT geobasisdaten_grundnutzung_zonenflaeche_pkey PRIMARY KEY (t_id),
  CONSTRAINT geobasisdaten_grundnutzung_zonenflaeche_typ FOREIGN KEY (typ)
    REFERENCES m_nutzungsplanung.geobasisdaten_typ (t_id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION DEFERRABLE INITIALLY DEFERRED
)
WITH (
  OIDS=FALSE
);
  
```



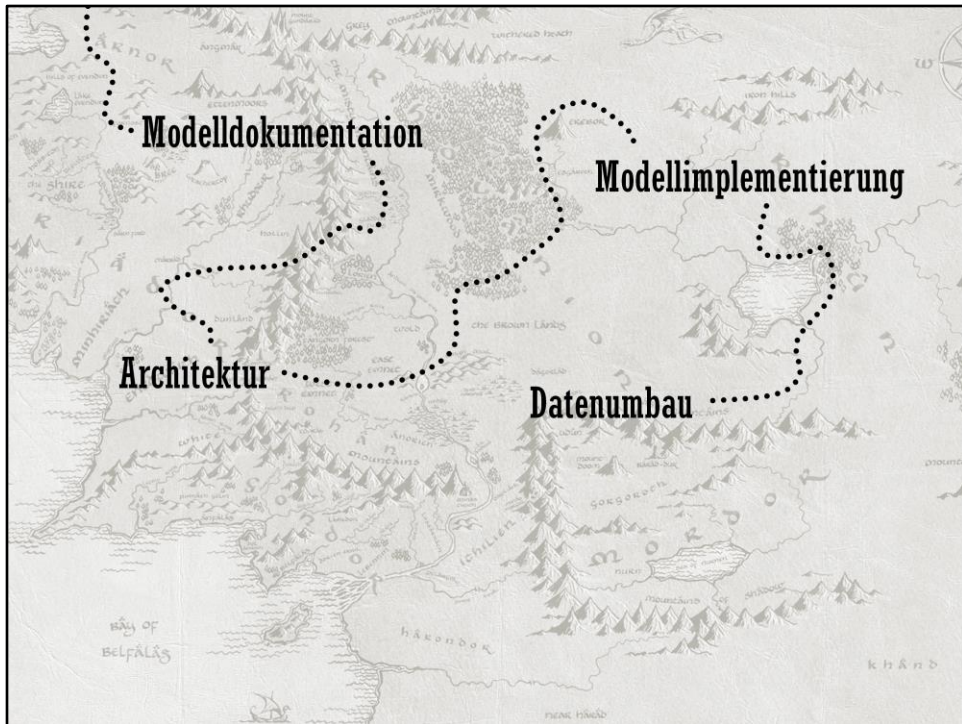
12

Es ist dabei völlig WURSCHT, *wie genau* das generische Schnittstellenprogramm konzeptionelle Datenmodelle in der Datenbank implementiert!

Zwei Aspekte sind wichtig:

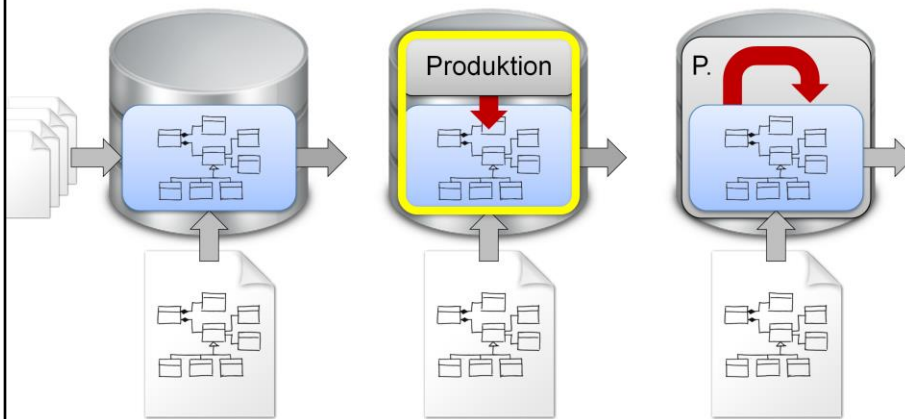
1. Ich muss ein einziges Mal genau *verstehen*, wie die einzelnen Modellkonstrukte umgesetzt werden (→ Dokumentierung des Programms/Anwenderhandbuch!).
Beispielsweise wie eine Vererbung in der Datenbank abgebildet wird, damit ich später die Objektbildung korrekt vornehmen kann;
2. Ich muss mich 100% darauf verlassen können, dass das generische Schnittstellenprogramm jedes Mal jedes beliebige Datenmodell exakt gleich gemäss definierten Regeln umsetzt.

Der Export von modellkonformen Transferdaten muss «Import⁻¹» entsprechen. Dann ist es zweitrangig, wie genau die Modellumsetzung datenbankintern abläuft.



DATENUMBAU

«Modell-isierung»



14

Fall A:

Datenmodell wird in der DB konfiguriert, externe Produktionsdaten werden modellkonform geliefert.

Fall B:

Datenmodell wird in der DB konfiguriert; bestehende interne Produktionsdaten werden mittels semantischer Transformation in das modelläquivalente Schema umgebaut.

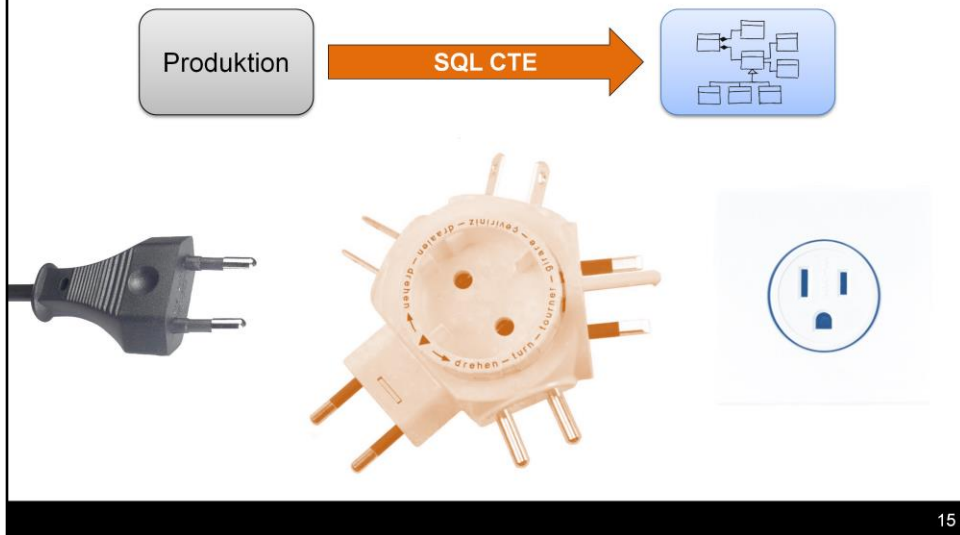
Fall C:

Datenmodell wird in der DB konfiguriert; interne Produktion wird originär auf die modelläquivalente Schemastruktur umgestellt.

Fall A—C:

Modellkonforme Transferdaten werden aus der modelläquivalenten Schemastruktur exportiert.

Datenumbau (bei Fall B)



15

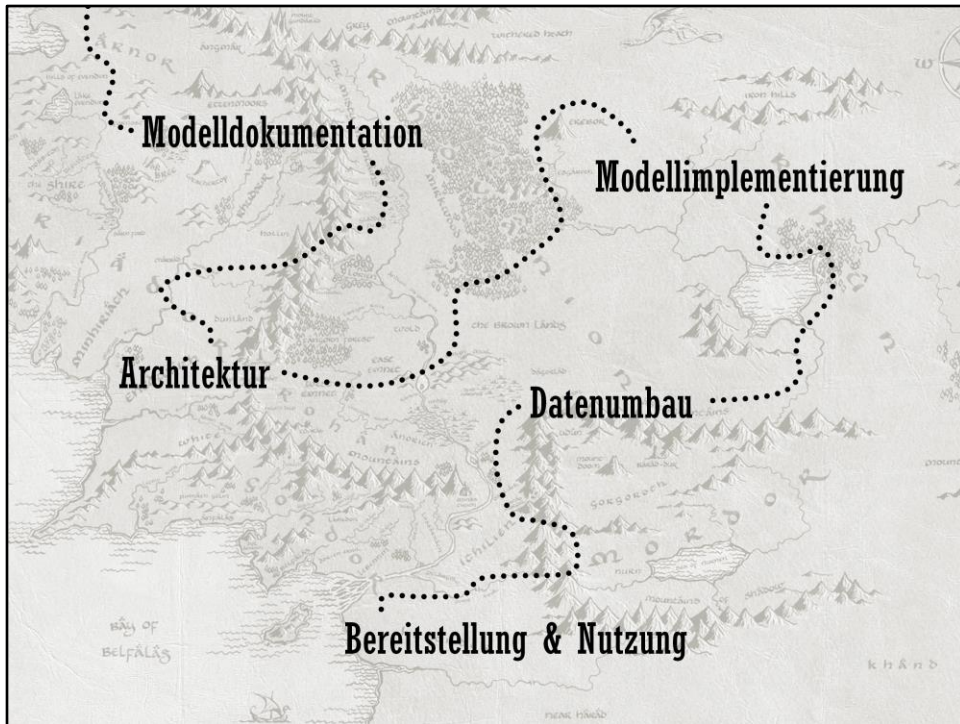
Der **Datenumbau** mittels semantischer Transformation erfolgt bei uns mit der Verwendung von PostGIS-Datenbanken mittels SQL-Skripten.

Eine sehr nützliche und mächtige Methode ist der Einsatz von so genannten «Common Table Expressions» (CTE) eine hierarchische, rekursive Datenbankabfrage-Methodik. Dabei werden vorbereitete Abfragen wiederum in Abfragekonstrukten eingebunden, um alternativ zu abgeleitete, temporären Datenbanktabellen direkt die gewünschten Resultate zu liefern.

Datenbank-extern kann der Datenumbau selbstverständlich auch sehr gut mit dem Datenverarbeitungswerkzeug FME erledigt (und über die Server-Version) automatisiert werden.

https://en.wikipedia.org/wiki/Hierarchical_and_recursive_queries_in_SQL#Common_table_expression

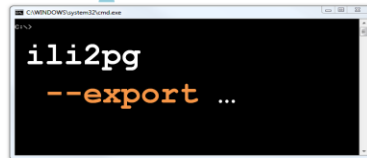
<https://www.postgresql.org/docs/9.6/static/queries-with.html>



BEREITSTELLUNG & NUTZUNG

Transferdaten aus der Datenbank!

models.geo....ch



17

Mit dem generischen Schnittstellenwerkzeug **ili2pg** können **modellkonforme Transferdaten** aus PostGIS **exportiert** werden. Dazu wird wiederum die Datenmodelldefinition aus der Datenmodellablage verwendet.

Als Beispiel folgt der Befehl, wie aus dem modelläquivalenten Datenbankschema `m_nutzungsplanung` modellkonforme Transferdaten exportiert werden (Kommandozeile unter Windows 7):

```
java -jar C:\PATH_TO\ili2pg.jar
--export
--dbhost MY_SERVER --dbusr MY_USER --dbpwd MY_PWD
--dbdatabase glarus
--dbschema m_nutzungsplanung
--models GL_Nutzungsplanung_V1
./nupla_export.xtf
```

Und genauso leicht können GML-Daten gemäss **eCH-0118** Version 2.0 erzeugt werden:

...

```
--ILIGML20
./nupla_export.gml
```

Transferdaten aus der Datenbank!

```
<TRANSFER xmlns="http://www.interlis.ch/INTERLIS2.3">
  <HEADERSECTION SENDER="ili2pg-3.6.0-20161230" VERSION="2.3">
    <MODELS> ...
    <MODEL NAME="GL_Nutzungsplanung_V1" VERSION="2016-12-07" URI="http://geo.gl.ch"/>
  </MODELS>
</HEADERSECTION>
  <DATASECTION>
    <GL_Nutzungsplanung_V1.Geobasisdaten BID="GL_Nutzungsplanung_V1.Geobasisdaten">
      ...
      <GL_Nutzungsplanung_V1.Geobasisdaten.Typ_Kt TID="5c768ffb-692e-4b46-9b13-f5aa40ab7215">
        <Code>42-01</Code>
        <Bezeichnung>Strasse ausserhalb Siedlung</Bezeichnung>
        <Abkuerzung>StrA</Abkuerzung>
        <Hauptnutzung_CH REF="ch.admin.are.nutzungsplanung.hauptnutzung_ch.42"/>
      </GL_Nutzungsplanung_V1.Geobasisdaten.Typ_Kt>
      ...
    </GL_Nutzungsplanung_V1.Geobasisdaten>
  </DATASECTION>
</TRANSFER>
```



BROWSE

INTERLIS transfer file

UPLOAD



<https://interlis2.ch/ilivalicator/>

Modellkonforme Transferdaten können (und sollen auch!) gegenüber dem konzeptionellen Datenmodell geprüft werden.

Ein neues Werkzeug ist der **ilivalicator** (OpenSource), welcher als Service betrieben wird.

Transferdaten können in den Service geladen werden und nach erfolgter Prüfung wird ein Prüfprotokoll zurückgegeben.

<https://interlis2.ch/ilivalicator/>

Automatisieren Standardisieren

curl://

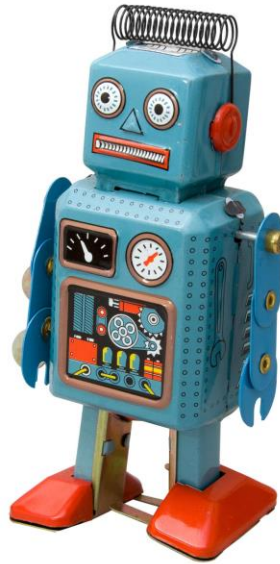
OpenSearch.org



RSS &
Atom



eCH



19

Um eine effiziente modellgetriebene Geodaten-Infrastruktur zu erzielen, müssen möglichst viele Prozesse automatisiert werden. Dazu gehören

- Regelmässige Datenimports
- Export modellkonformer Transferdaten
- Prüfung der exportierten Transferdaten
- Bereitstellung verschiedener Datenprodukte zum Download
- Abgabe von Geobasisdaten an amtliche Stellen (Bund)
- Integration von Geobasisdaten in die Aggregations-Infrastruktur (A.I.) der KKGeo

Zentral ist dabei die Standardisierung beziehungsweise der Einsatz standardisierter Methoden und Werkzeuge:

- cURL – <https://curl.haxx.se/>
- OpenSearch.org – <http://www.opensearch.org/>
- pyCSW – <http://pycsw.org/>
- RSS & Atom – <https://tools.ietf.org/html/rfc4287> | <https://tools.ietf.org/html/rfc5023>
- eCH-0031, eCH-0056, eCH-0118 – s. unter <http://www.ech.ch/>
- GeoPackage – <http://www.geopackage.org/>

and back again...



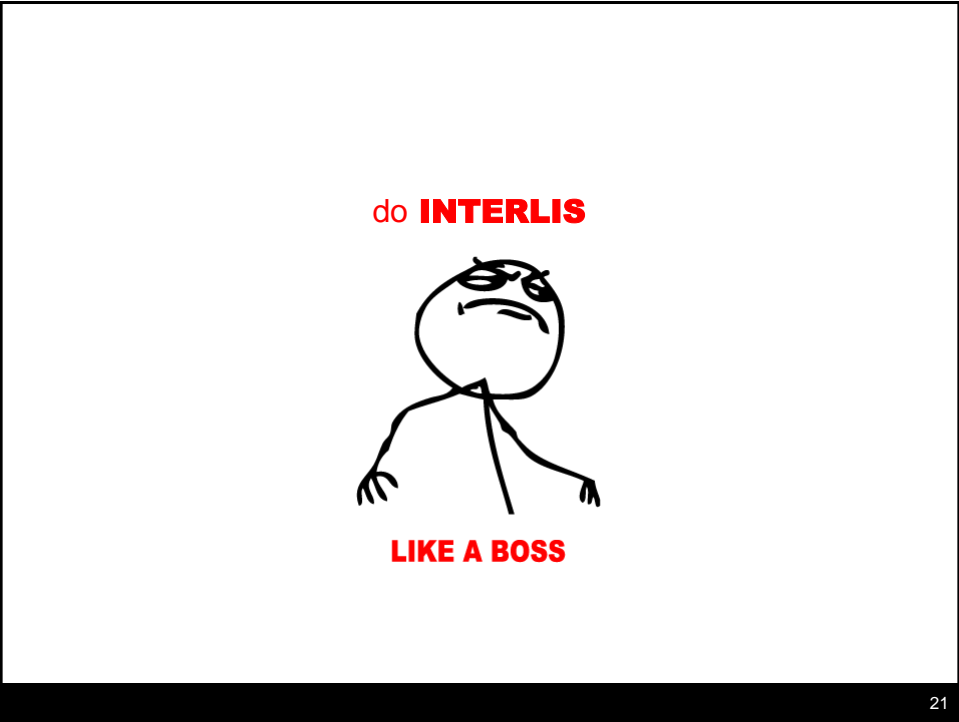
...auch für die Nutzung **modellnaher**
Datenprodukte ist die Modelldokumentation
sehr nützlich! → Metadaten


20

Die vollständige, saubere, verständliche Datenmodelldokumentation bringt auch für Nutzer von nicht-modellkonformen Datenprodukten einen Nutzen.


Wenn darauf geachtet wird, dass nutzerorientierte Datenprodukte «so modellnah wie möglich» ausgestaltet werden, also beispielsweise die Objekte und ihre Attribute gleich heißen wie im Datenmodell und möglichst die gleiche Objektstruktur vorhanden ist, dann sind diese Daten zusammen mit der Modelldokumentation besser verständlich und besser nutzbar.


Darum ist die Datenmodelldokumentation so wichtig.





kantonglarus 


Kanton Glarus
Bau und Umwelt
Raumentwicklung und Geoinformation

 Kirchstrasse 2
CH-8750 Glarus

 +41-55-646-6436

 geoinformation@gl.ch

 <http://www.geo.gl.ch>

 @GL_Geoportal

