

Objekt-Relationale Datenbanken zur Verwaltung von ATKIS-Daten

Carsten Kleiner, Udo W. Lipeck

Universität Hannover
Institut für Informatik
FG Datenbanken und Informationssysteme (dbis)
30159 Hannover

26.09.2000

Überblick

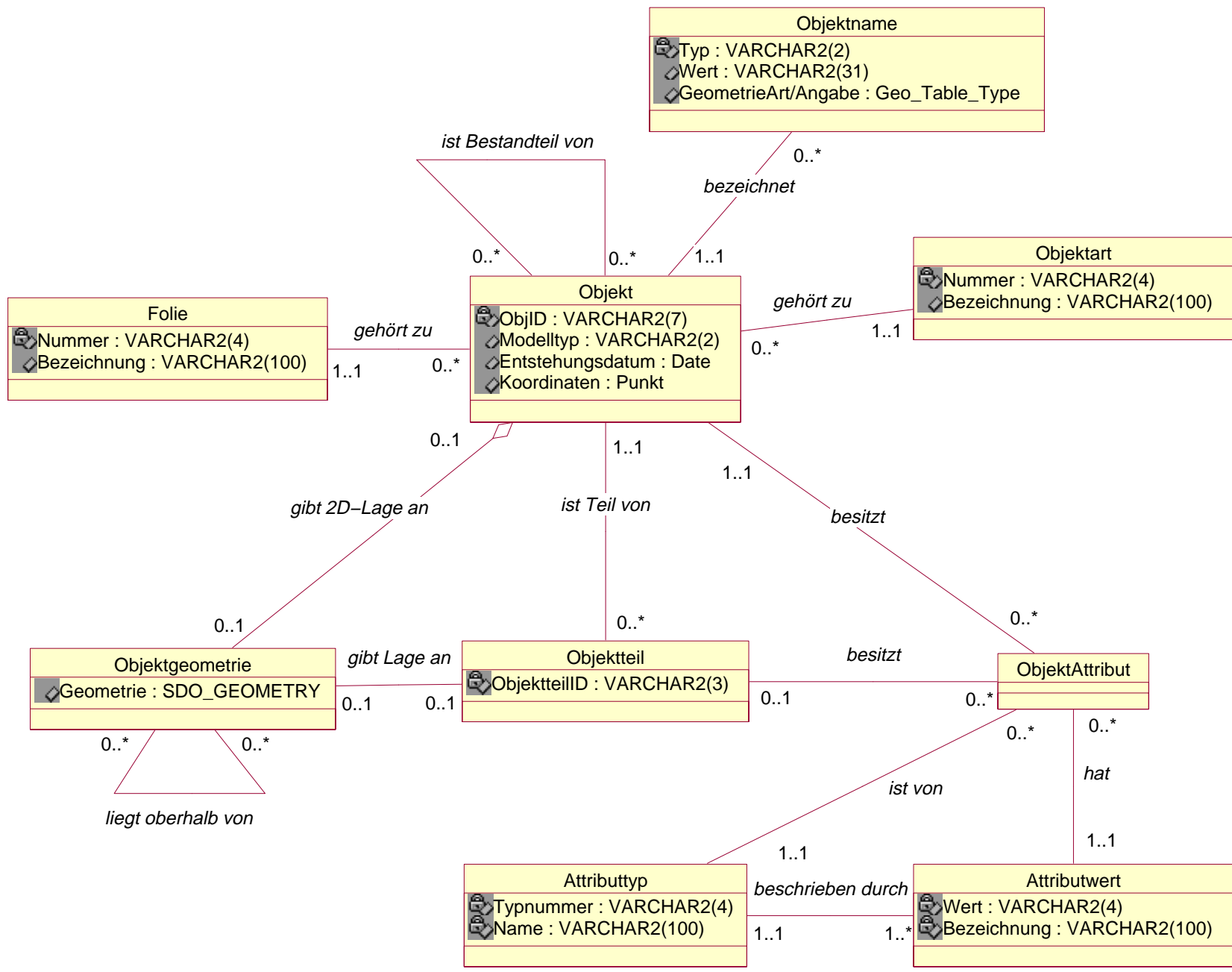
Überblick über den Vortrag:

1. Konzeptionelles Datenmodell für ATKIS-DLM
2. Datenimport aus EDBS-Dateien
3. Vorteile der Verwendung eines DBMS
4. Tools, Erweiterungen und allgemeiner Ausblick

Motivation

- Moderne GIS verwenden Datenbanksysteme zur Speicherung von Geo-Daten
- Integration von Geo-Daten mit anderen meist in DBMS gehaltenen Sachdaten
- Elegante und naheliegende Modellierung durch Integration komplexer Datentypen
- Einfache, effiziente und flexible Datenauswahl und -analyse
- Weitere Vorteile eines DBMS erhält man automatisch: Mehrbenutzerfähigkeit, Transaktionskontrolle, Backup, Recovery
- Gute Basis für Interoperabilität

Konzeptionelles Datenmodell



Relationenschema (1)

Relationen für **ATKIS-Standardkatalog**:

Layer (LayerNo:VARCHAR2(3), Name:VARCHAR2(100))

ObjectKind (ObjKindNo:VARCHAR2(4), Name:VARCHAR2(100))

AttributeType (AttTypeNo:VARCHAR2(4), Name:VARCHAR2(100))

AttributeValue (TypeNo→AttributeType, Value:VARCHAR2(7),
Name:VARCHAR2(100))

Relationenschema (2)

Relationen für **konkreten ATKIS-DLM-Datensatz**:

ATKISObject (ObjNo:VARCHAR2(7), LayerNo→*Layer*,
ObjectKindNo→*ObjectKind*, CoordinateX:NUMBER(9),
CoordinateY:NUMBER(9), Actual:VARCHAR2(2), ModType:VARCHAR2(2),
BuildDate:VARCHAR2(6))

ObjectGeometry (ObjectNo→*ATKISObject*, ObjectPartNo:VARCHAR2(3),
Geometry:SDO_GEOMETRY)

Overpass (TopNo→*ATKISObject*, TopPartNo:VARCHAR2(3),
DownNo→*ATKISObject*, DownPartNo:VARCHAR2(3))

ComplexObject (BigNo→*ATKISObject*, SmallNo→*ATKISObject*)

Relationenschema (3)

ObjectName (ObjectNo → *ATKISObject*, TypeNo:VARCHAR2(2),
Value:VARCHAR2(31), Geo:GEO_TABLE_TYPE)

ObjectAttribute (ObjectNo → *ATKISObject*, ObjectPartNo:VARCHAR2(3),
Type → *AttributeType*, Value:VARCHAR2(7))

Datenimport (1)

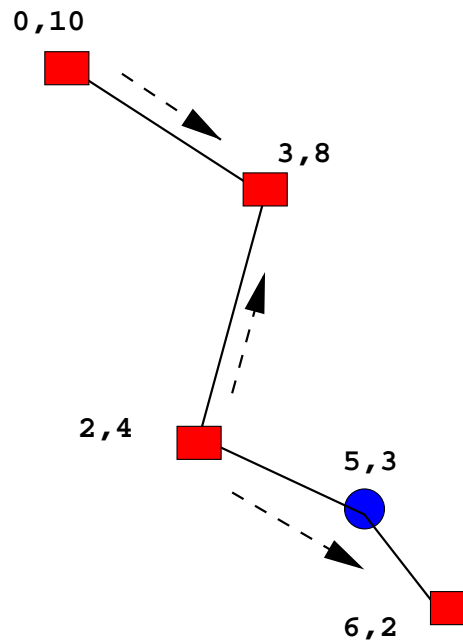
- Importroutine für **EDBS-Dateien** implementiert
- zunächst Erstellung und Import der Daten in Katalogrelationen
- **Temporäres Datenbankschema** zur (nahezu) direkten Übertragung der Daten aus EDBS-Datei in die Datenbank
- Aufbereitung der Daten zur Transformation in das endgültige Schema
- Viele Relationen können durch **einfache Auswahlanweisungen** gefüllt werden
- Wesentliche Ausnahme: **Geometrien** zu einzelnen Landschaftsobjekten **müssen rekonstruiert werden** !

Datenimport (2)

⇒ Verwende Algorithmus zur **Linienverfolgung**:

1. Suche Linie mit dem am weitesten links liegenden Endpunkt
2. Markiere diese Linie als Startlinie ($\text{myNo} := 0$)
3. $n := 1$;
4. Suche Linie mit Punkt == Zielpunkt der vorherigen Linie bzgl. der verfolgten Richtung
5. Falls vorhanden:
 - (a) Numeriere diese Linie mit n ($\text{myNo} := n$); trage Richtung ein
 - (b) $n++$;
 - (c) Weiter bei 4.
6. $n := -1$;
7. Weiter bei 4. mit $n--$ statt $n++$ bei 5b (andere Richtung von Startlinie aus)

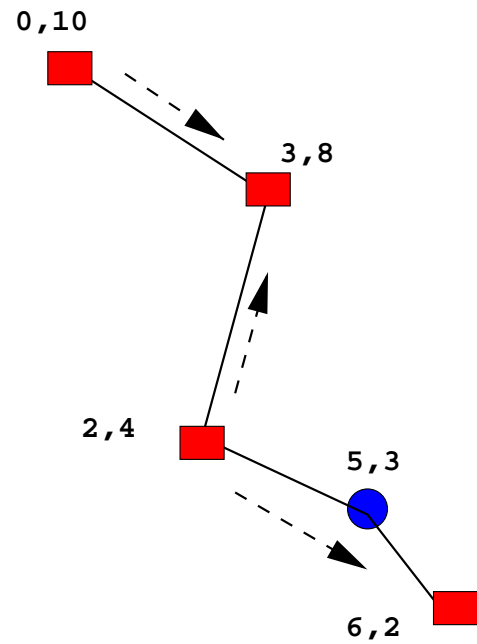
Datenimport (3)



- **linienförmige Objekte:**

L.Nr	O.Nr	BegX	BegY	EndX	EndY	Race	Dir	MyNo	P.Nr
10	17	0	10	3	8	11			
8	17	2	4	3	8	11			
13	17	2	4	6	2	15			

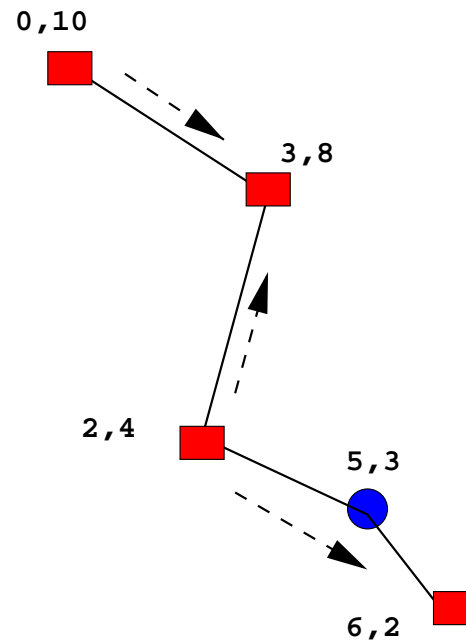
Datenimport (4)



- **linienförmige Objekte:**

L.Nr	O.Nr	BegX	BegY	EndX	EndY	Race	Dir	MyNo	P.Nr
10	17	0	10	3	8	11	0	0	0
8	17	2	4	3	8	11			
13	17	2	4	6	2	15			

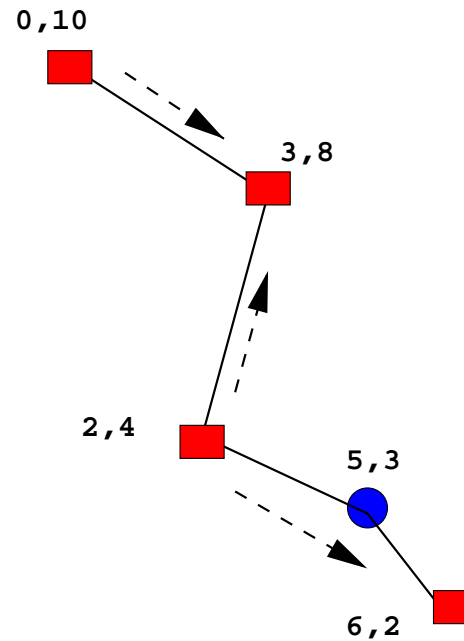
Datenimport (5)



- **linienförmige Objekte:**

L.Nr	O.Nr	BegX	BegY	EndX	EndY	Race	Dir	MyNo	P.Nr
10	17	0	10	3	8	11	0	0	0
8	17	2	4	3	8	11	1	1	0
13	17	2	4	6	2	15			

Datenimport (6)



- **linienförmige Objekte:**

L.Nr	O.Nr	BegX	BegY	EndX	EndY	Race	Dir	MyNo	P.Nr
10	17	0	10	3	8	11	0	0	0
8	17	2	4	3	8	11	1	1	0
13	17	2	4	6	2	15	0	2	0

Vorteile des DBMS (1)

- **Integrierte Datenhaltung** und -verwaltung mit Sachdaten
- Verfügbarkeit kommerzieller Werkzeuge für Verwaltung und Verwendung der Daten
- **Interoperabilität** wird begünstigt
- Möglichkeit zur **Zugriffsoptimierung**, z. B. durch benutzer-definierte Indexe
- Effiziente und **flexible Datenauswahlen** und -analysen (→ Beispiele folgen)

Vorteile des DBMS (2)

Wieviele Objekte gibt es pro Objektart ?

```
select count(o.objno), name
from   object_race r , objects o
where  o.objectraceno = r.objraceno
group  by name;
```

Ergebnis der Anfrage (1)

COUNT(O.OBJNO) NAME

3	Absetzbecken
171	Ackerland
1	Bahnhofsanlage
1	Bahnstrecke
4	Baum
28	Baumreihe
8	Binnensee
14	Brücke
7	Denkmal
6	Fahrbahn
46	Fläche
9	Fläche besonderer funktionaler Prägung
92	Fläche gemischter Nutzung
4	Freileitung
1	Freizeitanlage
3	Friedhof
4	Gartenland
7	Gehölz
2	Gemeindename
17	Graben
13	Grenze
10	Grünanlage
67	Grünland
2	Hecke

Ergebnis der Anfrage (2)

20	Industrie- und Gewerbefläche
4	Kläranlage
8	Mast
1	Moor
16	Ortslage
4	Platz
3	Pumpe
2	Schienenbahn
3	Schießstand
2	Schwimmbad
3	Schwimmbecken
10	Sportanlage
15	Sportplatz
1	Stollenmundloch
349	Straße
3	Straße (komplex)
3	Straßenkörper
12	Strom
1	Turm
1	Verkehrsknoten
93	Wald
2	Wasserschutzgebiet
367	Weg
255	Wohnbaufläche
6	Wohnplatzname

49 rows selected.

Vorteile des DBMS (3)

Welche Acker- oder Grünlandflächen gibt es in einem Umkreis von 300m um Waltringhausen ?

```
select o1.objno, o1.objecttraceno, g1.objectpartno
from   objects o1, object_geometries g1, object_geometries g2
where  g2.objectno = 'N00BN5L'
and    o1.objno = g1.objectno and o1.objecttraceno in ('4101','4102')
and    sdo_within_distance(g1.objectgeometry,g2.objectgeometry,
                           'distance = 30000') = 'TRUE';
-- N00BN5L ist Wohnplatzname von Waltringhausen
-- 30000cm = 300m
```

Vorteile des DBMS (4)

Welche Acker- oder Grünlandflächen gibt es in einem Umkreis von 300m um Waltringhausen ?

```
select o1.objno, o1.objectraceno, g1.objectpartno
from   objects o1, object_geometries g1, object_geometries g2
where  g2.objectno = 'N00BN5L'
and    o1.objno = g1.objectno and o1.objectraceno in ('4101','4102')
and    sdo_within_distance(g1.objectgeometry,g2.objectgeometry,
                           'distance = 30000') = 'TRUE';
-- N00BN5L ist Wohnplatzname von Waltringhausen
-- 30000cm = 300m
```

Ergebnis:

```
OBJNO  OBJE OBJECTP
-----
N016X3S 4101 001
N016X3X 4101 001
N016X93 4102 001
N016XAN 4101 001
N016XBJ 4101 001
N016XBL 4101 001
```

6 rows selected.

Vorteile des DBMS (5)

Wie groß ist die Fläche aller Moorgebiete im Untersuchungsraum ?

```
select sum(sdo_geom.sdo_area(g.objectgeometry,u.diminfo))/10000
from   object_geometries g, user_sdo_geom_metadata u, objects o
where  u.table_name = 'OBJECT_GEOMETRIES'
and    o.objno = g.objectno
and    o.objecttraceno = '4105';
```

-- Ausgabe in cm²; durch 10000 ergibt m²

Vorteile des DBMS (6)

Wie groß ist die Fläche aller Moorgebiete im Untersuchungsraum ?

```
select sum(sdo_geom.sdo_area(g.objectgeometry,u.diminfo))/10000
from   object_geometries g, user_sdo_geom_metadata u, objects o
where  u.table_name = 'OBJECT_GEOMETRIES'
and    o.objno = g.objectno
and    o.objecttraceno = '4105';
```

-- Ausgabe in cm²; durch 10000 ergibt m²

Ergebnis:

```
SUM(SDO_GEOM.SDO_AREA(G.OBJECTGEOMETRY,U.DIMINFO))/10000
-----
8884.1247
```

Vorteile des DBMS (7)

Welche Flüsse durchqueren ein Wasserschutzgebiet ?

```
select o1.objno, g1.objectpartno
from   objects o1, objects o2, object_geometries g1, object_geometries g2
where  o1.objno = g1.objectno and o2.objno = g2.objectno
and    o2.objectraceno='7311' and o1.objectraceno like '51%'
and    sdo_relate(g1.objectgeometry,g2.objectgeometry,
                  'mask=ANYINTERACT querytype=JOIN')='TRUE';
```

Ergebnis der Anfrage

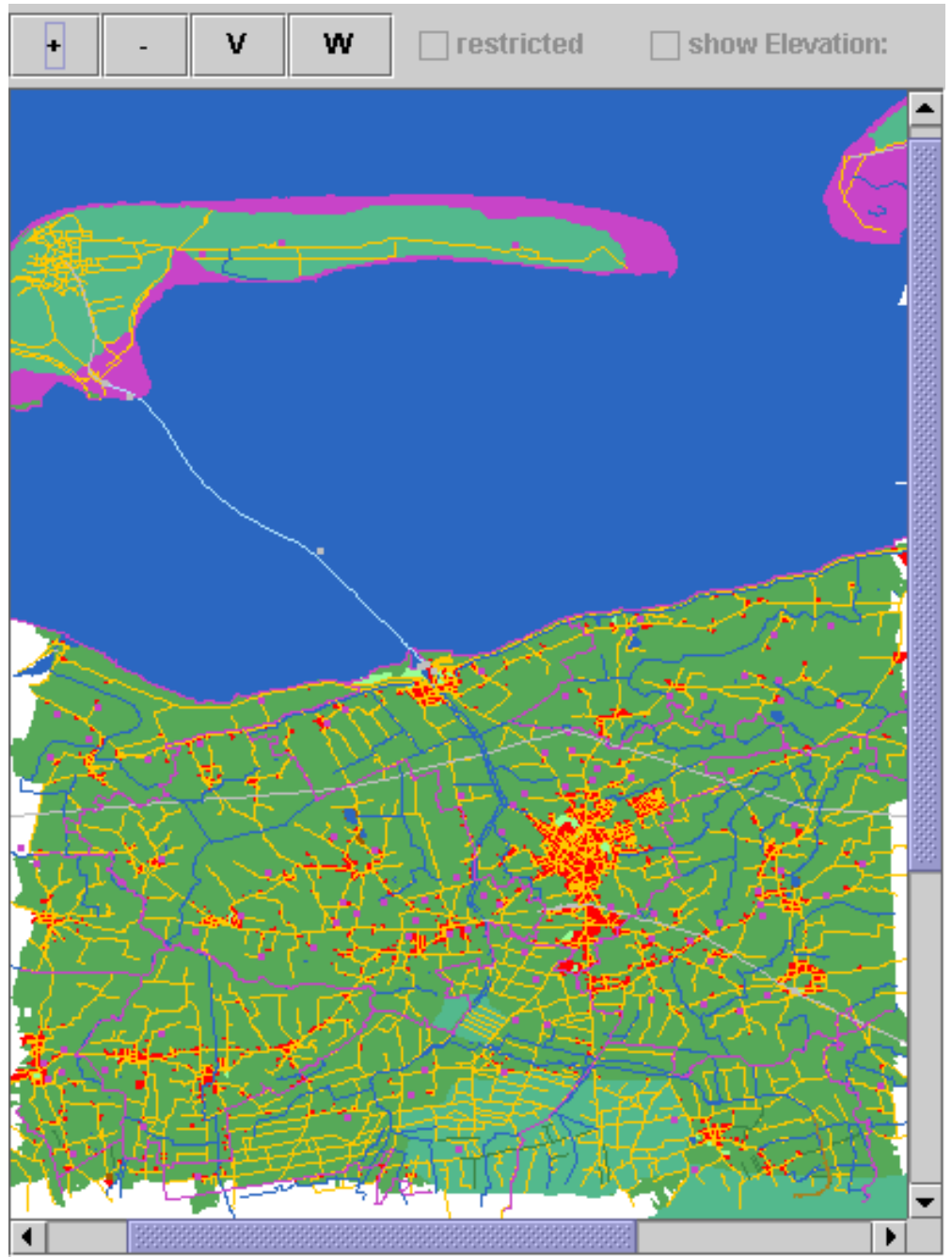
OBJNO	OBJECTP
N016VWS	003
N016VWS	002
N016VVQ	005
N016VRD	016
N016VRD	015
N016VRD	014
N016VRD	013
N016UXF	005
N016UXF	004
N016UXF	003
N016UXF	001
N016UVP	008
N00BNOC	001
N016UY4	001
N016X3T	004
N016X3T	005

16 rows selected.

Elapsed: 00:00:13.30

Visualisierungstool

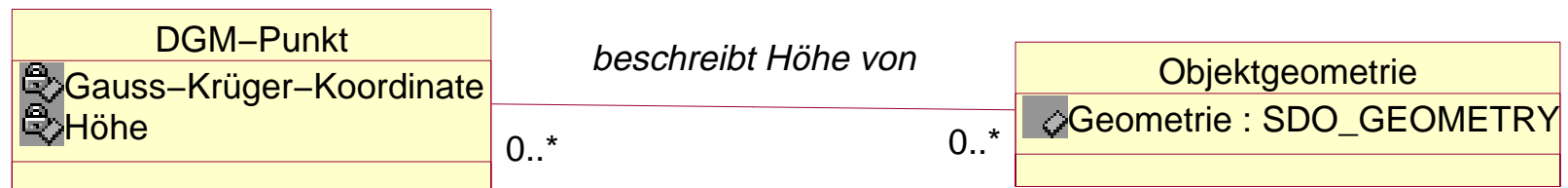
- Einfaches Tool zur [Visualisierung räumlicher Daten](#) aus der Datenbank
- Visualisierung des Ergebnisses einer SQL-Anfrage pro Layer
- Spezialfunktion zur Färbung ähnlich dem [ATKIS-Signaturenkatalog](#)
- Status: Prototyp





Integration von Höheninformationen

- DLM-Daten kommt noch grössere Bedeutung zu, wenn auch **Höheninformationen integriert** sind
- Als Objektart in Planung; bereits in anderem Format verfügbar: DGM
- **Integration von DGM-Daten** ins konzeptionelle Datenmodell:



- Beim physischen Entwurf ggf. Vorausberechnungen vorsehen, um Effizienz zu erhöhen

Ausblick

- Nach Entwurf geeigneter Indexstrukturen: effiziente Bearbeitung von **Kombinationsanfragen an DLM und DGM**
- Entwurf einer **XML-basierten Sprache** zum Austausch von ATKIS-Daten
- Verbesserung der Visualisierung durch weitergehende Integration des Signaturenkatalogs
- **Vereinfachung der Anfragen** an die Datenbank durch geeignete Benutzerschnittstellen