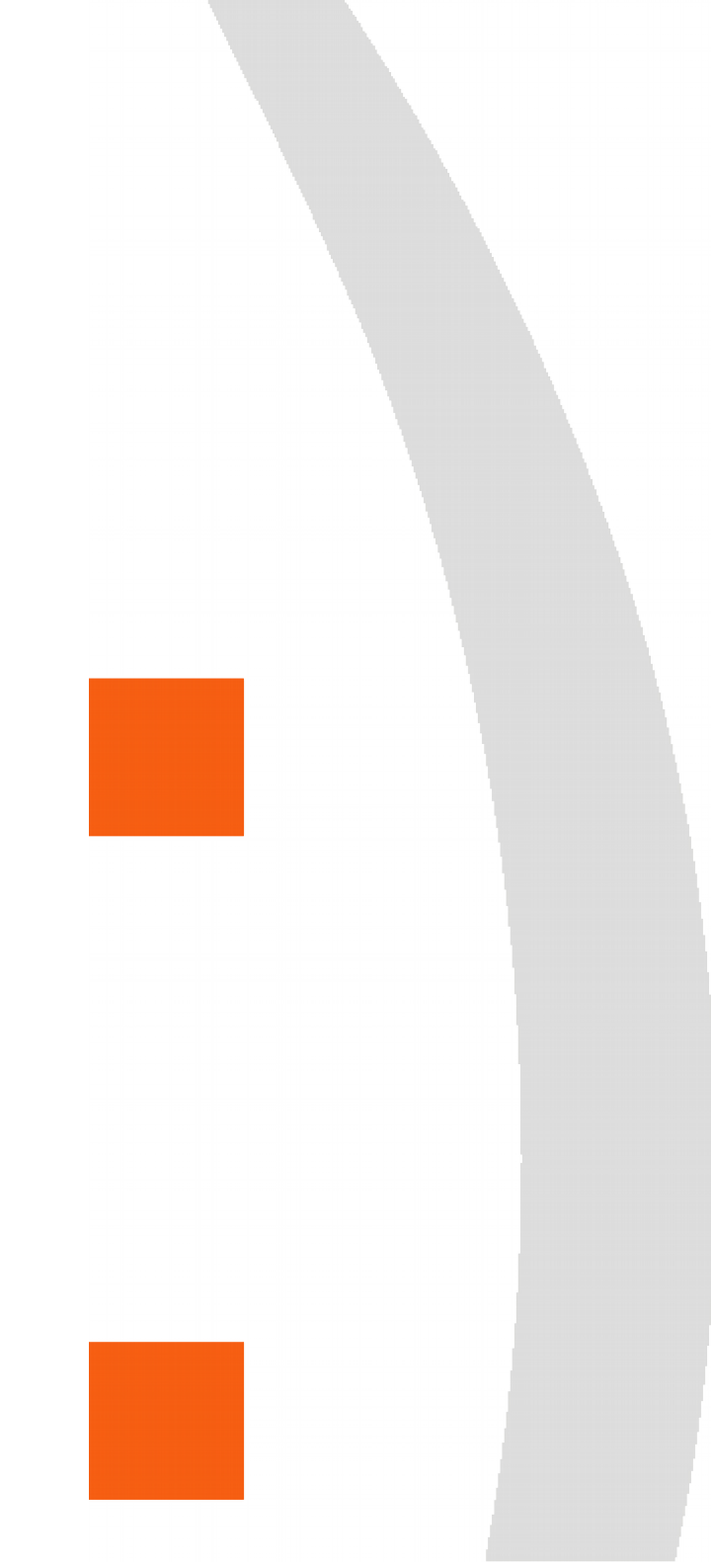




Studie einer alternativen Datenbankstruktur

Datenbankschnittstelle Version 3.0

Dipl.-Ing. Christoph Lippert
smile consult GmbH

Motivation für eine Studie

- Optimierung von Zugriffszeiten:
 - schreibender und lesender Zugriff sollte für weiter steigende Datenmengen performanter erfolgen
 - Metadaten-Recherche bei großen Datenmengen (zum Teil >80000 Datensätze) und speziell unter Oracle inperformant
- Optimierung von Datenbankgrößen
 - Datenvolumen der Datenbanken wird enorm groß (Fragestellung auch für Sicherungsstrategien und Datenaustausch relevant, vgl. Workshop-Diskussion)
- Datensicherheit
 - Vereinfachung der Datenbankstruktur für robustere Datenbankschnittstelle
 - Transaktionssicherheit
- Datensicherungsstrategien

Grundlegende Analysen und Betrachtungen

- Datenbankschnittstelle Version 2.0:
 - dynamisches Tabellenkonzept zur Speicherung beliebig strukturierter Daten (Peillinien, Fächerlot, etc., Raster/unstrukturiert, vermascht/unvermascht, mit oder ohne Attribute,...)
 - Strukturinformation des Datensatzes und Speicherort wird in den Metadaten abgelegt
 - jedes geometrische Objekt eines Datensatzes wird als ein Tabelleneintrag in einer Datentabelle gespeichert
 - für selektive Zugriffe auf die Daten werden diverse Spalten der Datentabellen indiziert

Grundlegende Analysen und Betrachtungen

Tabelle data2points (Basis-Tabellenname + points)

id	md_id	nr	x	y	z	status	spi	depthnumber	depthva	datared	datetime	cyclice	rawxnorth	rawyeast	rawlatitude
1	1	0	529685.99	6075004.97	1.705	1	0	161	1	0	2012-09-19 07...	0	6075004.97	529685.99	54.82130881
2	1	1	529855.03	6075145.61	1.205	1	0	290	1	0	2012-09-19 06...	0	6075145.61	529855.03	54.82256262
3	1	2	529612.79	6074869.85	1.405	1	0	252	1	0	2012-09-19 07...	0	6074869.85	529612.79	54.8200989

Tabelle data2elements (Basis-Tabellenname + elements)

id	md_id	nr	n0	n1	n2	n3	status	spi
1	3	0	4206	3992	3983	3983	128	110
2	3	1	3934	4206	3935	3935	128	110
3	3	2	3886	4230	3920	3920	128	110

Tabelle data2polygons (Basis-Tabellenname + polygons)

id	md_id	nr	poly_edge_nr	p0	p1	spi
1	3	0	0	427	963	8
2	3	0	1	963	1463	10
3	3	0	2	1463	2016	33

Punktnummern !

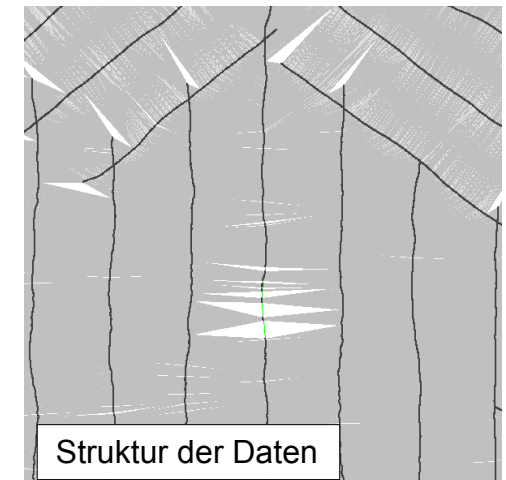


Tabelle data2polygondata (Basis-Tabellenname + polygondata)

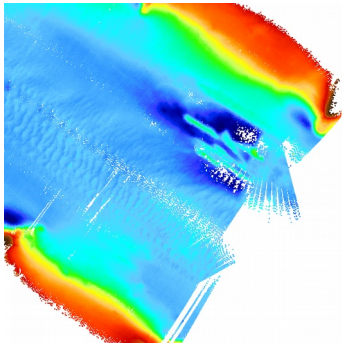
id	poly_id	md_id	verwinprofsha	profileguid	authority	projectname	profilenumber	profilevalid	profileisregula	profileforconsc	profileiscontrol	scale
1	0	1	3.11.13	93b21fa9-4...	BSH	0312	1	1	0	1	0	20000
2	1	1	3.11.13	e5b6988e-b...	BSH	0312	2	1	0	1	0	20000
3	2	1	3.11.13	16fcd52-c0...	BSH	0312	3	1	0	1	0	20000

Tabelle data2tree (Basis-Tabellenname + tree)

id	md_id	spi	depth	minx	miny	maxx	maxy	pointcount	edgecount	polygonedgecount
2	3	0	6	529694.5585540001	6075010.048557	529723.481446	6075038.911443001	356	0	0
3	3	2	7	529690.578996	6075027.6091665	529710.6610040001	6075044.2808335	96	0	0
4	3	3								

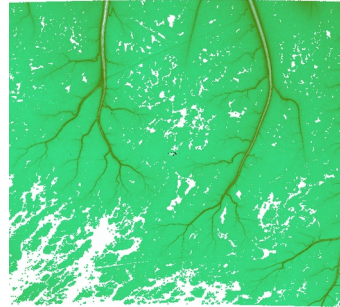
Datenbankschnittstelle 2.0: umfangreiche Tabellenstrukturen speziell zur Ablage triangulierter, unstrukturierter Daten erforderlich

Grundlegende Analysen und Betrachtungen



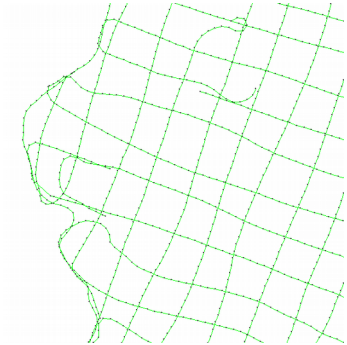
Fächerecholot-Daten

data?points
data?tree



Lidar-Daten mit Attributen

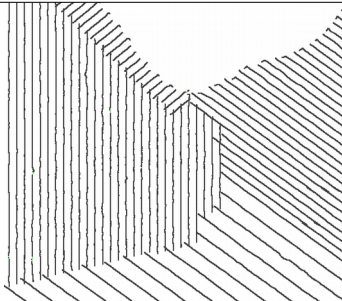
data?points
data?tree



Kreuzprofile

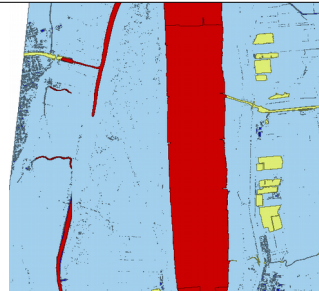
data?points
data?polygons
data?polygondata
data?tree

Datenbankschnittstelle 2.0: Beispiele für erforderliche Tabellen zur Abbildung unterschiedlicher Datenarten/Modelltypen



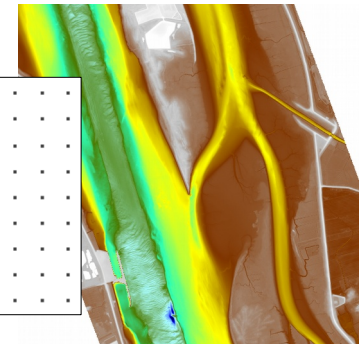
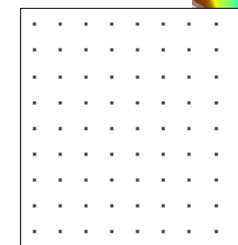
Linienpeilungen mit Attributen an den Vermessungspunkten und den Peillinien

data?points
data?polygons
data?polygondata
data?tree



Datenquellenkarte (Polygonflächen mit Attributen)

data?points
data?polygons
data?polygondata
data?tree



1m-Raster-Modell

data?points

Grundlegende Analysen und Betrachtungen

 data30polygondata0.MYD	 data30polygondata0.MYI	 data30polygons.frm	 data30polygons.MYD
 data30polygons.MYI	 data30tree.frm	 data30tree.MYD	 data30tree.MYI
 data32elements.frm	 data32elements.MYD	 data32elements.MYI	 data32points.frm
 data32points.MYD	 data32points.MYI	 data32polygondata0.frm	 data32polygondata0.MYD

Datenbankschnittstelle 2.0: dynamisches Tabellenkonzept resultiert i.d.R. in einer Vielzahl von verwalteten Tabellen innerhalb einer Datenbank

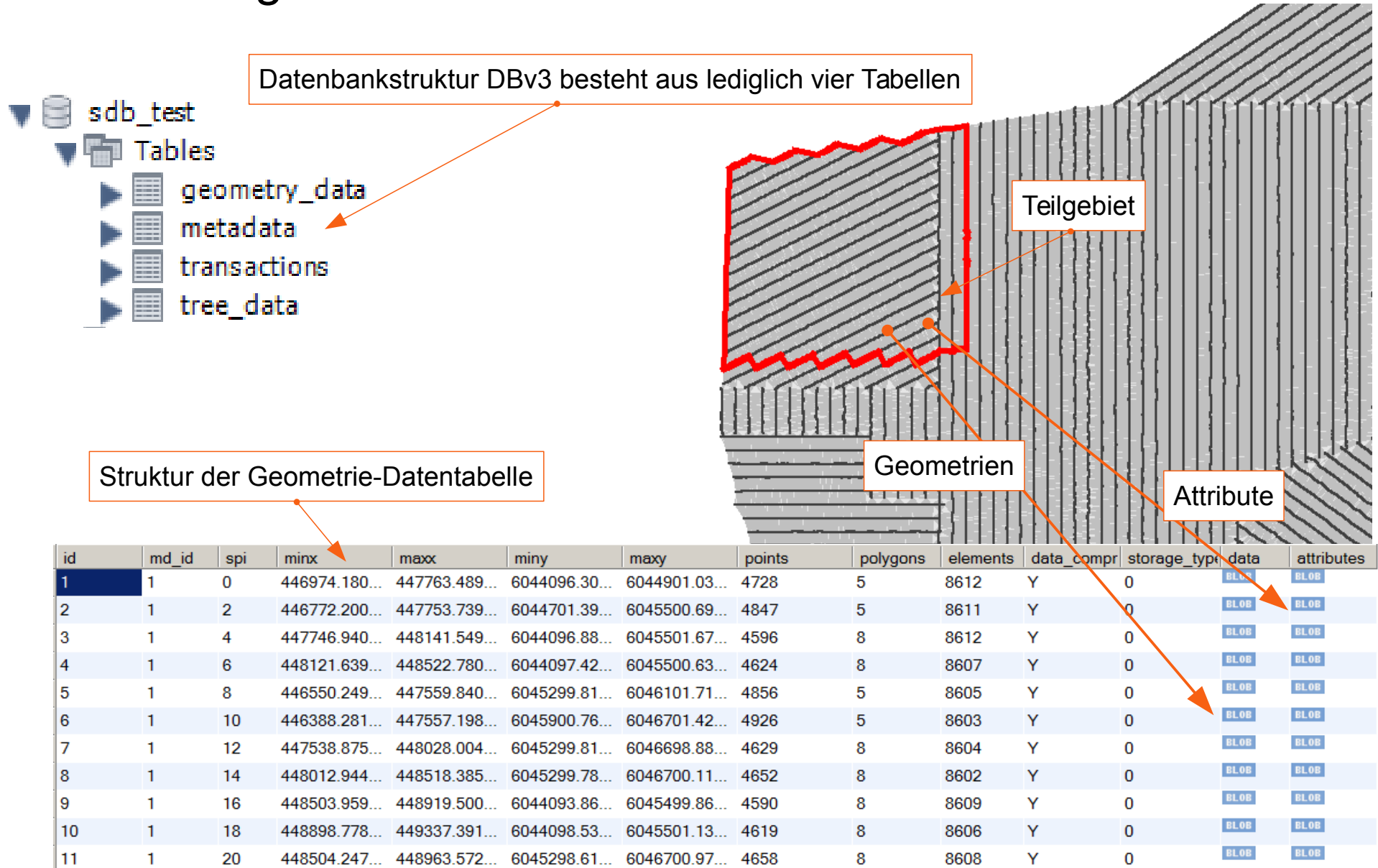
 data34points.MYI	 data34polygondata0.frm	 data34polygondata0.MYD	 data34polygondata0.MYI
 data34polygons.frm	 data34polygons.MYD	 data34polygons.MYI	 data34tree.frm
 data34tree.MYD	 data34tree.MYI	 data35elements.frm	 data35elements.MYD
 data35elements.MYI	 data35points.frm	 data35points.MYD	 data35points.MYI
 data35polygondata0.frm	 data35polygondata0.MYD	 data35polygondata0.MYI	 data35polygons.frm
 data35polygons.MYD	 data35polygons.MYI	 data35tree.frm	 data35tree.MYD
 data35tree.MYI	 data36elements.frm	 data36elements.MYD	 data36elements.MYI
 data36points.frm	 data36points.MYD	 data36points.MYI	 data36polygondata0.frm
 data36polygondata0.MYD	 data36polygondata0.MYI	 data36polygons.frm	 data36polygons.MYD
 data36polygons.MYI	 data36tree.frm	 data36tree.MYD	 data36tree.MYI
 data37elements.frm	 data37elements.MYD	 data37elements.MYI	 data37points.frm
 data37points.MYD	 data37points.MYI	 data37polygondata0.frm	 data37polygondata0.MYD
 data37polygondata0.MYI	 data37polygons.frm	 data37polygons.MYD	 data37polygons.MYI
 data37tree.frm	 data37tree.MYD	 data37tree.MYI	 db.opt
 metadataasciidata.frm	 metadataasciidata.MYD	 metadataasciidata.MYI	 metadatabinarydata.frm
 metadatabinarydata.MYD	 metadatabinarydata.MYI	 metadatacore.frm	 metadatacore.MYD
 metadatacore.MYI	 metadatadatadescriptor.frm	 metadatadatadescriptor.MYD	 metadatadatadescriptor.MYI
 metadatadatetimedata.frm	 metadatadatetimedata.MYD	 metadatadatetimedata.MYI	 metadatadoubledata.frm
 metadatadoubledata.MYD	 metadatadoubledata.MYI	 metadataaintdata.frm	 metadataaintdata.MYD
 metadataaintdata.MYI	 transactions.frm	 transactions.MYD	 transactions.MYI

Grundgedanken für eine alternative Datenbankstruktur

- Tabellenstruktur der Datenbankschnittstelle soll signifikant vereinfacht werden:
 - DBv2 : dynamische Tabellenverwaltung mit statischem Inhalt
 - DBv3 : statische, einfache Tabellenstruktur mit dynamischen Inhalt

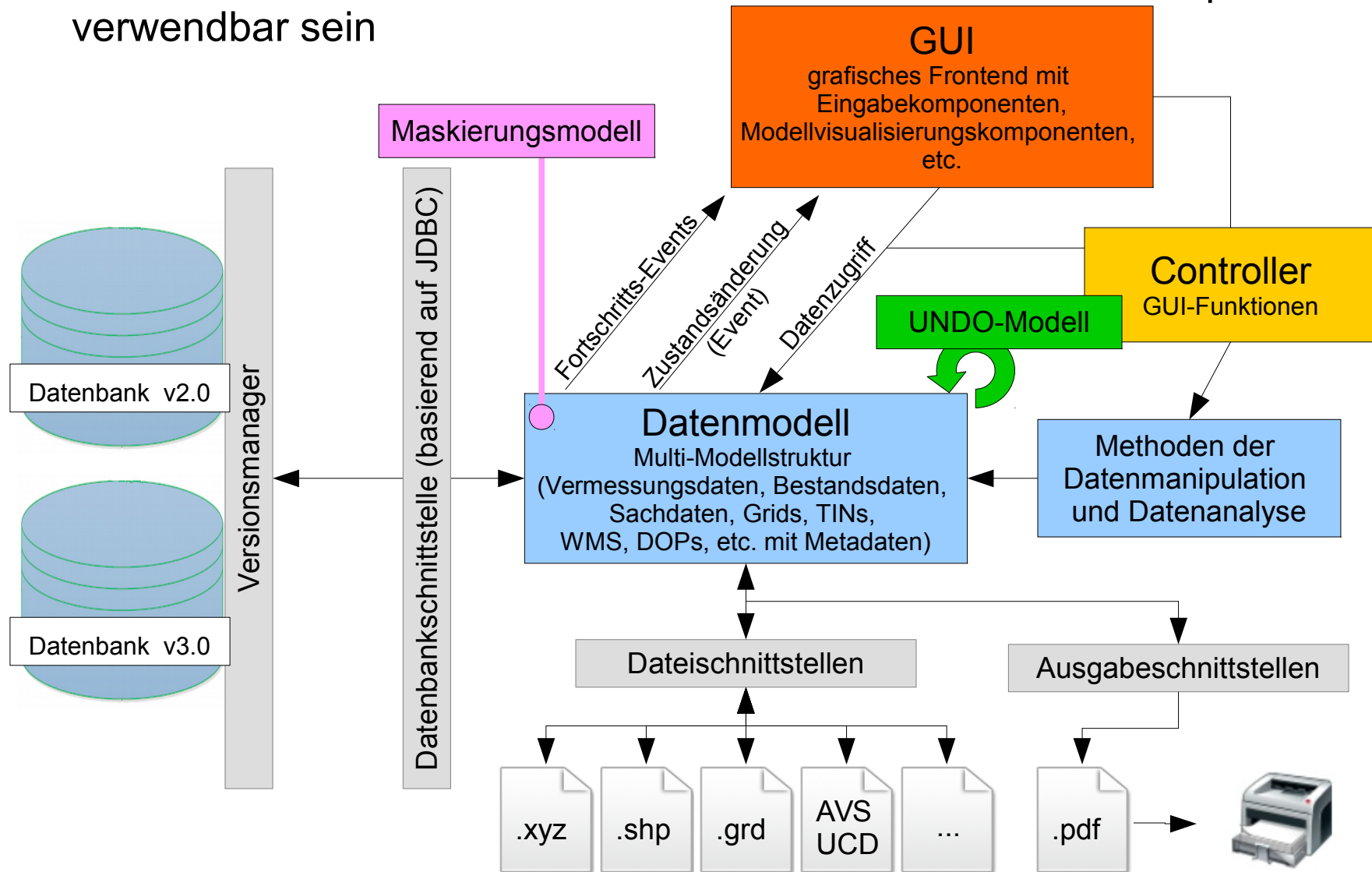
besonderer Vorteil: Speicherort des Datensatzes muss nicht mehr in Metadaten vorgehalten werden
- die Abbildung einzelner geometrischer Objekte in jeweils einer Tabellenzeile soll vermieden werden:
 - DBv3 : Speicherung von Teilgebieten / Teildatensätzen als binär-codierte, komprimierte Datenpakete innerhalb einer Tabellenzeile
 - Teilgebiet stellt kleinste Zugriffseinheit auf einen Datensatz dar
 - Aufteilung in Teilgebiet stellt sicher, dass einzelne Datenpakete nicht zu umfangreich werden und das ein selektiver Zugriff auf Teildatensätze (z.B. für eine Datenbank-Interpolation) möglich ist

Grundgedanken für die alternative Datenbankstruktur



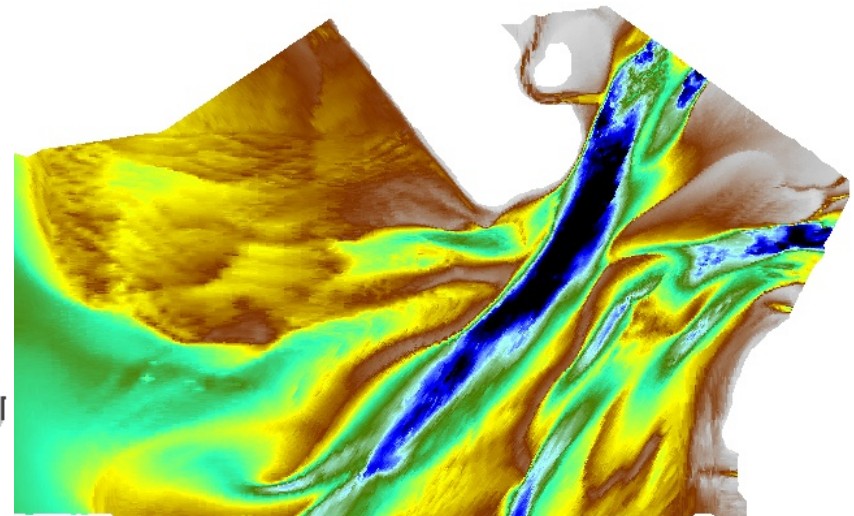
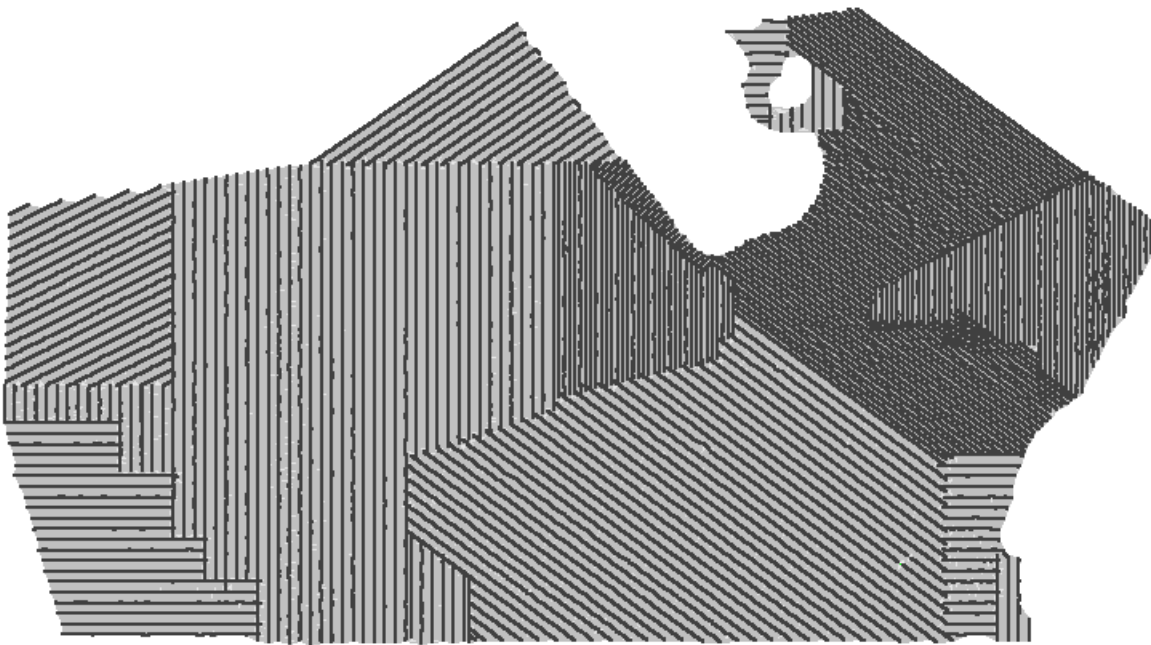
Grundgedanken für die alternative Datenbankstruktur

- Datenbankschnittstellen unterschiedlicher Versionen müssen parallel verwendbar sein



Anwendungsbeispiel

- SDB-Aufgabe „09132“
 - 2205094 Vermessungspunkte mit je 53 Attributen
 - 440 Peillinien mit je 64 Attributen
 - 4405448 Dreieckselemente
 - Dateigröße NetCDF: 792 MB



Anwendungsbeispiel

- SDB-Aufgabe „09132“ in Datenbank schreiben (Beispiel für PostgreSQL)

Datenbank	Zeit [sec]	Anmerkung
DB v2, PostgreSQL	1583.2	
DB v3, PostgreSQL	56.2	DB-Daten-Write: 9.0 sec

Faktor (v2/v3) = **28,2**

Anzahl Teilgebiete = Zeilen in Datentabelle

Gesamtgröße der Geometrie- und Attributdaten (komprimiert)

```
R-Tree, Depth: 11, Leafs: 522
Datensatzgroesse      257.6 MB, 522 Teilgebiete
Datensatzgroesse/TG   0.5 MB, 522 Teilgebiete
Zeit, RTree           15.2 sec
Zeit, Teilgitter      31.6 sec (236.4 sec)
Zeit, DB write        9.0 sec
Data stored in 56.2 sec
```

Zeit für Generierung der Teilgebiete (ohne Parallelisierung)

Zeit für Generierung der Teilgebiete (parallelisiert)

Datenbankschnittstelle 3.0,
Analysekennzahlen

Anwendungsbeispiel

- SDB-Aufgabe „09132“ aus Datenbank extrahieren (PostgreSQL)

Datenbank	Zeit [sec]	Anmerkung
DB v2, PostgreSQL	403.6	
DB v3, PostgreSQL	74.8	

Faktor (v2/v3) = **5,4**

- Datenbankgrößen (PostgreSQL)

Datenbank	Größe [MB]	Anmerkung
DB v2, PostgreSQL	1444	
DB v3, PostgreSQL	275	

(zum Vergleich: Dateigröße NetCDF: 792 MB)

Faktor (v2/v3) = **5,2**

Zwischenfazit der Studie

- alternative Datenbankstruktur ermöglicht
 - signifikant performantere Zugriffszeiten (zum Teil Beschleunigung Faktor >10) sowohl für Schreib- als auch für Leseoperationen
 - signifikant kleinere Datenbanken im Hinblick auf den physikalischen Speicherverbrauch
- parallele Verwendung von Datenbanken unterschiedlicher Versionen für
 - einfache Vergleiche der Ansätze
 - einen schrittweisen (möglichen) Umstieg
- Studienergebnisse bislang nur für MySQL, PostgreSQL, ähnliche Resultate werden jedoch auch für Oracle erwartet



Kontakt

Dipl.-Ing. Christoph Lippert

post: smile consult GmbH
Vahrenwalder Straße 4
30165 Hannover

tel: 0511 / 543617-43

fax: 0511 / 543617-66

mail: lippert@smileconsult.de

web: <http://www.smileconsult.de>

