



# Integration verteilter Datenquellen in GIS-Datenbanken

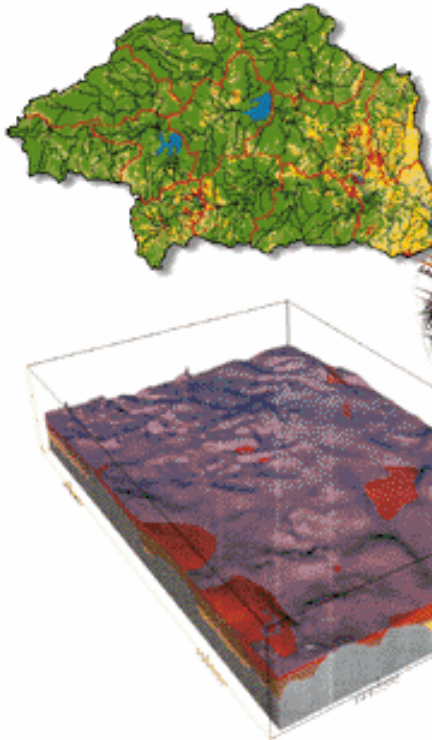
Seminar „Verteilung und Integration von Verkehrsdaten“

Am IPD – Lehrstuhl für Systeme der  
Informationsverwaltung

Sommersemester 2004

Christian Hennings

# Geographic Information System (GIS)



## Ziel:

- Datenintegration (nach außen hin einheitlich abrufbar)
  - Ausnutzen von angebotenen Features
  - Bereitstellung von zusätzlichen Features, die einzelne Quelle nicht liefern
- unterstützen



# Gliederung



1. Allgemeines zum Mediator-konzept von GIS
2. notwendige Ergänzungen (ICA+virtuelle Operationen)
3. Ablauf einer Anfrage an GIS
4. Zusammenfügen der einzelnen Ergebnisse
5. Ontologiebasierter Ansatz



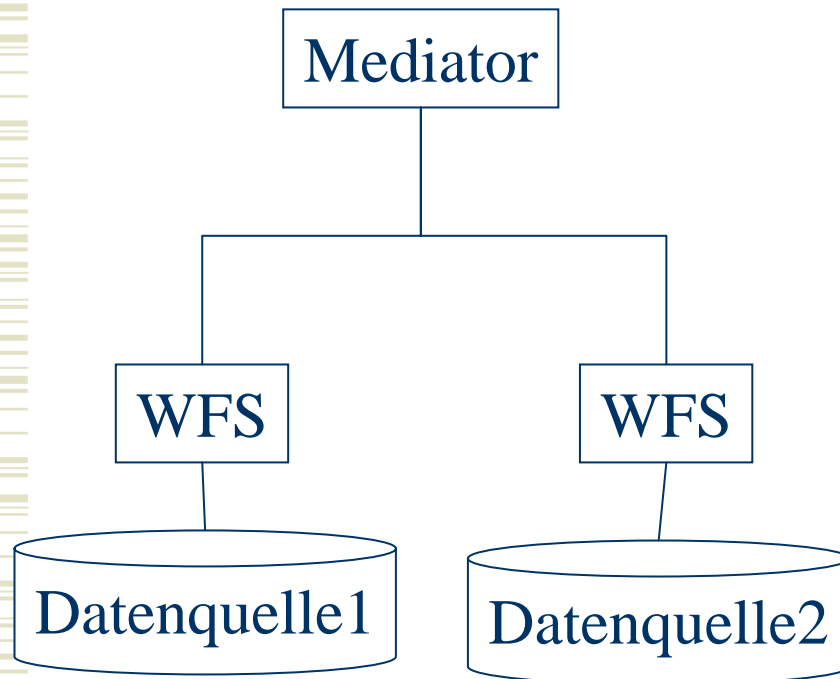
---

# Wrapper+Mediator

---

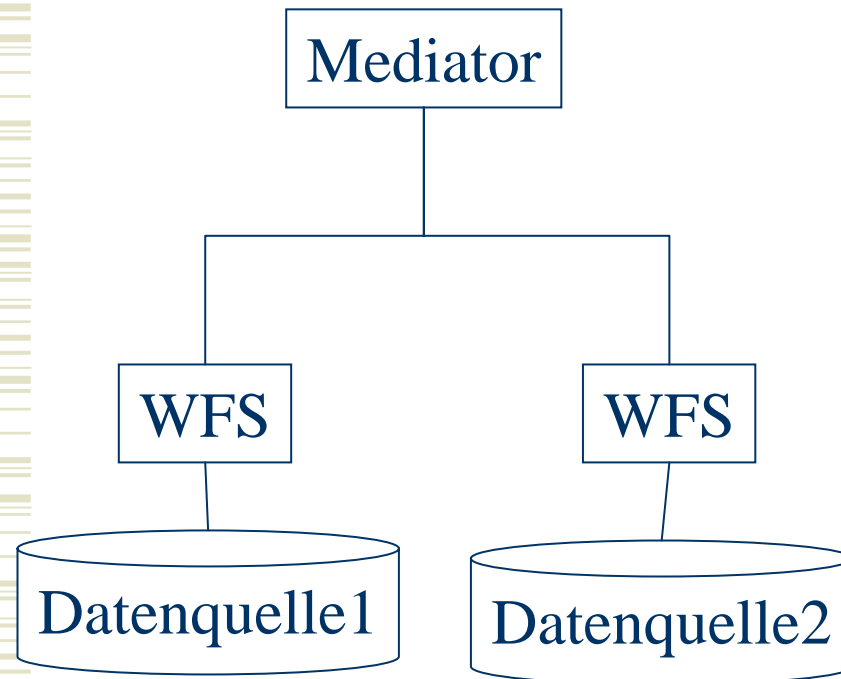
- ◆ Lösung allgemein mit Wrappern und einem Mediator
- ◆ kurze Wiederholung: Wrapper transformieren die Ausgaben einer Quelle entsprechend eines externen Standards und umgekehrt.

# Lösung bei GIS



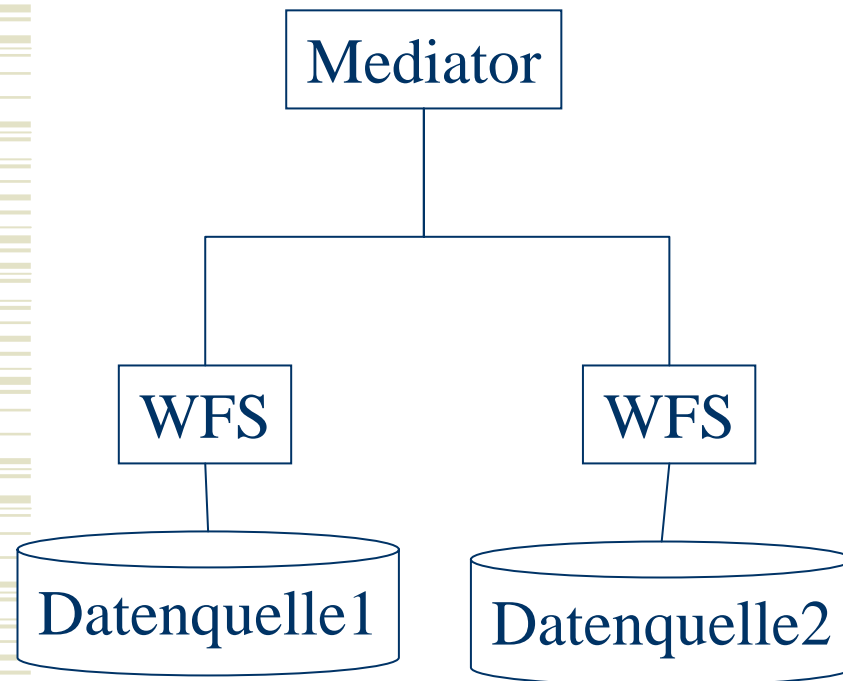
- ◆ 3-Schichten Client-Server-Architektur (Mediator greift als Client auf Datenserver zu)
- ◆ Der Wrapper ist als Web Feature Service umgesetzt

# Aufgaben des Mediators



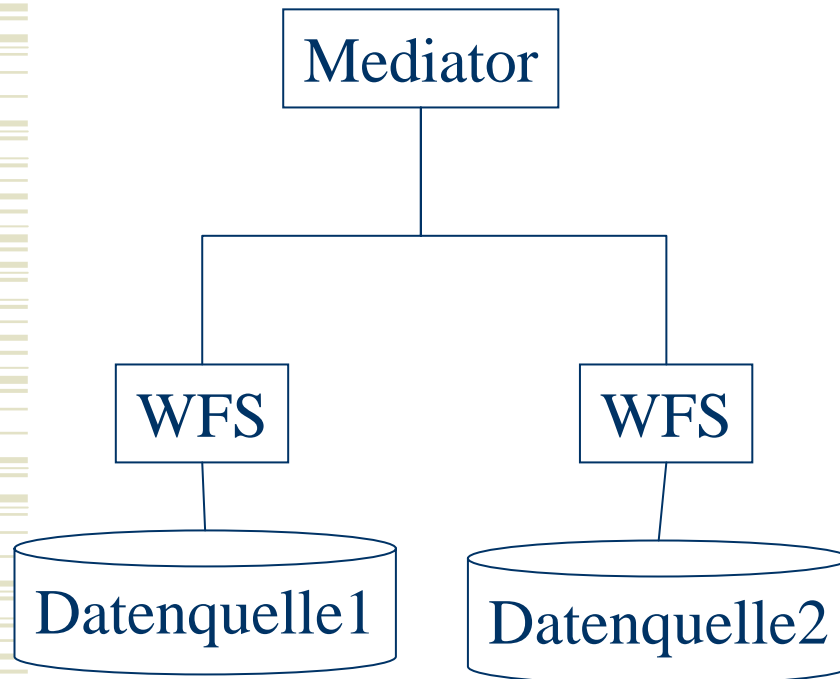
- ◆ Nimmt eine Analyse der Anfragen vor (u.a. zur Beseitigung von Konflikten)
- ◆ Durchführung einer Anfrage-Optimierung
- ◆ Aufsplitten einer Anfrage in mehrere Anfragen, die mittels WFS an die Datenquellen weitergereicht werden
- ◆ Späteres Zusammenfügen der Einzelergebnisse zum Gesamtergebnis

# Aufgaben von Web Feature Service (WFS)



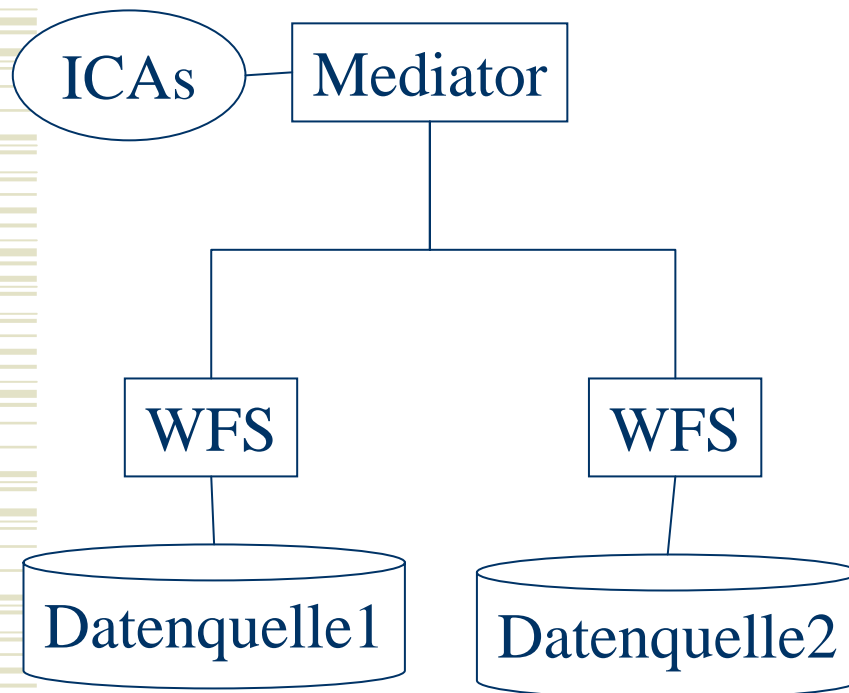
- ◆ Übersetzung der vom Mediator verwendeten Sprache in die Anfragesprache seiner Datenquelle+Transformation der Ergebnisse in umgekehrter Richtung
- ◆ beschreibt dem Mediator welche Features und Operationen er darauf anbietet
- ◆ beschreibt, wie darauf zugegriffen werden kann

# Aufgaben der Datenquellen



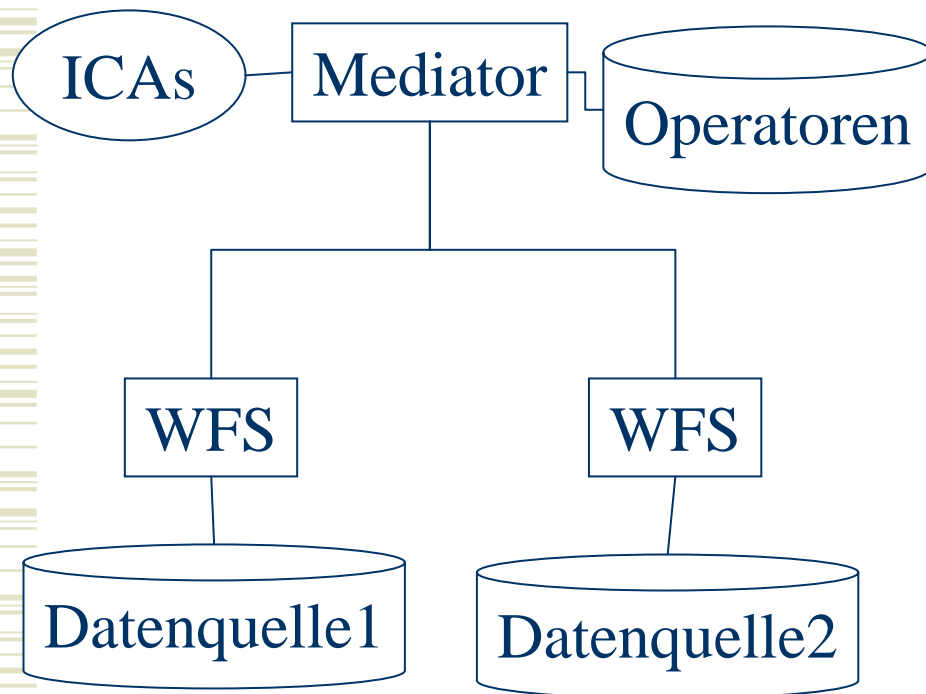
- ◆ Liefert die Daten
- ◆ führt gegebenenfalls bestimmte Operationen auf den Daten aus

# Ergänzungen – Inter-Schema Correspondence Assertions (ICA)



- ◆ Da der Mediator Konflikte lösen soll (z.B. Aggregationskonfl.: Kirche ist auch ein Gebäude), braucht der Mediator Regeln, die die Beziehungen zwischen den versch. Datenschemata abbilden.
- ◆ nötig bei der Analyse der Anfrage und dem Zusammenfügen der Ergebnisse

# Ergänzungen – Operators



- ◆ Ziel: Bereitstellung von Features, die einzelne (eventuell alle) Quellen nicht bereit stellen
- ◆ Beinhaltet Implementierungen von genau solchen Operatoren und von Operatoren, die sich erst aus den Regeln (ICA) ergeben



# Ablauf einer Anfrage an GIS



1. Analyse der Anfrage und Lösung von Konflikten durch den Mediator mit Hilfe der ICA-Regeln (Kirche->Gebäude)
2. Mediator erstellt einen Ablaufplan für die Anfrage:
  - Anfrage mittels WFS welche Features eine Quelle unterstützt
  - Aufteilung der Anfrage in Anfragen, die von den Quellen allein beantwortet werden können, und solche die einen oder mehrere Operatoren aus dem Vorrat benötigen



# Ablauf einer Anfrage an GIS



3. Wo nötig, wird für den Operator ein so genannter derived wrapper als weiterer WFS. (gleichwertig zu den anderen Quellen)
4. Sind die Ergebnisse aller benötigten Quellen eingetroffen, berechnet der Mediator schließlich das Gesamtergebnis



---

# Wie berechnet sich ein einheitliches Ergebnis?

---

Bzw. wie filtert man semantisch ähnliche Objekte?

Kirche ist auch Gebäude

Strasse und Street können Äquivalente sein.

Der Rinnstein ist ein Teil der Strasse

->Es ist nicht wünschenswert, eine Kirche und ein beliebiges Gebäude in einer Antwort zu erhalten, die an der selben Position stehen und offensichtlich korrespondieren, also das gleiche Gebäude beschreiben.



---

# Ontologiebasierterer Ansatz

---

unterschiedliche Quellen gebrauchen ähnliche  
Objekte der Realen Welt in einer  
verschiedenen Semantik.

→ Eine Ontologie aus  
Domänenontologie,  
Applikationsonthlogien und  
Abstraktionsregeln



---

# Domänenontologie

---

Enthält Definitionen für Konzepte, die  
Objekten aus der Realen Welt entsprechen.

Straßen, Gebäude, Wiesen



# Applikationsontologie



Beinhaltet, wie ein Konzept in einer Datenbasis benutzt wird.

## Abstraktionsregeln

Definieren die Beziehungen zwischen Domänen- und Applikationsontologie

# Umsetzung einer Ontologie in Prolog

Zunächst werden die Objektklassen taxonomiert:

```
taxon [SubClass, Class]
```

z.B. `taxon [wiese, gelaende]`

darauf aufbauend die subClass – Beziehung:

```
subClass [x, x].
```

```
subClass [x, z] :- (taxon [x, y]  
    && subClass [y, z]).
```

# Beziehung Domänen- und Applikationsontologie

Äquivalente Klassen:

```
referstoEquivalentClass [DomainClass,  
                          ApplClass].
```

Definition von Aggregatklassen:

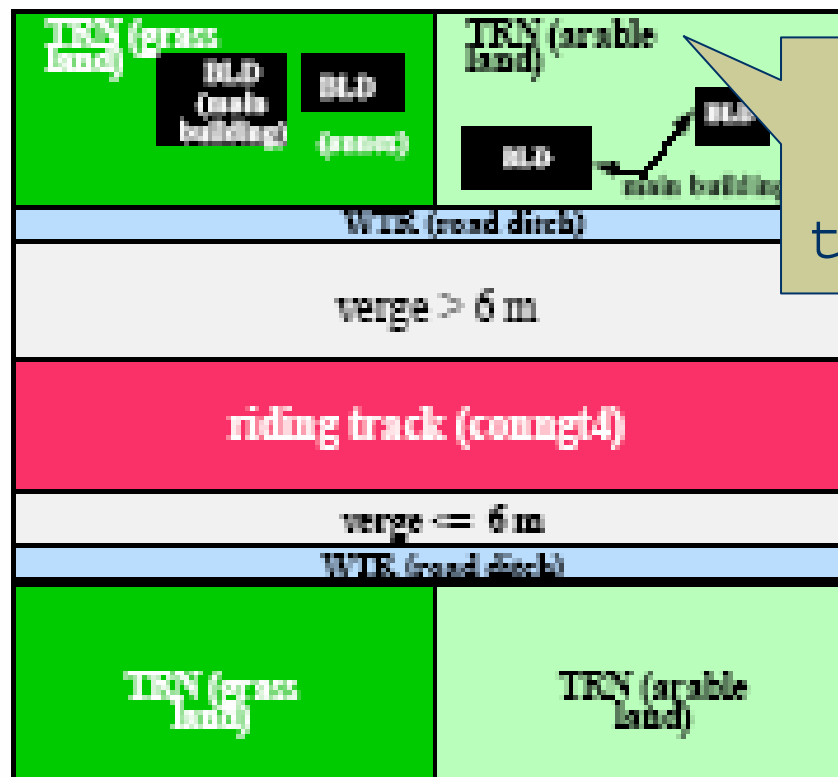
```
referstoAggregateClass [DomainClass,  
                        ApplClass].
```

# Beziehungen zwischen verschiedenen Objektklassen aus den Datensätzen

## Äquivalenz:

```
semanticEquivalentClass [App1Class1, App1Class2] :-  
    (refersToEquivalentClass [DomainClass,  
    App1Class1]  
    &&  
    refersToEquivalentClass [DomainClass,  
    App1Class2]) .
```

# Praktisches Beispiel



arableland Unterklasse von TRN:  
`taxon [arableland, trn] .`

Fig. 2. The Real World of the example with labels from the GTM domain ontology.

# Praktisches Beispiel

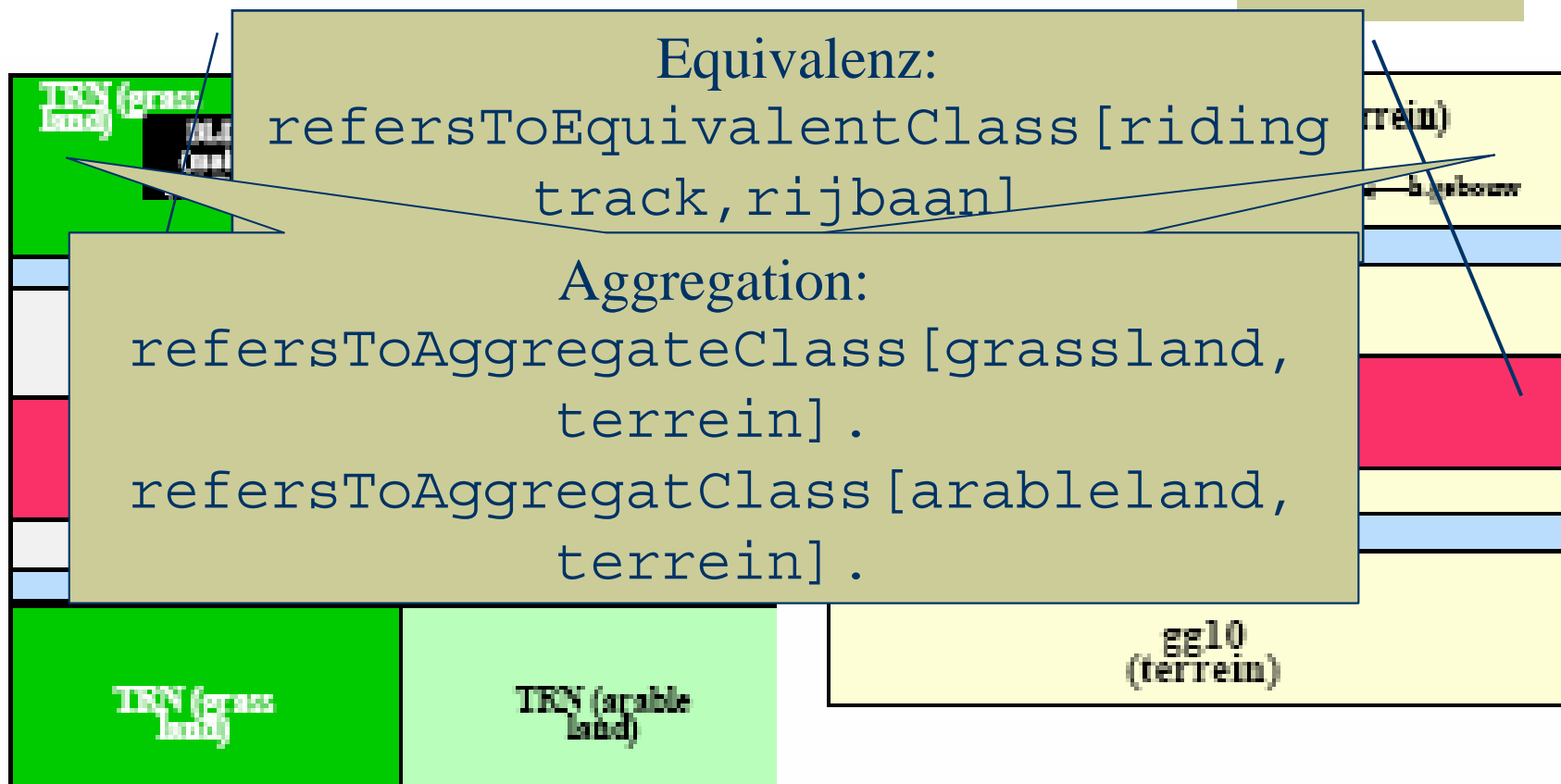


Fig. 2. The Real World of the example with labels from the GTM domain ontology.

Fig. 4. The GBKN as an abstraction from the Real World (Fig. 2). gg1, gg2, ..., gg10 are object identifiers (oid's).

# Zweite Datenquelle

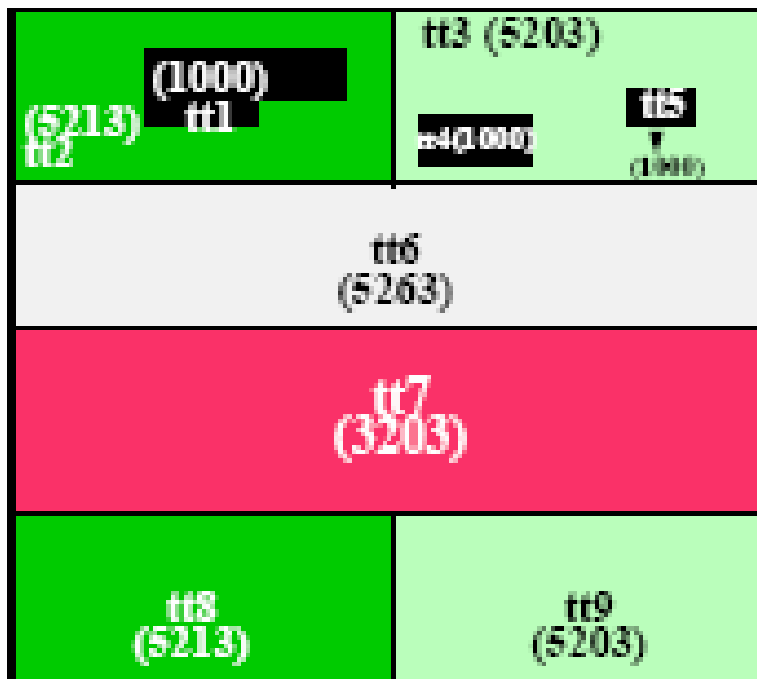


Fig. 5. The TOP10vector as an abstraction from the Real World (Fig. 2). tt1, tt2, ..., tt10 are object identifiers (oid's).

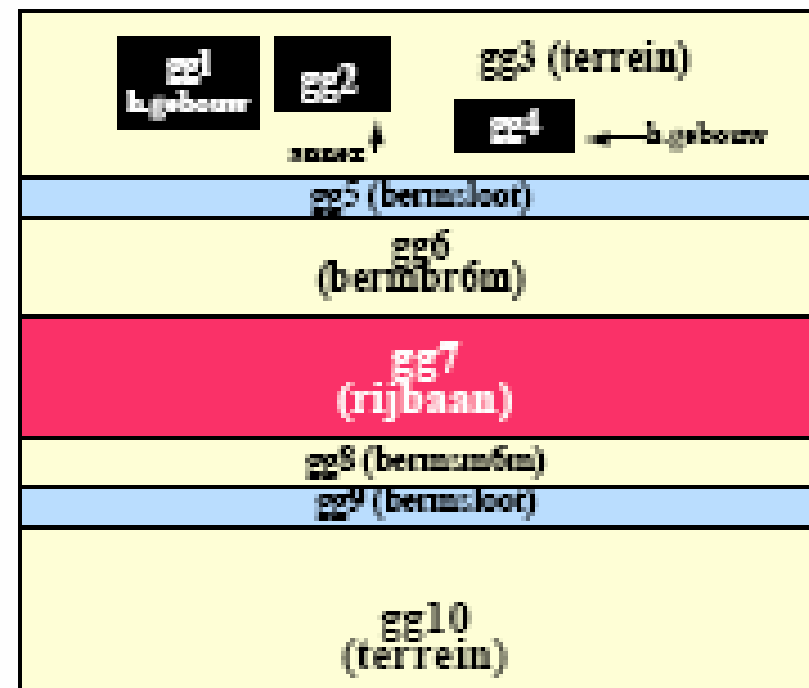


Fig. 4. The GBKN as an abstraction from the Real World (Fig. 2). gg1, gg2, ..., gg10 are object identifiers (oid's).

# Wie findet man korrespondierende Objekte?

1. Beide Datensätze werden anhand ihrer räumlichen Koordinaten übereinander gelegt.
2. Es entsteht eine neue Einteilung in verschiedene Oberflächen.
3. Jede dieser Oberflächen ist exakt einer Objektinstanz in jeder der beiden Quellen zugeordnet.

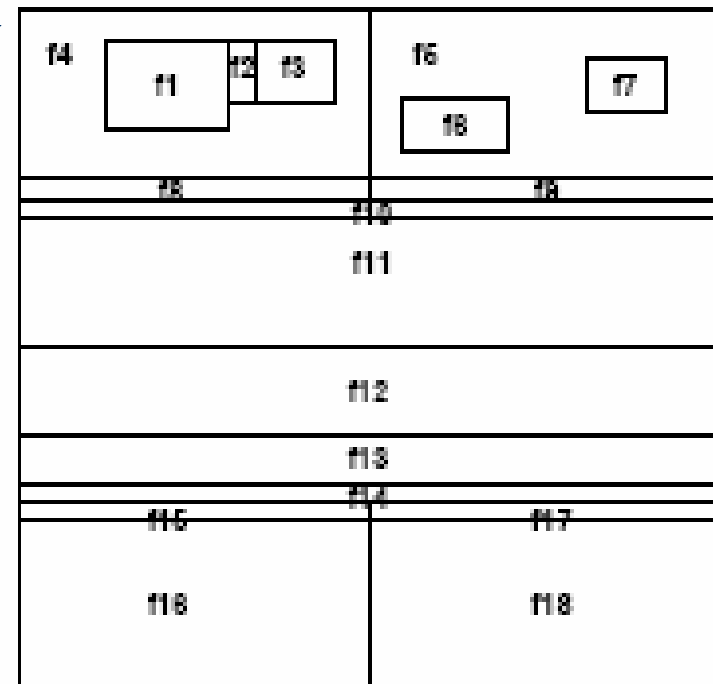


Fig. 7. The combination of GBKN (Fig. 4) and TOP10vector (Fig. 5).

# Wie findet man korrespondierende Objekte?

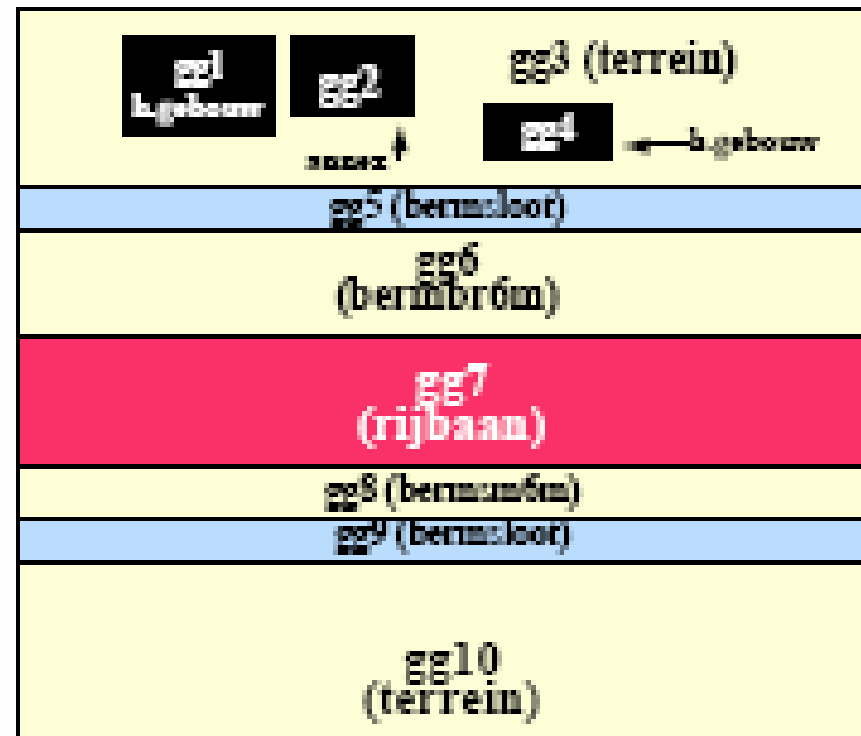
4. Erstellen einer Tabelle:
  - ◆ Jede Oberfläche erhält einen Eintrag
  - ◆ Es werden ihr die beiden Objektinstanzen und ihre jeweiligen Klassen zugeordnet
  - ◆ Es wird ein Eintrag über die semantische Übereinstimmung der Klassen aufgrund der in Prolog eingegebenen Daten hinzugefügt.

#1	#2	#3	#8	#7
#5	#6	#8		
		#10		
		#11		
		#12		
		#13		
		#14		
#15			#17	
#18			#18	

Fig. 7. The combination of GBKN (Fig. 4) and TOP10vector (Fig. 5).

# Tabelle entwickeln

14	f11	f12	f13	15	f17
				f18	
	f16			f16	
	f10				
	f11				
	f12				
	f13				
	f14				
	f15			f17	
	f18			f18	



# Tabelle

Oberflächen -id	GBKN-oid	Top10-id	semantische Übereinstim- mung
f1	gg1	tt1	related
f2	gg3	tt1	incompatible
f3	gg2	tt1	related

# Wie findet man korrespondierende Objekte

5. Semantisch ähnliche Objekte werden nun in einer Liste gespeichert.

```
{  
  { gg1,gg2 }, { tt1 } },  
  ... }
```

6. Objekte, zu denen kein korrespondierendes gefunden wurde, kommen in eine zweite.

```
{      { gg5 }, { } } ,      ...      }
```



Vielen Dank fürs Zuhören!



Fragen?