Informationsintegration

Antrittsvorlesung
am Tag der Informatik, 5.5.2004

Felix Naumann
naumann@informatik.hu-berlin.de

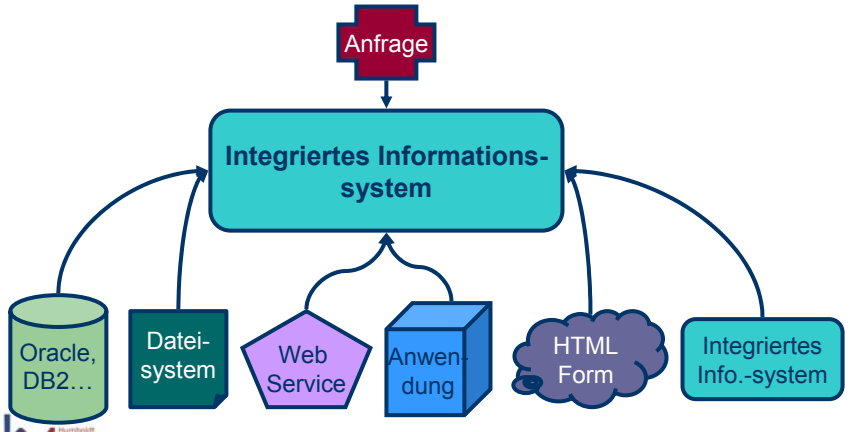
Arbeitsgruppe Informationsintegration

- Juniorprofessur
  - 2 x 3 Jahre
- Gefördert durch DFG
  - „Aktionsplan Informatik“
  - Seit 5. Mai 2003

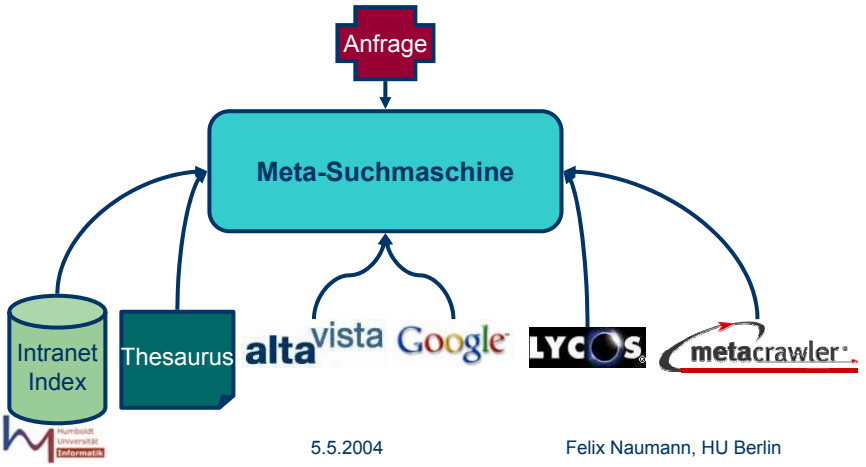
Einige Untertitel

- Content Merging
- Objekt/Data Fusion
- Data Amalgamation
- Data Consolidation
- Intelligent Information Integration
- Data Cleansing
- Datenintegration
- Datenverschmelzung

Integrierte Informationssysteme

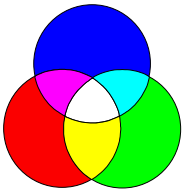


Integrierte Suchmaschinen

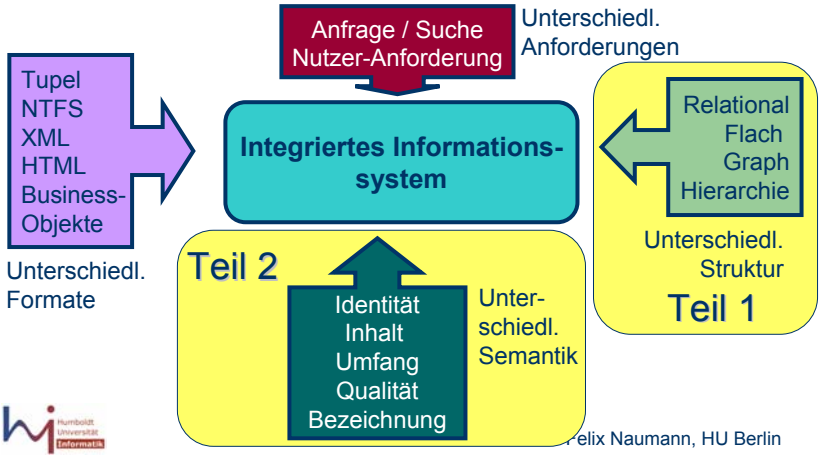


Redundanz – pro & contra

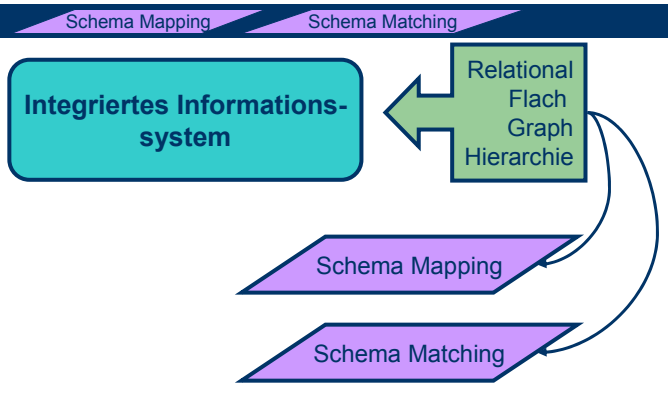
- Ist gut, denn sie bietet
  - mehr Informationen,
  - detaillierter Informationen,
  - und verifizierbare Informationen.
  - Deshalb sollten wir integrieren!
- Ist problematisch, denn
  - Redundanz herrscht nur konzeptionell.
  - Technische und strukturelle Schwierigkeiten
  - Konflikte und Alternativen
  - Deshalb ist Informationsintegration interessant.



Integrierte Informationssysteme



Lösung unterschiedlicher Struktur



Schematische Heterogenität

Schema Mapping | Schema Matching

```
Männer( Id, Vorname, Nachname)
Frauen( Id, Vorname, Nachname)
```

Relation vs. Attribut

```
Person( Id, Vorname,
        Nachname,männlich,
        weiblich)
```

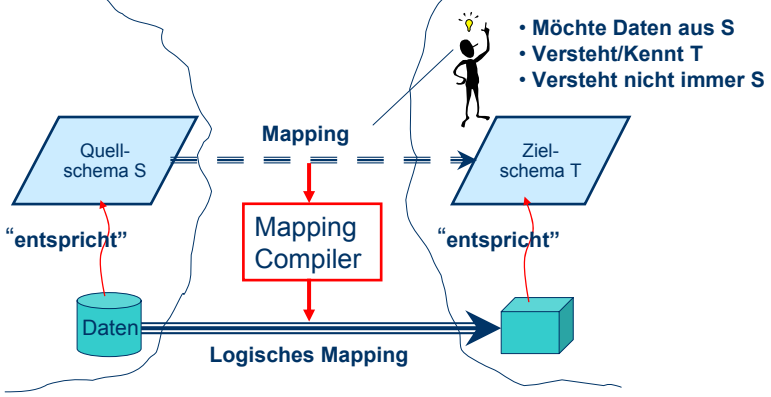
Relation vs. Wert

Attribut vs. Wert

```
Person( Id, Vorname,
        Nachname, Geschlecht)
```


Schema Mapping im Kontext

Schema Mapping | Schema Matching



Motivation: Warum ist Schema Mapping nützlich?

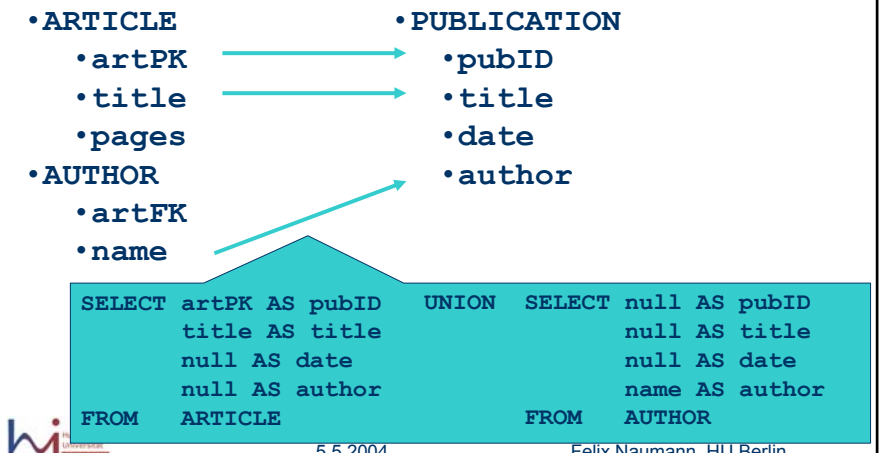
- Datentransformation zwischen heterogenen Schemata
  - Üblicherweise schreiben Experten komplexe Anfragen oder Programme
    - Zeitintensiv
    - Experte für die Domäne, für Schemata und für Anfrage
    - XML macht alles noch schwieriger
- Idee: Automatisierung
  - Gegeben: Zwei Schemata und ein high-level Mapping dazwischen
  - Gesucht: Anfrage zur Datentransformation

Motivation: Warum ist Schema Mapping schwierig?

- Generierung der “richtigen” Anfrage unter Berücksichtigung
  - des Quell und Ziel-Schemas,
  - des Mappings,
  - und der Nutzer-Intention.
  - Fagin et al. 2002
- Garantie, dass die transformierten Daten dem Zielschema entsprechen.
- Effiziente Datentransformation
  - Für materialisierte Integration
  - Für virtuelle Integration

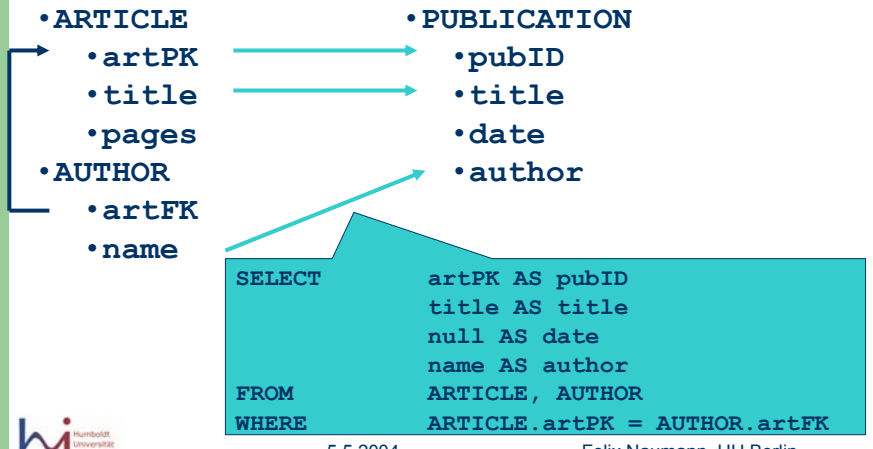
Schema Mapping Beispiel

Schema Mapping | Schema Matching



Schema Mapping Beispiel

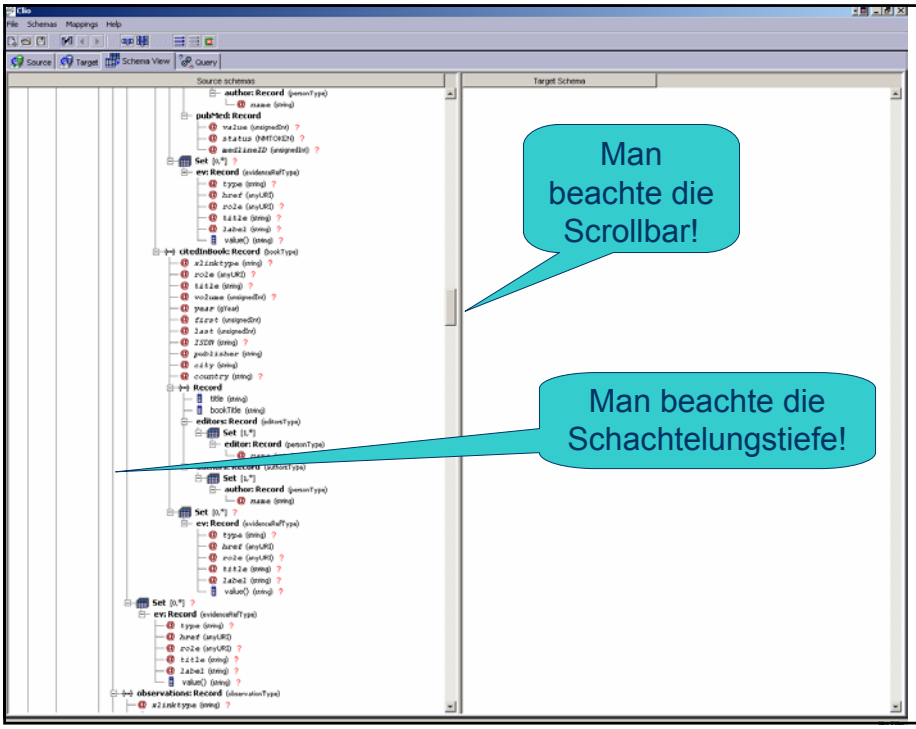
Schema Mapping | Schema Matching



Schema Matching – Motivation

Schema Mapping | Schema Matching

- Große Schemata
  - > 100 Tabellen, viele Attribute
  - Bildschirm nicht lang genug
- Unübersichtliche Schemata
  - Tiefe Schachtelungen
  - Fremdschlüssel
  - Bildschirm nicht breit genug
  - XML Schema
- Fremde Schemata
  - Unbekannte Synonyme
- Irreführende Schemata
  - Unbekannte Homonyme
- Fremdsprachliche Schemata
- Kryptische Schemata
  - |Attributnamen| ≤ 8 Zeichen
  - |Tabellennamen| ≤ 8 Zeichen



Schema Matching Klassifikation

- Schema Matching basierend auf
  - Namen der Schemaelemente (*label-based*)
  - Darunterliegende Daten (*instance-based*)
  - Struktur des Schemas (*structure-based*)
  - Mischformen, Meta-Matcher
  - (Rahm und Bernstein 2001)
- Kernalgorithmus des Schema Matching:
  - Gegeben zwei Schemata mit Attributmengen A und B.
  - Bilde Kreuzprodukt aller Attribute aus A und B.
  - Für jedes Paar vergleiche Ähnlichkeit.
  - Ähnlichste Paare sind Matches.

Schema Matching – Label-basiert

- Ähnlichkeitsmaß: Ähnlichkeit der Attributnamen
- Beispiele:
  - Ähnlichkeit(‚Name', ‚Name') = 1
  - Ähnlichkeit(‚Titel', ‚title') = 0.8
- Edit-distance
  - Anzahl der Edit-Operationen um String 1 in String 2 umzuwandeln.
- Probleme:
  - Effizienz
  - Auswahl der besten Matches
    - Iterativ,
    - Stable Marriage, etc.
  - Synonyme und Homonyme werden nicht erkannt

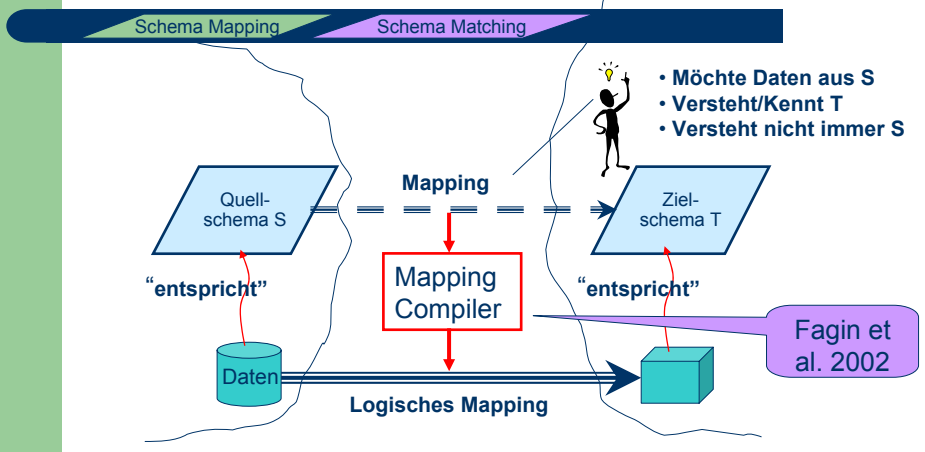
Schema Matching – Instanz-basiert

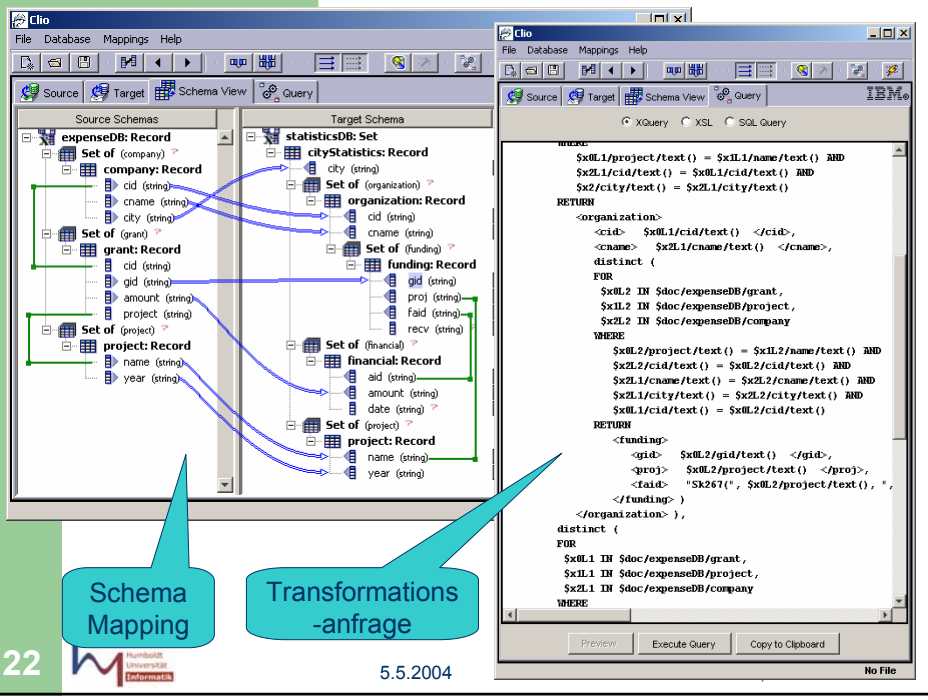
- Ähnlichkeitsmaß: Ähnlichkeit der Attributwerte
  - Typische Eigenschaften
    - Länge,
    - Existenz eines Trennzeichens,
    - Mischung Groß- und Kleinschreibung,
    - Buchstabenverteilung, etc.
- Probleme
  - Auswahl und Gewichtung der Eigenschaften
  - Datenmenge: Sampling
  - Vergleichsmethode, z.B. Naive Bayes

Schema Matching – Struktur-basiert

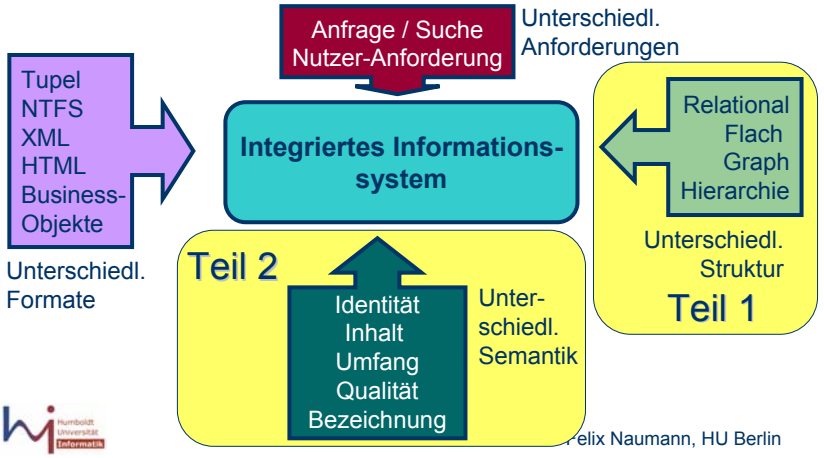
- Ähnlichkeitsmaß: Elemente der „Nachbarschaft“ von Attributen
- Kernidee
  - Nutze (komplexe) Struktur der Schemata aus.
    - Hierarchieebene
    - Elementtyp (Attribut, Relation, Label,...)
    - Nachbarschaftsbeziehungen
  - Wenn Element *x* ähnliche Nachbarn hat wie Element *y*, sind *x* und *y* ähnlich.
- Effektiv durch Kombination mit anderen Methoden

Schema Mapping im Kontext

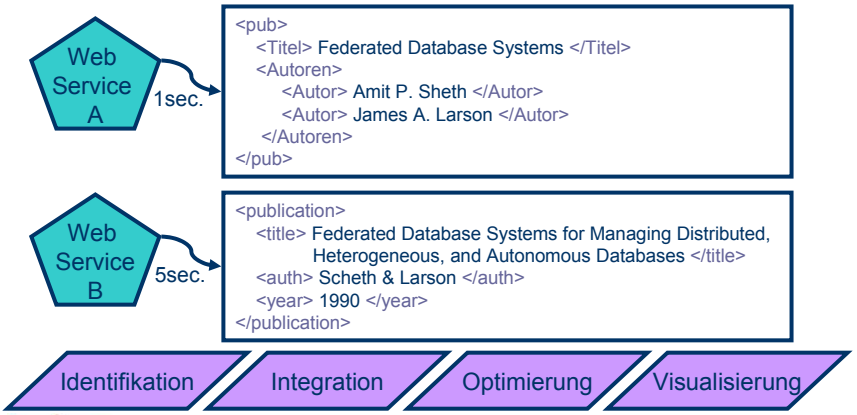




Integrierte Informationssysteme



Beispiel der Informationsintegration



Objektidentifikation

- Erkennung mehrere (ähnlicher) Darstellungen des gleichen Objekts
- Auch
  - Duplicate Detection
  - Data Cleansing
  - Record Linkage
- Domänenspezifische Algorithmen
  - Adressdaten
  - Mikrobiologische Daten
- Manchmal gibt es IDs
  - Bücher: ISBN
  - Personen: SSN / Personalausweisnummer
  - Webseiten: URL

Objektidentifikation

- Das klassische Problem
  - Finde Duplikate innerhalb einer Tabelle.
  - Sehr große Datenmenge
    - kein quadratischer Algorithmus
    - kein Hauptspeicher-Algorithmus
- Forschung
  - Sorted-Neighborhood Methode (Hernandez, Stolfo 1998)
  - Record Linkage (Fellegi, Sunter, 1969), etc.
- Industrie
  - Trillium, Vality, ETI, et al.

Vorname	Nachname	Adresse	ID
Sal	Stolpho	123 First St.	456780
Mauricio	Hernandez	321 Second Ave	123456
Felix	Naumann	Hauptstr. 11	987654
Sal	Stolfo	123 First Street	456789

Objektidentifikation in XML Daten

Identifikation | Integration | Optimierung | Visualisierung

- Data Warehouse Duplicates
  - Ausnutzung hierarchischer Daten
  - Nur Star-Schema
  - (Ananthakrishna, Chauduri & Ganti 2002)
- XML Duplikate
  - Kinder verschiedener Typen (Snowflake-Schema)
  - Was ist ein Duplikat?
  - Semistrukturierte Daten
  - Schemalose Daten

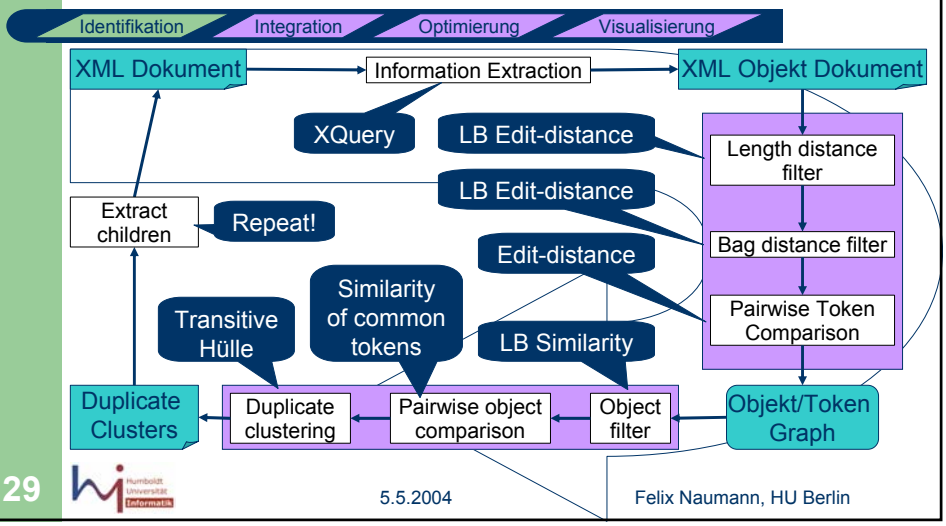
Objektidentifikation in XML Daten

Identifikation | Integration | Optimierung | Visualisierung

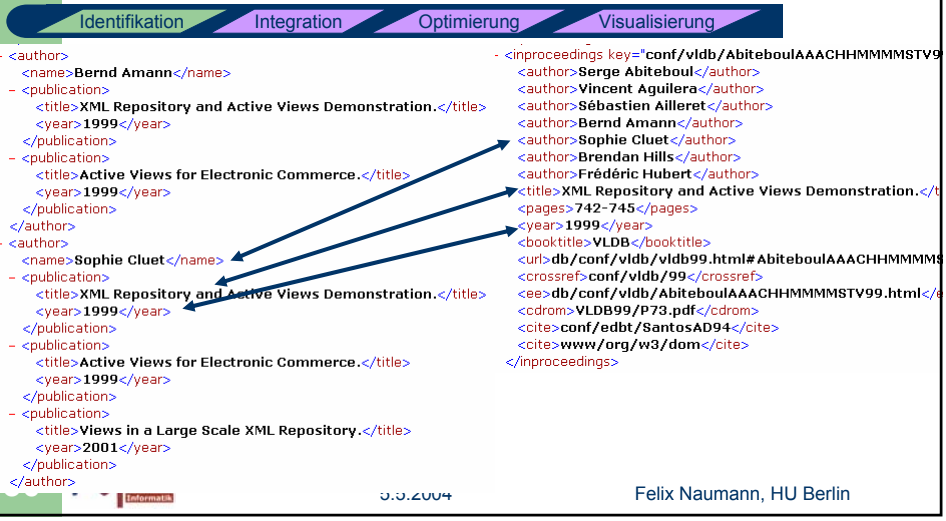
```
- <author>
    <name>Bernd Amann</name>
  - <publication>
      <title>XML Repository and Active Views Demonstration.</title>
      <year>1999</year>
    </publication>
  - <publication>
      <title>Active Views for Electronic Commerce.</title>
      <year>1999</year>
    </publication>
  </author>
- <author>
    <name>Sophie Cluet</name>
  - <publication>
      <title>XML Repository and Active Views Demonstration.</title>
      <year>1999</year>
    </publication>
  - <publication>
      <title>Active Views for Electronic Commerce.</title>
      <year>1999</year>
    </publication>
  - <publication>
      <title>Views in a Large Scale XML Repository.</title>
      <year>2001</year>
    </publication>
  </author>
```

- Vergleiche <author>
  - Mit Subelementen (<publication>)?
  - Wie tief?
- Vergleiche <publication>
  - Mit parallelen Elementen (<year>)?
  - Schema, oder Daten?
- Kurz: Was ist ein Duplikat?

Objektidentifikation in XML Daten (Melanie Weis)



Objektidentifikation in XML Daten



Duplikat-gesteuertes Schema Matching (Alexander Bilke)

Identifikation | Integration | Optimierung | Visualisierung

- Umgekehrte Idee
- Schema Matching
- Herkömmliche Lösungen (siehe zuvor)
  - Namen der Schemaelemente (*label-based*)
  - Darunterliegende Daten (*instance-based*)
    - Insbesondere Eigenschaften von Spalten
  - Struktur des Schemas (*structure-based*)
- Nun
  - Finde Duplikate (trotz mangelnder Schemata)
  - Korrespondenzen zwischen gleichen Attributwerten

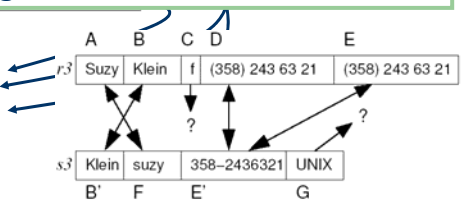
Duplikat-gesteuertes Schema Matching

Identifikation | Integration | Optimierung | Visualisierung

R	A	B	C	D	E
r_1	John	Doe	m	(408) 757 33 39	(408) 757 33 38
r_2	Joe	Smith	m	(249) 361 56 16	(249) 234 23 66
r_3	Suzy	Klein	f	(358) 243 63 21	(358) 243 63 21
r_4	Sam	Adams	m	(541) 812 71 00	(541) 812 11 64
r_5	Mark	Spitz	m	(901) 831 93 11	(901) 861 23 82
r_6	Jim	Beam	-	(782) 123 89 57	(781) 188 37 44

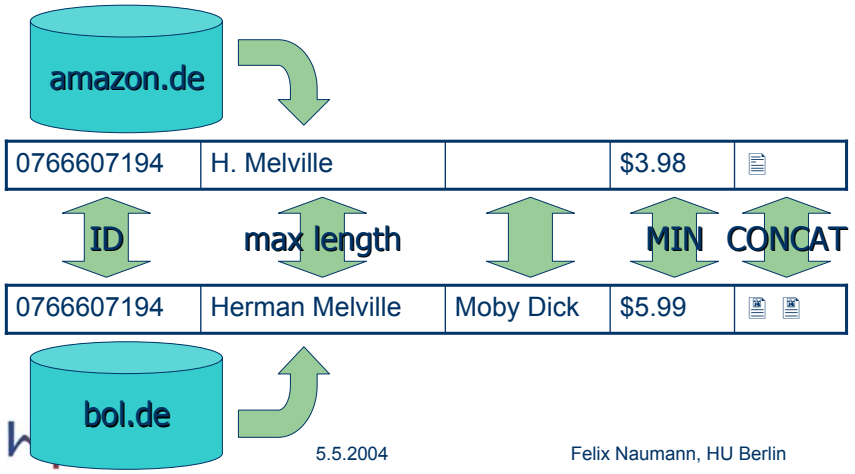
Jetzt wissen wir, WAS integriert werden soll. Bloß WIE?

S	B'	F	E'	G
s_1	Doe	jdoe	408-9182043	XP
s_2	Deen	jdean	369-3663625	XP
s_3	Klein	suzy	358-2436321	UNIX
s_4	Adams	adams	541-8121164	W2000
s_5	Wong	howard	923-6363443	Linux
s_6	Kurz	itsme	-	UNIX



Objektintegration

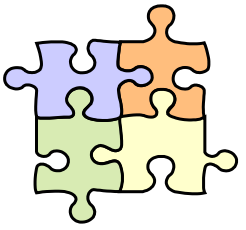
Identifikation | Integration | Optimierung | Visualisierung



Relationale Objektintegration

Identifikation | Integration | Optimierung | Visualisierung

- Union ∪
  - Duplikat-Eliminierung
- Outer union ⓤ (Codd 1979)
  - Union bei heterogenen Schemata
- Minimum union ⊕ (Ullmann 1989, Galindo-Legaria 1994)
  - Eliminierung subsummierter Tupel
- Merge union
  - Duplikatintegration
  - Konfliktlösung



Wie Union integriert

Quelle 1(A,B,C) →
Quelle 2(A,B,C) →

a, b, c a, b, c	→ a, b, c	⇒ Elimination identischer Tupel
a, b, c a, b, -	→ a, b, c a, b, -	⇒ Keine Elimination subsummierter Tupel
a, b, c a, b, d	→ a, b, c a, c, d	⇒ Keine Konfliktlösung
a, -, c a, b, -	→ a, -, c a, b, -	⇒ Keine Integration komplementärer Tupel

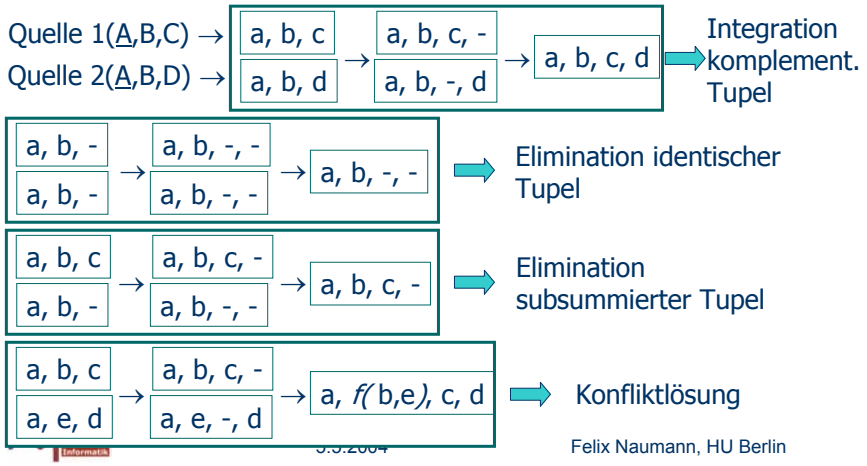
Wie Minimum Union integriert

Quelle 1(A,B,C) →
Quelle 2(A,B,D) →

a, b, c a, b, d	→ a, b, c a, b, -, d	→ a, b, c, - a, b, -, d ⇒ Keine Integr. komplement. Tupel
a, b, - a, b, -	→ a, b, -, - a, b, -, -	→ a, b, -, - ⇒ Elimination identischer Tupel
a, b, c a, b, -	→ a, b, c, - a, b, -, -	→ a, b, c, - ⇒ Elimination subsummierter Tupel
a, b, c a, e, d	→ a, b, c, - a, e, -, d	→ a, b, c, - a, e, -, d ⇒ Keine Konfliktlösung

Wie Merge Union integrieren sollte (Jens Bleiholder)

Identifikation | Integration | Optimierung | Visualisierung



Konfliktlösung

Identifikation | Integration | Optimierung | Visualisierung

$$f(x,y) := \begin{cases} \perp & if\ x = \perp\ and\ y = \perp \\ x & if\ y = \perp\ and\ x \neq \perp \\ y & if\ x = \perp\ and\ y \neq \perp \\ g(x,y) & else \end{cases}$$

null = unbekannt

interne Konfliktlösungsfunktion

Konfliktlösungsfunktionen

- Numerisch:

  SUM, AVG, MAX, MIN, …
- Nicht-numerisch:

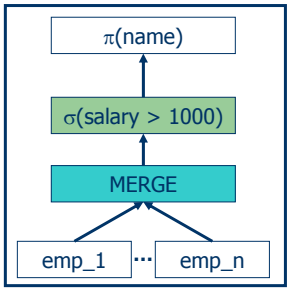
  …
- Spezialisiert:

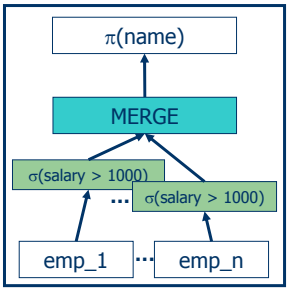
  RANDOM, COUNT, CHOOSE, FAVOR, MaxIQ,…
- Domänen-spezifisch

  …

Jetzt wissen wir, WIE integriert werden soll.
Aber WIE GUT schaffen wir das?

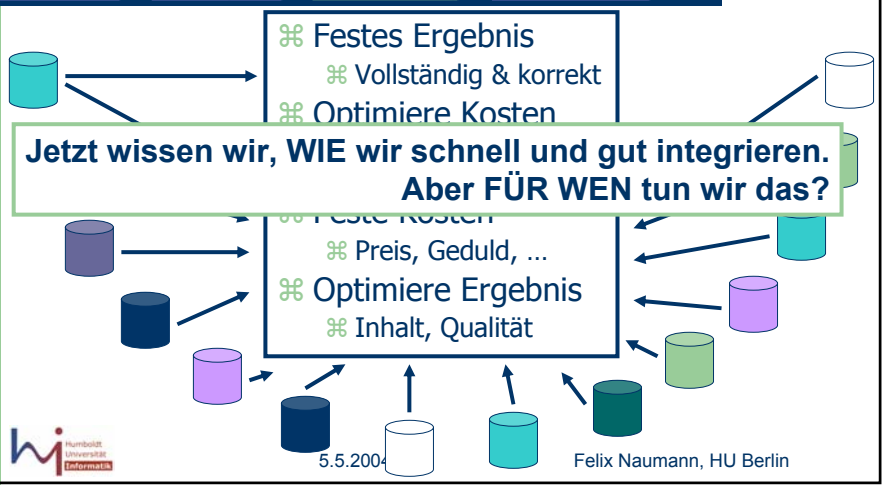
Anfrageoptimierung mit Merging





- Korrektheit?
- Vollständigkeit?
- Effizienz?

Optimierung – Ein Paradigmenwechsel

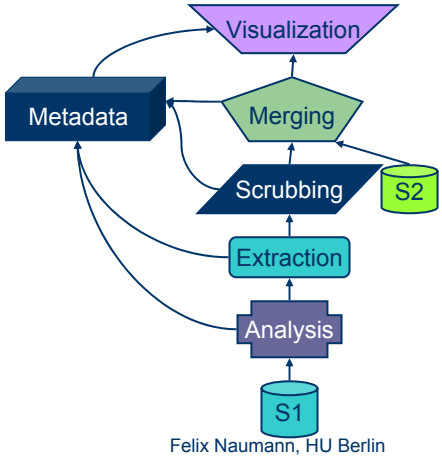


Visualisierung – Ziele

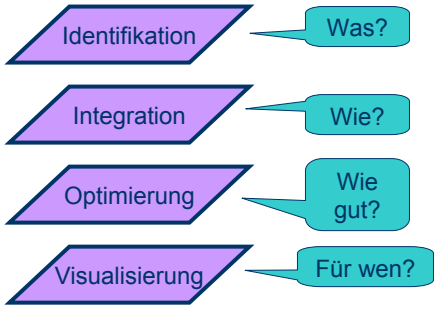
- Darstellung integrierter Ergebnisse
  - Viele Quellen
  - Viele Operationen / Transformationen
- Nutzerfreundliche Darstellung
- Drill-Down
  - Metadaten
- Kein kleinster gemeinsamer Nenner!

Metadaten der Integration

- Herkunft
  - Data lineage
    - z.B. (Cui & Widom 2001)
  - Durch Operatoren hindurch
- Transformationen
  - Aggregation / Integration
- Konflikte
- Datenqualität
- Ranking



Informationsintegration - Zusammenfassung



http://www.informatik.hu-berlin.de/mac/
naumann@informatik.hu-berlin.de

Dank an

- Doktoranden
  - Jens Bleiholder
  - Melanie Weis
  - Alexander Bilke (TU-Berlin)
  - Armin Roth (DaimlerChrysler)
- Studenten
  - Christoph Böhm
  - Karsten Draba

Literatur

- Fellegi & Sunter 1969
  - A theory of record linkage. Journal of the American Statistical Association, 64: 1183-1210.
- Codd 1979
  - Extending the Relational Database Model to Capture more Meaning. ACM Transactions on Database Systems (TODS), 4(4): 397-434.
- Galindo-Legaria 1994
  - Outerjoins as Disjunctions. ACM SIGMOD Conference, 348-358.
- Hernandez & Stolfo 1998
  - Real-world data is dirty: Data cleansing and the merge/purge problem. Data Mining and Knowledge Discovery, 2(1): 9-37.
- Rahm & Bernstein 2001
  - A survey of approaches to automatic schema matching. VLDB Journal 10(4), 334-350.
- Cui & Widom 2001
  - Lineage tracing for general data warehouse transformations. VLDB Jorunal 12(1): 41-58.
- Ananthakrishna, Chauduri & Ganti 2002
  - Eliminating Fuzzy Duplicates in Data Warehouses. VLDB Conference: 586-597.
- Erhard Rahm and Philip Bernstein 2001
  - A survey of approaches to automatic schema matching, VLDB Journal 10(4), 2001.
- Ron Fagin, Mauricio Hernandez, Lucian Popa, Renee Miller, and Yannis Velegrakis,
  - Translating Web Data, VLDB 2002, Hong Kong, China.