Schema Mappings in PDMS

Rostock, 9. Juni 2005
Prof. Felix Naumann
naumann@informatik.hu-berlin.de
Humboldt-Universität zu Berlin

Humboldt-Universität zu Berlin

Humboldt-Universität zu Berlin

- Wilhelm und Alexander von Humboldt
- Einheit von Lehre und Forschung
- Freiheit und Unabhängigkeit der Wissenschaft
- 29 Nobelpreisträger
  - Mommsen, Hertz, Koch, Hahn, Planck, Einstein,...
- 38,000 Studenten, (1100 Informatik)
- 500 Professoren (21 Informatik)

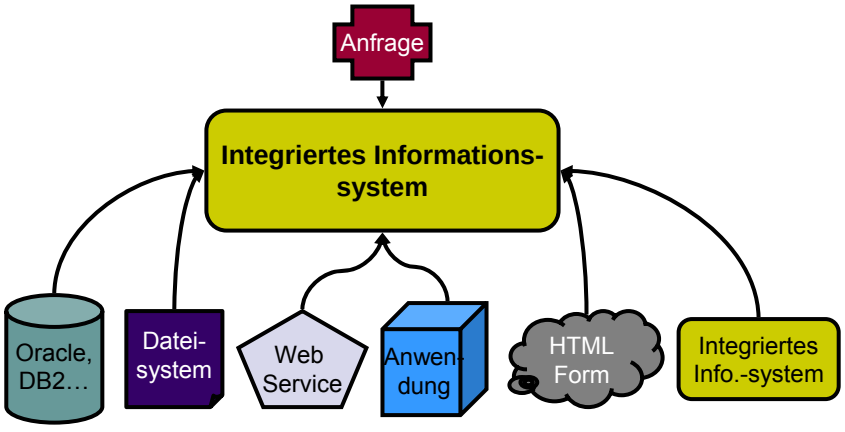
Forschungsgruppe Informationsintegration

- Leitung: Felix Naumann (naumann@informatik.hu-berlin.de)
- Mitarbeiter
  - Jens Bleiholder (bleiho@informatik.hu-berlin.de)
    - Informationsfusion in relationalen Daten
  - Melanie Weis (mweis@informatik.hu-berlin.de)
    - Objektidentifikation in XML Daten
- Affiliated
  - Armin Roth (aroth@informatik.hu-berlin.de)
    - Datenqualität in Peer-Data-Management-Systemen
  - Alexander Bilke (bilke@cs.tu-berlin.de)
    - Schema Matching
- Forschungsthemen
  - Objektidentifikation
  - Informationsfusion
  - Optimierung
  - Visualisierung

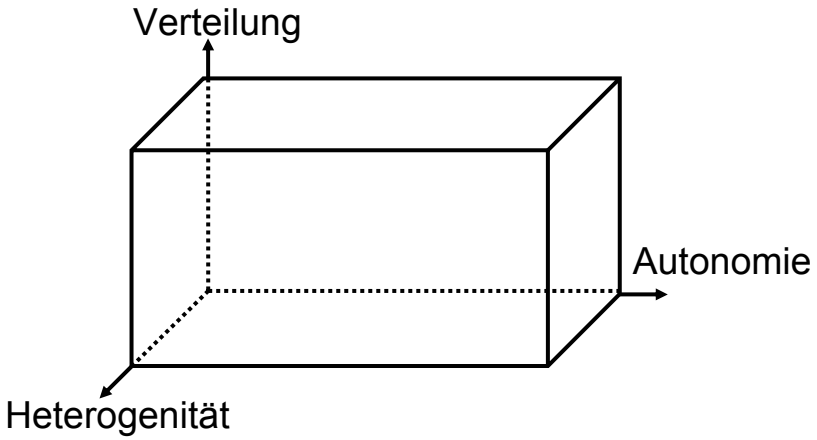
Überblick

1. Informationsintegration und schematische Heterogenität
2. Schema Mapping
3. Schema Matching
4. Peer Data Management (PDMS)
5. Mappings und Anfrage-bearbeitung in PDMS
6. Weitere Themen der Arbeitsgruppe

Integrierte Informationssysteme



Klassifikation von Informationssystemen [ÖV99]



Warum ist Informationsintegration so schwer? [Halevy04]

- Alon Halevy: „It's plain hard!"
- System-bedingte Gründe
  - Verschiedene Plattformen
  - Anfragebearbeitung über mehrere Systeme
- Soziale Gründe
  - Finden relevanter Daten in Unternehmen
  - Beschaffen relevanter Daten in Unternehmen
  - Menschen zur Zusammenarbeit überreden
- Logik-bedingte Gründe
  - Schema- und Datenheterogenität
  - Dies ist unabhängig von der jeweiligen Integrationsarchitektur.

Schematische Heterogenität - Beispiel

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)

Schematische Heterogenität - Beispiel

- Normalisiert vs. Denormalisiert
  - Assoziationen zwischen Werten wird unterschiedlich dargestellt
    - Durch Vorkommen im gleichen Tupel
    - Durch Schlüssel-Fremdschlüssel Beziehung

•ARTICLE		•PUBLICATION
•artPK	vs.	•pubID
•title		•title
•pages		•date
•AUTHOR		•author
•artFK		
•name		

Schematische Heterogenität - Beispiel

- Geschachtelt vs. Flach
  - Assoziationen werden unterschiedlich dargestellt
    - Als geschachtelte Elemente
    - Als Schlüssel-Fremdschlüssel Beziehung

•**ARTICLE**
  •**artPK**
  •**title**
  •**pages**
  •**AUTHOR**
  •**name**
	VS.	•**PUBLICATION**
		  •**pubID**
		  •**title**
		  •**author**

Schematische Heterogenität - Lösungen

- Zwei alternative Probleme
  1. Einheitlich auf beide Schemata zugreifen
     - Auf Schemaebene:
       - Schema-Sprachen (SchemaSQL, MSQL, CPL)
       - Schema Mapping (Clio, Rondo, Tools)
     - Auf Datenebene: Virtuelle Integration
  2. Beide Schemata in ein gemeinsames neues Schema integrieren
     - Auf Schemaebene: Schemaintegration
     - Auf Datenebene: Materialisierte Integration, ETL

Schematische Heterogenität – Lösungen

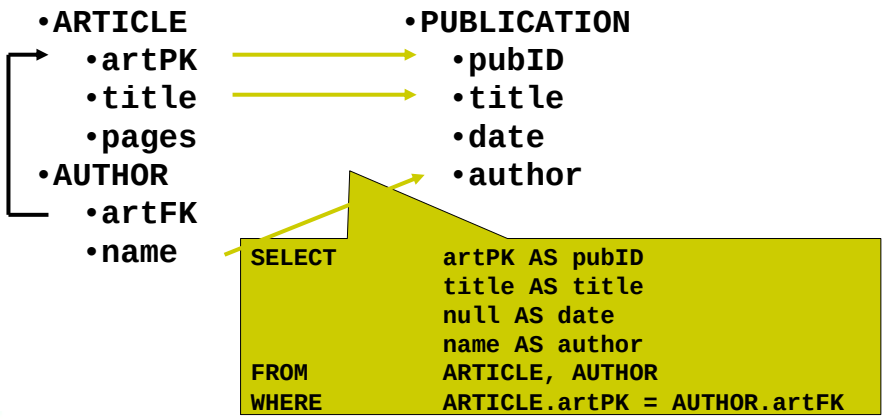
- SchemaSQL [LSS96]
  - Erweiterung von SQL
  - Daten und Metadaten werden gleich behandelt
  - Umstrukturierungen innerhalb der Anfrage
  - Dynamische Sicht-Definition
  - Horizontale Aggregation

```
SELECT RelA
FROM   uniA->RelA, uniA::RelA A, uniB::grundgehalt B
WHERE  RelA = B.institut
AND    A.Kategorie = „Student“
```

Join zwischen Relationenname und Attributwert

Schematische Heterogenität – Lösungen (Ausblick)

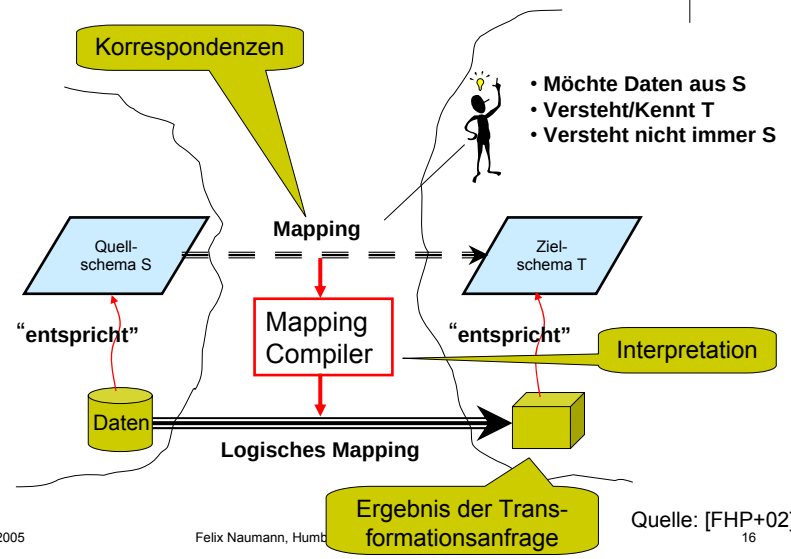
- Schema Mapping



Überblick

1. Informationsintegration und schematische Heterogenität
2. Schema Mapping
3. Schema Matching
4. Peer Data Management (PDMS)
5. Mappings und Anfrage-bearbeitung in PDMS
6. Weitere Themen der Arbeitsgruppe

Schema Mapping im Kontext



Quelle: [FHP+02]

Warum ist Schema Mapping nützlich?

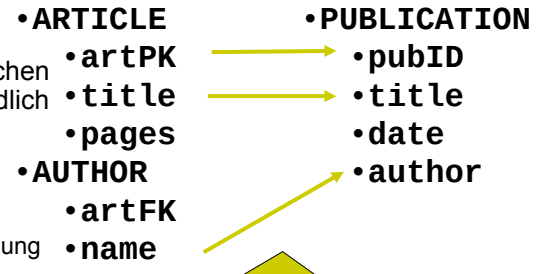
- Datentransformation zwischen heterogenen Schemata
  - Altes aber immer wiederkehrendes Problem
  - Üblicherweise schreiben Experten komplexe Anfragen oder Programme
    - Zeitintensiv
    - Experte für die Domäne, für Schemata und für Anfrage
    - XML macht alles noch schwieriger
      - XMLSchema, XQuery
- Idee: Automatisierung
  - Gegeben: Zwei Schemata und ein high-level Mapping dazwischen
  - Gesucht: Anfrage zur Datentransformation
  - Später: Schema Matching = automatisches Finden des high-level Mapping

Warum ist Schema Mapping schwierig?

- Generierung der „richtigen“ Anfrage unter Berücksichtigung
  - des Quell und Ziel-Schemas,
  - des Mappings
  - und der Nutzer-Intention
- Semantik der Daten erhalten
  - Assoziationen entdecken
  - Schemata und deren Integritätsbedingungen nutzen
  - Ggf. neue Datenwerte erzeugen
  - Korrekte Gruppierungen erzeugen
- Garantie, dass die transformierten Daten dem Zielschema entsprechen
- Effiziente Datentransformation
  - Für materialisierte Integration
  - Für virtuelle Integration

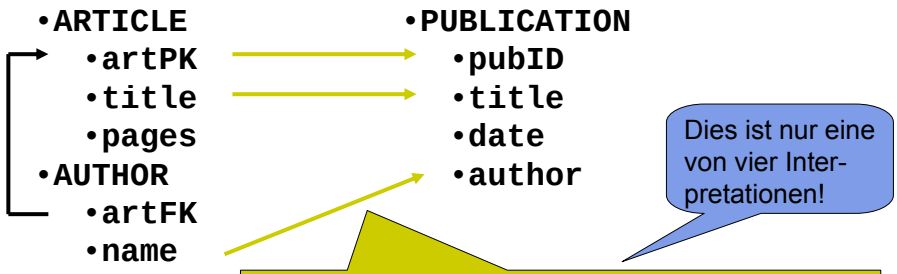
Schema Mapping Beispiel

- Normalisiert vs. Denormalisiert
  - 1:1 Assoziationen zwischen Werten wird unterschiedlich dargestellt
    - Durch Vorkommen im gleichen Tupel
    - Durch Schlüssel-Fremdschlüssel Beziehung




```
SELECT artPK AS pubID    UNION  SELECT null AS pubID
       title AS title                  null AS title
       null AS date                    null AS date
       null AS author                  name AS author
FROM   ARTICLE                 FROM   AUTHOR
```

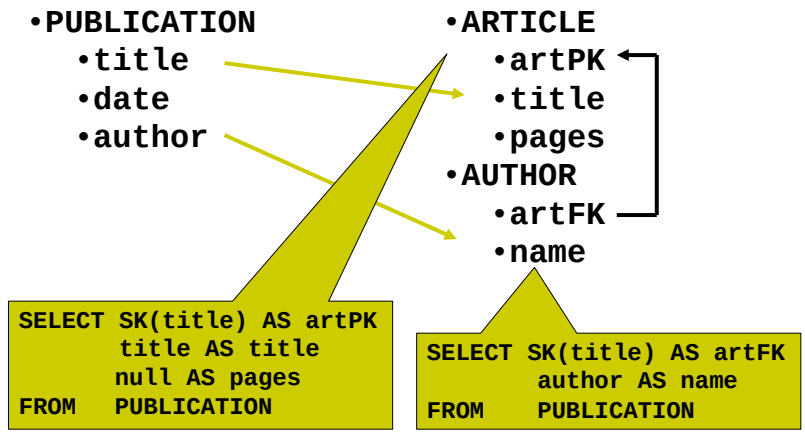

Schema Mapping Beispiel



Dies ist nur eine von vier Inter-pretationen!

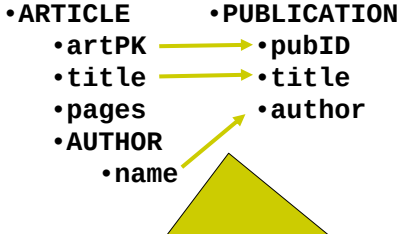
```
SELECT      artPK AS pubID
            title AS title
            null AS date
            name AS author
FROM        ARTICLE, AUTHOR
WHERE       ARTICLE.artPK = AUTHOR.artFK
```

Schema Mapping Beispiel



Schema Mapping Beispiel

- Geschachtelt vs. Flach
  - 1:n Assoziationen werden unterschiedlich dargestellt
    - Als geschachtelte Elemente
    - Als Schlüssel-Fremdschlüssel Beziehung



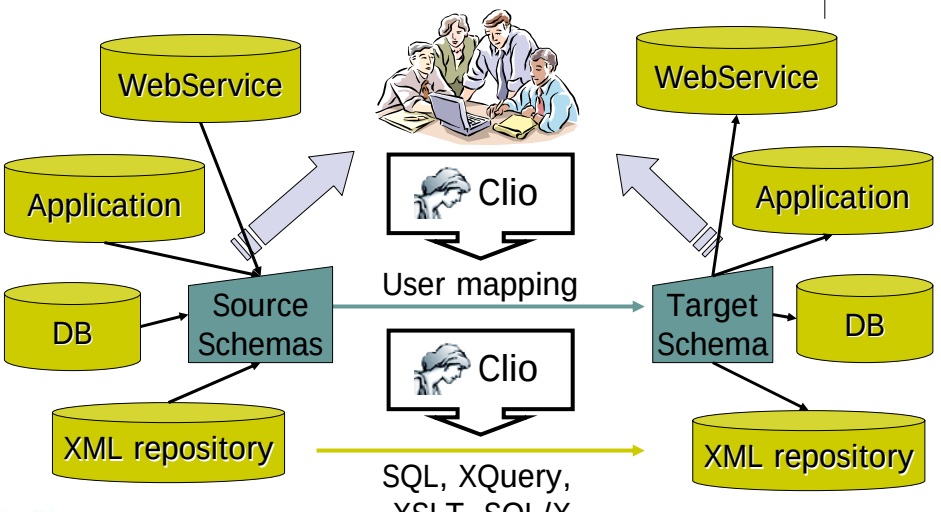

```
LET $doc0 := document("articls.xml") RETURN
<dblp> { distinct-values (
   FOR $x0 IN $doc0/authorDB/ARTICLE, $x1 IN $x0/AUTHOR
   RETURN
        <publication>
        <pubID>  {  $x0/artPK/text() }  </pubID>
        <title>  {  $x0/title/text() }  </title>
        <author>  {  $x1/name/text() }  </author>
        </publication> )
   } </dblp>
```

Schema Mapping Beispiel

- PUBLICATION
  - title → ARTICLE.title
  - date
  - author → ARTICLE.AUTHOR.name

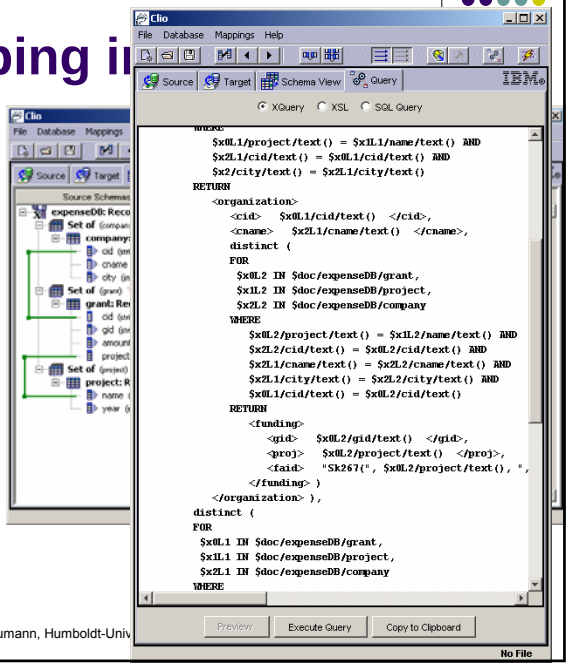
```
LET $doc0 := document("publication.xml")
RETURN
<articles> { distinct-values (
   FOR $x0 IN $doc0/dblp/publication  RETURN
      <ARTICLE>
         <title>  {  $x0/title/text() }  </title>
         { distinct-values (
         FOR $x0L1 IN $doc0/dblp/publications
         WHERE $x0/title/text() = $x0L1/title/text()
         RETURN
            <AUTHOR>
               <name>  {  $x0L1/author/text() }  </name>
            </AUTHOR> )}
      </ARTICLE> ) } </articles>
```


Schema Mapping mit Clio

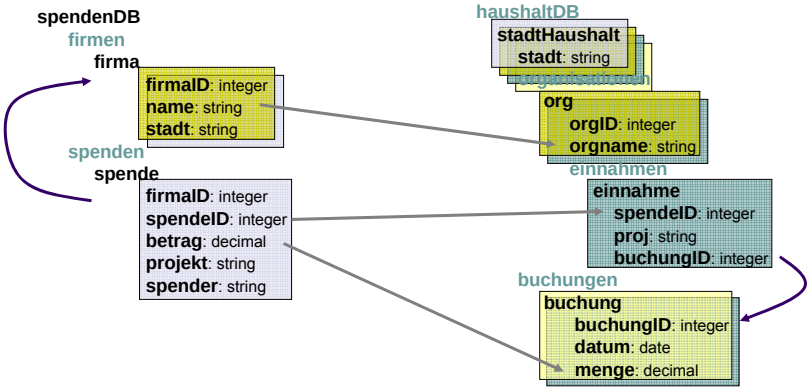


Schema Mapping i

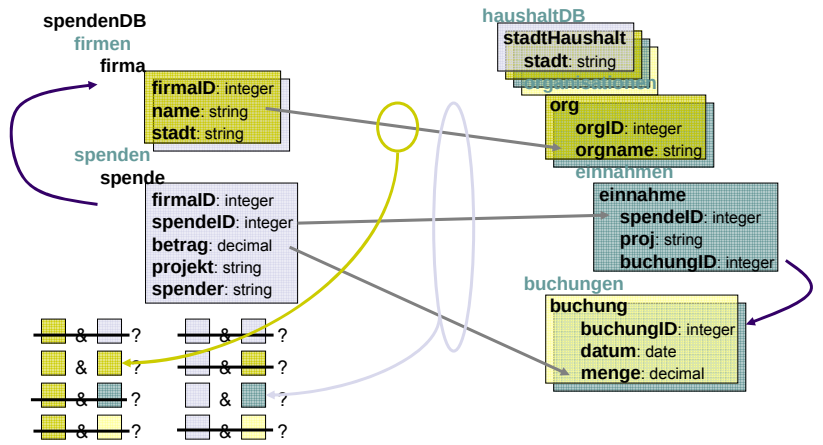
1. Schema Matching & Korrespondenzen
2. Schema Mapping
3. Mapping Interpretation
4. Daten-transformation



Entdeckung von Assoziationen



Entdeckung von logischen Mappings



Andere Tools

- Rondo: Eine Programmierplatform für Model-Management [MRB03]
  - Modelle
    - RDB Schema, Sichten, XML-Schema, DTD
  - Basis Operatoren
    - Domain, Invert, Compose, TransitiveClosure, SubGraph, All, Copy
  - Weitere Operatoren
    - Extract, Delete, Match, Merge
  - Szenarien, z.B. Change Propagation
- Industrie
  - Altova: MapForce
  - IBM: WebSphere Application Developer
  - Microsoft: BizTalk Mapper
  - u.v.a.m, z.B. Data Warehouse Manager und ETL Tools

Modellmanagement als Vision
[BLP00, Ber03]

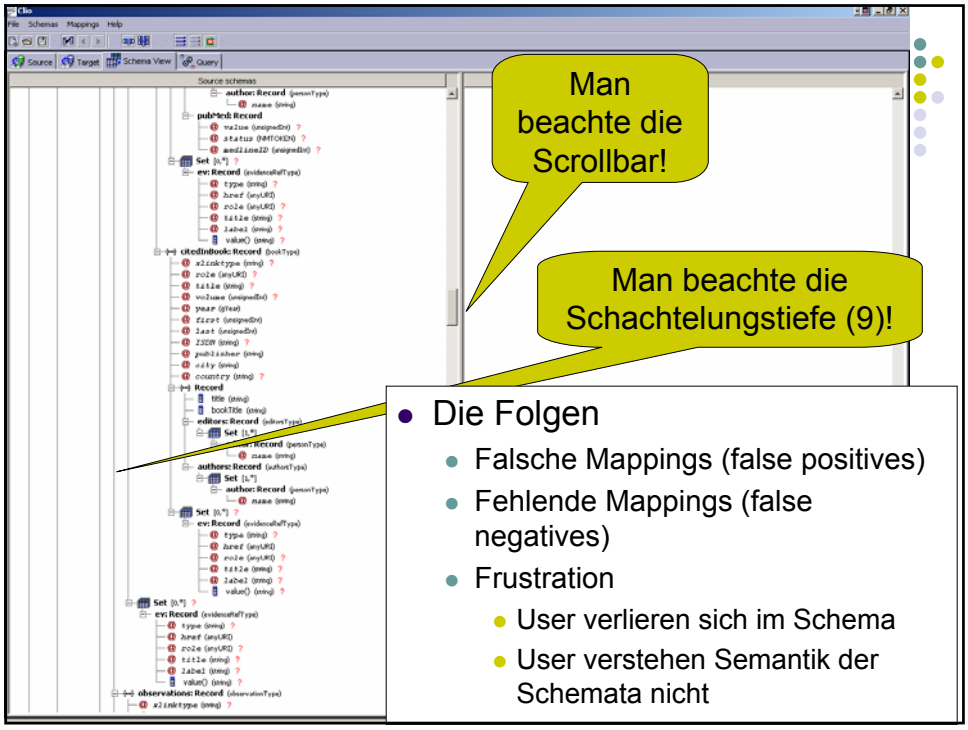
- Modelle als *First-Class-Citizens*
  - RDB Schema, Sichten, XMLSchema, DTD, Anfragen, Java Classen, HTML Seiten, usw.
  - Allgemein Graphen:
    - Objekte + Relationships
    - + Mappings
- Basis Algebra
  - Create, Update, Delete
  - Select, Project, SetDifference, ApplyFunction, Copy, Enumerate
- Weitere Operatoren der Algebra
  - Extract, Delete, Match, Merge, Compose
- Teilweise implementiert in RONDO
- Much to do!

Überblick

1. Informationsintegration und schematische Heterogenität
2. Schema Mapping
3. Schema Matching
4. Peer Data Management (PDMS)
5. Mappings und Anfrage-bearbeitung in PDMS
6. Weitere Themen der Arbeitsgruppe

Schema Matching – Motivation

- 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 – Der Kernalgorithmus

- Gegeben zwei Schemata mit Attributmengen A und B.
- Für jedes Attributpaar: Vergleiche Ähnlichkeit
  - bezgl. Attributnamen,
  - Bezgl. Daten, usw.
  - Ähnlichste Paare sind Matches
- Probleme:
  - Effizienz
  - Ähnlichkeitsmaß
  - Auswahl der besten globalen Matches
    - Iterativ?
    - Stable Marriage?

Schema Matching Klassifikation [RB01]

- Schema Matching basierend auf
  - Namen der Schemaelemente (*label-based*)
  - Darunterliegende Daten (*instance-based*)
  - Struktur des Schemas (*structure-based*)
  - Mischformen, Meta-Matcher

Duplicate-driven Schema Matching [BN05a]

- Instance-based Schema Matching:
  - Correspondences based on similar data values or their properties
- Conventional solution: Vertical
  - Comparison of columns
  - = Attribute classification
- Our solution: Horizontal
  - Comparison of rows
  - = Duplicate detection (despite missing attribute correspondences)

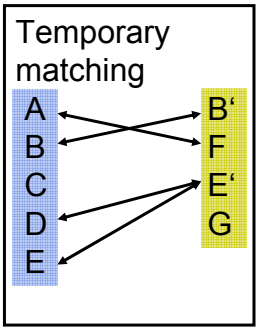
Duplicate-driven Schema Matching

A	B	C	D	E
Max	Michel	m	601- 4839204	601- 4839204
...	...	...	...	...

?

?

B‘	F	E‘	G
Michel	maxm	601- 4839204	UNIX
...	...	...	...

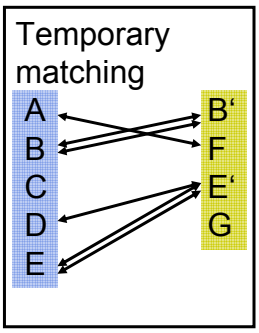


Duplicate-driven Schema Matching

A	B	C	D	E
Max	Michel	m	601- 4839204	601- 4839204
Sam	Adams	m	541- 8127100	541- 8121164

? ? ? ?

B‘	F	E‘	G
Michel	maxm	601- 4839204	UNIX
Adams	beer	541- 8127164	WinXP



- Assumptions
  - There is data in both DBs.
  - There are (at least a few) duplicates in both DBs.
  - Equal or similar values reflect same semantics of attributes.

Schema Matching – Erweiterungen

- 1:n, n:1 Matches
  - Vorname, Nachname → Name
    - Viele Kombinationsmöglichkeiten
    - Viele Funktionen denkbar: Mathematische Operatoren, Konkatenation, etc.
  - Name → Vorname, Nachname
    - Viele Kombinationsmöglichkeiten
    - Viele Parsingregeln
- Globales matching
  - Matche nicht nur einzelne Attribute (oder Attributmengen)
  - Sondern komplette Tabellen oder komplette Schemata
  - Stable Marriage Problem
  - Maximum Weighted Matching Problem

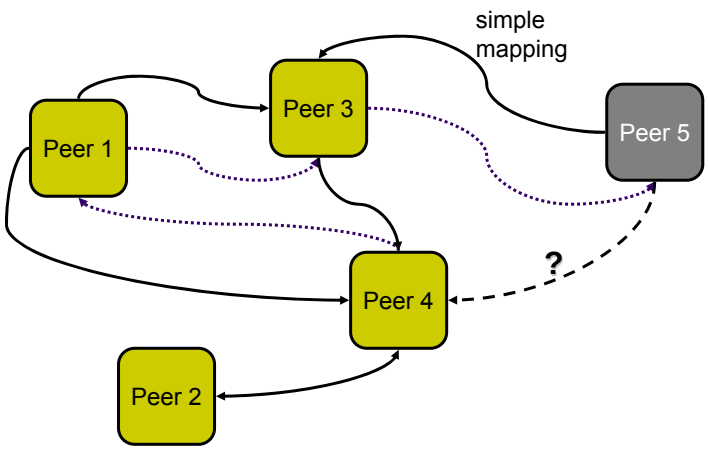
Überblick

1. Informationsintegration und schematische Heterogenität
2. Schema Mapping
3. Schema Matching
4. Peer Data Management (PDMS)
5. Mappings und Anfrage-bearbeitung in PDMS
6. Weitere Themen der Arbeitsgruppe

PDMS – Idea

- Idea: Peer network (P2P)
  - [HIST03], [HIMT03], [BGK+02]
- Each peer can
  - Export data (= data source)
  - Provide views on data (= wrapper)
  - Accept and forward queries of other peers (= mediator)
- Schema Mappings
  - Not between local and global schema
  - but between peers schemas

Peer-Data-Management Systems (PDMS)



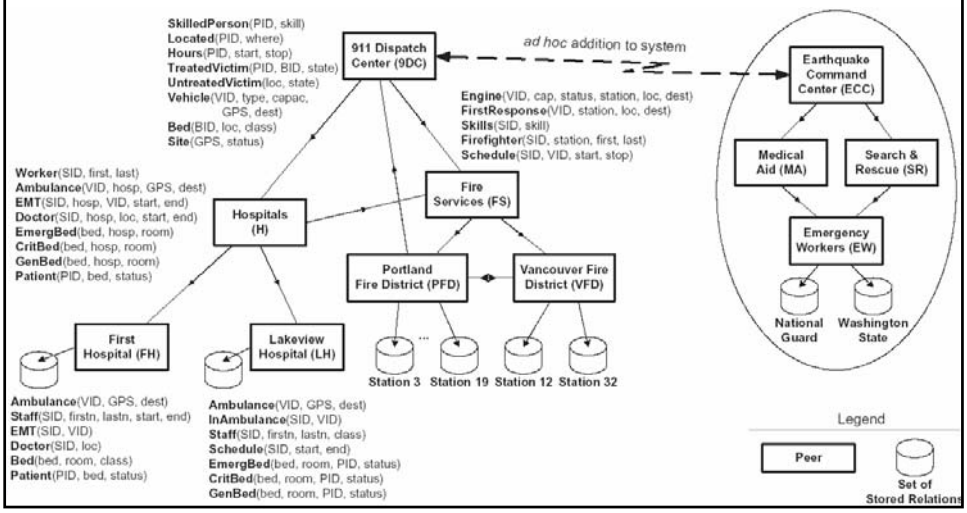
PDMS vs. P2P file sharing

P2P		PDMS
Only complete files (low granularity)	⟺	Objects (high granularity)
Simple queries • Filename	⟺	Complex queries • Query language (SQL, etc.) • Possibly search queries
Incomplete query response	⟺	Complete quers response expected
No schema • Exception: Napster for music files	⟺	Schema
Highly dynamic	⟺	Usual Assumption: Controlled dynamics

PDMS Applications

- Health information system
  - Hospital data distributed on many systems
  - Doctors want to distribute only parts of their data
  - *Content-management*-like search is important.
  - Complex and heterogeneous schemata
  - Added-values for patients through data sharing
- Life sciences data
  - Labs have the will and the duty to freely publish data.
  - Complex schemata and complex queries
  - Known relationships among the data and schemata
  - Creation of global schema difficult to impossible
- Automobile industry
- Catastrophe management

Piazza – Example [HIST03]



PDMS vs. FDBMS

- Advantages
  - Users need to know only their own schema.
  - All data is reachable (transitive closure of mappings).
  - Adding new schemata and peers is incremental and easy.
  - Mapping only to most similar schema
- Disadvantages / Problems
  - Finding mappings automatically (schema matching)
  - Mapping composition
  - Many mapping steps
    - Efficiency
    - Scalability
    - Data Quality
  - Efficient data placement
  - Read-only or updates?

Überblick

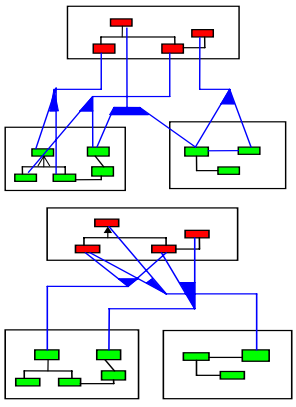
1. Informationsintegration und schematische Heterogenität
2. Schema Mapping
3. Schema Matching
4. Peer Data Management (PDMS)
5. Mappings und Anfrage-bearbeitung in PDMS
6. Weitere Themen der Arbeitsgruppe

Modeling Data Sources

- Main idea
  - Schema Mapping: Model structurally heterogeneous source schemas to a global schema as views.
  - Relational model
  - In general: A view combines multiple relations and produces one relation.
  - Here: A view on relations of one schema produces a relation of the other schema.

Global as View / Local as View

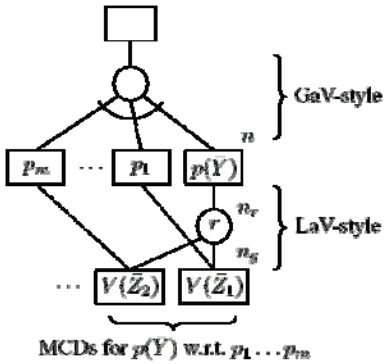
- Global as View
  - Relations of the global schema are expressed as views on the local sources.
- Local as View
  - Relations of the local source schemas are expressed a s views on the global schema.

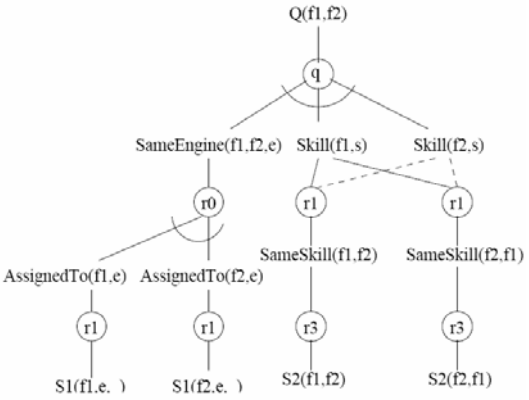


Query Answering in PDMS

- "Syntactic & structural" challenges (certain answers)
  - No centralized control
  - Long mapping paths
  - Mixed GaV/LaV query reformulation
  - Cycles
  - Routing query responses
  - Scalability
- "Semantic" challenges (best answers)
  - Peer selection & Data Lineage
  - Completeness
  - Information Quality

Query Answering in PDMS – The rule-goal tree





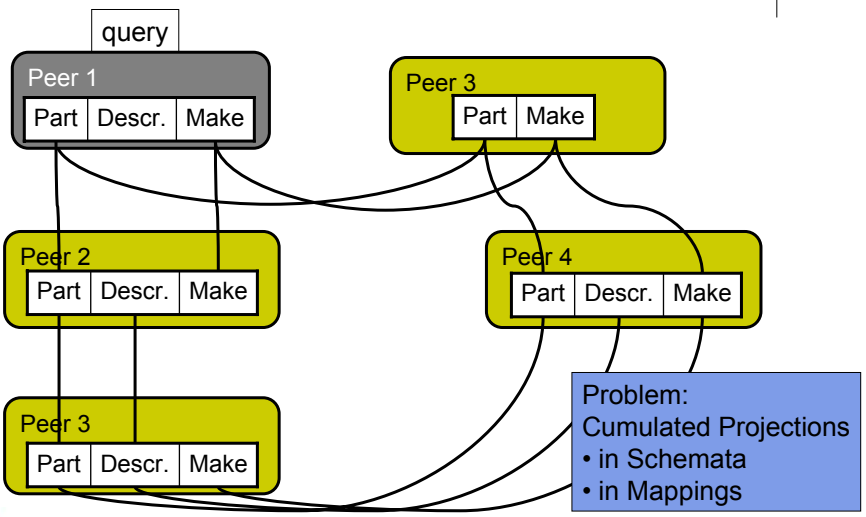
PDMS and Scalability

- Flexibility
  - Flexible and rapid modeling
    - In particular LaV
  - Mapping to one or more similar peers
    - Schema matching helps
  - Full query language on all data!
- Conceived for tens of sources, not more
  - Finding CERTAIN answers is complex.
  - Finding ALL certain answers is complex.

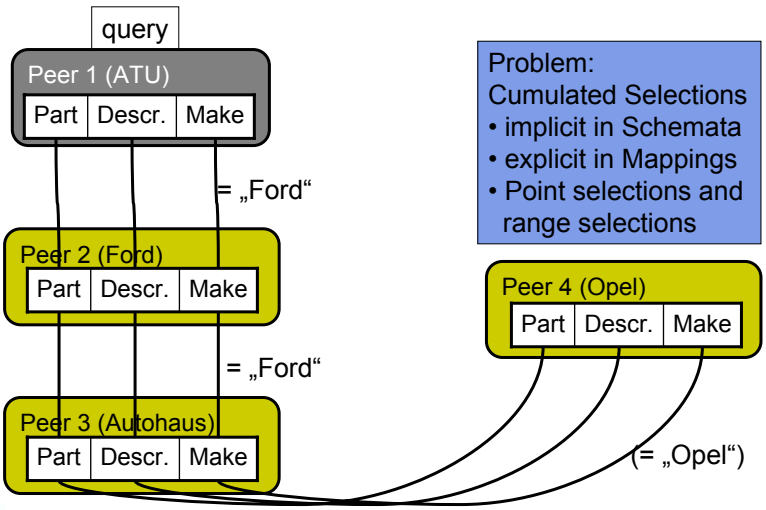
Improving Scalability

1. Concessions towards completeness
   - Not all certain answers (coverage)
   - Not all query attributes (density)
   - Prune the rule-goal-tree
     - Using completeness and other IQ criteria
   - I.e.: Making it more P2P-y (GRID-y)
2. Take P2P idea further
   - On demand source discovery using P2P Index
   - On the fly schema mapping using matching
     - Automatic data integration = autonomic data integration
   - „Ontology shortcuts“

Incomplete Mappings



Selective Mappings



The effect of selections and projections

- Completeness of data suffers
  - Extensional completeness (= coverage)
    - Number of tuples reached
    - Compared to all certain answers
  - Intensional completeness (= density)
    - Number of attributes reached
    - Measured for each relation and source
    - Compared to all attributes of the query
    - Conventional PDMS and Query Answering using views: density = 1

Optimization in PDMS

- Idea: Do not find all certain answers, just find some.
- Idea: Do not demand all attributes, just some.
- Optimization goal:
  - Maximize completeness given some cost constraint
- Cost:
  - Response time / latency
  - Number of peers
  - Number of bytes / network load
  - $$$
- Main problem: Strictly local optimization – no global knowledge!
- Opportunity for growth: Scalable PDMS

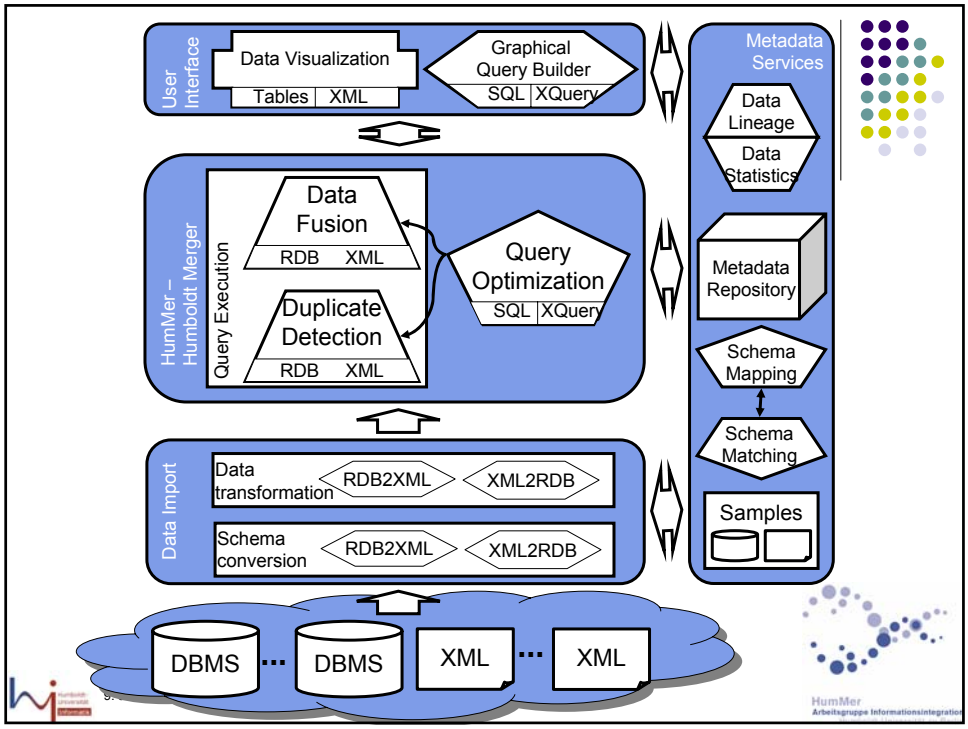
Pruning the rule-goal-tree

- Predict Completeness
  - Selectivity estimation on steroids
  - Complex formulas
  - Many assumptions (independence)
  - Many overlap variations
- Different strategies
  - Threshold for completeness – direct pruning
  - Budget-approach – direct budget along promising mappings

Ongoing work

Überblick

1. Informationsintegration und schematische Heterogenität
2. Schema Mapping
3. Schema Matching
4. Peer Data Management (PDMS)
5. Mappings und Anfrage-bearbeitung in PDMS
6. Weitere Themen der Arbeitsgruppe



Algorithm [WN05]

1. Create data structure
   - XQueries to extract relevant elements
     - Elements of one type at one level
     - plus descendants
2. Acceleration of similarity comparisons
   - Edit-distance filter
3. Avoidance of similarity comparisons
   - Element-similarity-filter
   - Connected components
4. Similarity comparisons
   - Among remaining elements
   - $sim(e_1,e_2) \geq t_{dup}$

Supported by graph-based data structure:

Similarity of tokens are edges between token-nodes

Similarity of elements are edges between element-nodes

Fuse By – Queries [BN05b]

```
SELECT *
FROM Q1
FUSE BY (Name)
```

Grouping with coalesce aggregation

```
SELECT *
FROM Q1
FUSE BY ()
```

Subsumption

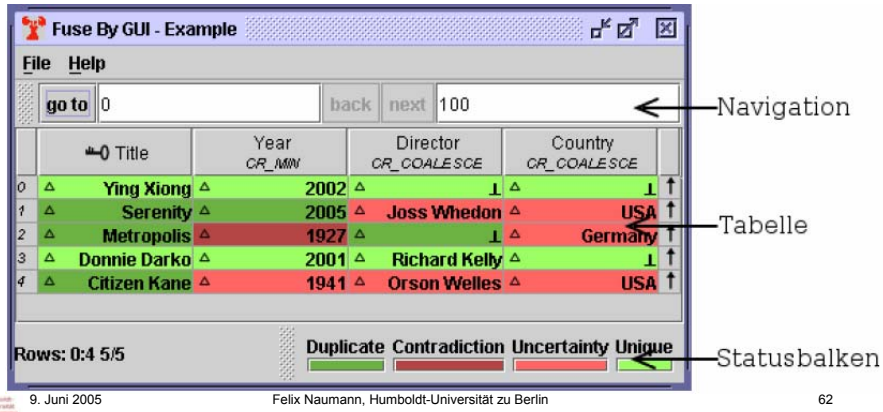
```
SELECT *
FROM Q1, Q2
FUSE BY ()
```

Minimum Union

```
SELECT Name, RESOLVE(Age, max), RESOLVE(Student,
       vote), RESOLVE(Place), RESOLVE(Phone)
FROM Q1, Q2
FUSE BY (Name) ON ORDER Q2.Age DESC
```


Visualisierung Integrierter Daten

- Why Provenance & Where Provenance
- Konflikte en detail und im Überblick (zoom out)



Research Group – Acknowledgments

- PhD students
  - Jens Bleiholder (bleiho@informatik.hu-berlin.de)
    - Information fusion for relational data
  - Melanie Weis (mweis@informatik.hu-berlin.de)
    - Object identification for XML data
  - Armin Roth (aroth@informatik.hu-berlin.de)
    - Data Quality in Peer-Data-Management-Systems
  - Alexander Bilke (bilke@cs.tu-berlin.de)
    - Schema Matching
- Topics of masters students (“Diplom”)
  - Classification of schema mapping tools
  - Schema integration using schema mappings
  - Meta Schema-matching
  - Sorted neighborhood in XML data
  - Metasearching using DB2 II

Fragen?

naumann@informatik.hu-berlin.de

Heterogenität
Schema Mapping
Schema Matching
Peer Data Management (PDMS)
Anfragebearbeitung in PDMS
Weitere Themen der Arbeitsgruppe

Literatur

- [BN05a] Alexander Bilke and Felix Naumann: Schema Matching using Duplicates, ICDE, 2005.
- [BN05b] Jens Bleiholder and Felix Naumann: Declarative Data Fusion - Syntax, Semantics, and Implementation, ADBIS, 2005.
- [WN05] Melanie Weis and Felix Naumann: DogmatiX Tracks down Duplicates in XML, SIGMOD, 2005.
- [BB+05] Alexander Bilke, Jens Bleiholder, Christoph Böhm, Karsten Draba, Felix Naumann and Melanie Weis: Automatic Data Fusion with HumMer, VLDB, 2005, Demonstration.

- [Ber03] Philip A. Bernstein: Applying Model Management to Classical Meta Data Problems. CIDR 2003
- [BLP00] A Vision for Management of Complex Models. Philip A. Bernstein, Alon Y. Levy, Rachel A. Pottinger, MSR-TR-2000-53, 2000.
- [FHP+02] Ron Fagin, Mauricio Hernandez, Lucian Popa, Renee Miller, and Yannis Velegrakis, Translating Web Data, VLDB 2002, Hong Kong, China.
- [Halevy04] Alon Halevy: SSS, Invited talk at VLDB 2004, Toronto.
- [LSS96] Lakshaman, Sadri, Subramanian, SchemaSQL – A Language for Interoperability in Relational Mulitdatabase Systems, in VLDB 1996
- [MRB03] S. Melnik, E. Rahm, P. A. Bernstein: Rondo: A Programming Platform for Model Management, in Proc. ACM SIGMOD 2003, San Diego, June 2003
- [ÖV99] Principles of Distributed Database Systems, M. Tamer Özsu, Patrick Valduriez, Prentice Hall, 1999.
- [RB01] Erhard Rahm and Philip Bernstein, A survey of approaches to automatic schema matching, VLDB Journal 10(4), 2001.