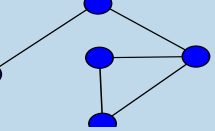
Vorlesung: P2P und Datenbanken (4)

Dr. Dieter Sosna

11. Juni 2008



Kapitel 4: P2P und DB

Widersprüche P2P - DB

Technische Hilfen

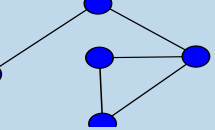
Integration auf welchem Level ?

Metadatenbasiert

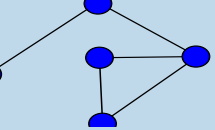
Instanzdatenbasiert

Schemaebene

MOMA (reuse)

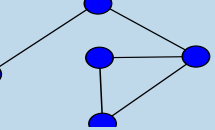


Widersprüche P2P - DB



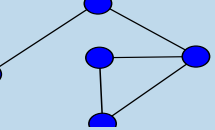
Anforderungen für P2P-DB:

- komplexes Schema, d.h.
  Definition: *komplex*
  es ex. eine Schemabeschreibungssprache, mit der Schemata definiert und erweitert werden können.
  $\mapsto$ verschiedene Knoten mit evt. versch. Schemata bzw. Sichten auf Daten - Heterogenität.
  $\mapsto$ im Schema Metainformationen über Daten (Autor, Gebiete, ..) in starkem Maße vorhanden und Gegenstand von *komplexen* Anfragen (incl. Anfragen nach Metadaten).
  $\mapsto$ verschiedene Operatoren, z.B. JOIN, Gruppierung, Aggregate-, Sortierfunktionen über mehrere Knoten (ggf. im gesamten Netz)
- Mehrschichtige Architektur, meist basierend auf Hash-Tabellen und Routingindices u.ä.; Mechanismen zur Anfrageweiterleitung, darüber Anfragebearbeitung und Nutzerinterface.
- aus P2P: Kein globales Wissen, Knotenautonomie, fluktuierende Teilnehmer und Datenbestände



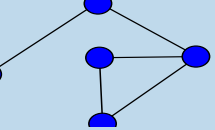
Grundprobleme

- Grundlegende Diskrepanz zwischen Transaktionseigenschaften einer DB und der Autonomie der Peers.
  Vergleichbar: Föderierte Datenbanksysteme mit Fluktuation und Autonomie der Komponenten
  Semantische Integrität i.a. nicht gesichert.
- Datenintegrationsprobleme (wie aus FDBS bekannt)
  Lösung 1: Wrapper, Mediatoren, Exportschemata
  bedingt geeignet, da globales Schema unterstellt wird.
  Lösung 2: Semistrukturierte Daten, z.B. XML
  Lit.: Risse, T. und Knesevic,P.: *A Peer-to-Peer XML Database.* Fraunhofer IPSI. 2003.
  Integration auf Instanz- oder Schemaebene.
- Ressourcenallocation: ggf. komplexer, da Metadaten mit verwaltet.
  Nutzung von Hash-Funktionen: Ausrechnen, welcher/welche Server welche Resource verwalten (ähnlich CHORD)
- Anfragebearbeitung: Zerlegung einer Anfrage in Teile für einzelne Peers, Zusammenfügen (UNION, JOIN) der Teilantworten.



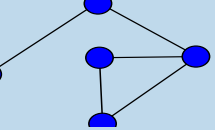
Gegenüberstellung

	P2P (2007)	?	FDBMS
Granularität	Vollständige Dateien		Objekte, Tupel, Attribute
Anfragen	Simpel: Dateinamen, Hashwerte		Komplex: SQL, XQuery
Antworten	meist unvollständig		vollständig, exakt
Update, Version	meist mangelhaft		DB-artig
Schema	höchstens Techn. Tab.		Exportschema
Systemkenntnis	kein glob. Wissen		Komponenten bek.
Anfrage-bearbeitung	Fluten, Hash, part. Kenntnis (Nachbarn)		Optimierend, 2PC
Dynamik	Persistenzproblem		Verfügbarkeit wie DBMS
Kardinalität	sehr groß		wenige DBVS

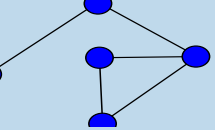


Gegenüberstellung (Fortsetzung)

	P2P (2007)	P2P-DB	FDBMS
Granul.	Vollst. Dateien	Obj., Tub., Attr.	Obj., Tub., Attr.
Anfrage	Simpel: Namen, Hashwerte	Komplex: SQL, XQuery, (?) ...	SQL, XQuery,...
Antwort	meist unvoll.	vollst. (?), exakt Qual.-Bewert.	v., e.
Versionen	meist mangelh.	DB-artig	DB-artig
Schema	Techn. Tab.	paarw.vorberechn. Integr. on the fly ?	Exp.-schema.
Sys-kenntn.	kein glob. Wissen	mind. Nachbarn glob. Struktur ?	Komponenten bekannt
Anfrage-bearb.	Fluten, Hash, part. Kenntnis (Nachbarn)	Mappings, Hash Serverfunkt. Koord. 2PC o.ä.	Optimierend, 2PC
Dynamik	keine Persistenz	Persistenz (?)	DBMS
Kardinalität	sehr groß	mittel (derzeit)	wenige DBVS



Technische Hilfen



Technische Hilfen: RDF

RDF *Resource Description Framework* (*http://www.w3.org/RDF/*)
ermöglicht die Beschreibung von WEB-Resourcen durch Metadaten,
automatisch verarbeitbar.

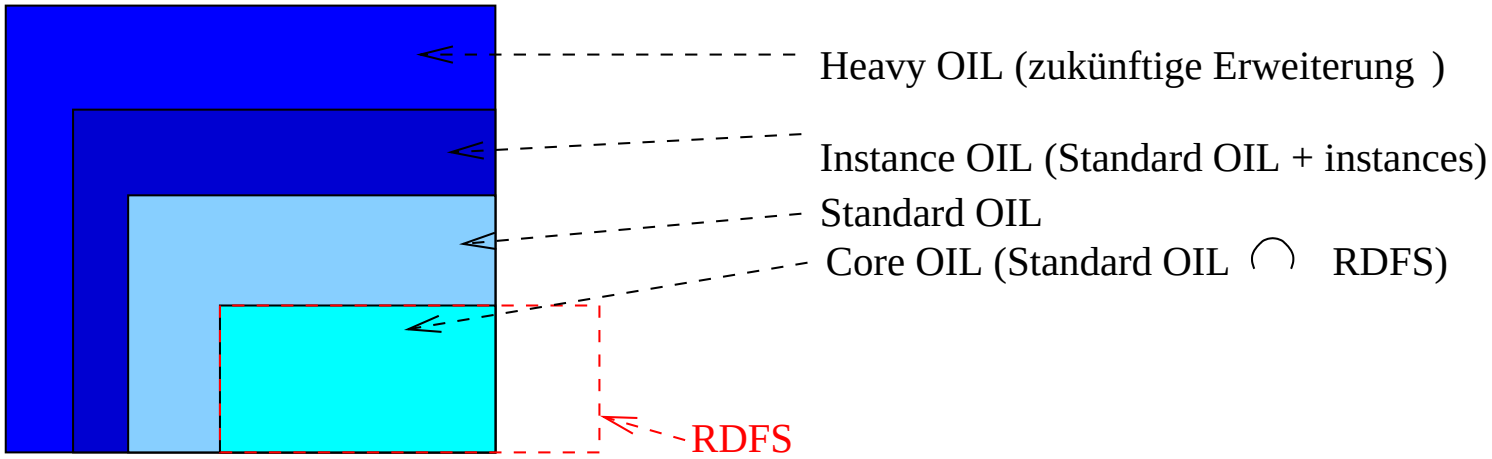
- Datenmodell: Beschreibung einer Ressource durch *properties*:

$$< property, subject, value >$$

  $subject$: die zu beschreibende Resource
  $property$: die zum Subjekt gehörige Eigenschaft
  $value$: der Wert der Eigenschaft.
- Beispiel:
  $\{Autor, [http://www.informatik.uni-leipzig.de/\sim sosna], DieterSos$
  in XML:
  <rdf:Description about='http://www.informatik.uni-leipzig.de/~sosna'>
  <autor>Dieter Sosna</autor>
  </rdf:Description>

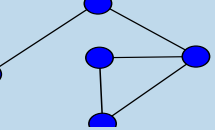
Technische Hilfen (2): OIL

OIL *Ontology Interchange Language*, *http://www.ontoknowledge.org/oil/* ermöglicht formale Definitionen für Ontologien



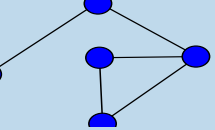
Schichtenarchitektur, abwärtskompatibel: Automaten, die untere Schichten verstehen, verstehen auch Teile aus den oberen Schichten.

- OIL-Definition besteht aus Klassen und Beziehungen. Klassen durch Eigenschaften(Attribute) beschrieben, Beziehungen = Verhalten der Instanzen.
- Core Oil: kann von RDF-Schema Agenten verarbeitet werden
  Instance OIL: Integration über Datenbankfähigkeiten.



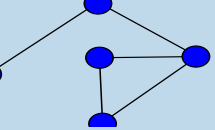
Ontologien

- το ον = das Seiende (philos.Begriff).
- ca. seit 1990 Informatik Beschreibung eines Anwendungsbereiches, der Begriffe und der Beziehungen untereinander.
  Eigenschaften:



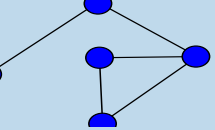
Ontologien

- το ον = das Seiende (philos.Begriff).
- ca. seit 1990 Informatik Beschreibung eines Anwendungsbereiches, der Begriffe und der Beziehungen untereinander.
  Eigenschaften:
  (1) Begriffe und Beziehungen eindeutig und unstrittig definiert.



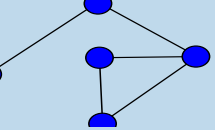
Ontologien

- το ον = das Seiende (philos.Begriff).
- ca. seit 1990 Informatik Beschreibung eines Anwendungsbereiches, der Begriffe und der Beziehungen untereinander.
  Eigenschaften:
  (1) Begriffe und Beziehungen eindeutig und unstrittig definiert.
  (2) Formal und genau: neues Wissen durch log. Schlüsse ableitbar.



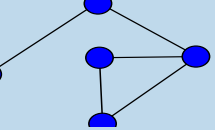
Ontologien

- το ον = das Seiende (philos.Begriff).
- ca. seit 1990 Informatik Beschreibung eines Anwendungsbereiches, der Begriffe und der Beziehungen untereinander.
  Eigenschaften:
  (1) Begriffe und Beziehungen eindeutig und unstrittig definiert.
  (2) Formal und genau: neues Wissen durch log. Schlüsse ableitbar.
  (3) Beziehungstypen mind.: is-a, part-of (, element).
- Top-Level-Ontologie: Fundamentale Beziehungen (nicht in dieser Vorlesung)
  Domänspezifische O. (Fachterminologie) - Metabeschreibung !.
  Überschneidungen der Gebiete $\rightarrow$ Anpassungen nötig. $\rightarrow$



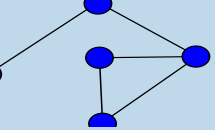
Ontologien

- το ον = das Seiende (philos.Begriff).
- ca. seit 1990 Informatik Beschreibung eines Anwendungsbereiches, der Begriffe und der Beziehungen untereinander.
  Eigenschaften:
  (1) Begriffe und Beziehungen eindeutig und unstrittig definiert.
  (2) Formal und genau: neues Wissen durch log. Schlüsse ableitbar.
  (3) Beziehungstypen mind.: is-a, part-of (, element).
- Top-Level-Ontologie: Fundamentale Beziehungen (nicht in dieser Vorlesung)
  Domänspezifische O. (Fachterminologie) - Metabeschreibung !.
  Überschneidungen der Gebiete $\rightarrow$ Anpassungen nötig. $\rightarrow$ Ontologiematching



Nutzen der Ontologien

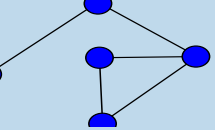
- Einheitliche Begriffswelt (kontrolliertes Vokabular):
  Übername des Vok. in Daten sichert Vergleichbarkeit von Daten,
  Hilfe bei Überwindung von Heterogenität (z.B. Synonyme)
  14:00 Uhr kann bei Matchprozeduren helfen, die Semantik zu erhennen:
  Matching gegen Standard.
- Gene Ontologie: ca. 17000 Begriffe (Molekülchemie, (molekular-) biolog.
  Prozesse.
  Struktur: Konzepte, is-a- und part-of- Beziehung.
  Inhalte: von Experten erzeugt, Internationale Konsortium, sehr gut akzeptiert, da Nutzen offensichtlich - Quasistandard.
  Benutzung: tool-unterstützt.
- Praktisch wird Begriff der O. im stark erweiterten Sinn genutzt:
  Liste von Konzepten, Taxonomien, Tessauri, Polyhierarchien, Graphen



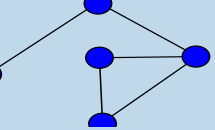
Ontologiebasierte Informationsintegration

Lit: Leser, Naumann: *Informationsintegration*. dpunkt,
ISBN 978-3-89864-400-6 42

- 3 Schritte:
  Erstellung der globalen Ontologie
  Einordnung der Datenquellen
  Subsumption zur Anfragebearbeitung:
  Anfragen (z.B. nach Gleichheit , ...) als Konzepte formuliert. Alle Konzepte, die spezieller als das Anfragekonzept sind und eine Datenquelle repräsentieren, enthalten dann nur semantisch korrekte Objekte.
- Ontologie als Klammer über die Atonomie ?!



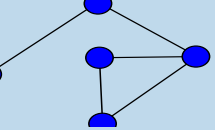
Integration auf welchem Level ?



Integration auf Instanzebene

Local Relational Model (LRM)
Lit.: Bernstein,A. Ph. u.a.: *Data Management for Peer-to-Peer Computing: A Vision*. http://www.db.ucsd.edu/webdb2002/papers/15.pdf

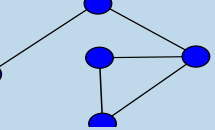
- Architektur:
  auf lokalem DBVS in allen Knoten eine identische Schnittstelle (LPM-Ebene) mit User-Interface (UI), Query-Manager (QM), Update-Manager (UM) und Wrapper(zwischen UM,QM und der lokalen DB).
- Bekanntschaften, Koordinierungsformeln:
  2 Peers haben Bekanntschaft, wenn es Koordinierungsformeln gibt, die semantische Beziehungen / Abhängigkeiten zwischen deren Daten beschreiben, die zeigen, wie sich die eigenen Daten für den Bekannten darstellen, wie die Elemente der einen DB in Elemente der anderen DB übersetz werden müssen (Binäre Domainbeziehungen).
- Regeln formal beschrieben.



Integration auf Instanzebene (2)

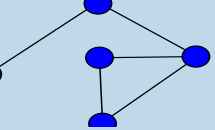
Lit.: Arenas, M. u.a.: *The Hyperion Project: From Data Integration to Data Coordination*. ACM SIGMOD Record, 32(3), 2003

- Architektur ähnlich LRM, statt UM jetzt Rulemanager (RM), der Konsistenzregeln, die über Nutzerschnittstelle eingegeben werden, durchsetzt
- Mappingtabellen definieren die Überführung von Werten zwischen je zwei DB.
- Das Aufstellen der Mappingtabellen erfordert Kooperation der Partner, geschieht mit Nutzerinteraktion.



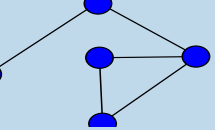
Schemamatching

- Voraussetzung: Gegeben 2 Quellen mit zugehörigen Metadaten und Instanzdaten.
  Ziel: Finden von semantisch gleichen Konzepten.
- 2 Ansätze:
  - Metadatenbasiert (Namen, Beschreibungen, Ontologie, Struktur (z.B. auch Fremdschlüsselbeziehungen))



Schemamatching

- Voraussetzung: Gegeben 2 Quellen mit zugehörigen Metadaten und Instanzdaten.
  Ziel: Finden von semantisch gleichen Konzepten.
- 2 Ansätze:
  - Metadatenbasiert (Namen, Beschreibungen, Ontologie, Struktur (z.B. auch Fremdschlüsselbeziehungen))
  - Instanzbasiert
    Grundannahme: Zwei Konzepte sind ähnlich, wenn sie eine hinreichend große Anzahl gleicher oder zumindest (sehr) ähnlicher Elemente haben.

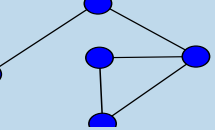


Mißerfolg erwartet

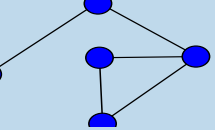
Gundproblem:
Aus formalen Merkmalen (Namensgleichheit, Strukturgleichheit, Häufigkeitsverteilung, ...) soll

auf semantische Ähnlichkeit

geschlossen werden.
Mit anderen Worten:
Schemamatching ist Forschungsgegenstand. Das Ziel der automatisierten Verfahren ist noch nicht erreicht.
Für jedes Verfahren lassen sich Negativbeispiele finden
$\rightarrow$ Kombination von Verfahren könnte Resultate verbessern. Problem: Wie kombinieren ?



Metadatenbasiert

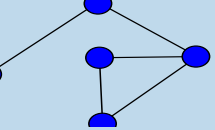


Metadaten: Indikatoren /Berechnungen für Gleichheit

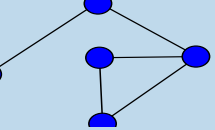
- Metadaten in (relat.) DB unzureichend dargestellt.
- Namensvergleiche:
  Gleichheit (! Homonyme, bei XML Lösung durch Namensräume)
  Gleichheit nach Normalisierung (Großschreibung, stemming, Übersetzung)
  Hyperonymie (hierarch. Beziehung is-a , Thessausus, Ontologie,Taxonomie)
  Ähnlichkeit
- Strukturvergleiche:
  Cupid (1): Schemata $\rightarrow$ Bäume . Konzepte ähnlich, wenn Eltern, Kinder, Brüder ähnlich sind; bei Blättern: Namensähnlichkeit.
  Similiarity-Flooding (2) : Schematapaar $\rightarrow$ Graphen. Startwert: Matrix der Ähnlichkeit. Iteration: Ähnlichkeit auf Nachbarn übertagen $\rightarrow$ Fixpunktproblem. Lsg. abh. von Anfangswerten! Unabh,. von Semantik.

(1) Madhavan, Bernstein, Rahm: *Generic Schema matching with Cupid*. Proc. VLDB, 2001.

(2) Melnik, Garcia-Moulina, Rahm: *Similiarity Flooding: A Versatile Graph Matching Algorithm*. Proc. Int. Conf. Data Eng. (ICDE), 2002.



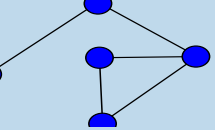
Instanzdatenbasiert



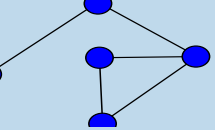
Instanzbasiertes Maching

setzt die Existenz von Instanzen in beiden Schemata voraus.

- Horizontale Matcher: Gleiche Konzepte in den Schemata durch Finden von Duplikaten erkannt.
  Vertikale Matcher: Extraktion von vorher definierten Merkmalen aus den Instanzen und Vergleich: z.B. statistische Merkmale (max, min, avg, var, covar, Clusterbildung, ...) aus den Werten der Attribute, aus Merkmalen der Attribute (Länge von Zeichenketten, ...)
- Erfahrungswert (Leser, Naumann, a.a.O) :
  Sind hinreichend viele (Statistik) Instanzen vorhanden (oder bei vert. Matchern theoret. Werte bekannt), so sind instanzbasierte Matcher (derzeit noch - D.S.) den metadatenbasierten überlegen.



Schemaebene



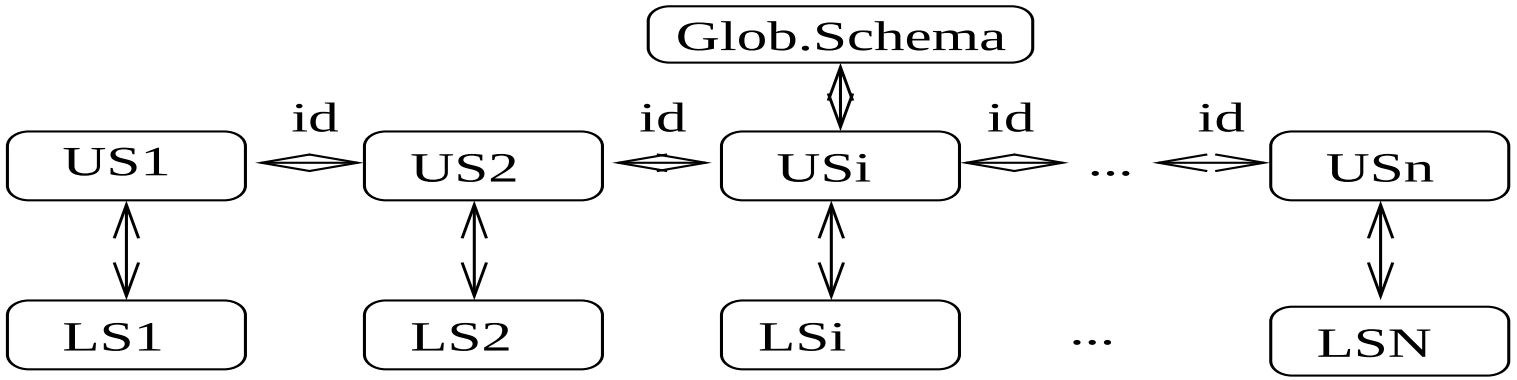
Integration auf Schemaebene

AutoMed: Automatic Generation of Mediator Tools for Heterogeneous Database Integration
Lit.: Poulovassilis, A.: *AutoMed: ....*
http://www.dcs.bbk.ac.uk/ ap/talks/abdn2003AutoMrdPres.ppt

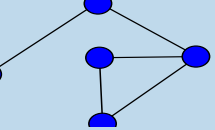
- Zwei Transformationsrichtungen:
  (1) GAV - global-as-view: Die Relationen des vermittelten (globalen) Schemas werden als Menge von Sichten der Relationen der Datenquelle beschrieben.
  (2) LAV - local-as-view: Die Relationen der (lokalen) Datenquelle werden als Sichten auf dem (globalen) vermittelten Schema dargestellt. (3) BAV - both-as-view: Kombination von LAV und GAV.
- Superpeer liefert ein globales Schema.

AutoMed (2)

Schematransformation: (US_i union-compatible Schemata)



- Grundoperationen: add, delete, rename, expand, contract; aus denen alle Umformungsregeln zusammengesetzt. Alle Modellierkonstrukte in Hypergraph Data Model (HDM) Termen dargestellt, einheitliche Beschreibungssprache.
- Model Definition Repository (MDR) - Modellierungskonstrukte Schemas & Transformations Repository (STR) - Speicher Schemata und Transformationen, Schema Transformation & Integration Tool Schaffung neuer Zwischen-Schemata und Transfomation Pathways.



Integration auf Schemaebene Piazza

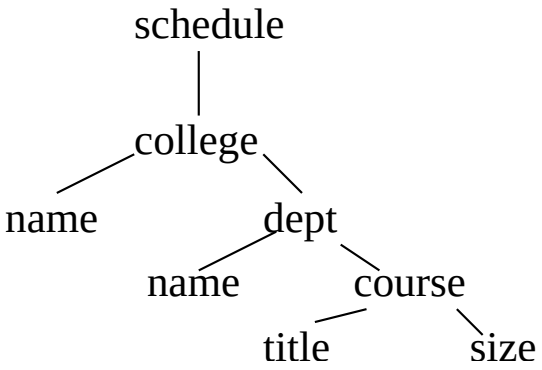
Lit.: Halevy,A. u.a.: *Piazza: Data Management Infrasructure for Semantic Web Applications*.
http://www.cis.upenn.edu/ zives/rearch/piazza-www03.pdf

- Piazza nutzt transitive Beziehungen zwischen den Schemata der beteiligten Peers.
- Kooperative Peers vorausgesetzt, die paarweise Mappings ihrer Schemata definieren. Eingebrachte Resourcen:
  (1) Daten (XML-, RDF-Dateninstanzen) (2) Metadaten (XML-Schema, Ontologien,..)
- Mappings:
  (1) vermittelndes Mapping: Daten durch verm. Schema od. Ontologie verbunden
  (2) Punkt-zu-Punkt Mapping: direkte Transformation zum anderen Peer beschrieben.
- GAV- und LAV-Verfahren auf XML verallgemeinert.
  Mappingsprache: Teilmenge von XQuery.

Piazza Beispiel

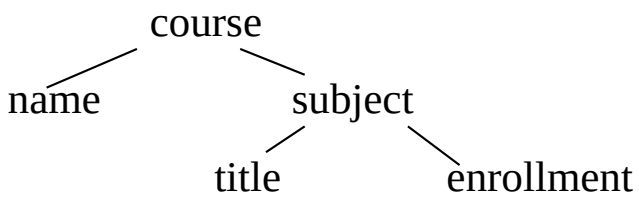
Berkley peer schema (XML DTD):

Elemant schedule(college*)
Element college(name, dept*)
Element dept(name, course*)
Elemant course(title, size)



MIT peer schema:

Element catalogue(course*)
Element course(name, subject*)
Element subject(title,enrollment)



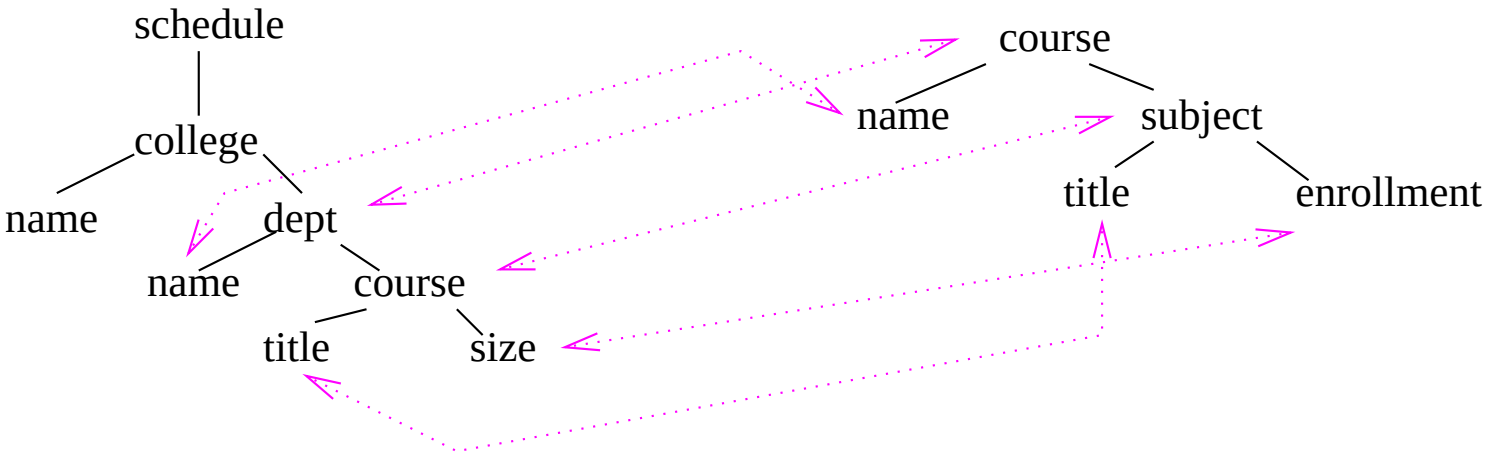
Piazza Beispiel (2)

Berkley peer schema (XML DTD):

Elemant schedule(college*)
Element college(name, dept*)
Element dept(name, course*)
Elemant course(title, size)

MIT peer schema:

Element catalogue(course*)
Element course(name, subject*)
Element subject(title,enrollment)



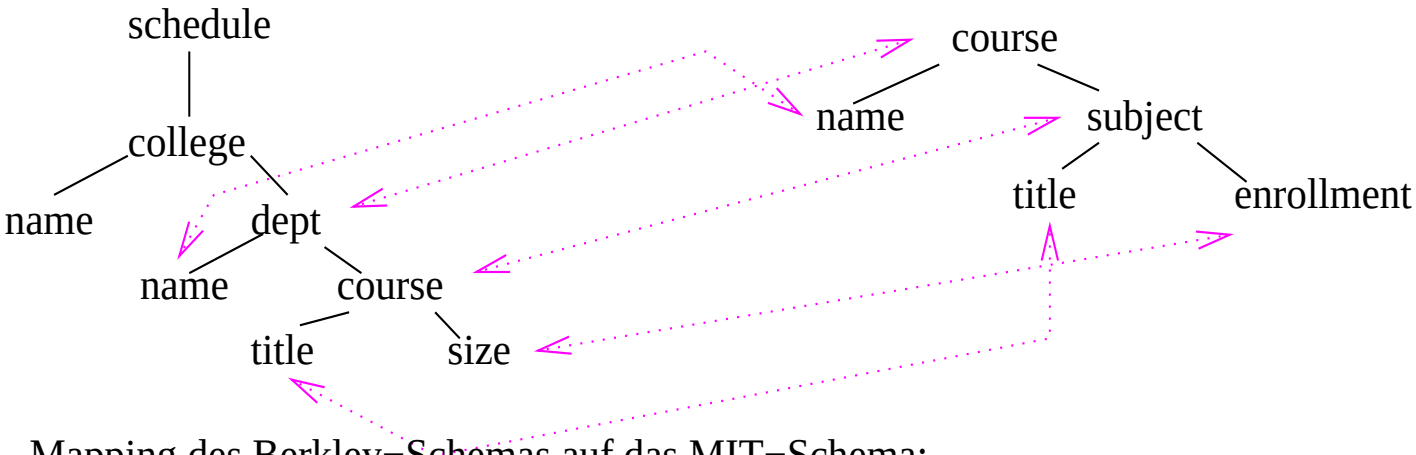
Piazza Beispiel (3)

Berkley peer schema (XML DTD):

Elemant schedule(college*)
Element college(name, dept*)
Element dept(name, course*)
Elemant course(title, size)

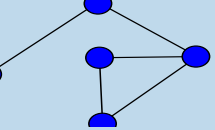
MIT peer schema:

Element catalogue(course*)
Element course(name, subject*)
Element subject(title,enrollment)

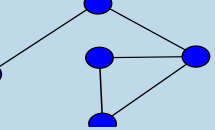


Mapping des Berkley−Schemas auf das MIT−Schema:

```
<catalog>
    <course> {$c=document("Berkley.xml")"/schedule/college/dept}
        <name> $c/name/text() </name>
        <subject> {$s = $c/course}
            <title> $s/title/text() </title>
            <enrollment> $s/size/text() </enrollment>     </subject></course></catalog>
```



MOMA (reuse)



Mapping-Verarbeitung

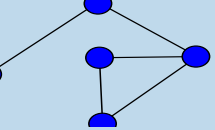
Motivation

- Matching ist i.A. sehr aufwändig: Viele Ähnlichkeitsvergleiche, manuelle Überprüfung, ...
- Matching ist i.A. sehr schwierig: Welcher Match-Algorithmus? Welche Parameter? ...
- Match-Ergebnis ist "wertvoll" und sollte wiederverwendet werden

Ziele

- Wiederverwendung von Match-Ergebnissen zur effizienten Berechnung neuer Match-Ergebnisse
- Kombination von Match-Ergebnissen zur Qualitätsverbesserung
- Bestimmung von Match-Ergebnissen, wenn kein geeignetes Ähnlichkeitsmaß zur Verfügung steht

Dank an Herrn A. Thor für die Bilder und Informationen zu MOMA.

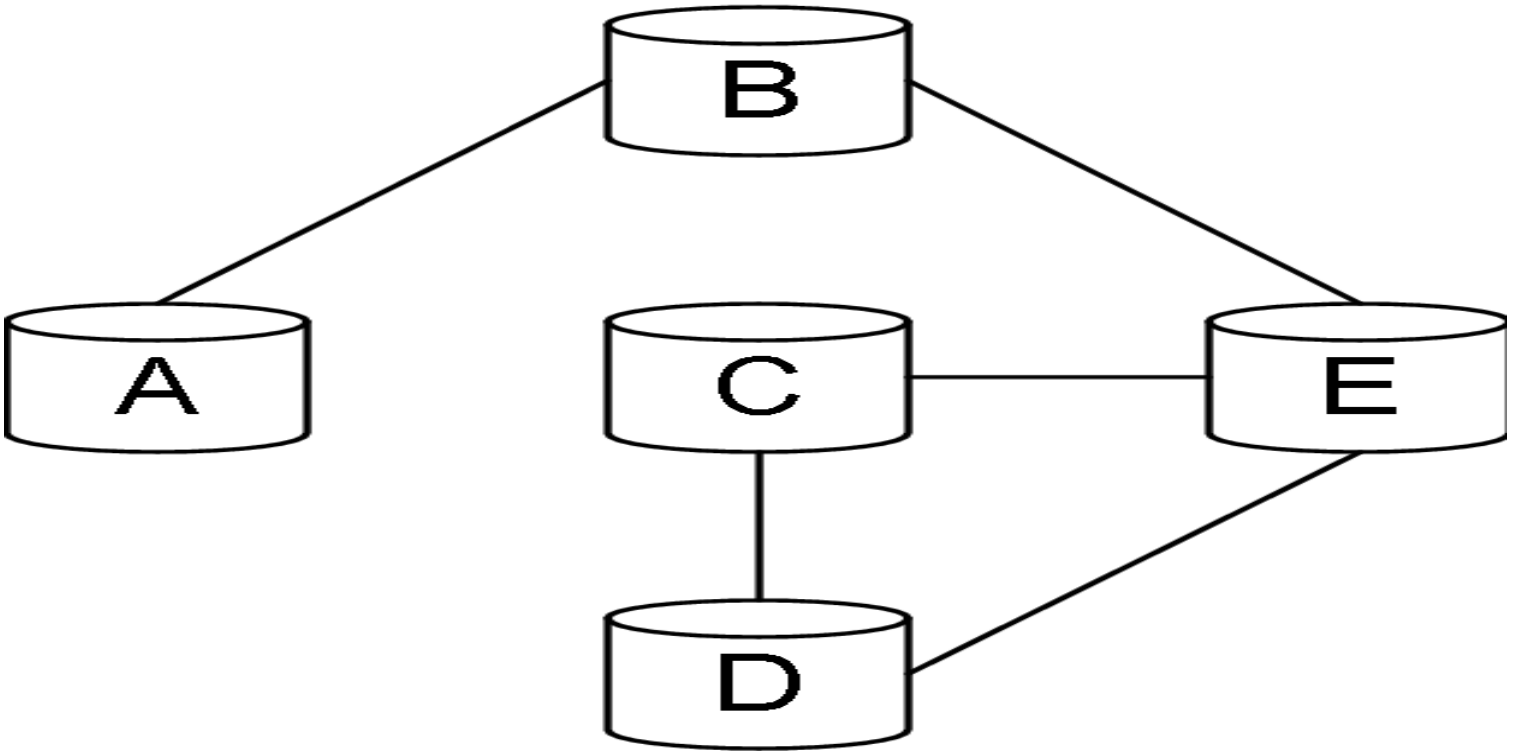


Moma und P2P

- Bisher P2P-DB als Forschungsansatz
- Kombination von Match-Ergebnissen bietet Ansatz:
  - Neuer Knoten meldet sich an, bekommt dadurch Nachbarn, Durchführung Match mit Nachbarn.
  - Services: Verteilung von Matches, Kombination von Matches, Speicherung von Matches Berechnung optimaler Wege.
- **Diplomthemen: Qualität von Matchkombinationen**

Mapping-Verarbeitung: Beispiel

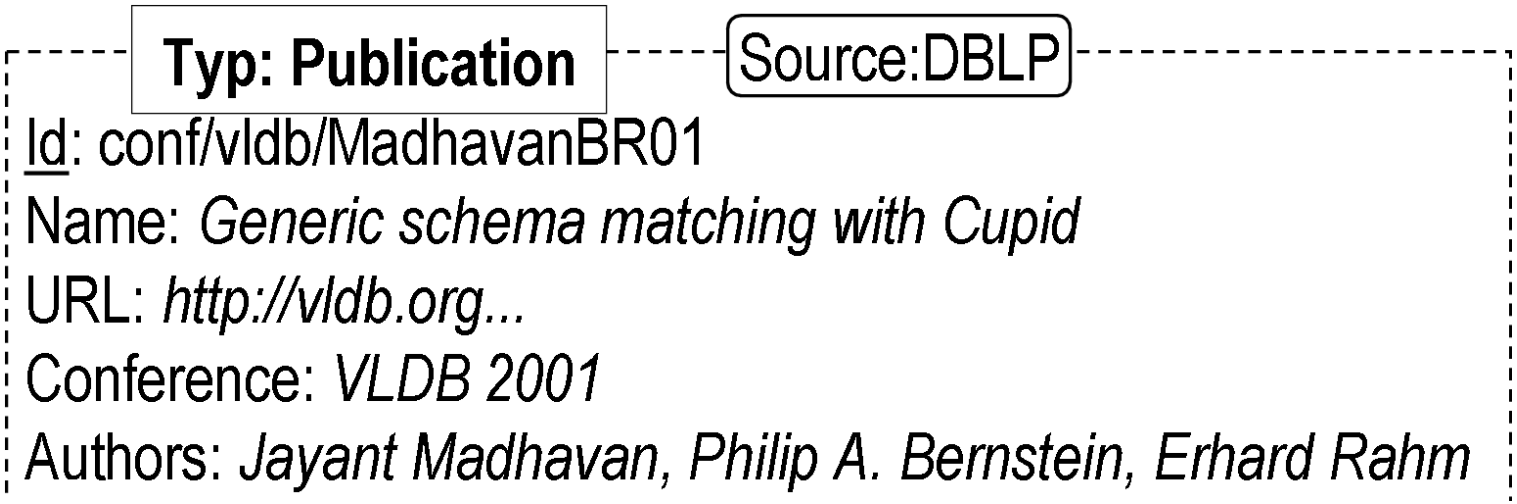
- Effiziente Berechnung: (A,E) mittels (A,B) und (B,E)
- Qualitätsverbesserung: Kombination von (D,E) direkt mit (D,C) + (C,E)
- Kein geeignetes Ähnlichkeitsmaß: (A,D) mittels (A,B) + (B,E) + (E,D)



MOMA-Ansatz: Begriffe (1)

Definition: Datenquelle (Logische Datenquelle, LDS)

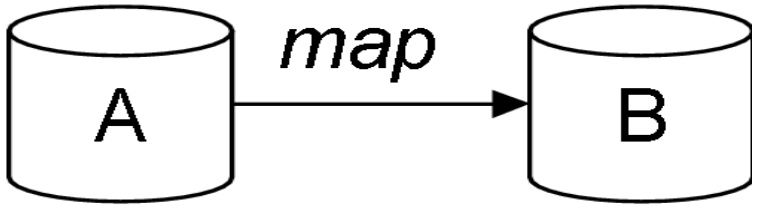
- Menge von Objektinstanzen
- Alle Objekte haben den gleichen semantischen Typ (z.B. Publikation)
- Jedes Objekt hat eine (innerhalb der Datenquelle) eindeutige Id und beliebige zusätzliche weitere Attribute
- Beispiel: Datenbanktabelle, Website, XML-Dokument, ...



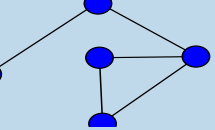
MOMA-Ansatz: Begriffe (2)

Definition: Same-Mapping

- $\{(a, b, s) | a \in A, b \in B, s \in [0, 1]\}$
- A und B sind Datenquellen, s ist Ähnlichkeitswert der Korrespondenz (a,b)
- Beispiel: Mapping-Tabelle, Web-Service, ...



A	B	s
a1	b1	1
a2	b2	0.9
a2	b3	0.3



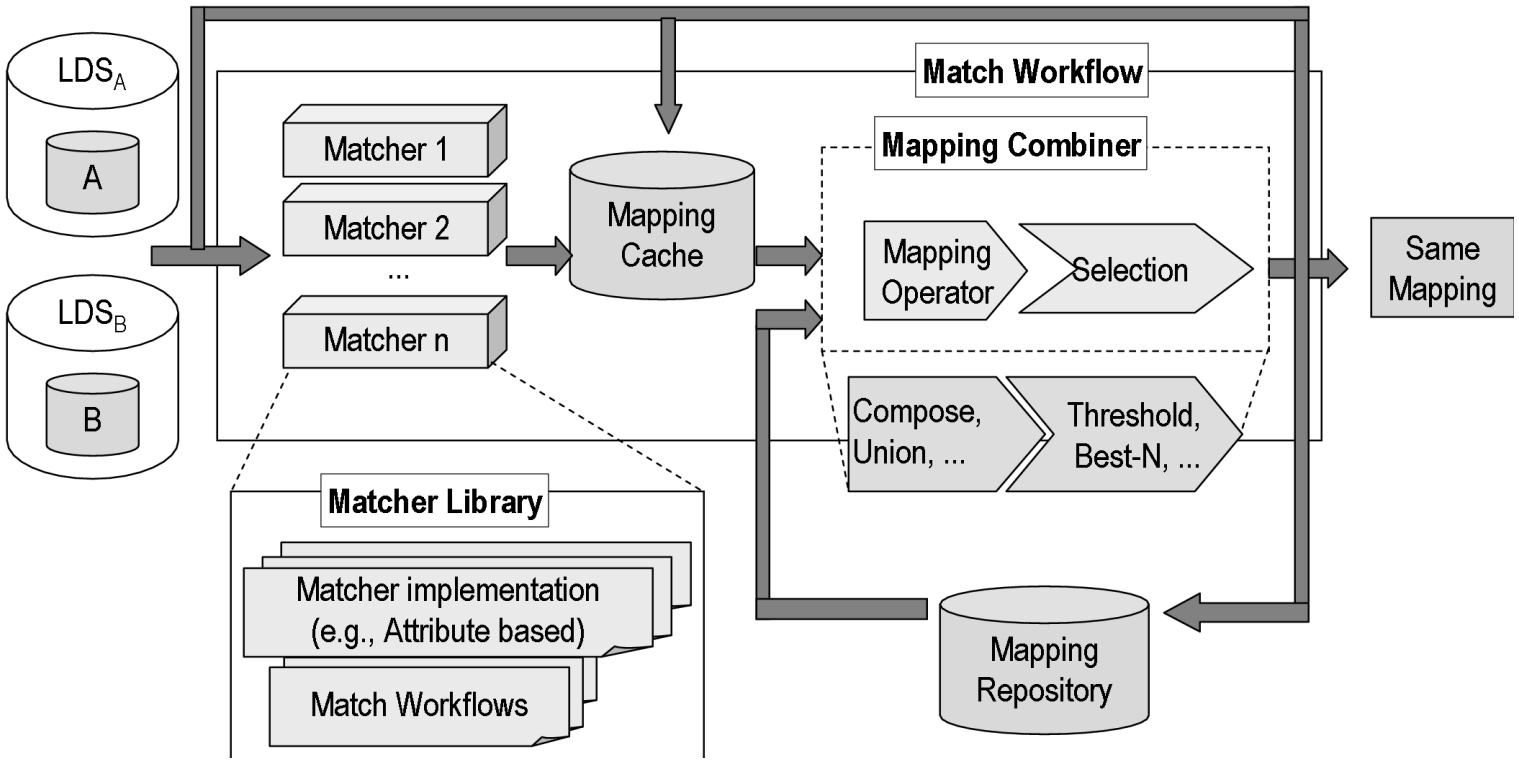
Mapping-Verarbeitung: MOMA-Ansatz

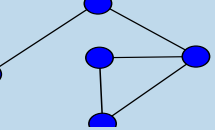
- Verarbeitung von Mappings und Objektinstanzen durch Operatoren
- Kombination der Operatorergebnisse durch Skriptsprache (iFuice*)
  - Prozedurale Programmiersprache mit Kontrollstrukturen (IF-THEN-ELSE, WHILE-DO)
  - Ergebnisse werden in Variablen gespeichert
  - Definition und Aufruf von Unterprozeduren
- MOMA = Mapping-based Object Matching
  - Definition und Ausführung von Match-Workflows
  - Eingabe: Objektinstanzen und Mappings, Ausgabe: Same-Mapping

* Rahm, E. et. al.: *iFuice - Information Fusion utilizing Instance Correspondences and Mappings*. WebDB, 2005

MOMA-Framework: Architektur

Thor, A., Rahm, E.: *MOMA - A Mapping-based Object Matching System*. CIDR, 2007





Operatoren: Übersicht (vereinfacht)

- Attributvergleich: $match(O_1, O_2, f) = map$
  - $\{(a, b, s) | a \in O_1, b \in O_2, s = f(a, b)\}$
  - f ist eine Match-Funktion, die für zwei Objekte den Ähnlichkeitswert ermittelt.
- Vereingung: $union(map_1, map_2) = map$
  - $\{(a, b, s) | (a, b, s_1) \in map_1 \vee (a, b, s_2) \in map_2\}$
- Durchschnitt: $intersect(map_1, map_2) = map$
  - $\{(a, b, s) | (a, b, s_1) \in map_1 \wedge (a, b, s_2) \in map_2\}$
- Komposition: $compose(map_1, map_2) = map$
  - $\{(a, b, s) | (a, x, s_1) \in map_1, (x, b, s_2) \in map_2\}$
- Weitere (Hilfs-)Operatoren
  - Selektion, z.B. alle Korrespondenzen deren Ähnlichkeitswert über einem Schwellwert liegen

Kombination: Vereinigung / Durchschnitt

- Ermittlung des kombinierten Ähnlichkeitswertes s durch Ähnlichkeitsfunktion $f(s_1, s_2)$
- Funktionen
  - Maximum (Max), Durchschnitt (Avg), Minimum (Min)
  - Ranked: $f(s_1, s_2) = s_1$, wenn $(a, b, s_1) \in map_1$, sonst s_2
- Umgang mit fehlenden Ähnlicheitswerten (relevant für Avg und Min)
  - Ignorieren oder "gleich Null setzen"

map1		
A	B	s
a1	b1	1
a2	b2	0.8

map2		
A	B	s
a1	b1	0.6
a3	b3	0.6

union (Max)		
A	B	s
a1	b1	1
a2	b2	0.8
a3	b3	0.6

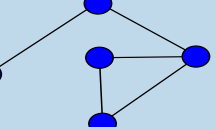
union (Ranked)		
A	B	s
a1	b1	1
a2	b2	0.8
a3	b3	0.6

union (Avg)		
A	B	s
a1	b1	0.8
a2	b2	0.8
a3	b3	0.6

union (Avg-0)		
A	B	s
a1	b1	0.8
a2	b2	0.4
a3	b3	0.3

union (Min)		
A	B	s
a1	b1	0.6
a2	b2	0.8
a3	b3	0.6

union (Min-0)		
A	B	s
a1	b1	0.6
a2	b2	0
a3	b3	0



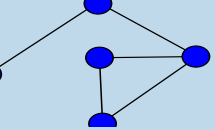
Kombination: Vereinigung / Durchschnitt (2)

- Evaluation für Publikationen von DBLP und ACM für drei attributbasierte Match-Verfahren

	Titel (Trigram)	Autoren (Trigram)	Jahr (Gleichheit)	Union-Avg (Filter:80%)
Precision	86,7%	38,0%	0,4%	97,3%
Recall	97,7%	87,9%	100,0%	93,9%
F-Measure	91,9%	53,1%	0,8%	95,5%

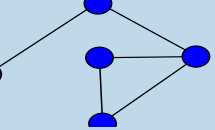
F-Measure: folgende Folie

- Fazit
  - Kombination kann Match-Qualität steigern
  - Vereinigung verbessert Recall (evtl. auf Kosten der Precision)
  - Durchschnitt verbessert Precision (evtl. auf Kosten des Recalls)
  - Wahl der Ähnlichkeitsfunktion von Match-Problem abhängig



F-Mesure

- Das $f_\alpha - Mesure$ ist ein
  gewichtetes harmonisches Mittel aus $precision$ und $recall$:
  Seien $a, b > 0$ und $\alpha = a/b$, dann wird f_α definiert durch
  .
  $(a+b) \times \frac{1}{f_\alpha} = a \times \frac{1}{precision} + b \times \frac{1}{recall}$
  .
- $f_\alpha = \frac{(1+\alpha) \times precision \times recall}{\alpha \times precision + recall}$
- Aus $\alpha \to \infty$ folgt $f_\alpha \to recall$,
  aus $\alpha \to 0$ folgt $f_\alpha \to precision$.



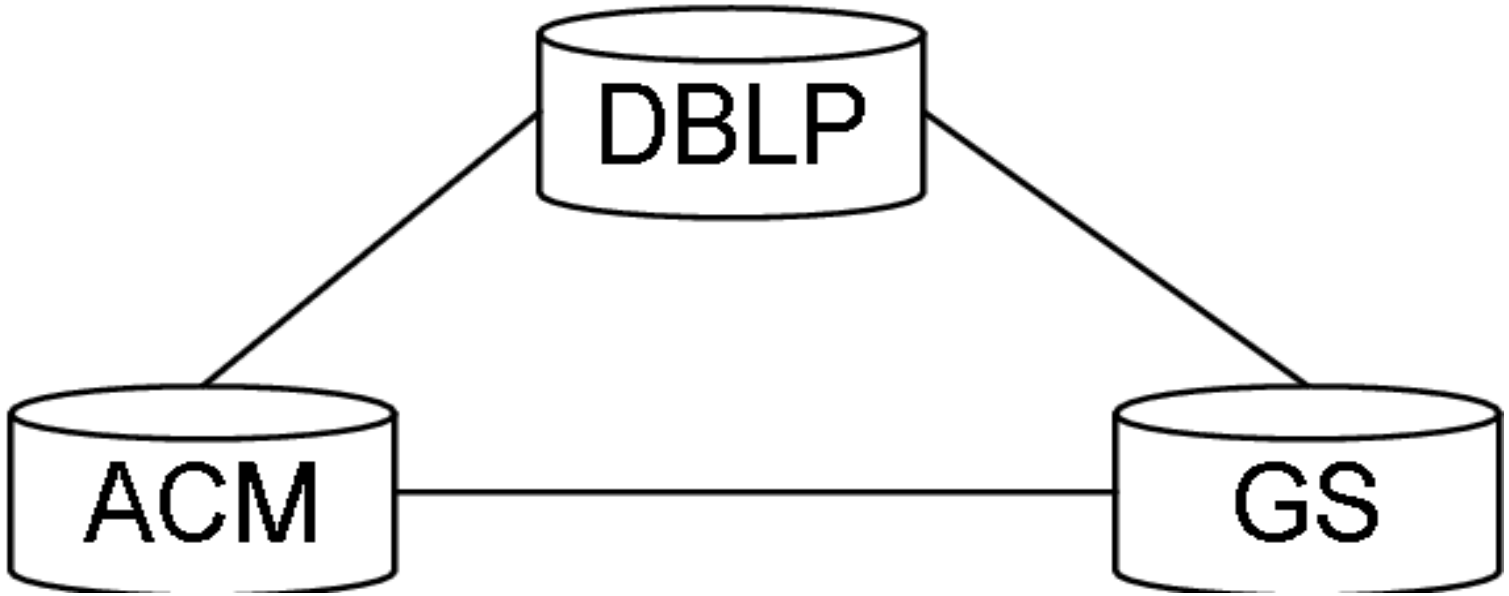
Komposition

- $compose(map_1, map_2) = \{(a, b, s') | (a, x, s_1) \in map_1, (x, b, s_2) \in map_2\}$
- Ermittlung des kombinierten Ähnlichkeitswertes s durch zwei Ähnlichkeitsfunktionen, da Korrespondenz zwischen zwei Objekten bei Komposition durch mehrere Pfade erreicht werden kann
  - Horizontal: Bestimmung des Ähnlichkeitswerts eines Pfades
    - Min, Max, Avg, Left ($= s_1$), Right ($= s_2$)
  - Vertikal: Bestimmung des Ähnlichkeitswerts einer Korrespondenz aus den zugehörigen Pfad-Ähnlichkeitswerten
    - $Dice = 2 \cdot \frac{s(a,b)}{n(a)+n(b)}$
    - $DiceLeft = \frac{s(a,b)}{n(a)}$, $DiceRight = \frac{s(a,b)}{n(b)}$
    - $DiceMin = \frac{s(a,b)}{min(n(a)+n(b))}$
- Dabei sei
  - $s(a, b) =$ Summe der Ähnlichkeitswerte aller Pfade (a, b)
  - $n(a) =$ Anzahl der Korrespondenzen $(a, x) \in map_1$
  - $n(b) =$ Anzahl der Korrespondenzen $(x, b) \in map_2$

Komposition (2)

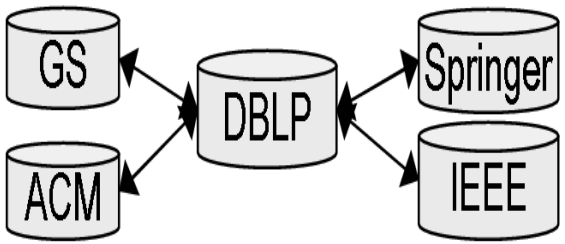
- Evaluation für Publikationen von DBLP, ACM und GS (F-Measure)

Mapping Compose via	DBLP - GS ACM	DBLP - ACM GS	GS - ACM DBLP
Direkt	81,3%	91,9%	35,3%
Compose	33,9%	63,7%	83,9%
Union	81,3%	91,6%	83,7%

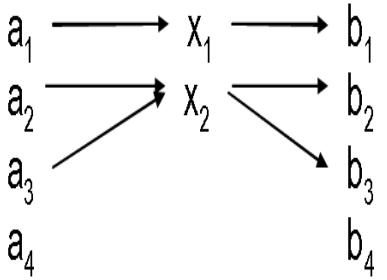


Komposition (3)

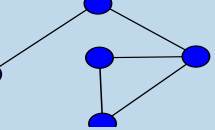
- Fazit
  - Komposition von Mappings ermöglicht effiziente Berechnung neuer Mappings
  - Besonders gut geeignet, falls Hub-Datenquelle vorhanden ist (Sternstruktur)
  - Fehlende Objekte in "mittlerer" Quelle führen zu fehlenden Korrespondenzen (Bsp: $a_4 - b_4$)
  - Komposition kann zu falschen Korrespondenzen führen (Bsp: $a_2 - b_3$)



Hub-Struktur

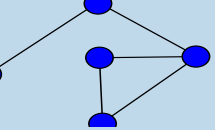


Problemfälle bei Komposition



Neighborhood-Matcher: Motivation und Idee

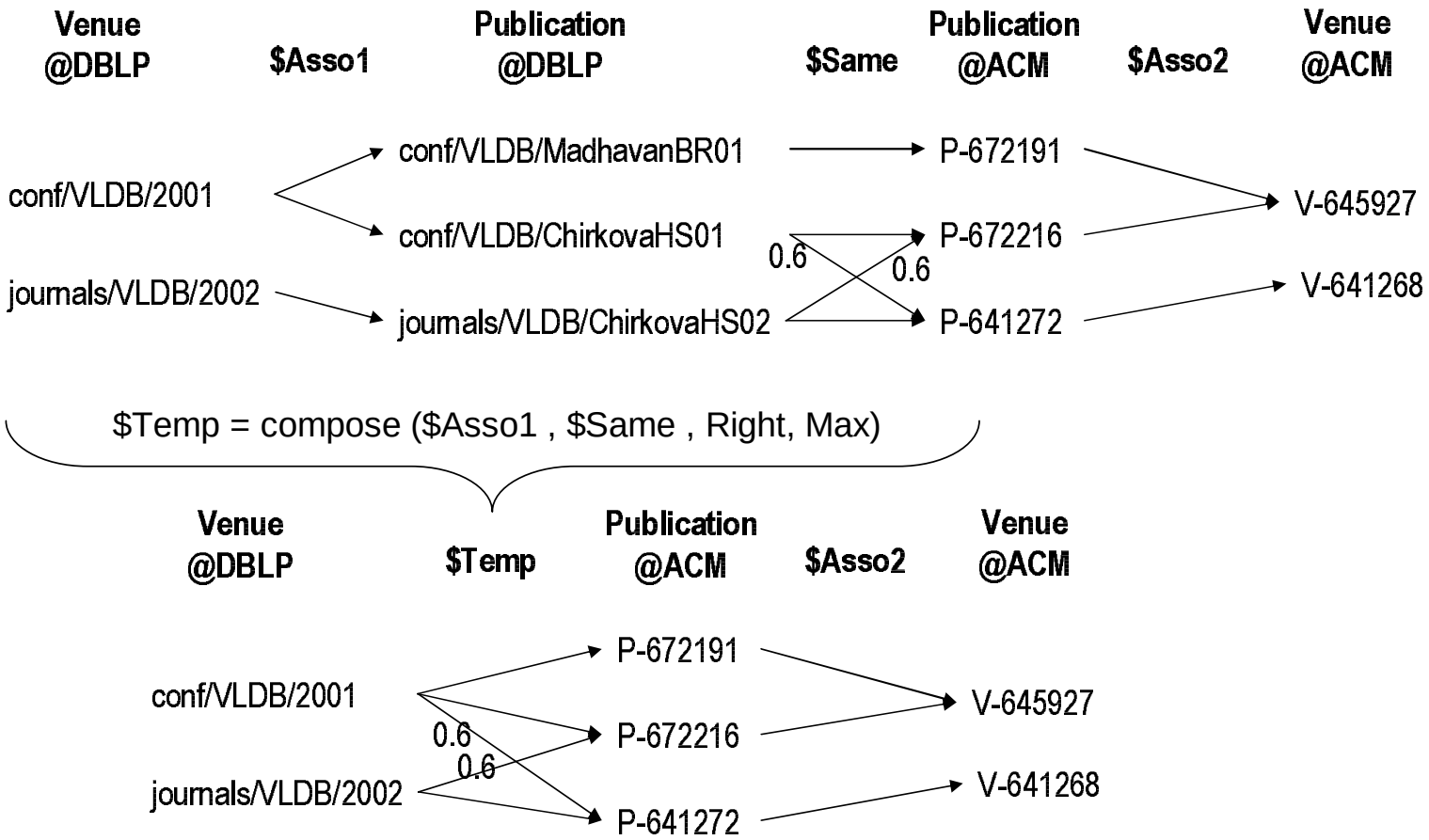
- Motivation: Wertevergleich für heterogene Objekte schwierig
- Beispiel für gleiche Konferenzen
  - "Proceedings of the 27th International Conference on Very Large Databases" vs. "Proc. of VLDB 2001, Italy"
- Lösung 1: Match-Verfahren mittels Domänenwissen
  - Abkürzungen, z.B. VLDB = Very Large Databases
  - Zuordnungen, z.B. "VLDB 2001" = "27. VLDB"
  - ...
- Problem: Woher kommt Domänenwissen? Bei jeder Domäne anders!
- Lösung 2: Verwendung assoziierter Informationen
  - Beispiel: "Zwei Konferenzen sind gleich, wenn die Menge der zugehörigen Publikationen gleich sind."
  - Mögliche Abschwächungen: alle $\rightarrow$ viele, gleich $\rightarrow$ ähnlich

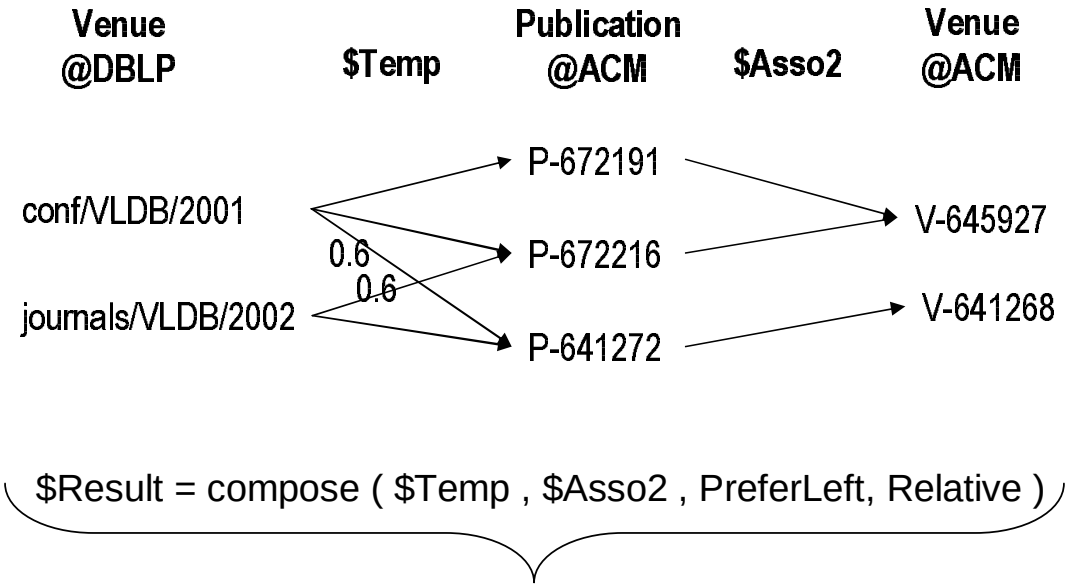


Neighborhood-Matcher: Match-Workflow

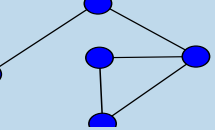
- Verwendung von Assoziations-Mappings
  - Syntax: Gleicher Struktur wie Same-Mappings; fester "Ähnlichkeitswert" $= 1$
  - Semantik: Korrespondenzen zwischen assoziierten Objekten, z.B. Publikationen - Venue
- Match-Workflow als Kompositon von drei Mappings
  - map1 und map3 sind Assoziations-Mappings; map2 ist ein Same-Mapping
- Idealfall (rechts) nicht immer erreicht, da
  - Assoziations-Mappings unvollständig, z.B. nicht alle Publikationen in jeder Datenquelle zu jedem Venue verfügbar
  - Same-Mapping fehlerhaft, z.B. als Ergebnis eines automatischen Match-Verfahrens

- Ähnlichkeitswerte = 1 (solange nicht anders angegeben)





Venue@DBLP	Publication@ACM	Ähnlichkeitswert s			
		DiceMin	DiceLeft	DiceRight	Dice
conf/VLDB/2001	V-645927	(1+1) / 2 = 1	(1+1) / 3 = 0.67	(1+1) / 2 = 1	2*(1+1) / (3+2) = 0.8
conf/VLDB/2001	V-641268	0.6 / 1 = 0.6	0.6 / 3= 0.2	0.6 / 1 = 0.6	2*0.6 / (3+1) = 0.3
journals/VLDB/2002	V-645927	0.6 / 2 = 0.3	0.6 / 2 = 0.3	0.6 / 2 = 0.3	2*0.6 / (2+2) = 0.3
journals/VLDB/2002	V-641268	1 / 1 = 1	1 / 2 = 0.5	1 / 1 = 1	2*1 / (2+1) = 0.67



Zusammenfassung

- 2008: Konzepte und Realisierungen von P2P und DBS stark unterschiedlich
- Zusammenführung erfordert (z.T. neue) Lösungen für
  - Schema- und Datenintegration, -transformation (automatisiert, große Schemata, kurzfristig - „Echtzeit“) oder Serverdienst?).
  - Anfrageverteilung, Anfrageoptimierung
  - Adäquates Transaktionskonzept (P2P-Variante des 2PC ?)
  - Qualitätsbewertung: Akzeptiert der Peer DB-Merkmale ?
- Kompromisse
  - Die letzten 3 Punkte - klare Merkmale eines DBVS.
  - Die letzten 2 Punkte - Autonomieeinschränkung der Peers.
  - Einschränkungen von DB-Merkmalen ? (Vollständigkeit d. Ergebnisse, Wiederholbarkeit einer Anfrage ?)