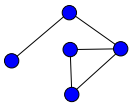
Gliederung

Peer-to-peer Systeme und Datenbanken(SS07)

- Kapitel 1: Einführung
- Kapitel 2: Beispiele
- Kapitel 3: Routing
- Kapitel 4: Schemabasierte p2p-Netzwerke
- Kapitel 5: Integrationsprobleme
  Teil 5-1: Einführung, Gleichheit
  Teil 5-2: Ähnlichkeit - 1
  Teil 5-3: Ähnlichkeit - 2
  Teil 5-4: Mappingbasierte Datenintegration
- Kapitel 6: Anonymität, Authentifikation
- Kapitel 7: Reputation

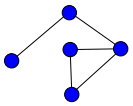
Version vom 26. Juni 2007



Kapitel 5-4

Mapping-basierte Datenintegration

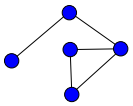
- Matching
  - Beispiel und Definition
  - Überblick über Ansätze
- Mapping-Verarbeitung
  - Motivation und Ziele
  - Grundbegriffe
  - MOMA-Framework
  - Operatoren zur Mapping-Verarbeitung
- Match-Workflows
  - Neighborhood-Matcher
  - Anwendung für Ontologie-Matching
- Qualitätsbewertung
- Zusammenfassung



Matching: Beispieldomäne

Bibliografische Domäne: Informationen über wissenschaftliche Publikationen

- DBLP Bibliography (http://www.informatik.uni-trier.de/ ley/db/index.html)
  - manuelle gepflegte Biographie wissenschaftlicher Publikationen aus dem Informatik-Bereich
  - enthält komplette Publikationslisten von Tagungen, Journals und Workshops
  - sehr hohe Datenqualität, vollständige biografische Angaben
- Google Scholar (http://scholar.google.com)
  - Suchmaschine für wissenschaftliche Publikationen
  - automatische Extraktion aus durch Crawling ermittelten PDF-Files
  - schlechte Datenqualität, unvollständige biografische Angaben
  - automatische Bestimmung der Zitierungszahl (“Wieviele andere Publikationen zitieren eine Publikation?”)



Matching: Beispieldomäne (2)

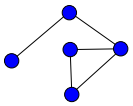
Erhard Rahm, Philip A. Bernstein: A survey of approaches to automatic schema matching. VLDB J. 10(4): 334-350 (2001)

[CITATION] **A survey of approaches to automatic schema matching**
PA Bernstein, E Rahm - VLDB Journal, 2001
Cited by 12 - Web Search

[CITATION] **A survey of approaches to automatic schema matching**
E Raham, PA Bernstein - The VLDB Journal, 2001
Cited by 4 - Web Search

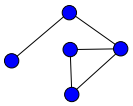
[CITATION] **A survey of approaches to automatic schema matching**
E Rahn, PA Bernstein - Very Large Database J, 2001
Cited by 1 - Web Search

[CITATION] 1, Philip A. Bernstein,"**A survey of approaches to automatic schema matching**."
E Rahm - The VLDB Journal
Cited by 1 - Web Search



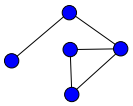
Matching: Beispieldomäne (3)

- Identifizieren gleicher Informationen in (verschiedenen) Datenquellen ist Voraussetzung für Datenintegration (Beispiel: Zitierungsanalyse)
  - “Die Publikationen welcher Konferenz haben durchschnittlich die meisten Zitierungen?”
  - DBLP enthält manuell erstellte und vollständige Publkationslisten von Konferenzen
  - Google Scholar ist Suchmaschine: automatische extrahierte Referenzen inkl. Zitierungszahl
  - “Kombiniere bibliografische Angaben von DBLP mit Zitierungszahlen von GS”
- Probleme (Beispiel: Google Scholar)
  - Heterogene Attributwerte (”VLDB Journal” vs. “Very Large Databases J.“)
  - Reihenfolge der Autoren, fehlende Autoren
  - Extraktionsfehler (Autoren mit im Titel)
  - Tippfehler (”Rahm” $\rightarrow$ “Rahn” oder “Raham”)
  - ...



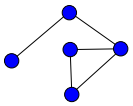
Definition: Instanz/Objekt-Matching

- Eingabe
  - 2 Mengen von Objekten/Entitäten (Datenbanksätze, XML-Dokumente, ...) O1 und O2
  - Optional: Assoziierte Informationen (z.B. per Fremdschlüssel referenzierte/referenzierende Instanzen), Hintergrundwissen (automatisch oder manuell erstellt, z.B. Synonymtabellen)
- Ausgabe:
  - Mapping zwischen O1 und O2, d.h. Korrespondenzen zwischen semantisch gleichen Instanzen, die z.B. “das gleiche Objekt der realen Welt repräsentieren”



Definition: Schema/Ontologie-Matching

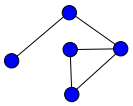
- Eingabe:
  - 2 Schemata (XML-Schema, DB-Schema, ...) S1 und S2
  - Optional: Instanzen der Schemata, Hintergrundwissen
- Ausgabe:
  - Mapping zwischen S1 und S2, d.h. Korrespondenzen zwischen semantisch gleichen Schemaelementen



Matching

Matching ist schwieriges Problem

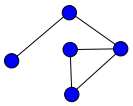
- Wichtiger Bestandteil der Datenintegration
- Forschungsthema seit 1980er Jahre
- Ziel: “korrekt und vollständig und automatisch” $\rightarrow$ weitgehend unerreicht
- Probleme
  - Namenskonflikte (z.B. Synonyme, Homonyme, Hypernyme)
  - Strukturelle Konflikte (z.B. unterschiedliche / fehlende Attribute, unterschiedliche Abstraktionsebenen)
  - ...



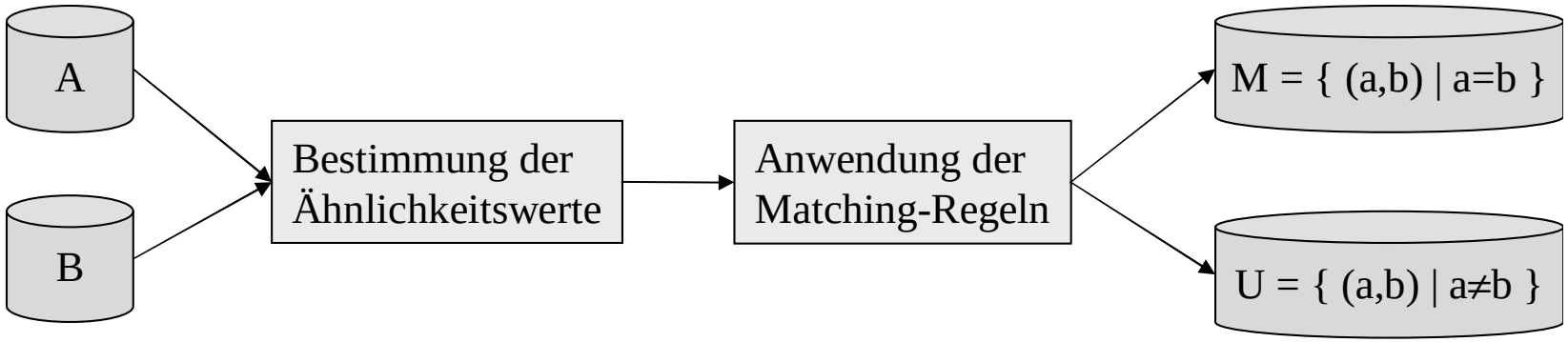
Object Matching: Allgemeines Verfahren

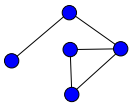
Paarweiser Vergleich zweier Instanzen $a = (a_1, \ldots, a_m) \in A$ und $b = (b_1, \ldots, b_n) \in B$

1. Bestimmung der Ähnlichkeitswerte
   - Ähnlichkeitsfunktionen zum Vergleich von Attributwerten
   - Verschiedene Funktionen, verschiedene Attributvergleiche möglich $\rightarrow$ mehrere Ähnlichkeitswerte
2. Anwendung der Matching-Regeln
   - Regel, die an Hand der Ähnlichkeitswerte bestimmt “Match” oder “kein Match”
   - Bsp: “Wenn Ähnlichkeit der Familienenamen 100% und Ähnlichkeit des Vornamens 80%, dann sind zwei Personen gleich.”



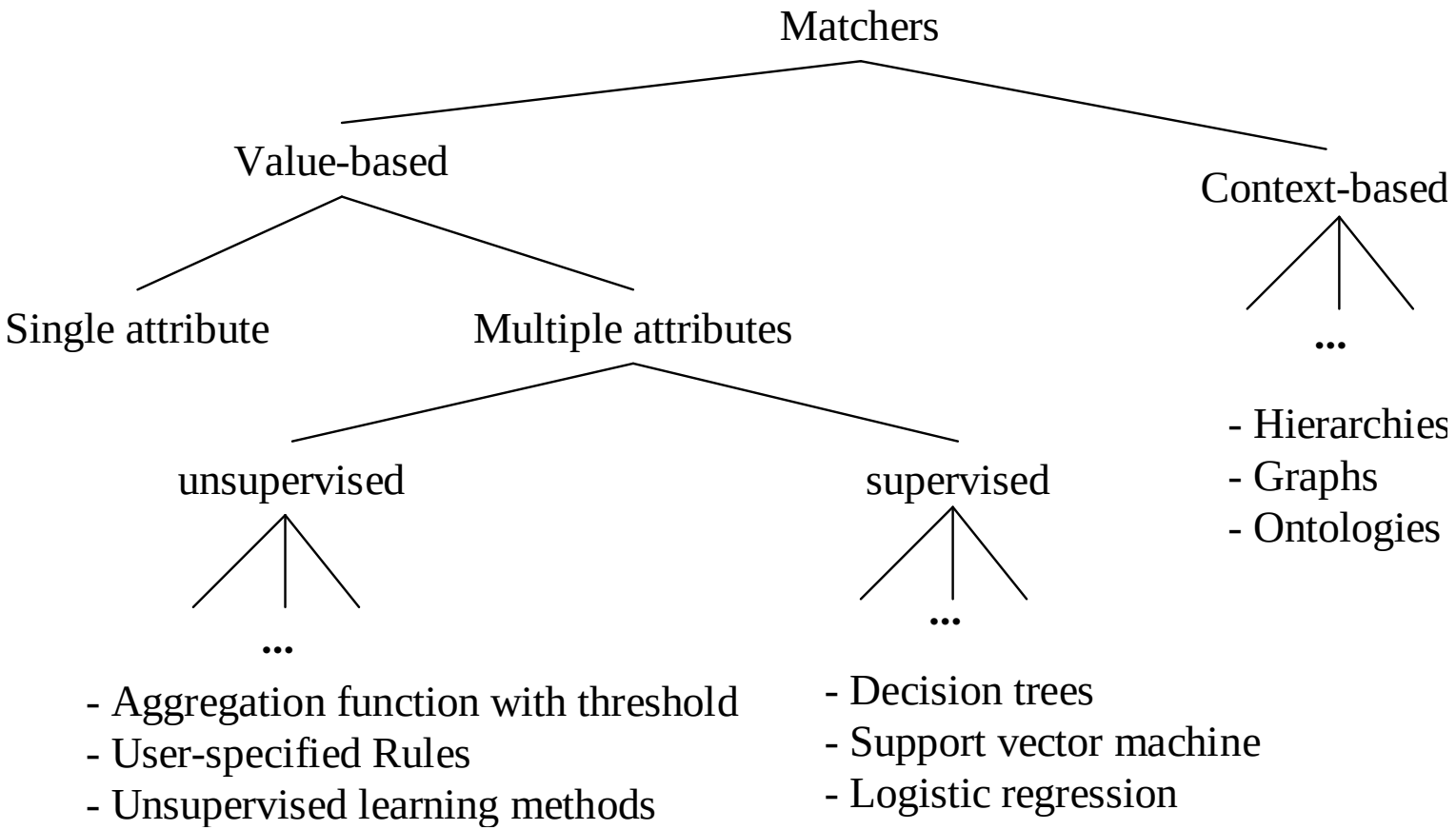
Object Matching: Allgemeines Verfahren (2)

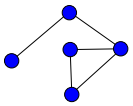




Ansätze für Object Matching

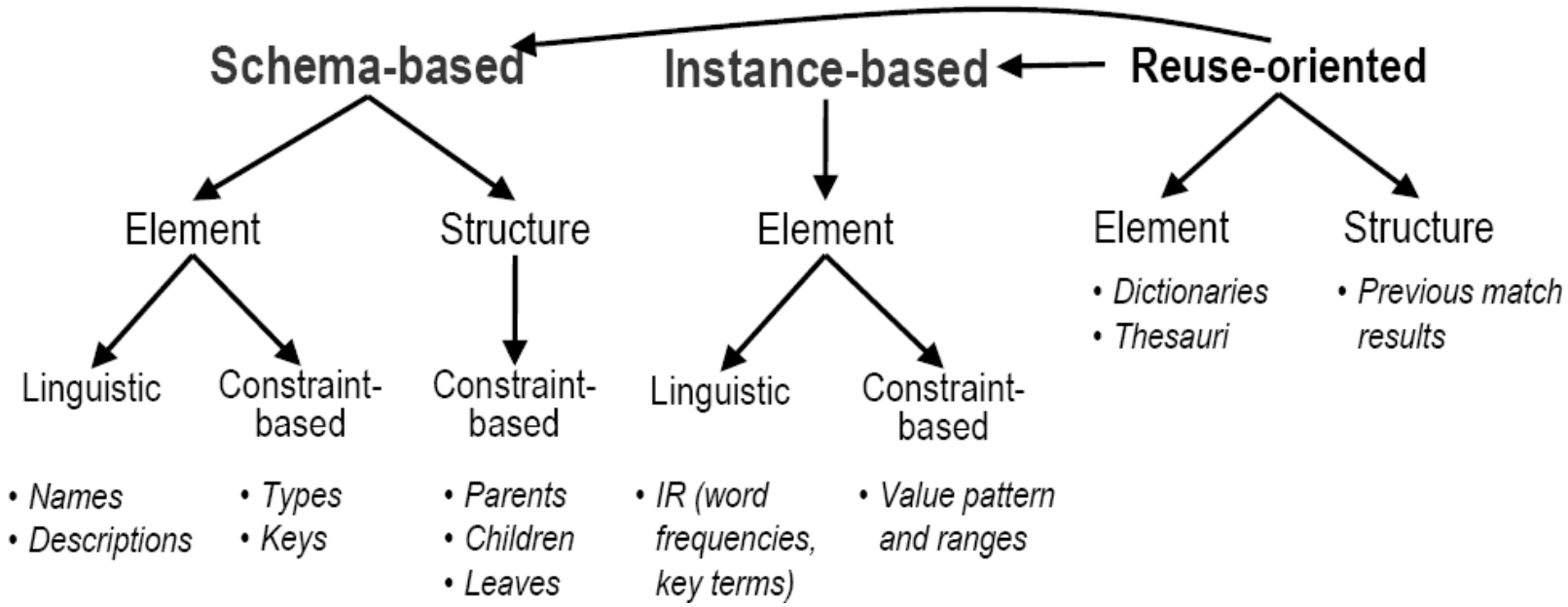
Viele verschiedene automatische Ansätze, die auch kombiniert werden können





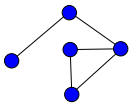
Ansätze für Schema Matching

Viele verschiedene automatische Ansätze, die auch kombiniert werden können



Publikationen

- Rahm, E., P.A. Bernstein: A Survey of Approaches to Automatic Schema Matching. VLDB Journal 10 (4), 2001
- Do, H.-H., Rahm, E.: COMA - A System for Flexible Combination of Schema Matching Approaches. VLDB, 2002



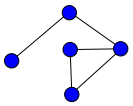
Mapping-Verarbeitung

Motivation

- Matching ist i.A. sehr aufwändig: Viele Ähnlichkeitsvergleiche (“jedes mit jedem” $\rightarrow O(n^2)$), manuelle Überprüfung (Arbeitszeit, Kosten), ...
- Matching ist i.A. sehr schwierig: Welcher Match-Algorithmus? Welche Parameter? ...
- Match-Ergebnis ist “wertvoll” und sollte wiederverwendet werden

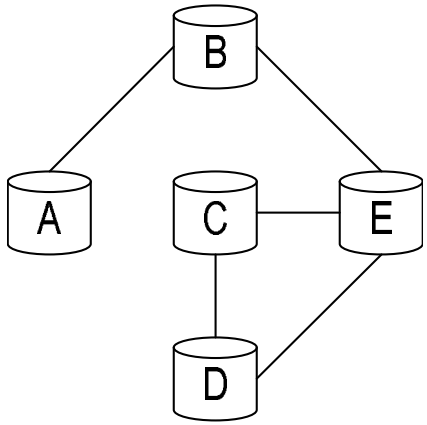
Ziele

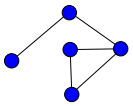
- Wiederverwendung von Match-Ergebnissen zur effizienten Berechnung neuer Match-Ergebnisse
- Kombination von Match-Ergebnissen zur Qualitätsverbesserung
- Bestimmung von Match-Ergebnissen, wenn kein geeignetes Ähnlichkeitsmaß zur Verfügung steht



Mapping-Verarbeitung: Beispiel

- Effiziente Berechnung: (A,E) mittels (A,B) und (B,E)
  - Annahme: Kosten für Matching zwischen A und E sind höher als bei Mapping-Verarbeitung
- Qualitätsverbesserung: Kombination von (D,E) direkt mit (D,C) + (C,E)
  - Annahme: Kombination mehrerer Mapping-Ergebnisse steigert Mapping-Qualität
- Kein geeignetes Ähnlichkeitsmaß: (A,D) mittels (A,B) + (B,E) + (E,D)
  - Annahme: Objekte aus A und D können mittels Attributwerten nicht (ausreichend) verglichen werden





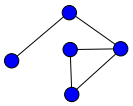
MOMA-Ansatz: Begriffe (1)

Definition: Peer-Datenquelle (Logische Datenquelle, LDS)

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

Typ: Publication Source:DBLP

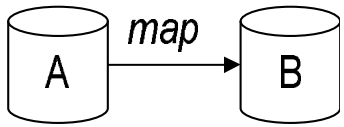
Id: conf/vldb/MadhavanBR01
Name: *Generic schema matching with Cupid*
URL: *http://vldb.org...*
Conference: *VLDB 2001*
Authors: *Jayant Madhavan, Philip A. Bernstein, Erhard Rahm*



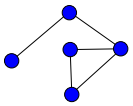
MOMA-Ansatz: Begriffe (2)

Definition: Same-Mapping

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



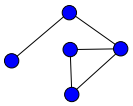
A	B	s
a1	b1	1
a2	b2	0.9
a2	b3	0.3



Mapping-Verarbeitung: MOMA-Ansatz

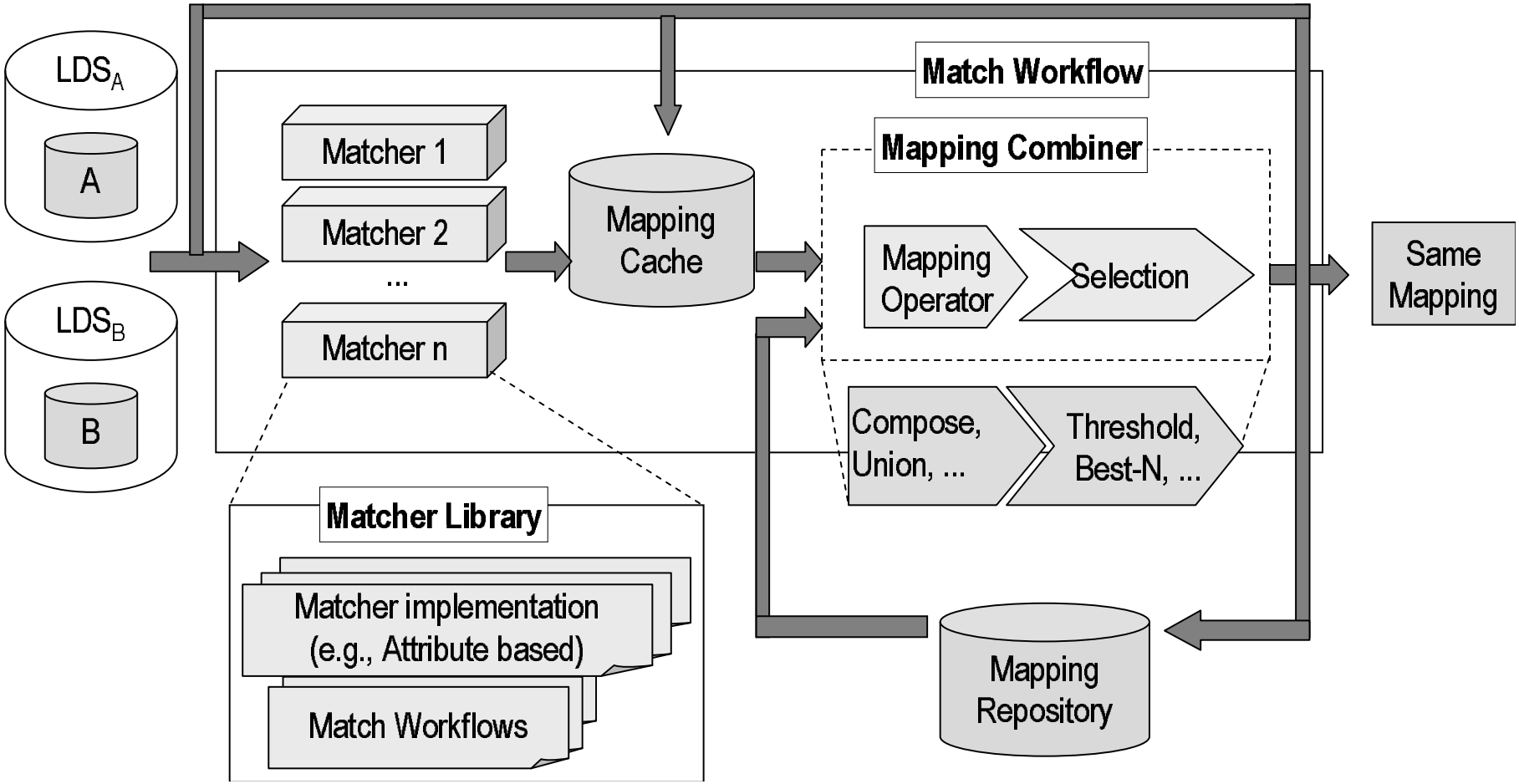
- Verarbeitung von Mappings und Objektinstanzen durch Operatoren
- Kombination der Operatorergebnisse durch Skriptsprache (iFuice*)
  - Prozedurale Programmiersprache mit Kontrollstrukturen (IF-THEN-ELSE, WHILE-DO)
  - Ergebnisse werden in Variablen gespeichert
  - Definition und Aufruf von Unterprozeduren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

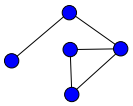
* Rahm, E. et. al.: iFuice - Information Fusion utilizing Instance Correspondences and Peer Mappings. WebDB, 2005



MOMA-Framework: Architektur

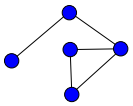
Thor, A., Rahm, E.: MOMA - A Mapping-based Object Matching System. CIDR, 2007





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
  - $inverse$: “Umkehrung” der Korrespondenzen, d.h. $(a, b, s) \rightarrow (b, a, s)$ (Semantik?)



Kombination: Vereinigung / Durchschnitt

- Ermittlung des kombinierten Ähnlichkeitswertes s durch Kombinationsfunktion $f(s_1, s_2)$
- 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

union (Max)		
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

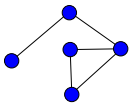
union (Min)		
A	B	s
a1	b1	0.6
a2	b2	0.8
a3	b3	0.6

map2		
A	B	s
a1	b1	0.6
a3	b3	0.6

union (Ranked)		
A	B	s
a1	b1	1
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-0)		
A	B	s
a1	b1	0.6
a2	b2	0
a3	b3	0

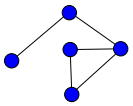


Kombination: Vereinigung / Durchschnitt (2)

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

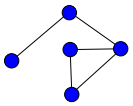
	Titel (Trigram)	Autoren (Trigram)	Jahr (Gleichheit)	Union-Avg (Filter:80%)
Precision	86,7%	38,0%	0,4%	97,3%
Recall	97,7%	87,9%	100,0%	93,9%
F-Measure	91,9%	53,1%	0,8%	95,5%

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



Komposition

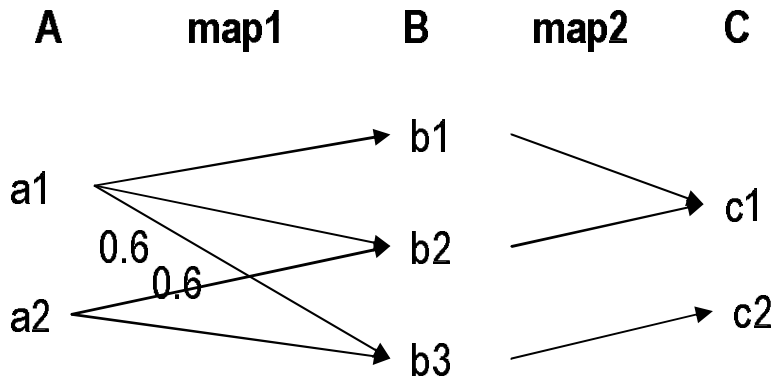
- $compose(map_1, map_2) = \{(a, b, s') | (a, x, s_1) \in map_1, (x, b, s_2) \in map_2\}$
- Ermittlung des kombinierten Ähnlichkeitswertes s durch zwei Funktionen, da Korrespondenz zwischen zwei Objekten bei Komposition durch mehrere Pfade erreicht werden kann
  - Horizontal: Bestimmung des Ähnlichkeitswerts 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 $(x, b) \in map_2$



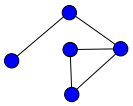
Komposition: Beispiel

map1		
A	B	s
a1	b1	1
a1	b2	1
a1	b3	0.6
a2	b2	0.6
a2	b3	1

map2		
B	C	s
b1	c1	1
b2	c1	1
b3	c2	1



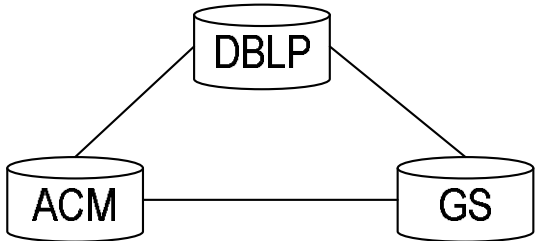
compose (map1, map2, Left, Dice)		
A	C	s
a1	c1	2*(1+1) / (3+2) = 0.8
a1	c2	2*0.6 / (3+1) = 0.3
a2	c1	2*0.6 / (2+2) = 0.3
a2	c2	2*1 / (2+1) = 0.67

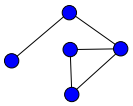


Komposition (2)

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

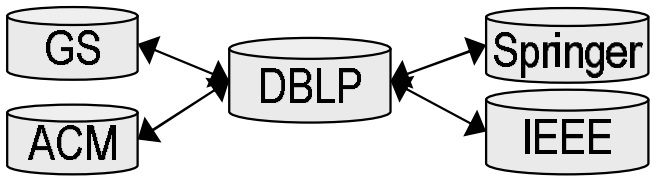
Mapping Compose via	DBLP - GS ACM	DBLP - ACM GS	GS - ACM DBLP
Direkt	81,3%	91,9%	35,3%
Compose	33,9%	63,7%	83,9%
Union	81,3%	91,6%	83,7%



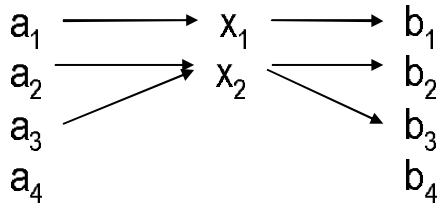


Komposition (3)

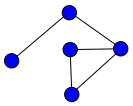
- Fazit
  - Komposition von Mappings ermöglicht effiziente Berechnung neuer Mappings
  - Besonders gut geeignet, falls Hub-Peer-Datenquelle vorhanden ist (Sternstruktur)
  - Fehlende Objekte in “mittlerem” Peer führen zu fehlenden Korrespondenzen (Bsp: $a_4 - b_4$)
  - Komposition kann zu falschen Korrespondenzen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



Hub-Struktur

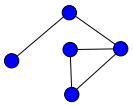


Problemfälle bei Komposition



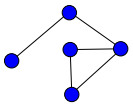
Neighborhood-Matcher: Motivation und Idee

- 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

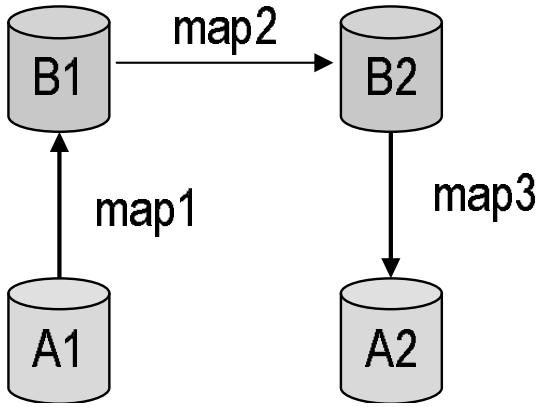


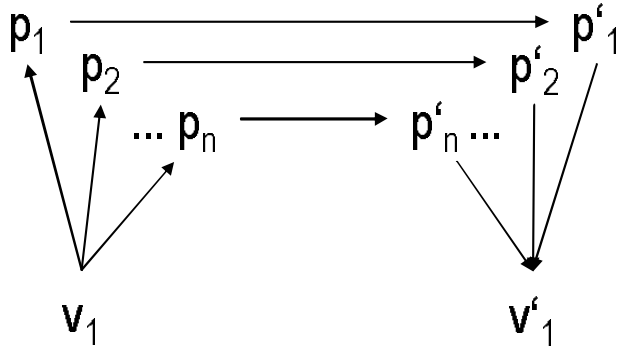
Neighborhood-Matcher: Match-Workflow

- 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

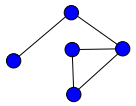


Neighborhood-Matcher: Match-Workflow (2)



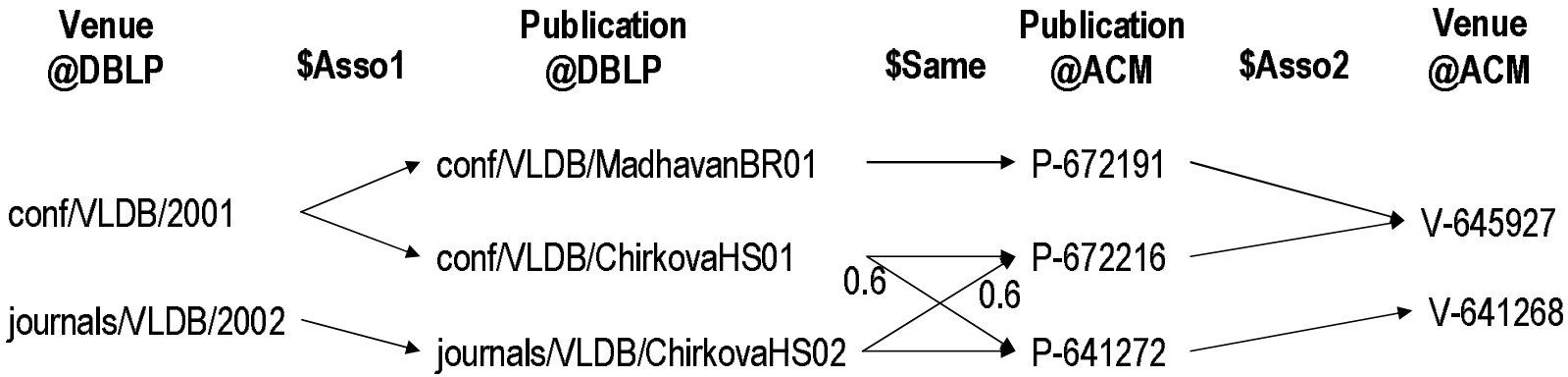


- Idealfall (rechts) nicht immer erreicht, da
  - Assoziations-Mappings unvollständig, z.B. nicht alle Publikationen in jeder Datenquelle zu jedem Venue verfügbar
  - Same-Mapping fehlerhaft, z.B. als Ergebnis eines automatischen Match-Verfahrens

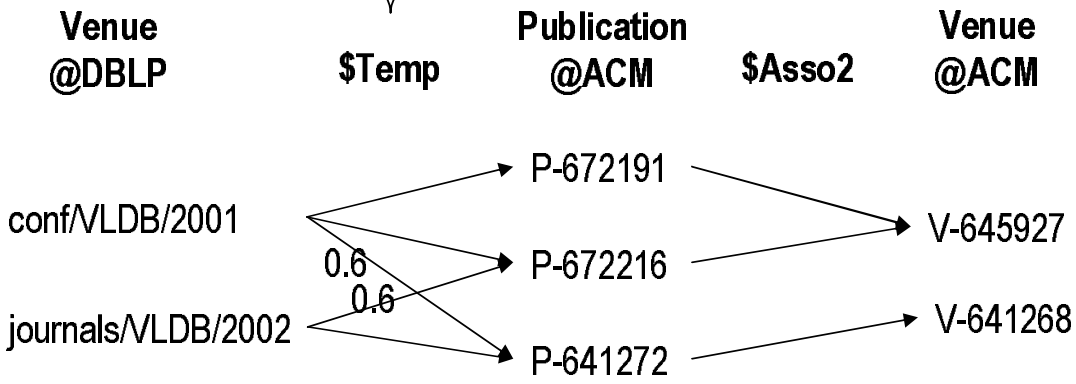


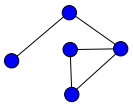
Neighborhood-Matcher: Beispiel (1)

- Ähnlichkeitswerte = 1 (solange nicht anders angegeben)

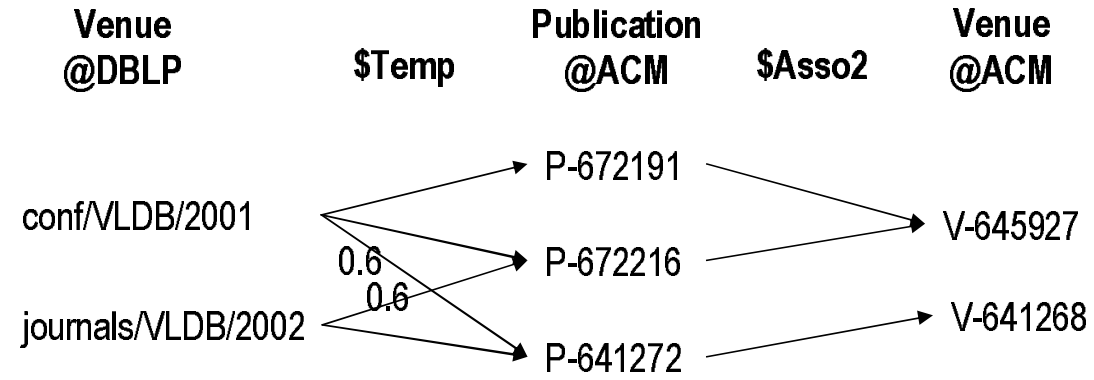


$Temp = compose ($Asso1 , $Same , Right, Max)



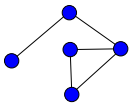


Neighborhood-Matcher: Beispiel (2)



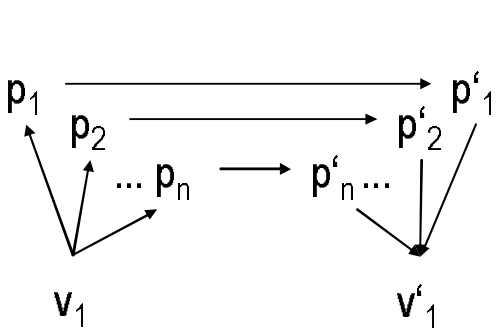
$Result = compose ($Temp , $Asso2 , PreferLeft, Relative)

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

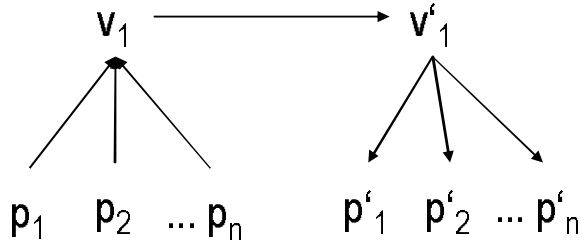


Neighborhood-Matcher: Kardinalitäten

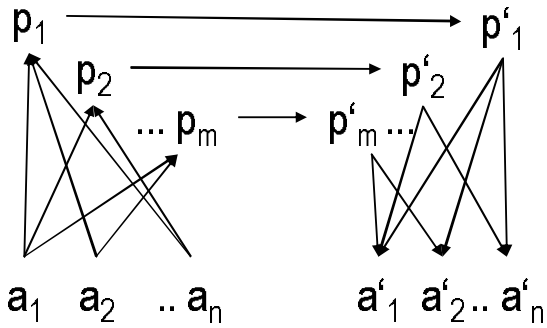
- Kardinalität des Assoziations-Mappings beeinflusst Match-Qualität



1:N

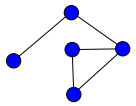


N:1



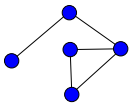
N:M

- Verarbeitung des Match-Ergebnisses (nach entsprechender Filterung)
  - 1:N (z.B. Venue-Publikation): meist keine Weiterverarbeitung nötig (bei großem N)
  - N:1 (z.B. Publikation-Venue): Kombination mit weiteren Matchern nötig, aber Einschränkung des “Match-Raums”, d.h. z.B. weniger Attribut-Vergleiche nötig
  - N:M (z.B. Publikation-Autor): für einzelne Instanzen keine Weiterverarbeitung nötig, ansonsten wie Fall N:1



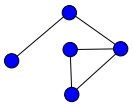
Ontologien

- Definition: Ontologie
  - “An ontology is an explicit specification of a conceptualization.” (Gruber: A translation approach to portable ontologies. In: Knowledge Acquisition, 1993)
  - Mittel zur Strukturierung eines Wissensbereiches
  - Bestandteile: Klassen, Relationen, Funktionen und Axiome
- Arten von Ontologien
  - Typ: Glossare, Thesauri, Hierarchien, Taxonomien, UML-Diagramme, RDF-Graphen, ...
  - Scope: Top-Level-Ontologie, Domänen-Ontologien, Anwendungs-Ontologie

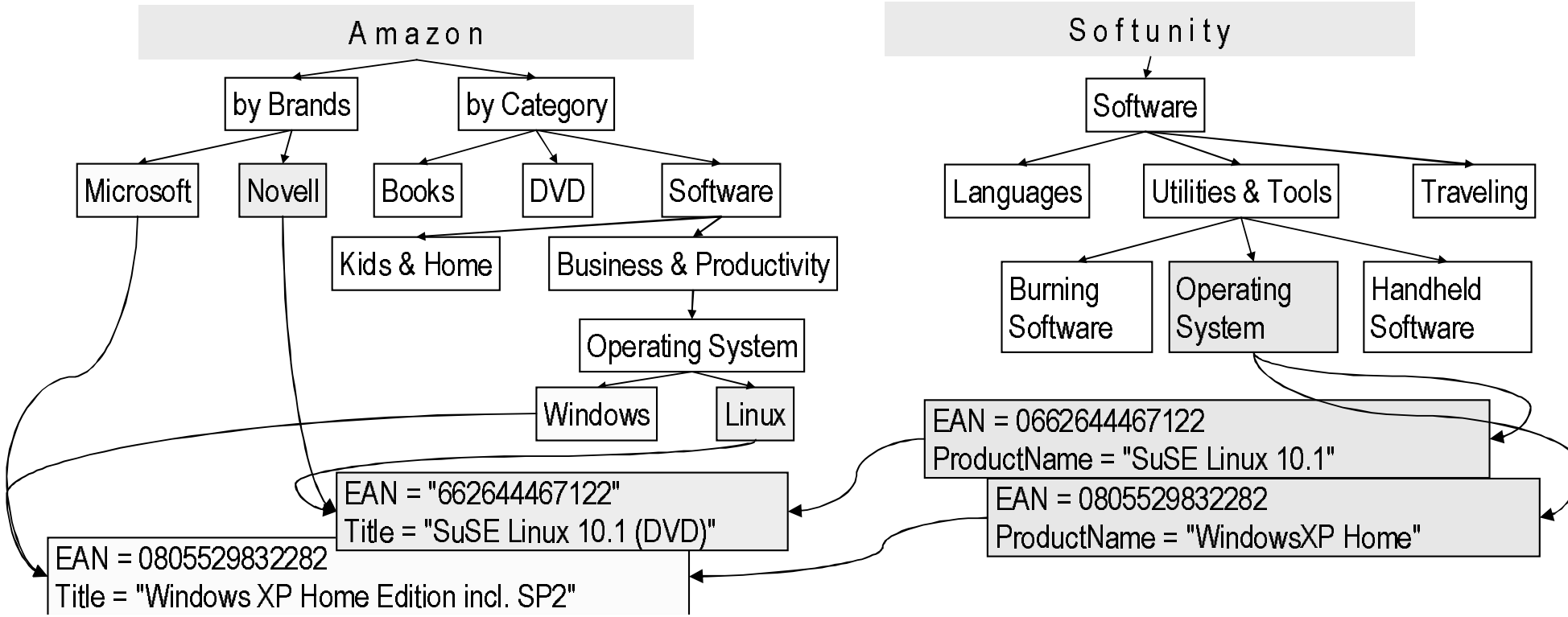


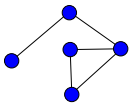
Ontologie-Matching: Beispiel

- Im Folgenden werden Taxonomien betrachtet (Beispiel: Produktkatalog)
  - Klasse “Konzept” (z.B. Produktgruppe “Kinderbücher”)
  - Klasse “Instanz” (z.B. Produkt “Harry Potter”)
  - (1:N-) Beziehung “is-a” zwischen Konzepten (z.B. hierarchische Anordnung von Produktgruppen)
  - (N:M-) Beziehung “element-of” zwischen Instanzen und Konzepten (z.B. “Harry Potter” element-of “Kinderbücher”)
- Anwendungsfälle
  - E-Commerce: Vereinigung von Produktkatalogen
  - Bioinformatik: Annotation von Daten



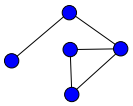
Ontologie-Matching: Beispiel (2)





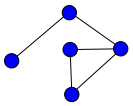
Ontologie-Matching: Ansatz

- Idee
  - Erstellung eines Instanz-Mappings (Objekt-Matching)
  - Ähnlichkeit zweier Konzepte = Ähnlichkeit der Menge der zugeordneten Instanzen
  - “Zwei Produktklassen sind umso ähnlicher, je mehr gleiche Produkte sie enthalten”
- Objekt-Matching meist einfacher als Ontologie-Matching
  - mehr Attribute bei Objektinstanzen als bei Konzepten (meist nur [kurzer] Name)
  - z.T. eindeutige Identifikatoren (EAN für Produkte, ISBN für Bücher) für Objekte
  - Namen der Konzepte entspringen meist verschiedenen Gesichtspunkten/Modellierungen



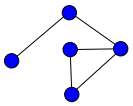
Ontologie-Matching: Ansatz (2)

- Ausnutzen der bekannten Ähnlichkeitsmaße für Mengen
  - $Sim_{Base}(A, B) = 1$, wenn $A \cap B \neq \emptyset$, sonst 0
  - $Sim_{Min}(A, B) = \frac{|A \cap B|}{min(|A|,|B|)}$
  - $Sim_{Dice}(A, B) = 2 \cdot \frac{|A \cap B|}{|A|+|B|}$
- Anwendung des Neighborhood-Matchers möglich
  - Modellierung von Konzepthierarchie und Instanzzuordnungen als Mappings
- Erweiterungen
  - Matching “innerer” Konzepte (implizite Zuordnung der Instanzen der Kindkonzepte)
  - Matching von Konzeptmengen



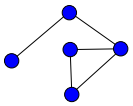
Qualitätsbewertung: Precision und Recall

- Annahme: Es gibt ein (z.B. manuell erstelltes) perfektes Match-Ergebnis (Mapping) map_{Perf}
- Verwendung der Qualitätsmaße aus dem Information Retrieval zur Bewertung von map_{Match}
  - $Precision = |map_{Match} \cap map_{Perf}|/|map_{Perf}|$
  - $Recall = |map_{Match} \cap map_{Perf}|/|map_{Match}|$
  - $F - Measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$
- Vorteile
  - Effektive Vergleichbareit verschiedener Match-Verfahren
  - Möglichkeit zur Optimierung von Match-Verfahren (z.B. Schwellwert bei Filterung)
- Nachteile
  - Vorhandensein des perfekten Mappings erforderlich



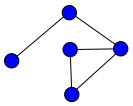
Qualitätsbewertung: Match Ratio und Match Coverage

- Perfektes Mapping nicht immer vorhanden
  - viele Datenquellen (P2P-Umfeld) + große Datenquellen → viele, große, manuell zu erstellende/verifizierende perfekte Mappings → großer Aufwand
  - perfektes Mapping nicht immer eindeutig, z.B. auf Grund unterschiedlicher Sichtweisen der Experten
- Abschätzung von Precision und Recall durch Match Ratio und Match Coverage
  - keine mathematische Abschätzung, sondern Ausnutzen einer “intuitiven Korrelation” zu anderen Maßen
  - “Wenn ein Konzept im Durchschnitt sehr viele Match-Partner besitzt, ist die Precision wahrscheinlich schlecht” (Match Ratio)
  - “Wenn nur ein kleiner Teil der Konzepte (mindestens) einen Match-Partner besitzt, ist der Recall wahrscheinlich schlecht” (Match Coverage)



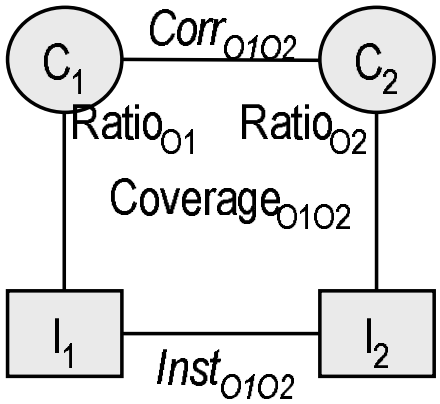
Qualitätsbewertung: Match Ratio und Match Coverage (2)

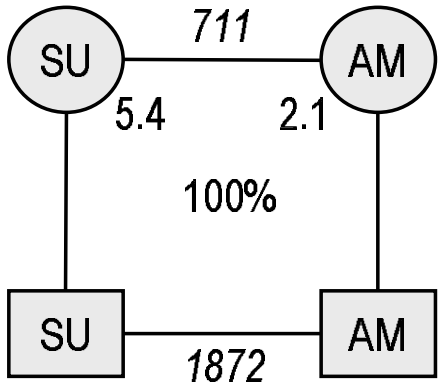
- Match Ratio ("Precision") = $\frac{|Corr_{O_1-O_2}|}{|C_{O_1}|}$ bzw. = $\frac{|Corr_{O_1-O_2}|}{|C_{O_2}|}$
  - Durchschnittliche Anzahl der Korrespondenzen (= Match-Partner) pro Konzept in O_i, dass einen mindestens einen Match-Partner hat
- Match Coverage ("Recall") = $\frac{|C_{O_1}|+|C_{O_2}|}{|C_{Base-O_1}|+|C_{Base-O_2}|}$
  - Anteil der Konzepte mit mind. einem Match-Partner im Vergleich zum Matching mit Baseline-Ähnlichkeit
- Dabei bedeuten
  - $|Corr_{O_1-O_2}|$ = Anzahl der Korrespondenzen des Mappings
  - $|C_{O_i}|$ = Anzahl der "gematchten" Konzepte (d.h. mind. ein Match-Partner) in O_i
  - $|C_{Base-O_i}|$ = Anzahl der "gematchten" Konzepte in O_i mit Baseline-Ähnlichkeit (d.h. Korrespondenz zwischen zwei Konzepten g.d.w. mind. eine gleiche Instanz zugeordnet)



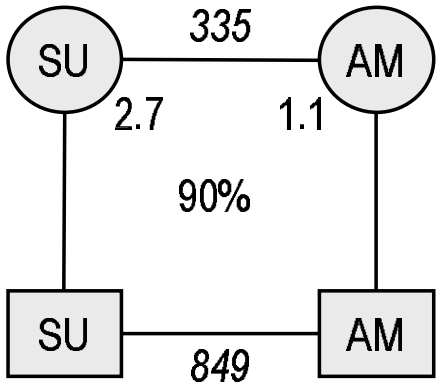
Qualitätsbewertung: Match Ratio und Match Coverage (Beispiel)

- Matching zwischen Produktkatalogen von Softunity und Amazon
  - Min-Ähnlichkeit sehr gut: ähnliche Match Coverage wie Baseline, geringere Match Ratio
  - Dice-Ähnlichkeit sehr restriktiv: Match Ratio ≈ 1, geringe Match Coverage

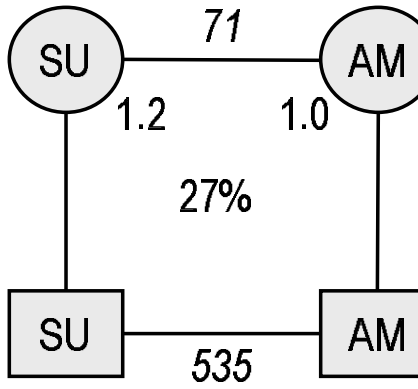




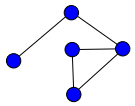
Baseline



Min (100%)

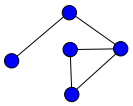


Dice (50%)



Zusammenfassung

- Matching ist wichtiger Aspekt der Datenintegration
- Mapping-Verarbeitung im P2P-Umfeld zur
  - effizienten Berechnung von Mappings
  - Qualitätsverbesserung von Mappings
  - Bestimmung von Mappings, wenn Attributvergleich unzureichend
- Beispiel: MOMA
  - Skriptsprache mit Operatoren
  - Mapping-Verarbeitung durch Match-Workflows
- Abschätzung der Match-Qualität bei vielen und großen Datenquellen (Peers) nötig



Offene Fragen - Diplomthemen

- Optimierung von Match-Workflows
  - Welches Match-Verfahren für welches Problem?
  - Welche Parameter (Ähnlichkeitsfunktion, Schwellwerte)?
- Anfragestrategien zur Integration von Suchmaschinen
  - Effiziente Bestimmung von Objektinstanzen durch Suchanfragen
  - Beispiel: Finde potentielle Match-Partner in Google Scholar für DBLP-Publikationen
- Web-basiertes System zur Analyse biologischer Daten
  - Mapping-basierte Integration und Auswertung der Ergebnisse
  - Nutzergesteuerte Überprüfung der Same-Mappings
- Vergleich von Scientific-Workflow-Management-Systemen in der Bioinformatik
  - Datenintegration ist Teil einer Analyse innerhalb eines Workflows
  - Verwendung iFuice / MOMA im Vergleich zu anderen Systemen