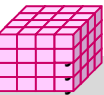


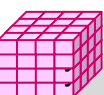
4. ETL: Schemaintegration + Data Cleaning

- ETL-Überblick
- Schemaintegration
- Schema Matching
 - Verfahren
 - COMA++
- Data Cleaning
 - Probleme
 - Teilaufgaben
- Objekt-Matching (Entity resolution)
 - Techniken
 - MS SQL-Server
 - Prototypen U Leipzig



ETL-Prozess

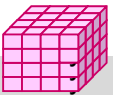
- Data Warehousing und ETL: materialisierter Ansatz zur Datenintegration
 - Erzeugung einer aggregierten, materialisierten Datenquelle für Online-Analysen
 - komplexer, aufwändiger Integrationsprozess
 - Offline-Durchführung erlaubt höhere Datenqualität gegenüber virtueller Datenintegration (Datentransformation während Query-Verarbeitung)
- *Extraktion*: Selektion eines Ausschnitts der Daten aus Quellen
 - ausgeführt an den entsprechenden Quellen
- *Transformation*: Aufbereitung und Anpassung der Daten an vorgegebene Schema- und Qualitätsanforderungen
 - ausgeführt im temporären Arbeitsbereich (Data Staging Area)
- *Laden*: physisches Einbringen der Daten aus Arbeitsbereich in das Data Warehouse, einschließlich evtl. notwendiger Aggregationen



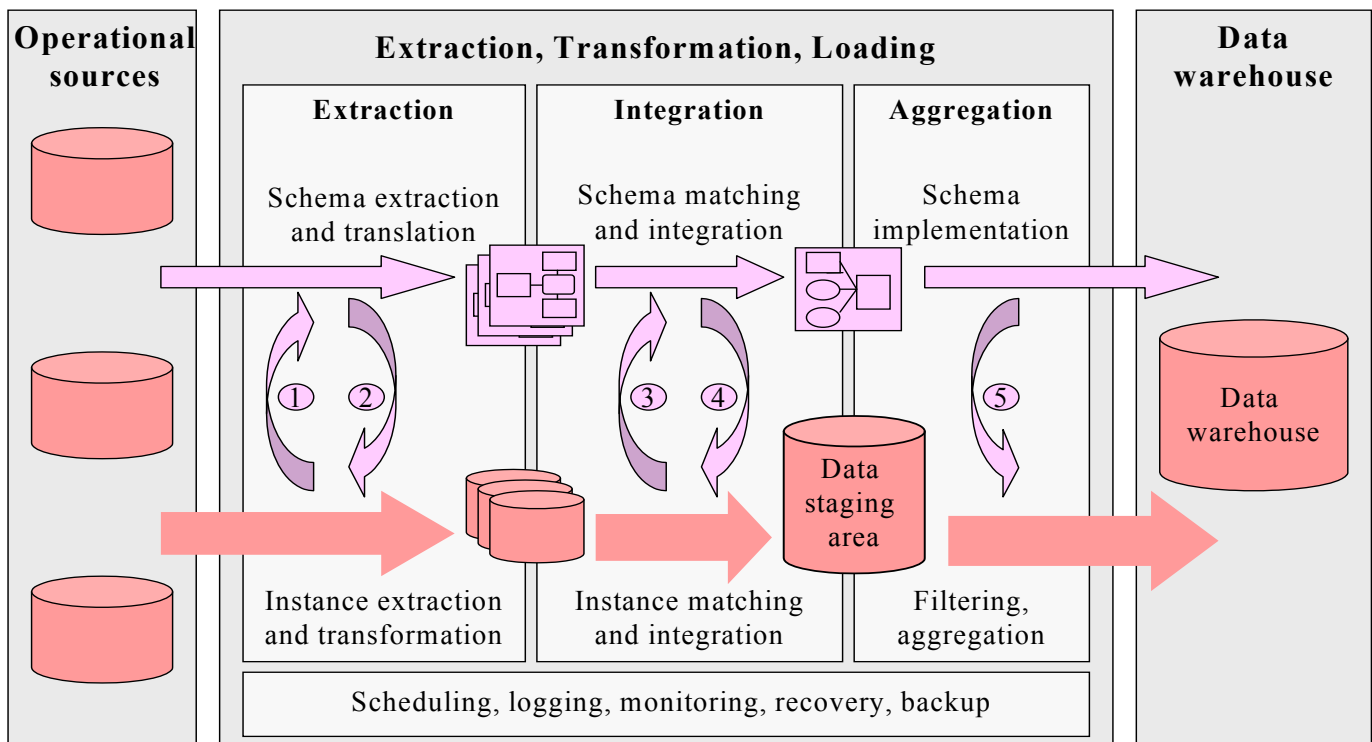
ETL-Prozess (2)

- Aufwändigster Teil des Data Warehousing
 - Vielzahl von operativen Quellen
 - Heterogenität der Datenquellen (DBMS, Schemata, Daten)
 - Gewährleistung hoher Qualität der Warehouse-Daten

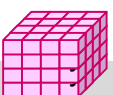
- Entscheidende Rolle im Data Warehousing, da großer Einfluss auf
 - Genauigkeit und Richtigkeit der später durchgeführten Analysen
 - die darauf basierenden Entscheidungen: „Garbage In, Garbage Out“



ETL-Prozess: Ablauf

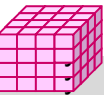


- Legends:**
- Metadata flow
 - Data flow
 - ① Instance characteristics (real metadata)
 - ② Translation rules
 - ③ Instance characteristics (real metadata)
 - ④ Mappings between source and target schema
 - ⑤ Filtering and aggregation rules



ETL als Integrationsprozess

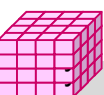
- ETL: Integration auf 2 Ebenen, Schemaintegration und Datenintegration
- Schemaintegration
 - Konstruktion eines Data Warehouse-Schemas aus existierenden Quellschemata
 - Ableitung von Korrespondenzen zwischen dem Data Warehouse-Schema und existierenden Quellschemata: *Schema Matching*
- Datenintegration / Data Cleaning
 - Transformation heterogener Daten in die einheitliche, durch das Data Warehouse-Schema vorgeschriebene Repräsentation
 - Entdeckung und Behebung von Datenqualitätsproblemen
 - Entdeckung äquivalenter Objekte/Sätze (Korrespondenzen auf Instanzenebene): *Objekt-Matching* / Duplikaterkennung



Schemaintegration - Anforderungen

- Minimalität
 - keine Redundanz im integrierten Schema
 - Abbildung mehrerer gleicher/ähnlicher Konzepte in lokalen Schemata auf ein Konzept im integrierten Schema
- Korrektheit
 - Äquivalenz der im integrierten Schema enthaltenen Informationen mit denen in den lokalen Schemata
 - Konsistenz der während der Integration ergänzten Informationen, z.B. Beziehungen zwischen Konzepten im integrierten Schema
- Verständlichkeit

Vollständigkeit (Beibehaltung aller Informationen aus Quellschemas) ?



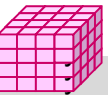
Schemaintegration (2)

■ Probleme der Schemaintegration

- Heterogenität der Schemarepräsentationen, z.B. relational (SQL), XML, Entity-Relationship (ER), objekt-orientiert (UML), ...
- semantische Heterogenität der Schemaelemente (Namenskonflikte, strukturelle Konflikte)

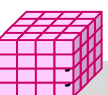
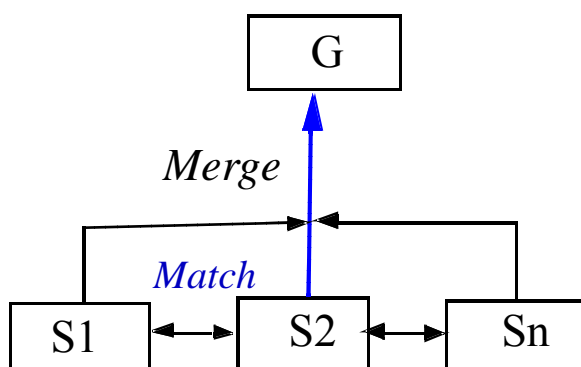
■ Alternativen

- Bottom-Up-Schemaintegration
- Top-Down-Schemaintegration

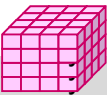
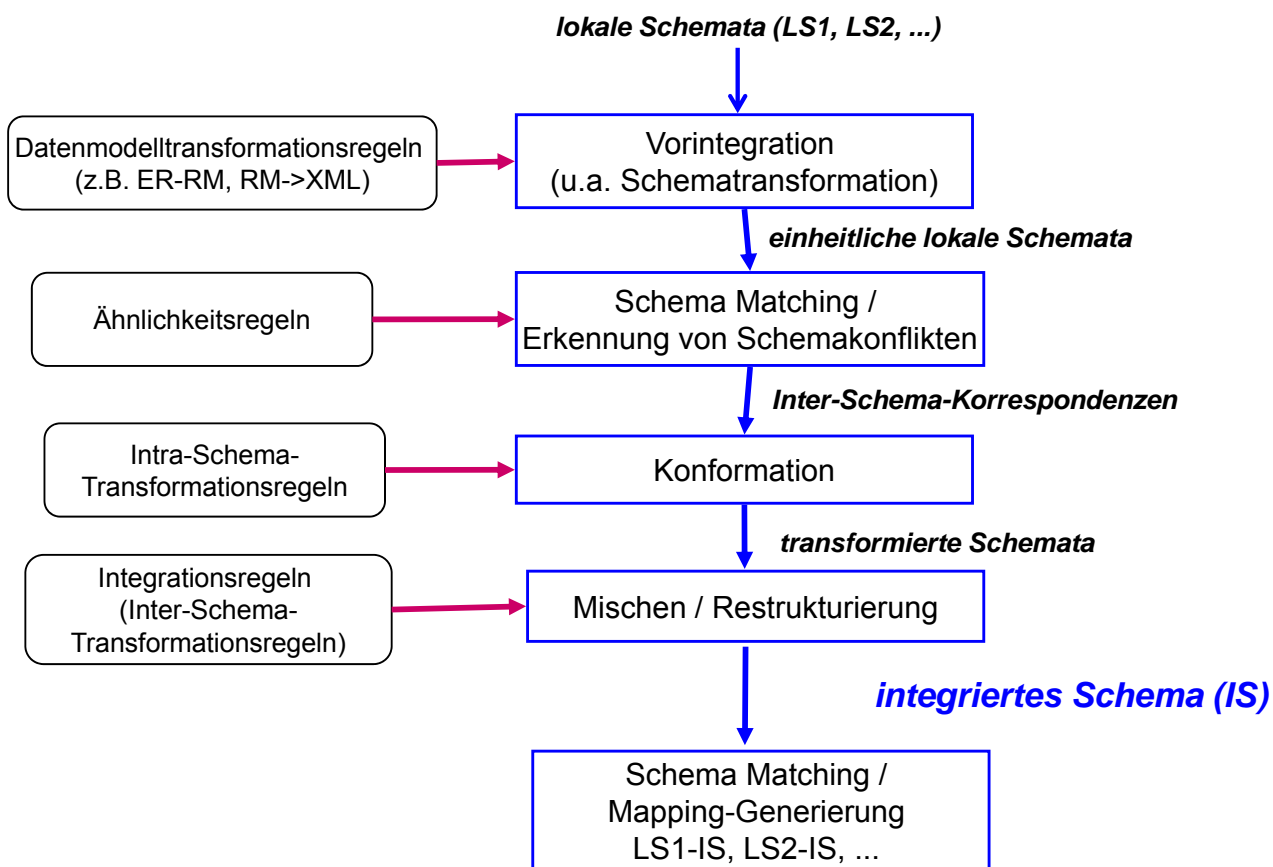


Bottom-Up-Integration (*Global as View*)

- (vollständiges) Mischen aller Source-Schemata in globales Schema
- setzt Abgleich zwischen Source-Schemas voraus, insbesondere Bestimmung von Korrespondenzen / Konflikten
- globales Schema entspricht gemeinsamer Sicht (View) auf die zugrundeliegenden Quellen
- neue Quelle ändert meist globales Schema

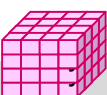
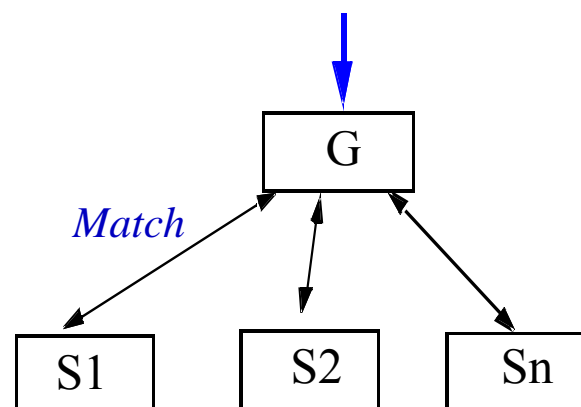


Bottom-Up-Schemaintegration (2)

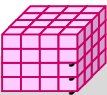
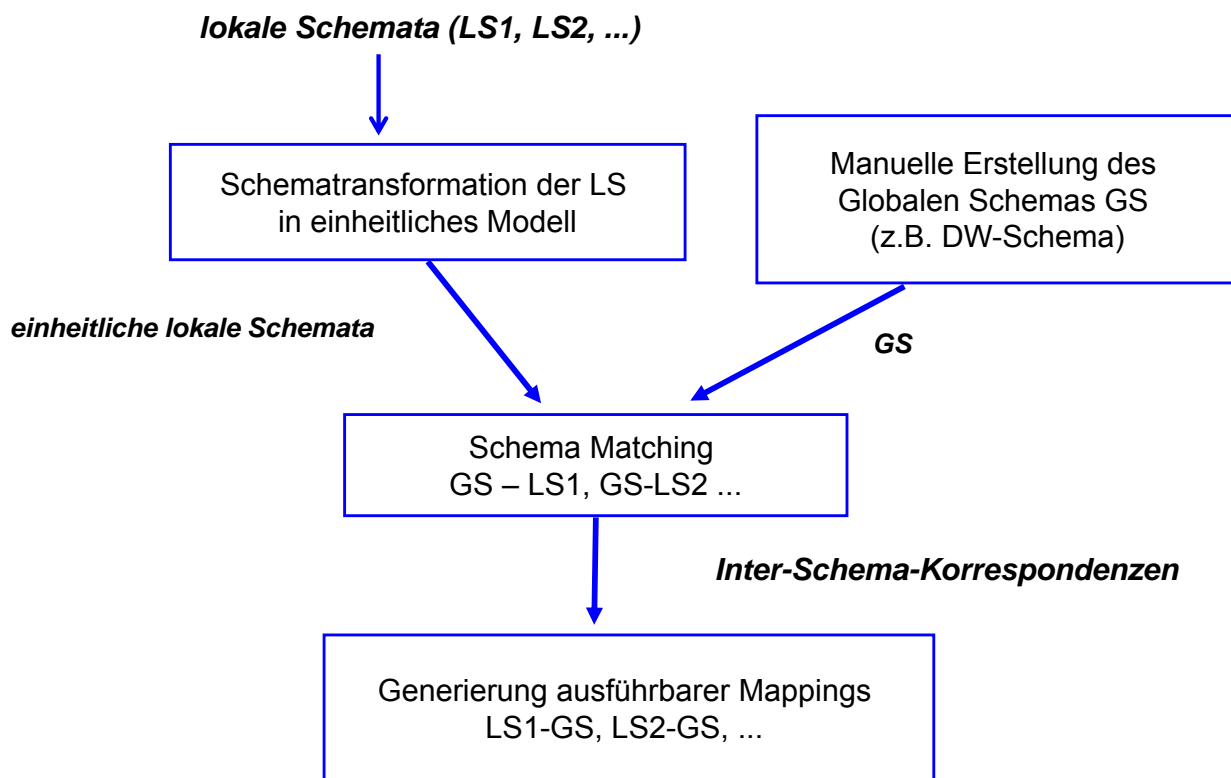


Top-Down-Integration (Local as View)

- globales Schema G ist vorgegeben
- jede Source S wird unabhängig von anderen Sources mit globalem Schema abgeglichen, d.h. ein Mapping $G - S$ erstellt (Mapping beschreibt Inhalt der Quelle)
- aufwändige Query-Verarbeitung bei virtueller Integration
- G berücksichtigt i.a. nur Teile der lokalen Schemata



Top-Down-Schemaintegration (2)



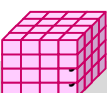
Namenskonflikte

- **Synonyme:** Repräsentation ein und desselben Konzepts durch unterschiedliche Namen:
- **Homonyme:** Nutzung gleicher Namen für verschiedene Konzepte
- **Hyponyme/Hyperonyme:** Unter-/Oberbegriffe

Mitarbeiter
Name
Adresse

Firma
Name
Adresse

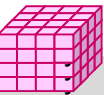
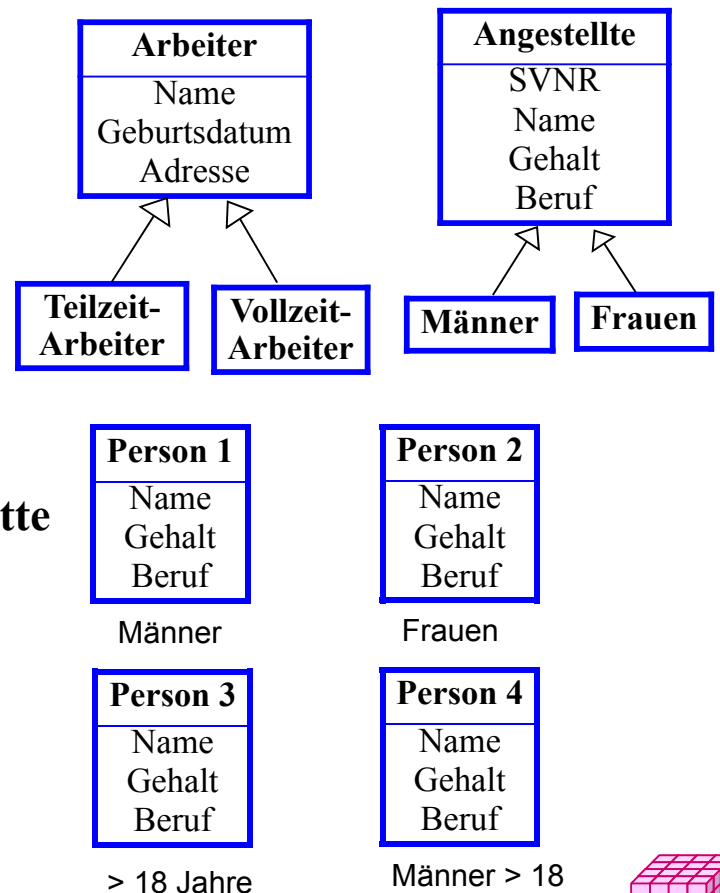
Angestellte
Name
Anschrift



Strukturelle Konflikte

Entity vs. Entity

- unterschiedliche Schlüssel
- unterschiedliche Attributmengen, fehlende Attribute
- unterschiedliche Abstraktionsebenen (Generalisierung, Aggregation)
- **unterschiedliche Realitätsausschnitte** (RWS, real world states)
 - disjunkt (disjoint):
 - überlappend (overlaps):
 - enthalten (contains):



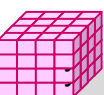
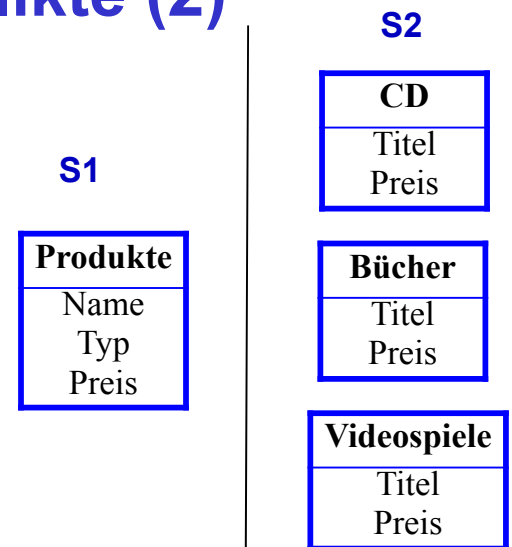
Strukturelle Konflikte (2)

Attribut vs. Entity-Konflikte

- Repräsentation von Attributen als eigenständige Entities/Relationen

Attribut vs. Attribut-Konflikte

- unterschiedliche Datentypen
Preis (Float) vs. Preis (String)
- unterschiedliche Detailgrade
Name vs. Vorname und Nachname
- unterschiedliche Einheiten: *\$ vs. Euro*
- unterschiedliche Genauigkeiten: *Tausend Euro vs. Euro*
- unterschiedliche Integritätsbedingungen, Wertebereiche, Default-Werte ...
Alter > 18 vs. Alter > 21
- unterschiedliche Behandlung von Nullwerten
- unterschiedliche Verwaltung der referentieller Integrität



Behandlung von Konflikten

■ Konflikterkennung: Vergleich der Schemata

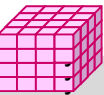
- Identifikation der ähnlichen/gleichen in den Schemata enthaltenen Information
- Identifikation verschiedener Strukturen, die ähnliche Informationen repräsentieren

■ Repräsentation der Inter-Schema-Korrespondenzen

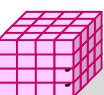
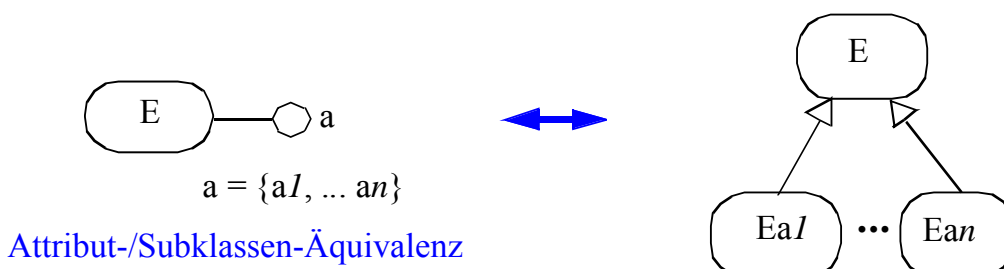
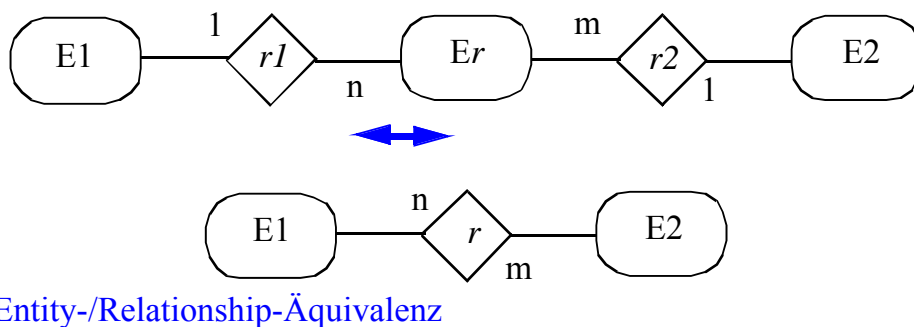
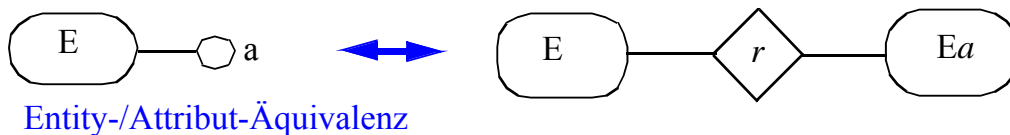
- Synonyme, Matches: Kunde = Klient
- Is-A-Korrespondenzen: Angestellte is-a Person
- RWS-Korrespondenzen: RWS(Produkte) contains RWS(Bücher)

■ Behebung von Schemakonflikten v.a. bei Bottom-Up-Integration (Merge) erforderlich

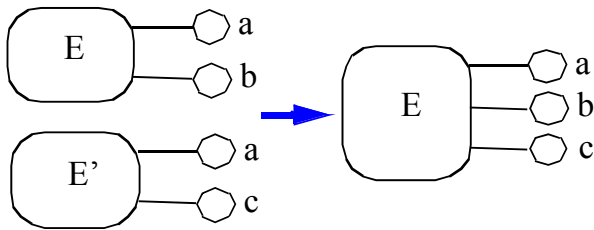
- Umbenennungen zur Behebung von Namenskonflikten
- Schemakonformation und –restrukturierungen
- Top-Down-Integration: Spezifikation notwendiger Transformationen bei Mapping-Erzeugung



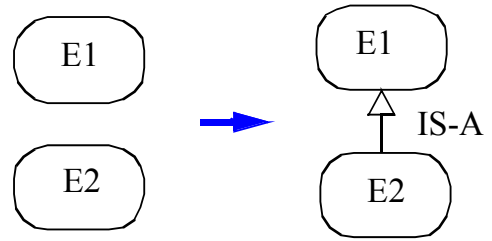
Schema-Konformationstransformationen



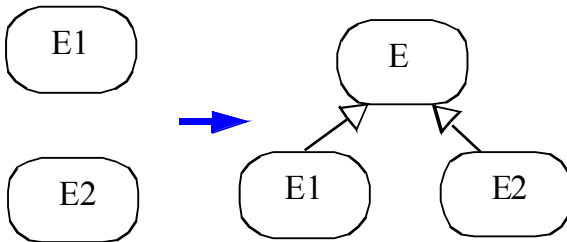
Schema-Merging-Transformationen



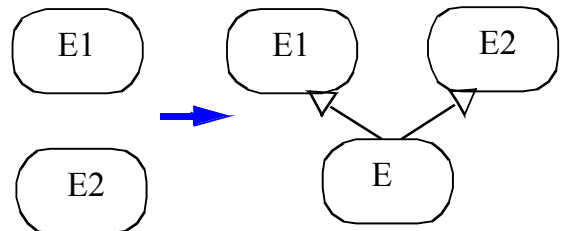
Vereinigung der Attribute



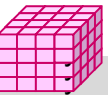
Einführung einer IS-A-Beziehung
 $RWS(E1)$ contains $RWS(E2)$



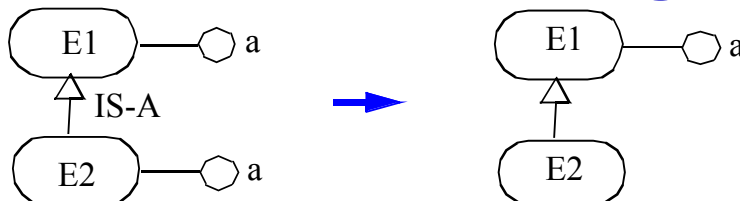
Einführung einer Superklasse
 $RWS(E1)$ disjoint $RWS(E2)$



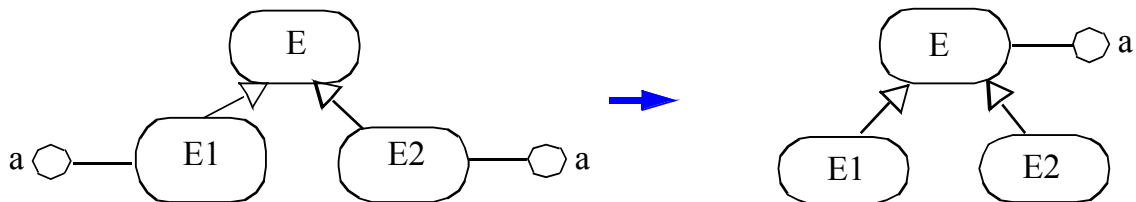
Einführung einer Subklasse
 $RWS(E1)$ overlaps $RWS(E2)$



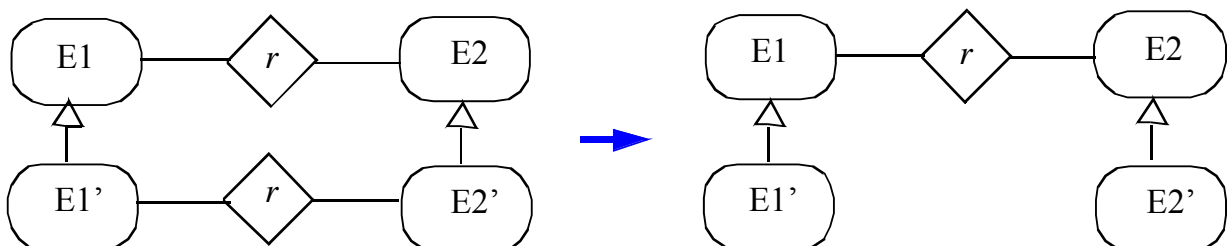
Schema-Restrukturierungstransformationen



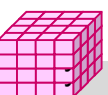
Entfernung redundanter Attribute



Generalisierung von Attributen

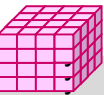


Generalisierung von Beziehungen



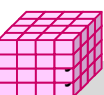
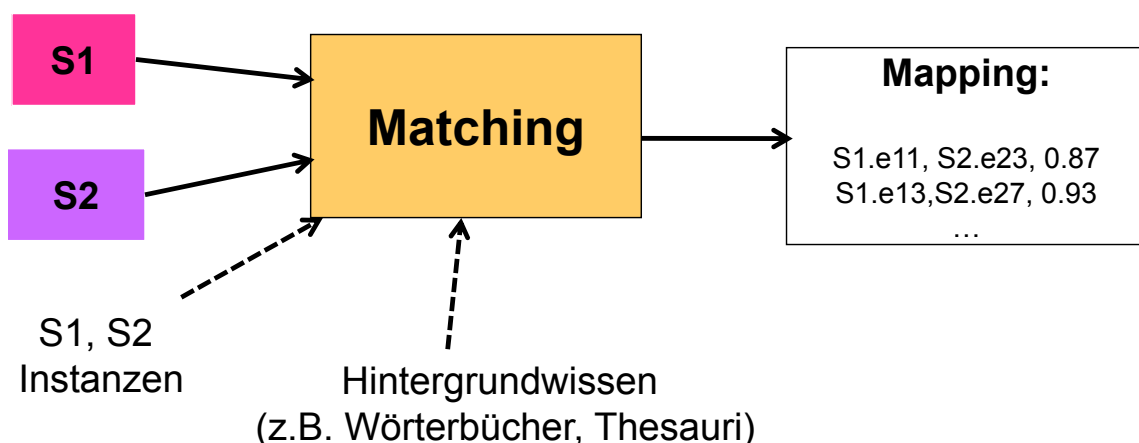
Automatisierungsbedarf

- bisherige Schemaintegrationsansätze weitgehend manuell
 - nutzerdefinierte Korrespondenzen und Konfliktbehandlung
 - Nutzung spezifischen Schema-/Domain-Wissens
 - aufwändig / fehleranfällig vor allem für größere Schemata
 - nicht skalierbar auf viele Schemata
 - Hoher Anpassungsaufwand bei Schemaänderungen
- Skalierbarkeit erfordert semi-automatische Lösungen / Tools!
 - vollautomatische Lösungen aufgrund semantischer Heterogenität nicht möglich
 - Namensproblematik (Synonyme, Homonyme)
 - begrenzte Mächtigkeit von Metadaten / Schemasprachen
- (Teil-)Automatisches Schema-Matching
 - v.a. für große Schemata wichtig
 - Nutzer-Feedback notwendig, jedoch im begrenzten Umfang

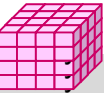
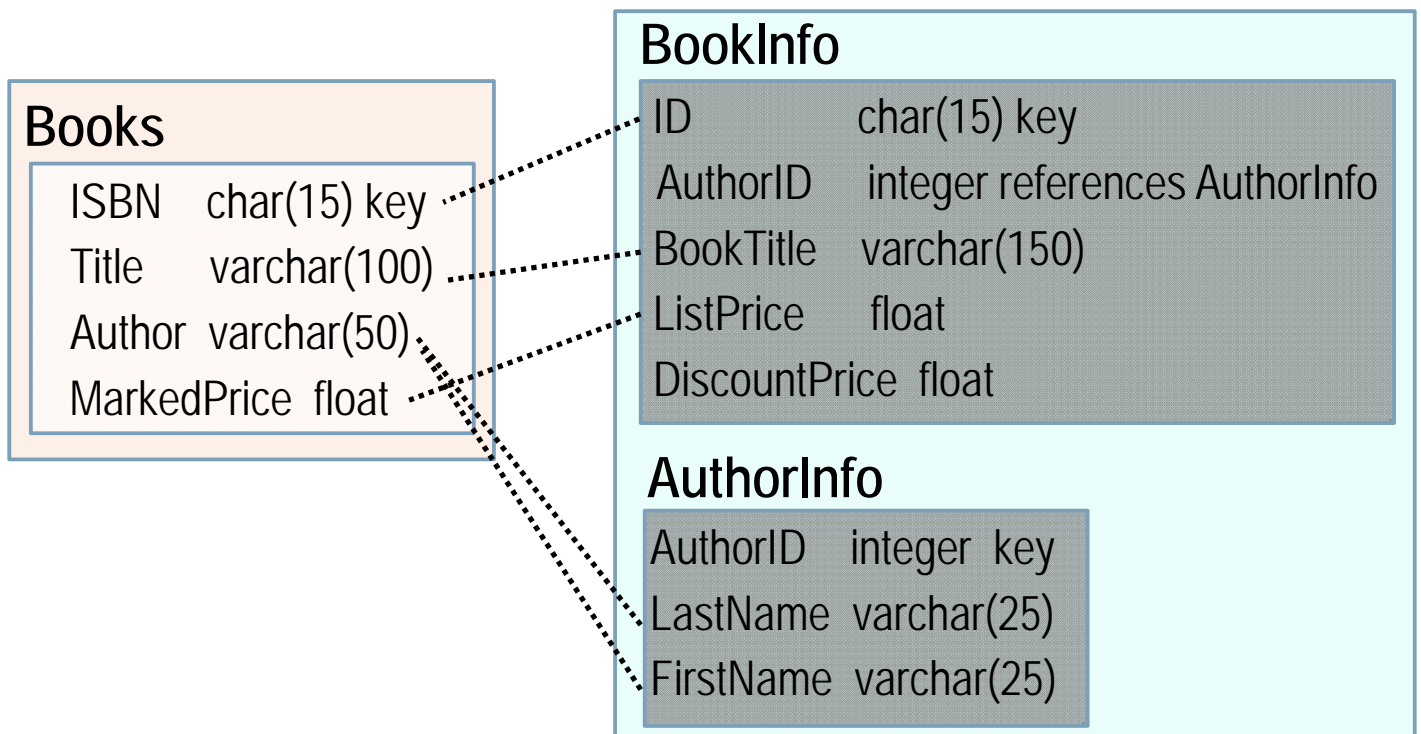


Schema Matching

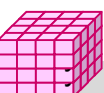
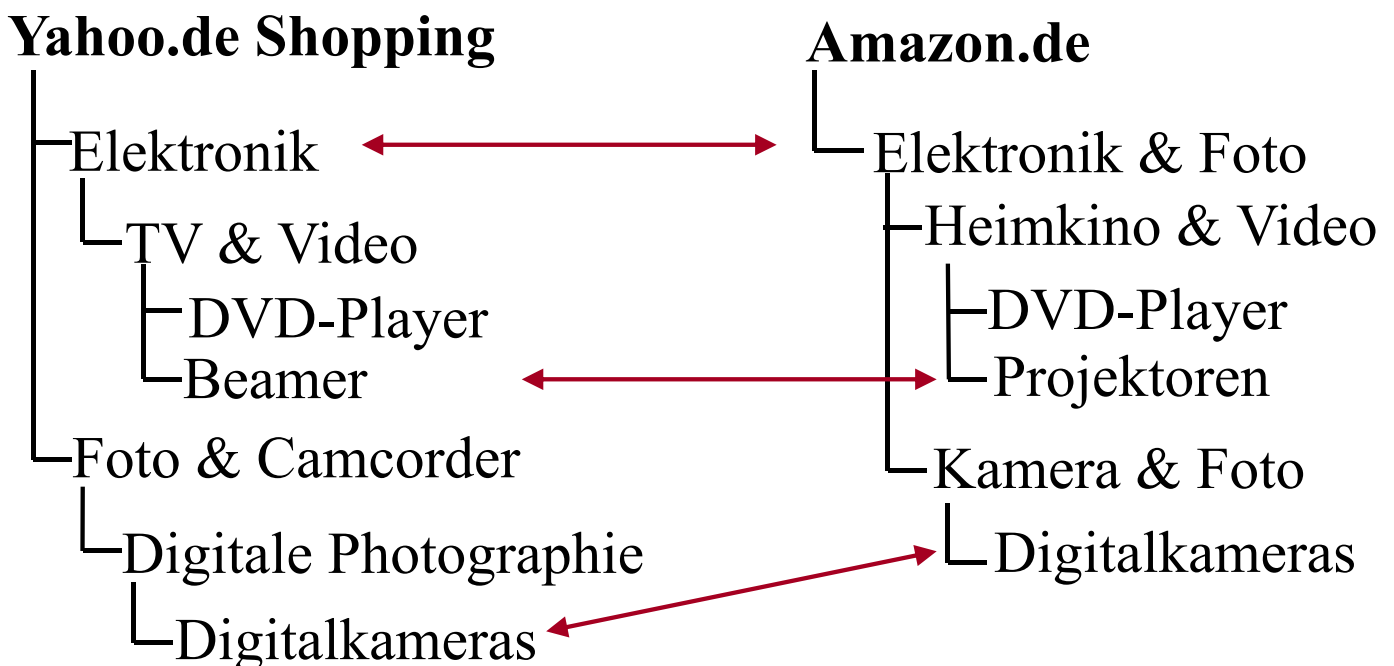
- Finden semantischer Korrespondenzen zwischen 2 Schemas
 - DB-Schemas, XML-Schemas, Ontologien, ...
- Kritischer Schritt in zahlreichen Applikationen
 - Datenintegration: Data Warehouses, Mediatoren, P2P
 - E-Business: XML Message Mapping; Katalogintegration
 - Semantic Web: Ontology Matching



Match-Beispiel 1: relationale Schemas



Match-Beispiel 2: Produktkataloge



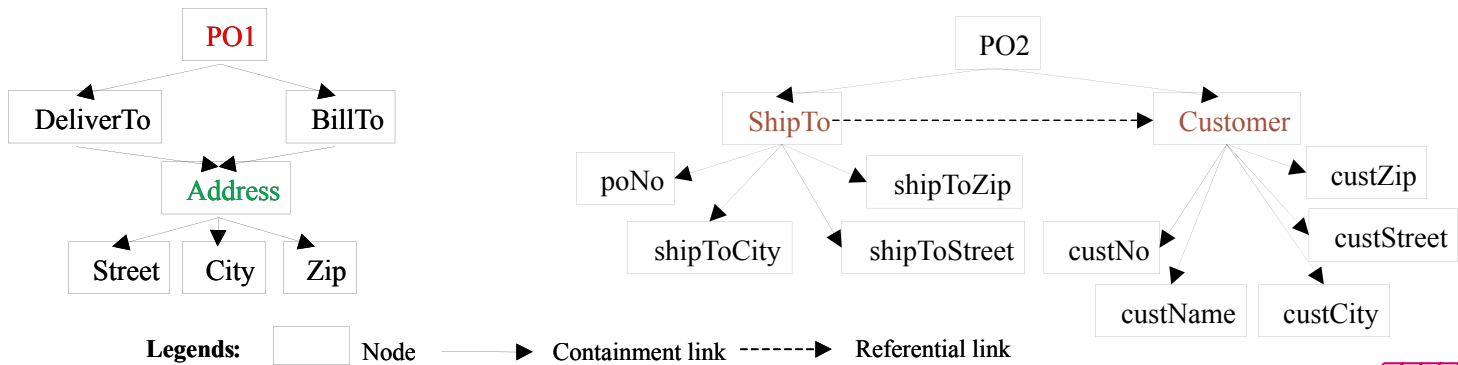
Match-Beispiel 3: Schemas (XML, relational)

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:complexType name="PO1">
  <xsd:sequence>
    <xsd:element name="DeliverTo" type="Address"/>
    <xsd:element name="BillTo" type="Address"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Address">
  <xsd:sequence>
    <xsd:element name="Street" type="xsd:string"/>
    <xsd:element name="City" type="xsd:string"/>
    <xsd:element name="Zip" type="xsd:decimal"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

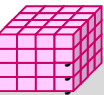
PO schema 1 (XML)

```
CREATE TABLE PO2. ShipTo (
  poNo      INT,
  custNo    INT REFERENCES PO2.Customer,
  shipToStreet VARCHAR(200),
  shipToCity  VARCHAR(200),
  shipToZip   VARCHAR(20),
  PRIMARY KEY (poNo)
);
CREATE TABLE PO2.Customer (
  custNo    INT,
  custName  VARCHAR(200),
  custStreet VARCHAR(200),
  custCity  VARCHAR(200),
  custZip   VARCHAR(20),
  PRIMARY KEY (custNo)
);
```

PO schema 2 (relational)

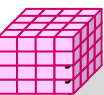


Graph representation of schemas (COMA++)



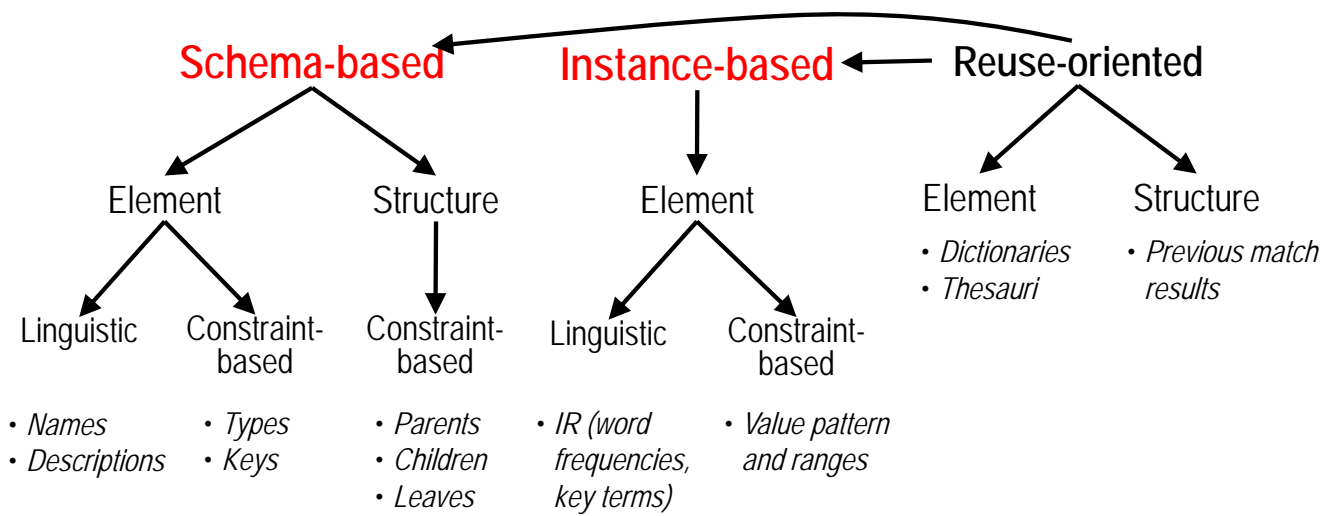
Match Tool: Altova MapForce

The screenshot shows the Altova MapForce interface. On the left is a 'Libraries' pane with categories like 'xpath2', 'accessors', 'boolean functions', and 'constructors'. The main workspace displays a 'Mapping Project' with three source schemas: 'ShortPO', 'Customers', and 'CompletePO'. 'ShortPO' has nodes 'CustomerNr', 'Lineltms', 'Lineltm', 'ArticleNr', and 'Amount'. 'Customers' has nodes 'Customer', 'Number', 'FirstName', 'LastName', 'Address', 'Street', 'City', and 'ZIP'. 'CompletePO' has nodes 'Customer', 'Number', 'FirstName', 'LastName', 'Address', 'Street', 'City', 'ZIP', 'State', 'Lineltms', 'Lineltm', 'Article', 'Number', 'Name', 'SinglePri', and 'Amount'. The mapping is visualized with arrows and function blocks like 'filter', 'equal', and 'result'. A red arrow highlights a specific mapping path.



Automatische Match-Ansätze*

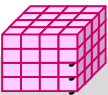
■ Einzelne Ansätze



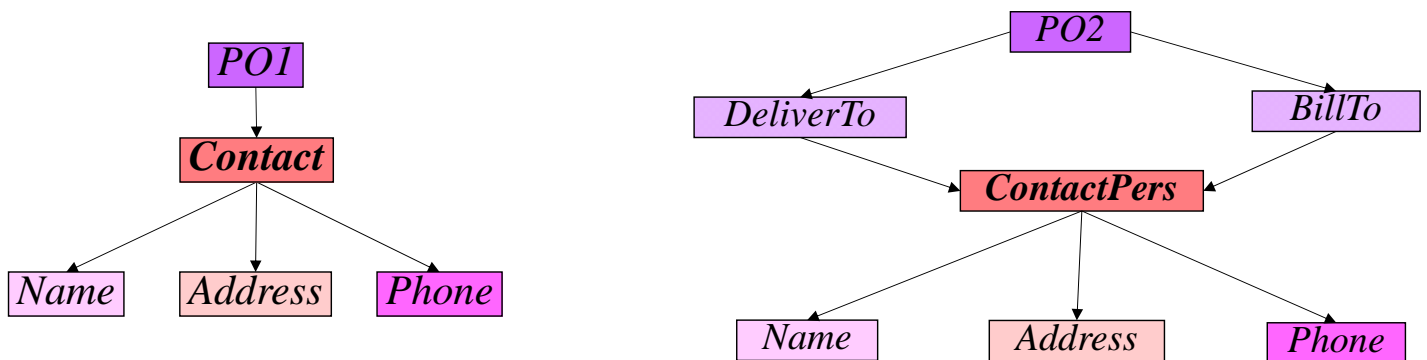
■ Kombinerende Ansätze

- Hybride Matcher (z.B. Name +Type)
- Kombination unabhängiger Matcher (Match-Workflows)

* Rahm, E., P.A. Bernstein: *A Survey of Approaches to Automatic Schema Matching*. VLDB Journal 10 (4), 2001



Match-Granularität

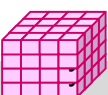


■ Element Matching

- Matching zwischen einzelnen Schema-Elementen (Blätter, innere Knoten)
- Knoten- vs. Pfad-Repräsentation von Elementen

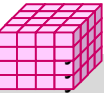
■ Strukturelles Matching

- Matching zwischen Teilschemas / Elementgruppen



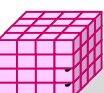
Linguistisches Matching

- einfachster Ansatz: Namensgleichheit
- syntaktische Ansätze: approximate String-Ähnlichkeit von Namen
 - N-grams, Edit distance, TF/IDF ...
- semantische Ähnlichkeit durch Nutzung terminologischer Beziehungen
 - Synonyme: *KFZ* ~ *Auto*
 - Hyponyme/Hyperonyme: *Buch* is-a *Publikation*, *Artikel* is-a *Publikation*
 - Nutzung von Wörterbüchern / Thesauri, z.B. WordNet
- **Vorverarbeitung** zur Behandlung kryptischer Namen, Auflösung von Abkürzungen, etc.
 - Tokenisierung von Namen: *PO_OrderNum* → {*PO*, *Order*, *Num*}
 - Expansion von Akronymen, Kurzformen: *PO* →
Num →



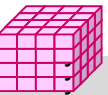
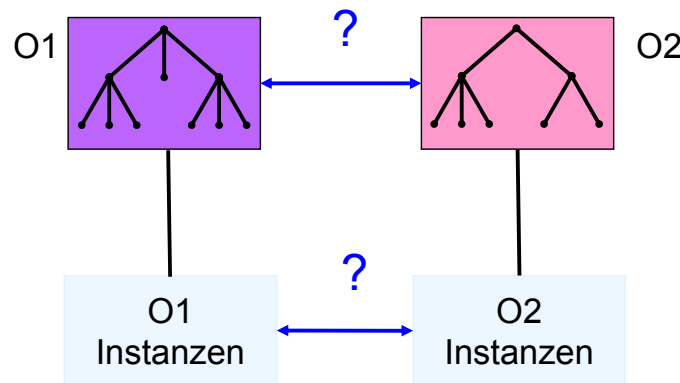
String-Matching: Beispiel

- „Street“ vs. „ShipToStreet“
- Edit Distance:
 - Distanz entspricht Anzahl notwendiger Einfüge-, Lösch- bzw. Änderungsschritte auf Zeichen bezogen auf maximale Stringlänge
 - Ähnlichkeit: $1 - \text{Distanz}$
- n-Gram (z.B. $n=3$ / Trigram):
 - Bestimme alle Zeichensequenzen der Länge n (n-grams)
 - Ähnlichkeit entspricht dem Anteil übereinstimmender n-grams



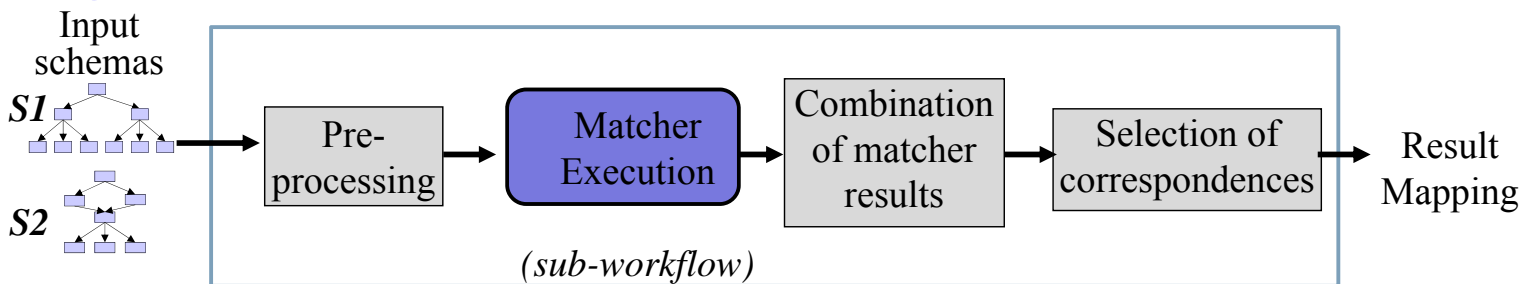
Instanz-basiertes Schema Matching

- Semantik von Schemaelementen / Ontologiekonzepten wird oft besser durch die zugeordneten Instanzdaten als durch Namen, Beschreibung etc. ausgedrückt
- Match für Elemente mit den ähnlichsten Instanzdaten
 - erfordert jedoch ähnliche Instanzdaten für viele Elemente/ Konzepte
- Überführung des Schema-Matching- in ein Objekt-Matching-Problem

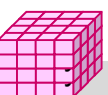
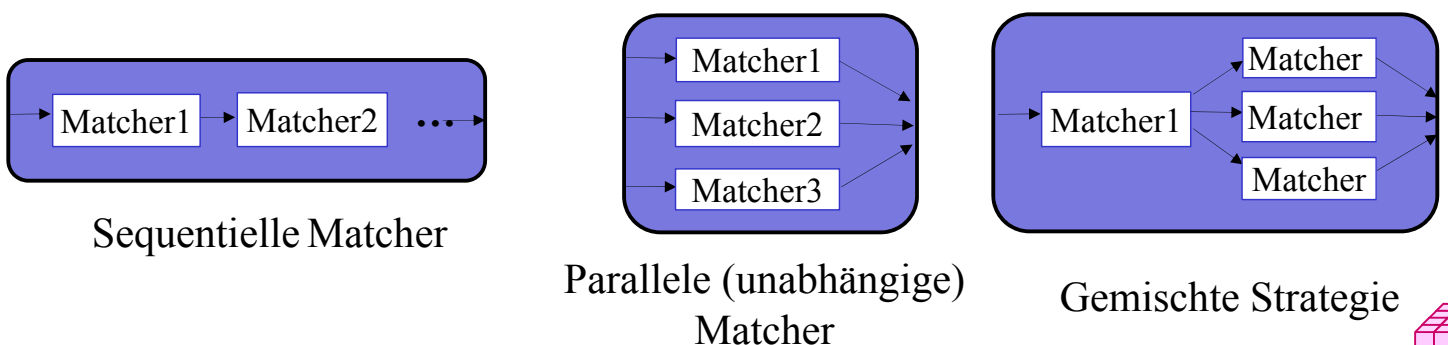


Kombination von Matchern

allgemeiner Match-Workflow (COMA, ...)



Matcher-Ausführung:



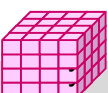
Match-Prototypen



Vergleich bekannter Match-Prototypen *

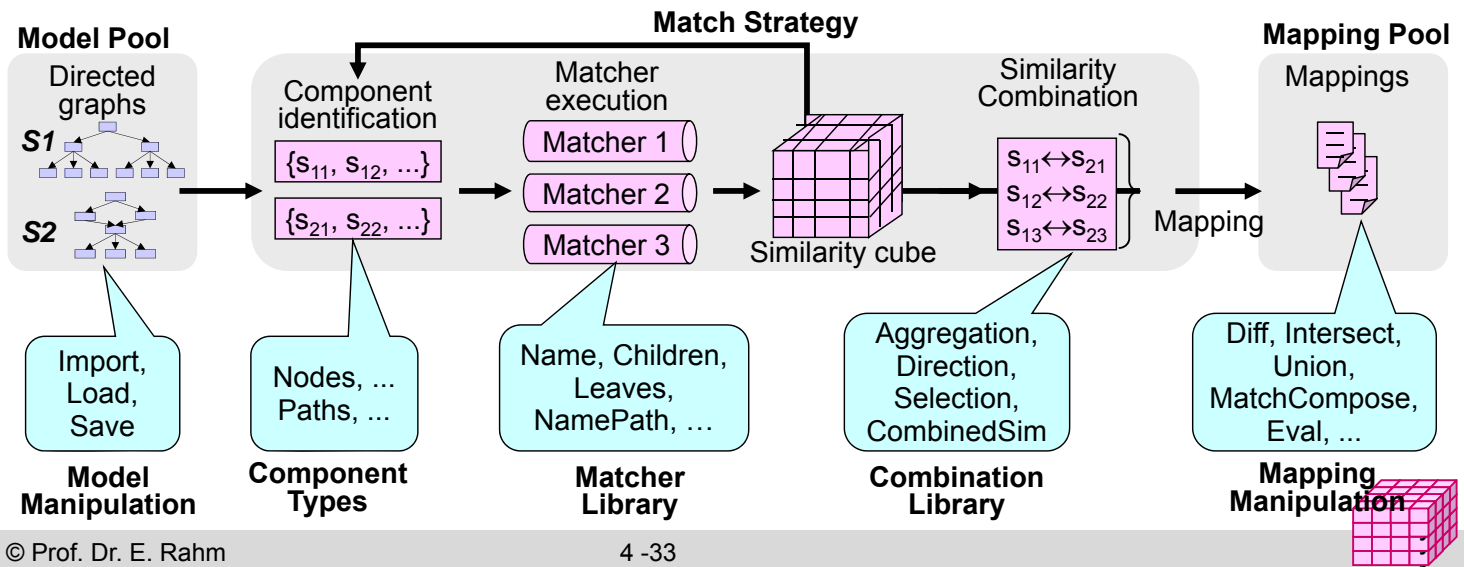
	Cupid	COMA++	Falcon	Rimom	Asmov	Agr. Maker	Oll Harmony	GOMMA
Year of introduction	2001	2002/2005	2006	2006	2007	2007	2008	2010
Input	<i>relational</i>	✓	✓	-	-	-	✓	-
schemas	<i>XML</i>	✓	✓	-	-	✓	✓	-
	<i>ontologies</i>	-	✓	✓	✓	✓	✓	✓
OAEI participation	-	✓	✓	✓	✓	✓	-	✓
Comprehensive GUI	-	✓	(✓)	?	?	✓	✓	-
Matchers	<i>linguistic</i>	✓	✓	✓	✓	✓	✓	✓
	<i>structure</i>	✓	✓	✓	✓	✓	✓	✓
	<i>instance</i>	-	✓	-	✓	✓	-	✓
Use of external dictionaries	✓	✓	?	✓	✓	✓	✓	(-)
Schema partitioning	-	✓	✓	-	-	-	-	✓
Parallel matching	-	-	-	-	-	-	-	✓
Dynamic matcher selection	-	-	-	✓	-	-	-	-
Mapping reuse	-	✓	-	-	-	-	-	✓

*Rahm, E.: Towards large-scale schema and ontology matching. In: Schema Matching and Mapping, Springer-Verlag, 2011



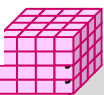


- generische Plattform für Schema- und Ontologie-Matching (XML, rel.,OWL)
- Schema- und Instanz-Matcher;
Parallele u. sequentielle Matcher-Kombination,
- Match-Strategien für große Schemas: Fragment-basiert / Wiederverwendung vorhandener Match-Ergebnisse
- umfassendes GUI
- 2012: Version COMA 3.0



COMA++ Screenshot

Source Information		Target Information	
All Nodes/Paths	117 / 117	Fisch	Name
Inner Nodes/Paths	14 / 14	Verzeichnis_Einkaufen-Sparen.Nahrung...	Path
Leaf Nodes/Paths	103 / 103		
Shared/Root No...	0 / 59		



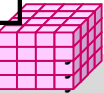
COMA: Matcher-Bibliothek

■ Basis-Matcher:

- String-Matcher: Synonym, Trigram, Affix, EditDistance
- Type-Matcher
- Instanz-Matcher
- Reuse-Matcher: Wiederverwendung von Mappings

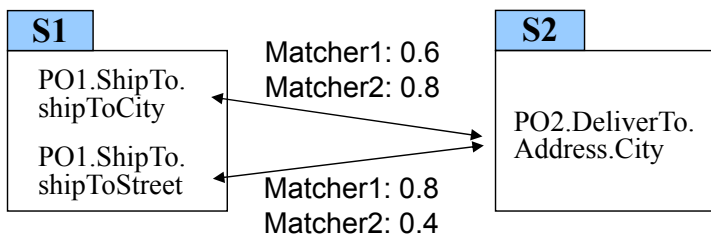
■ Hybrid-Matcher: feste Kombination anderer Matcher

Name	Constituents	Matchers/Sim measures	Combination
Name	Name tokens	Synonym/Taxonomy, Trigram	Avg, Both, Max1, Avg
NameType	Node	Name, Type	Wgt(0.7,03), Both, Max1, Avg
NameStat	Node	Name, Statistics	
Children	Children	NameType	Avg, Both, Max1, Avg
Leaves	Leaves	NameType	
Parents	Parents	Leaves	
Siblings	Siblings	Leaves	
NamePath	Ascendants	Name	

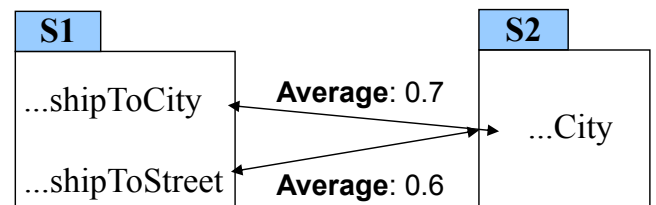


Kombination von Match-Ergebnissen: Beispiel

1. Matcher execution



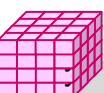
2. Aggregation



3. Selection

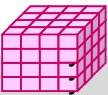
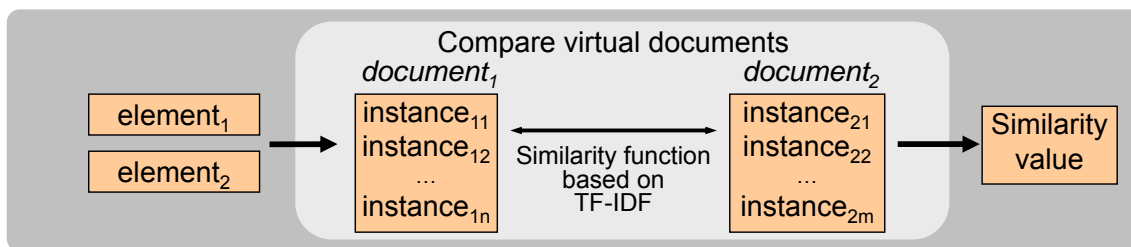
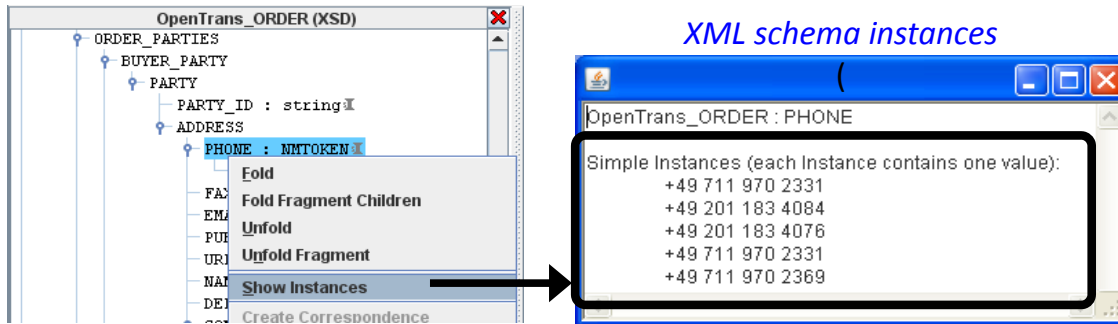
Max1		
S2 elements	S1 elements	Sim
...City	...shipToCity	0.7

Threshold(0.5)		
S2 elements	S1 elements	Sim
...City	...shipToCity	0.7
...City	...shipToStreet	0.6



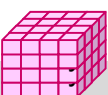
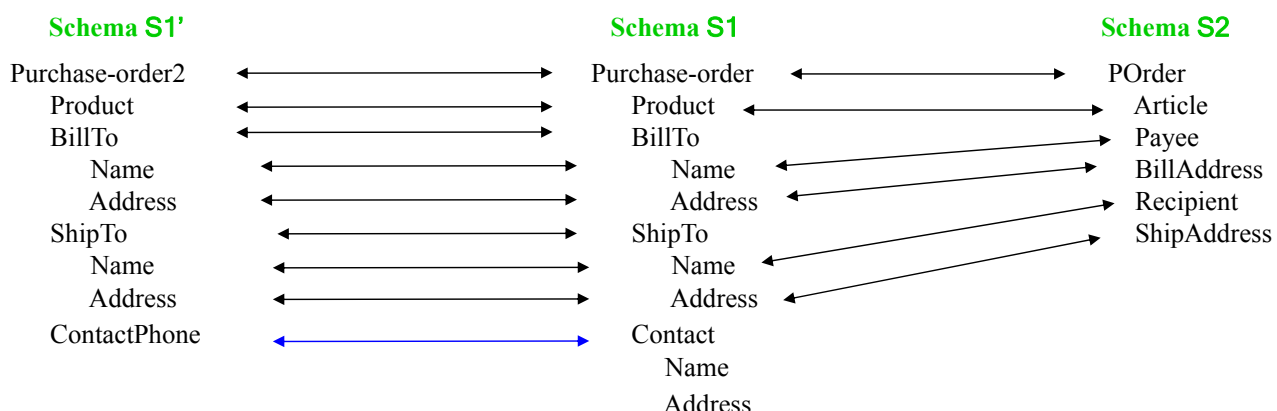
Instanz-basiertes Matching in COMA++

- ▶ Coma++ verwaltet pro Element Menge von Instanzen
- ▶ “Dokument-Ähnlichkeit”: Ähnlichkeitsvergleich für gesamte Instanzmenge basierend auf TF/IDF



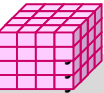
Wiederverwendung (Reuse)

- Nutzung von Hilfsquellen
 - nutzerspezifizierte Synonymtabellen
 - allgemeine/Domänenspezifische Vokabulare, Wörterbücher
 - Gemeinsame Ontologien
- Nutzung bereits bestätigter Match-Ergebnisse für ähnliche Match-Probleme
 - Speichern von Schemas und Mappings in Repository
 - besonders vorteilhaft für Abgleich neuer Schema-Versionen (Schema-Evolution)
- Beispiel: Wiederverwendung des vorhandenen (bestätigten) Mappings $S1—S2$ zur Lösung des neuen Match-Problems $S1'—S2$

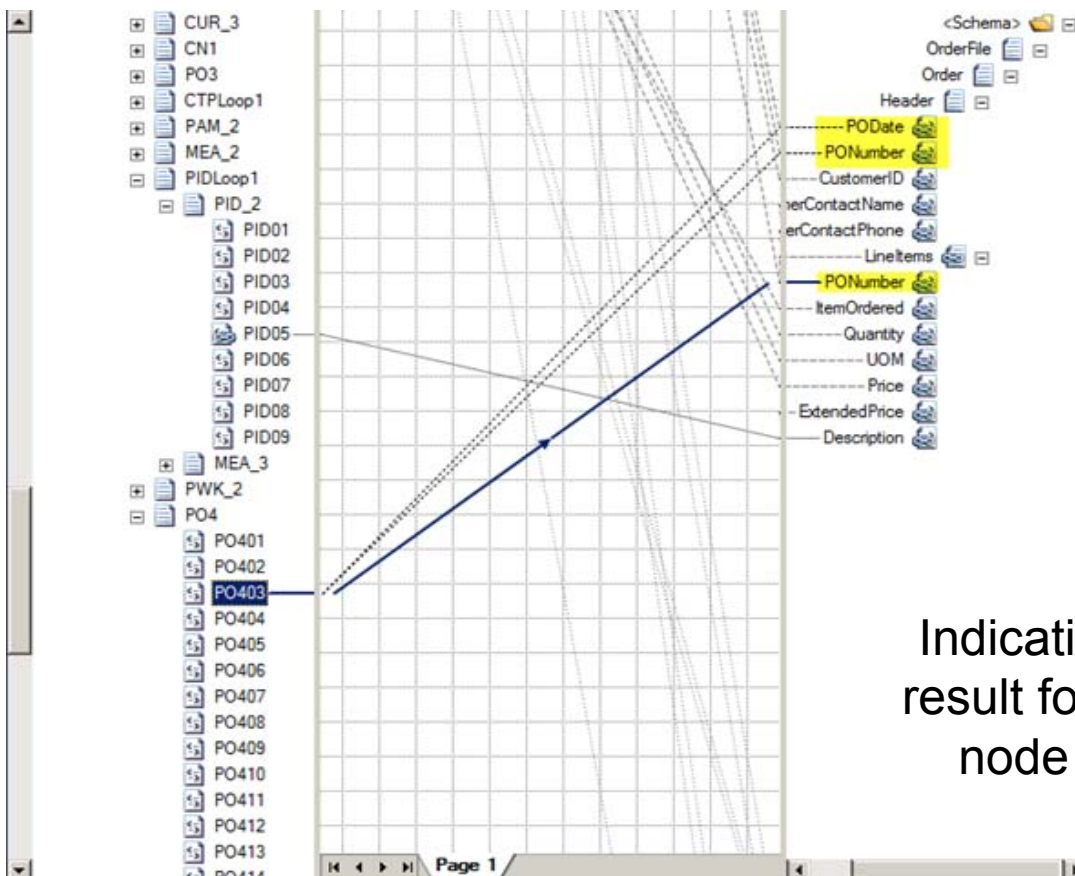


Kommerzielle Match Tools

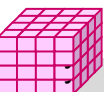
- Viele GUI-basierte Mapping Editors zur manuellen Spezifikation von Korrespondenzen und Mappings
- Zunehmende Unterstützung für semi-automatisches Matching, v.a. linguistisches Matching
 - Altova MapForce
 - MS BizTalk Server
 - SAP Netweaver
 - IBM Infosphere
- Viele weitere Verbesserungen noch möglich
 - strukturelles / instanz-basiertes Matching
 - Techniken für große Schemas



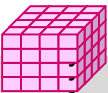
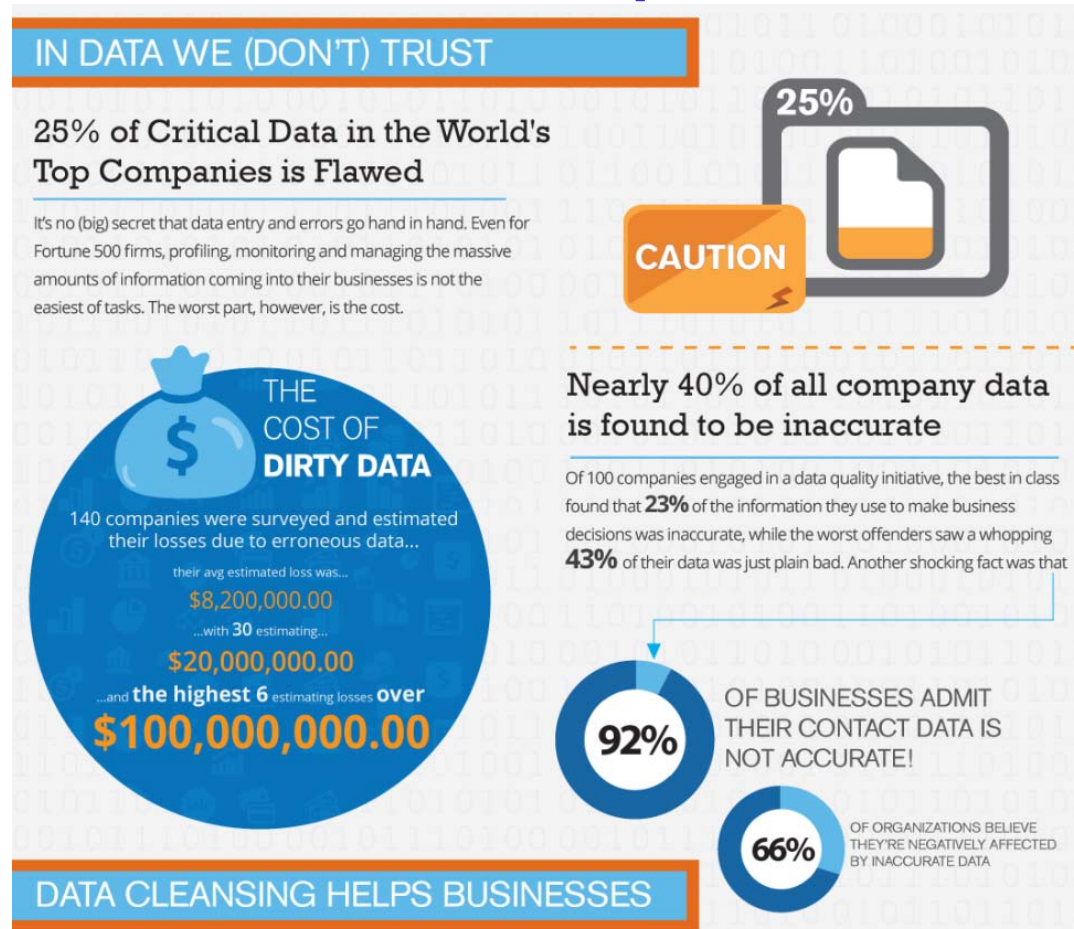
BizTalk 2010 Screenshot



Indicative match result for selected node PO403



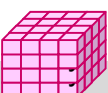
Datenqualität



Data Cleaning*

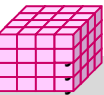
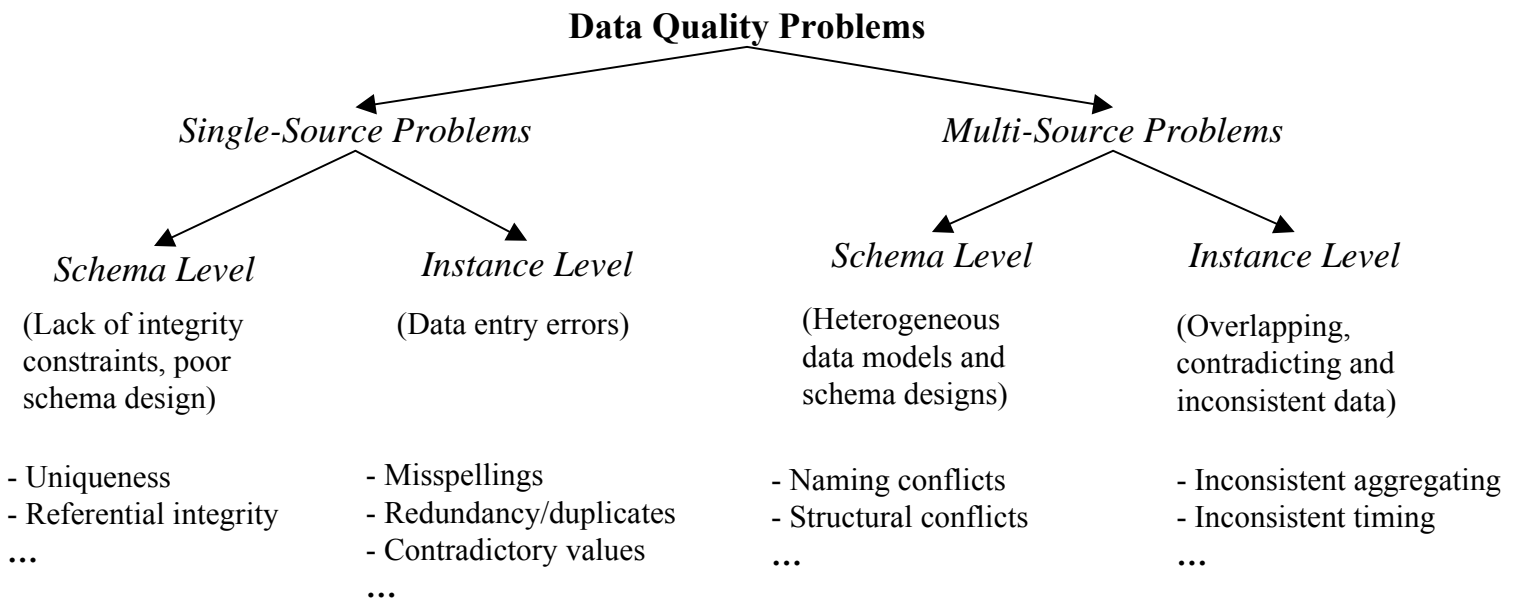
- **Datenanalyse**
 - Entdeckung von Datenfehlern und -inkonsistenzen
 - manuell bzw. Einsatz von Analyse-Tools
- **Definition von Mapping-Regeln und Transformations-Workflows**
 - Datentransformationen auf Schemaebene
 - Cleaning-Schritte zur Behandlung von Instanzdaten
 - deklarative Spezifikation erlaubt automatische Generierung von ausführbaren Skripts
- **Test / Verifizierung der Transformations-Workflows**
 - Korrektheit und Effektivität auf Kopien/Ausschnitt der Daten
- **Transformation**
 - regelmäßige Ausführung der geprüften Transformationsschritte
- **ggf. Rückfluss korrigierter Daten in operative Quellsysteme**

* E. Rahm, H. H. Do: *Data Cleaning: Problems and Current Approaches*.
IEEE Techn. Bulletin on Data Engineering, Dec. 2000



Probleme bezüglich Datenqualität

- Probleme auf Schema- und auf Instanzebene
- Probleme bezüglich einer oder mehrerer Datenquellen (Single-Source vs. Multi-Source)



Single-Source Probleme

■ Ursachen:

- Fehlen von Schemata (z.B. bei Dateien) und von Integritäts-Constraints
- Eingabefehler
- unterschiedliche Änderungsstände
- Unvollständigkeit ...

Name	Adresse	Phone	Erfahrung	Beruf
Peter Meier	Humboldtstr. 12, 04173 Leipzig	9999- 999999	A	Dipl- Informatiker
Schmitt, Ingo	Lessingplatz 1, 98321 Berlin	030- 9583014	M	Dipl.-Inf.
..

Multivalue-
Feld

Misspelling

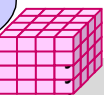
Fehlender
Wert

Transposition

Attributwert-
abhängigkeit

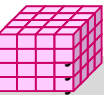
Kryptische
Werte

Uneinheitliche
Bezeichnungen



Multi-Source-Probleme

- überlappende, widersprüchliche bzw. inkonsistente Daten
 - aufgrund unabhängiger Erzeugung / Speicherung in verschiedenen Quellen
- Hauptproblem: Behandlung überlappender Daten
 - gängige Bezeichnungen: *Duplikate*, *Merge/Purge-Problem*, *Object Identity Problem*, *Record Linkage*
 - Beschreibung einer Instanz der realen Welt durch mehrere Datensätze unterschiedlicher Quellen
 - oft nur teilweise Redundanz (einzelne Attribute, nur in Teilmenge der Datenquellen) -> Fusion der Instanzen notwendig
- Unterschiedliche Repräsentationen der Instanzdaten
 - versch. Wertebereiche (z.B. *Geschlecht* = {1,2} vs. *Gender* = {m,w})
 - verschiedene Einheiten (z.B. *Verkauf in EUR* vs. *Verkauf in Tsd.EUR*)
 - verschiedene Genauigkeiten
- unterschiedliche Änderungsstände und Aggregationsstufen der Quelldaten



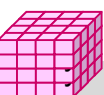
Multi-Source-Dateninkonsistenzen: Beispiel

Source1: Customer

<i>CID</i>	<i>Name</i>	<i>Street</i>	<i>City</i>	<i>Sex</i>
11	Kristen Smith	2 Hurley Pl	South Fork, MN 48503	0
24	Christian Smith	Hurley St 2	S Fork MN	1

Source2: Client

<i>Cno</i>	<i>LastName</i>	<i>FirstName</i>	<i>Gender</i>	<i>Address</i>	<i>Phone/Fax</i>
24	Smith	Christoph	M	23 Harley St, Chicago IL, 60633-2394	333-222-6542 / 333-222-6599
493	Smith	Kris L.	F	2 Hurley Place, South Fork MN, 48503-5998	444-555-6666



Beispiel (2)

<i>CID</i>	<i>Name</i>	<i>Street</i>	<i>City</i>	<i>Sex</i>
11	Kristen Smith	2 Hurley Pl	South Fork, MN 48503	0
24	Christian Smith	Hurley St 2	S Fork MN	1

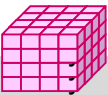
Source1:
Customer

<i>No</i>	<i>LName</i>	<i>FName</i>	<i>Gender</i>	<i>Street</i>	<i>City</i>	<i>State</i>	<i>ZIP</i>	<i>Phone</i>	<i>Fax</i>	<i>CID</i>	<i>Cno</i>
1	Smith	Kristen L.	F	2 Hurley Place	South Fork	MN	48503-5998	444-555-6666		11	493
2	Smith	Christian	M	2 Hurley Place	South Fork	MN	48503-5998			24	
3	Smith	Christoph	M	23 Harley Street	Chicago	IL	60633-2394	333-222-6542	333-222-6599		24

Customers (Integrierte und bereinigte Daten)

<i>Cno</i>	<i>LastName</i>	<i>FirstName</i>	<i>Gender</i>	<i>Address</i>	<i>Phone/Fax</i>
24	Smith	Christoph	M	23 Harley St, Chicago IL, 60633-2394	333-222-6542 / 333-222-6599
493	Smith	Kris L.	F	2 Hurley Place, South Fork MN, 48503-5998	444-555-6666

Source2:
Client

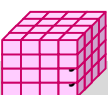


Datenanalyse

- Entdeckung von Fehlern / Verifikation korrekter Werte
- Ableitung von (wirklichen) Metadaten
- Berechnung der Statistiken zu Attributen auf Basis ihrer Instanzen
 - Datentyp, Länge, Maximum und Minimum, Null-, Default-Werte, Kardinalität, ...
 - Ermitteln von Wertebereichen, Häufigkeiten und Mustern von Attributwerte
- Erkennung von Ausreißern, funktionalen Abhängigkeiten

Attribute Values	#occurrences
IBM	3000
I.B.M.	360
Intel Bus Mach	213
International Business Machine	36

Instanzwerte	Pattern	Identifizierte Datenkategorie
(978) 555-1212	(nnn) nnn-nnnn	Telefonnummer
036-55-1234	nnn-nn-nnnn	Social Security Number
abc@web.de	aaa@aaa.aa	Email-Adresse
12.03.2008	nn.nn.nnnn	Datum

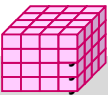


Behandlung von Single-Source-Problemen

- Definition und Einführung von Standardrepräsentationen
 - einheitliches Format für Datums-/Zeit-Angaben
 - einheitliche Groß/Kleinschreibungsform für Namen / String-Attribute
 - einheitliche Abkürzungen, Kodierungsschemas
- Bereitstellung von (Konversions-)Tabellen zur expliziten Werteabbildung

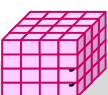
Legacy Value	New Value
IBM	IBM
I.B.M	IBM
Intel Bus Mach	IBM
...	...

- Extraktion von individuellen Werten aus Freiform-Attributen
 - Parsing und Attribut-Splitting, z.B. *Name* -> *Vorname* / *Nachname*
 - Reorganisierung der Wortreihenfolge
- Validierung / Korrektur mit Hintergrundwissen
 - Überprüfung/Spell checking mit Wörterbüchern Datenbanken mit Adressen, Produktbezeichnungen, Akronymen/Abkürzungen, etc.
 - Nutzung bekannter Attributabhängigkeiten zur Korrektur von fehlenden / falschen Attributwerten



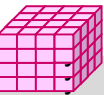
Behandlung von Multi-Source-Problemen

- Hauptproblem: Entdecken von Duplikaten bzw. korrespondierender Objekte (Objekt Matching)
- Durchführung auf aufbereiteten und gesäuberten Quellen
- Hauptschritte: Identifikation von “ähnlichen” Records (Matching) und Mischen (Merge) zu einem Record mit allen relevanten Attribute ohne Redundanz
- *Exact Matching*: Existenz eines Attributs oder eine Attributkombination zur eindeutigen Identifikation der einzelnen Records
 - Nutzung der Standard-Equijoin-Operationen zur Zusammenführung der zugehörigen Records
 - Sortierung der Records über die Schlüsselattribute und Vergleich der benachbarten Records zur Duplikatidentifikation
- *Fuzzy Object Matching*: keine gemeinsamen Schlüsselattribute (Normalfall)

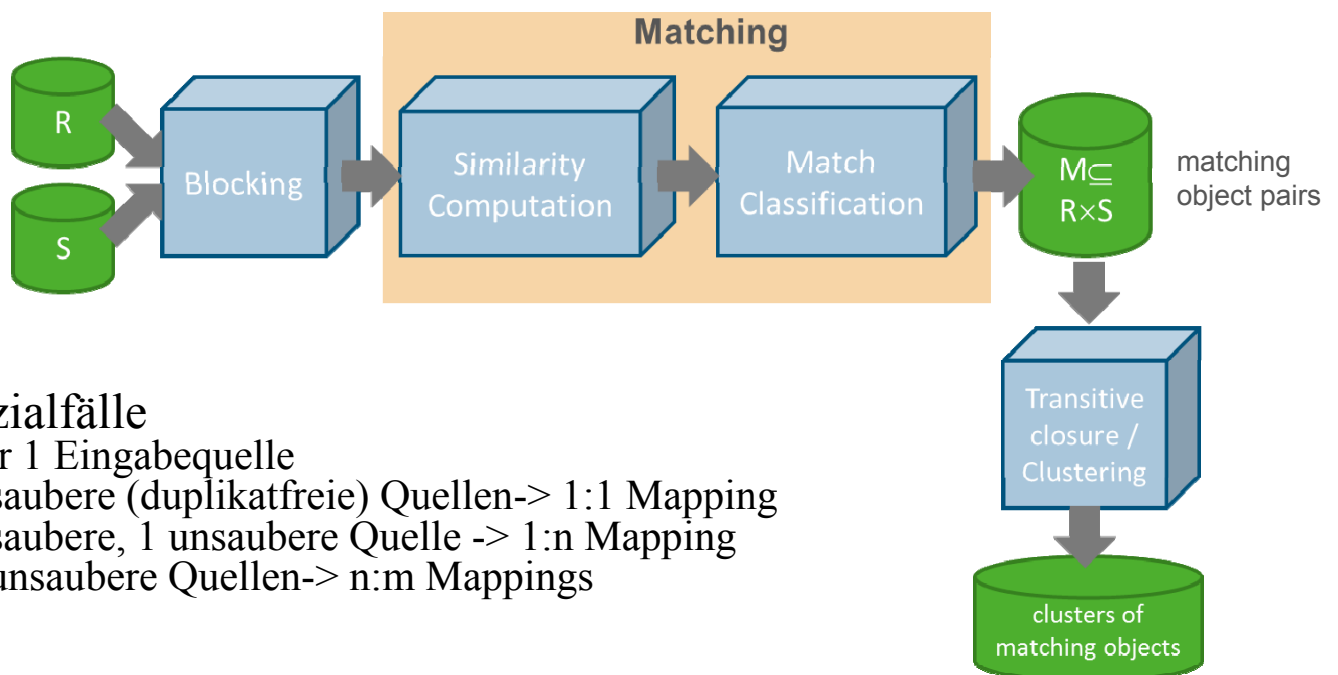


Fuzzy Object Matching

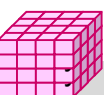
- Suche und Ranking ähnlicher Records auf Basis von Match-Verfahren bzw. (nutzerdefinierten) Matching-Regeln
 - Berechnung von Ähnlichkeitsmaßen zwischen Objektinstanzen / Sätzen zwischen 0 und 1
 - Robustheit gegenüber leicht variierenden Schreibweisen / Tippfehlern
 - Verwendung von Distanzfunktionen für String-Vergleiche: Edit-, Keyboard-Distanz, n-gram, TF/IDF, ...
 - Kombination der Attribut-Ähnlichkeiten zur Abschätzung der Satz-Ähnlichkeit
 - ggf. Gewichtung der Felder (z.B. hohe Gewichte für Namens-, Adressenfelder beim Matching von Personen-Records)
 - Berücksichtigung von Kontextinformationen (z.B. Gatte bei Personen, Koautoren bei Autoren, etc.)
- aktives Forschungsproblem zur Datenintegration



Allgemeiner Match-Workflow

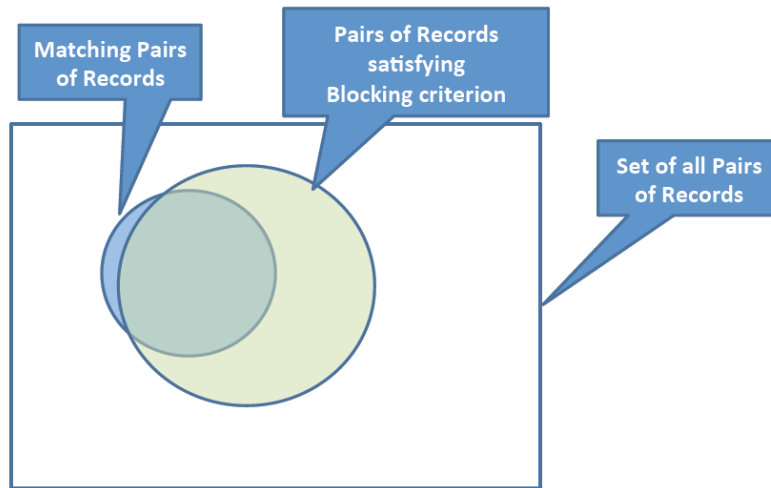


- Spezialfälle
 - nur 1 Eingabequelle
 - 2 saubere (duplikatfreie) Quellen-> 1:1 Mapping
 - 1 saubere, 1 unsaubere Quelle -> 1:n Mapping
 - 2 unsaubere Quellen-> n:m Mappings

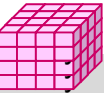


Fuzzy Object Matching (2)

- Performance-Probleme für große Datenmengen
 - Auswertung des kartesischen Produkts von Objektpaaren skaliert nicht
- Einsatz von **Blocking-Techniken**
 - vorhergehendes Ausfiltern sehr unähnlicher Objektpaare zur Einschränkung zu prüfender Match-Kandidaten

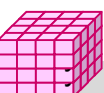


- Nutzung von Parallelität, z.B. über MapReduce/Hadoop



Blocking-Ansätze

- Indexing: nur Objektpaare mit gemeinsamen Teilstring werden geprüft
- Standard-Blocking
 - Partitionierung des Suchraums über Blocking Key
 - Ggf. mehrere Durchgänge mit unterschiedlichen Keys
- Sorted Neighborhood
 - Nachbarschaftssuche auf vorsortierten Daten



MS SQL-Server: Data Cleaning Operatoren

■ Seit 2005 Bestandteil von SQL-Server Integration Services (SSIS; vormals DTS)*

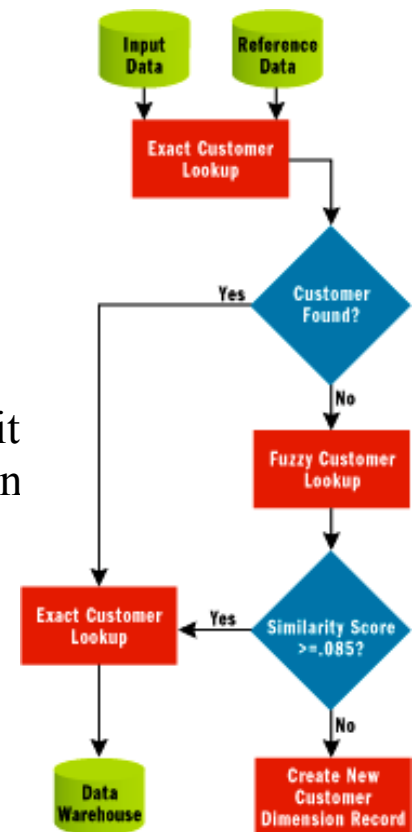
- Definition komplexer ETL-Workflows
- zahlreiche Operatoren

■ Fuzzy Lookup

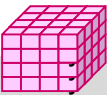
- “Fuzzy Join” zwischen Eingaberelation und sauberen Sätzen einer Referenztable
- Parameter: Schwellwerte bzgl. String-Ähnlichkeit (Edit Distance) sowie Gewichte zur Kombination von Ähnlichkeiten

■ Fuzzy Grouping

- Gruppierung ähnlicher Sätze (potentielle Duplikate) innerhalb einer Tabelle über String-Matching (Edit Distance)



* <http://msdn.microsoft.com/en-us/library/ms345128.aspx>



Match-Prototypen (U Leipzig)

■ MOMA (Mapping based Object Matching, 2007)

■ FEVER (2009)

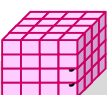
- Weiterentwicklung im WDI-Lab / Produkt-Matching
- Kommerzialisierung durch Spinoff

■ DEDOOP (2012)

- Paralleles Objekt-Matching auf Hadoop
- Lastbalancierung zur Behandlung von Skew-Effekten

■ Eigenschaften

- Unterstützung komplexer Match-Workflows mit mehreren Matchern
- Unterstützung lernbasierter Verfahren zur vereinfachten Konfigurierung
- Matching für Unternehmens- und Web-Daten



Duplikate in Webdaten: Beispiel

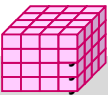
Canon VIXIA HF S10 Camcorder - 1080p - 8.59 MP - 10 x optical zoom \$975 new
Flash card, 32 GB, 1y warranty, F/1.8-3.0
The VIXIA HF S10 delivers brilliant video and photos through a Canon exclusive 8.59 megapixel CMOS image sensor and the latest version of Canon's advanced image processor, ...
★★★★☆ 12 reviews - [Add to Shopping List](#)
[Compare prices](#)

Canon (VIXIA) HF S10 iVIS Dual Flash Memory Camcorder \$899.00 new
Canon HF S10 iVIS Dual Flash Memory Camcorders SPECIAL SALE PRICE: \$899
Display both English/Japanese + we supply all English manuals in English as PDF.
[Add to Shopping List](#)
Made in Japan Online

Canon VIXIA HF S10 Dual Flash Memory High Definition Camcorder \$999.00 new
The Next Step Forward in HD Video
Canon has a well-known and highly-regarded reputation for optical excellence,
[Add to Shopping List](#)
Performance Audio
2 seller ratings

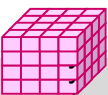
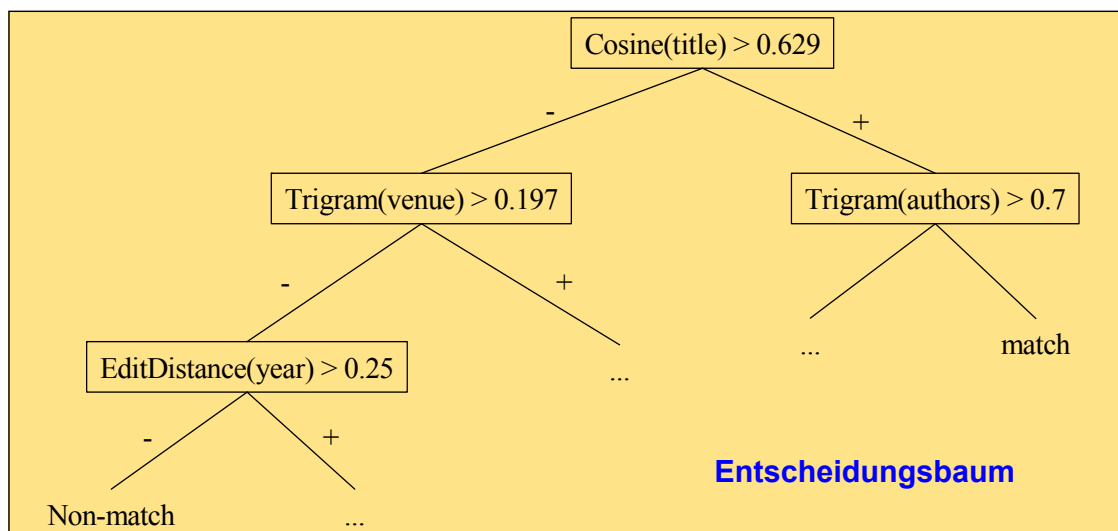
Canon VIXIA HF S100 Flash Memory Camcorder \$899.95 new
***Canon Video HF S100 Instant Rebate Receive \$200 with your purchase of a new Canon VIXIA HF S100 Flash Memory Camcorder. (Price above includes \$200
[Add to Shopping List](#)
Arlingtoncamera.com
5 seller ratings

Canon Vixia Hf S10 Care & Cleaning \$2.99 new
Care & Cleaning Digital Camera/Camcorder Deluxe Cleaning Kit with LCD Screen Guard Canon VIXIA HF S10 Camcorders Care & Cleaning.
[Add to Shopping List](#)
shop.com
★★★★☆ 38 seller ratings

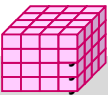
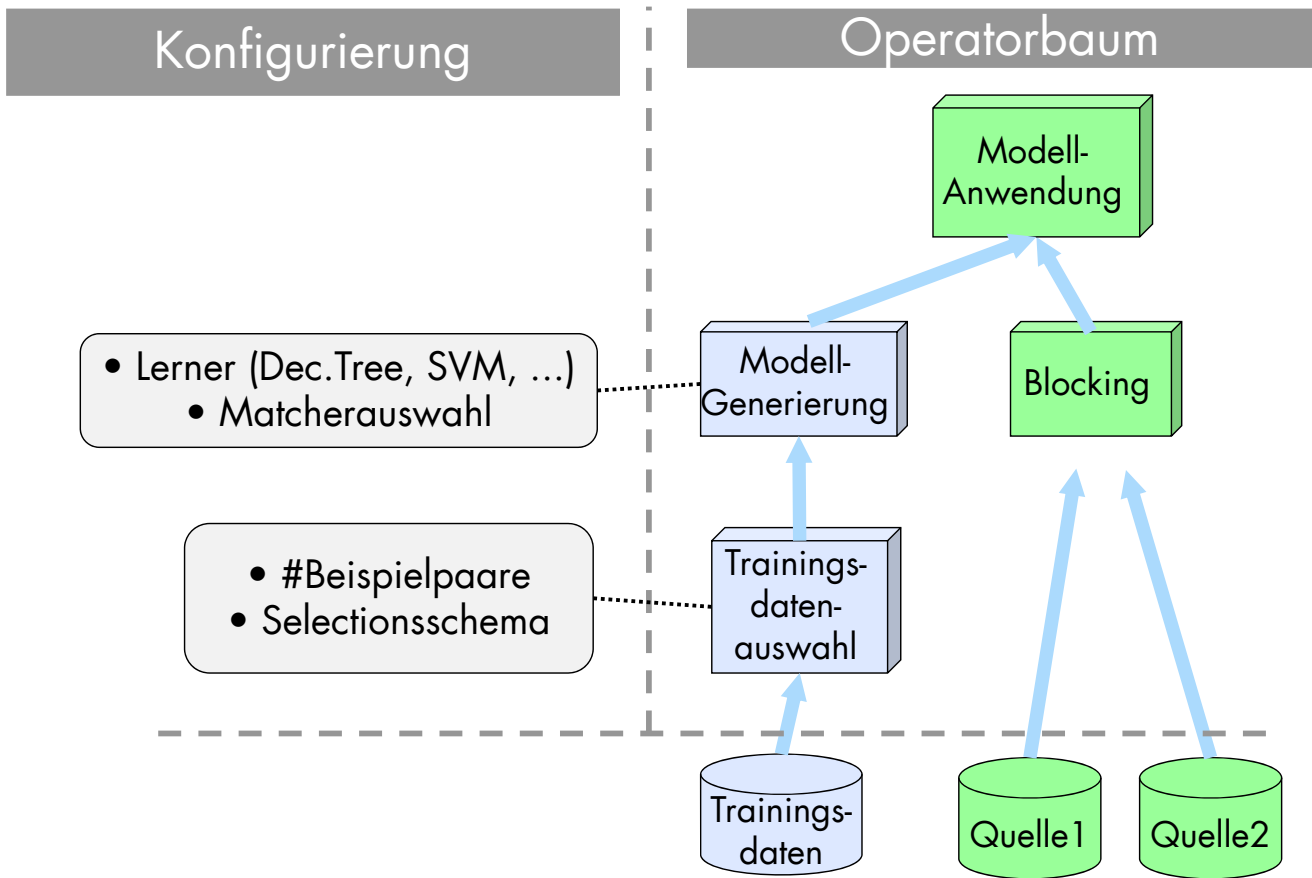


Trainingsbasiertes Objekt-Matching

- Finden effektiver Match-Einstellungen ist schwierig
 - Auswahl der Attribute, Matcher, Einstellungen
- Machine Learning verspricht Verbesserung
 - manuell spezifizierte Trainingsdatenmenge
 - Lernen von Match-Kriterien (z.B. mit Entscheidungsbaum)
 - erfordert gute Trainingsdaten mit vertretbarem manuellem Aufwand

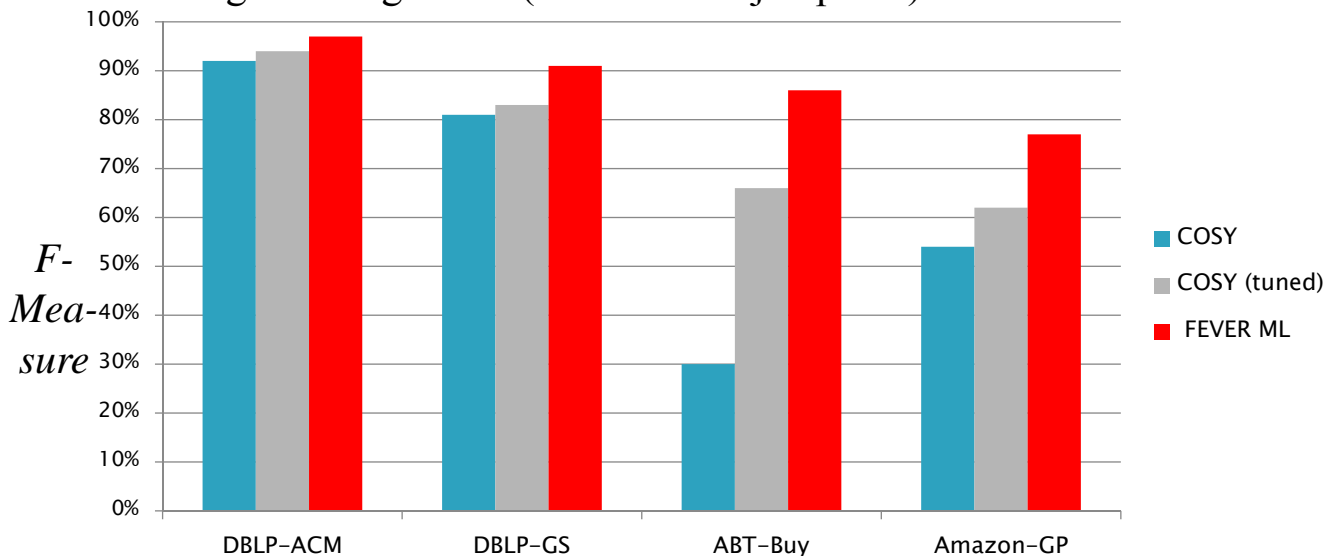


Lern-basierte Match-Strategien in FEVER

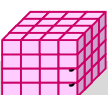


Evaluierungsergebnisse*

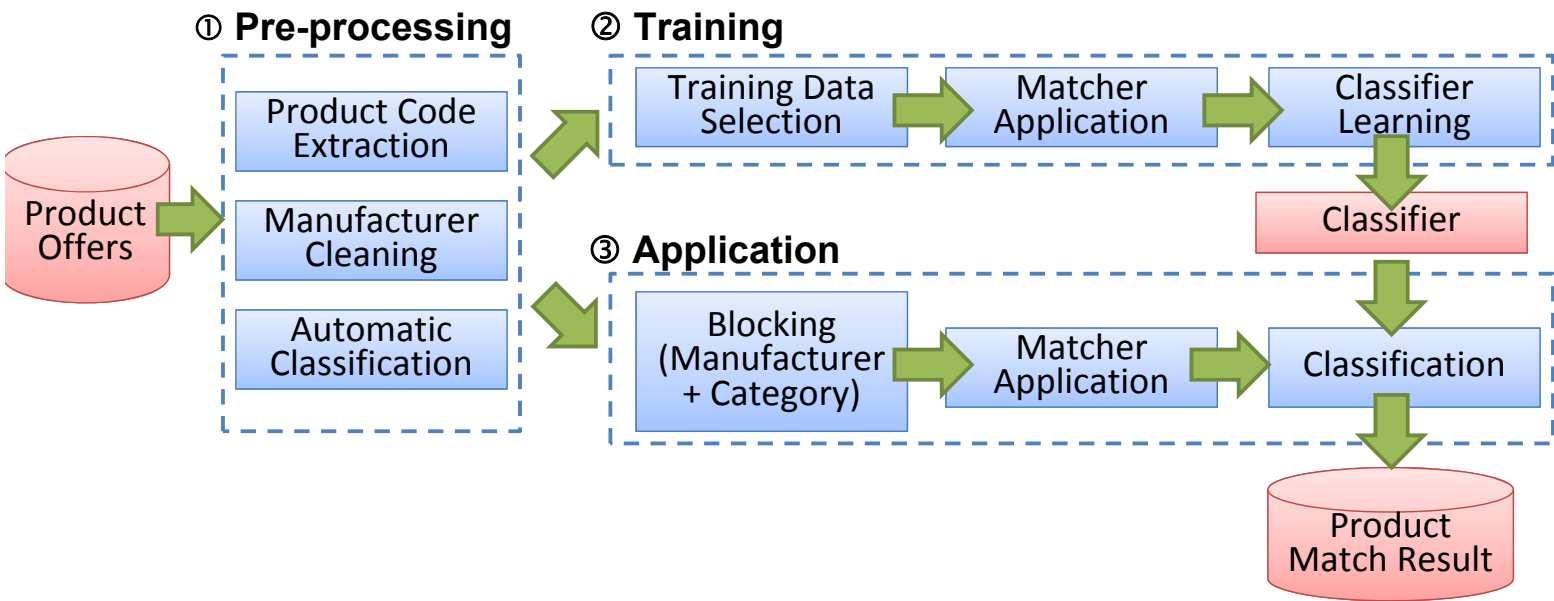
- 4 Matchaufgaben (bibliographisch, E-commerce)
- Matching auf 2 Attributen
- Kommerz. System („COSY“) mit Default-Einstellungen und durch FEVER optimierte Schwellwerte
- ML mit wenig Trainingsdaten (max. 500 Objektpaare)



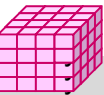
*Köpcke, H.; Thor, A.; Rahm, E.: *Learning-based approaches for matching web data entities*. IEEE Internet Computing, July 2010



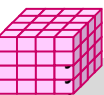
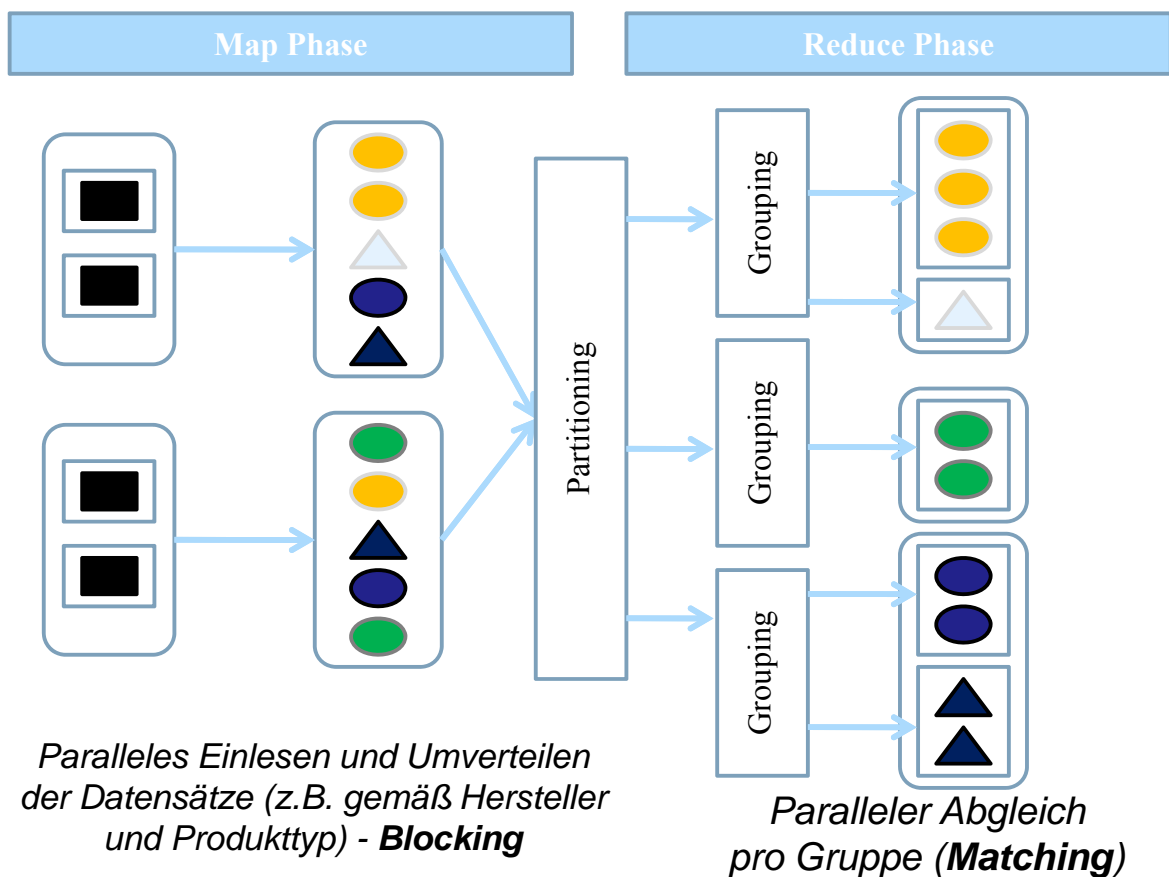
Workflow zum Matching von Produktangeboten aus Web-Shops



Koepcke, Thor, Thomas, Rahm: *Tailoring entity resolution for matching product offers*. Proc. EDBT, 2012



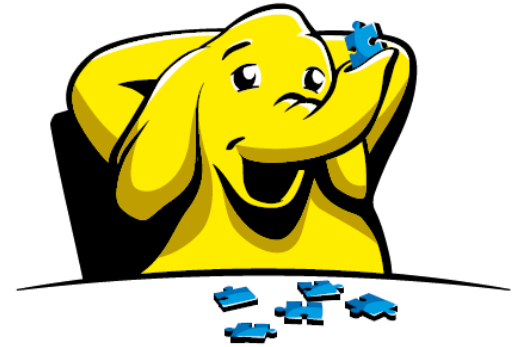
Matching mit MapReduce



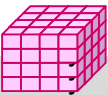
Dedoop: Efficient Deduplication with Hadoop

Parallele Ausführung von Datenintegrations/Match-Workflows mit Hadoop

- Browser-basiertes GUI
- Mächtige Funktionsbibliothek mit
 - vielen Match-Techniken
 - lernbasierte Konfiguration
- Automatische Generieren und Starten von Map/Reduce-Jobs auf unterschiedlichen Clustern
- Automatische Lastbalancierung
- Monitoring der Ausführung

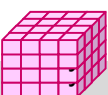


"This tool by far shows the most mature use of MapReduce for data deduplication"
www.hadoopsphere.com



Zusammenfassung (1)

- ETL als komplexer, aufwendiger Integrationsprozess
- Schema- und Datenintegration / Data Cleaning
 - zahlreiche Schema- und Datenkonflikte
 - begrenzte Automatisierbarkeit bezüglich Konflikterkennung und -behandlung
 - möglichst deskriptive Spezifikation aller Datentransformationen zur Behandlung von Schema- und Datenkonflikten
- Fokussierung auf Data Warehouse-spezifisches Zielschema erleichtert Schemaintegration
 - Top-Down-Schemaintegration
 - keine vollständige Integration aller Quell-Schemata erforderlich
- wichtiges Teilproblem: Schema-Matching
 - Nutzung und Kombination mehrerer Matcher, u.a. metadaten- und instanzbasierter Verfahren, linguistisches und strukturelles Matching
 - Reuse früherer Match-Ergebnisse
 - GUI
 - besondere Probleme für sehr große Schemas / Ontologien



Zusammenfassung (2)

- Unterscheidung quell-lokaler und -übergreifender Datenkonflikte
- Data Cleaning zunächst auf einzelnen Quellen
- zentrales Problem: Duplikat-Identifikation und –Behandlung (Objekt Matching)
 - hohe Effizienzanforderungen (-> Nutzung von Blockingverfahren und ggf. Parallelisierung)
 - kombinierte Nutzung mehrerer Match-Verfahren
 - besondere Herausforderungen für Webdaten
- Trainingsbasierte Lernverfahren helfen bei der Konfigurierung
 - ermöglicht automatische Auswahl/Gewichtung von Matchern sowie Ähnlichkeitsschwellwerten
 - manuelle Match-Entscheidung für ausgewählte Objektpaare
 - intelligente Auswahl der Trainingsdaten erforderlich

