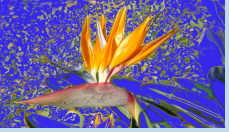
Vorlesung: Bio-Datenbanken

Kapitel 6: Schemaintegration

Dr. Dieter Sosna

30. Juni 2009



Kapitel 6: Schemaintegration

Allgemeines

Metadaten, Ontologien

Schemaintegration

Stabilität

Schemamatching

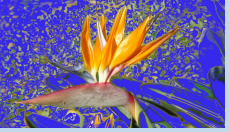
Metadatenbasiert

Instanzdatenbasiert

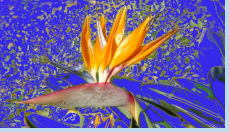
MOMA (reuse)

Qualität

Einige Folien, Graphiken wurden von Herrn A. Thor zur Verfügung gestellt. Danke.

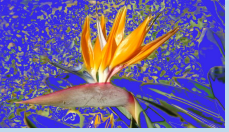


Allgemeines



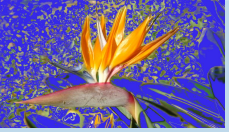
Zwischenstand und Ziel

- Bisher Integration von Instanzen: Gegeben eine Instanz in Quelle A, welche Einträge aus Quelle B gehören *semantisch* dazu?
  Lösung durch Vergleich charakterischer Werte (in der Art der Schlüsselkandidaten).
  Gleichheit - Ähnlichkeit.
  A (bzw. **B**). Komplexe Datentypen: Ähnlichkeit bei weitgehend übereinstimmendem Inhalt.



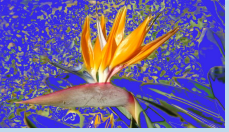
Zwischenstand und Ziel (2)

- Neuer Ansatz: In der Sprache der OO: Vergleich der Konzepte. Welche Konzepte in Datenquelle A entsprechen semantisch welchen Konzepten in Quelle B?
  Fragen:
  Konzepte in Ontologien ?
  Konzepte in rel. DB durch Tabellen, in ... beschrieben.
  Also: Welche Bestandteile (mit welchen Daten) aus DB 1 entsprechen welchem Teil von DB 2?
- Vorgehensweisen bei komplexen Datenstrukturen ?
  top down - bottom up als erster Ansatz;
  Kombinierte und andere Ansätze evt. besser.
  Welche Rolle spielt die Struktur (Relationen)?

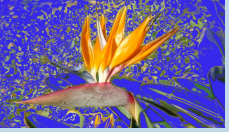


Literatur

Leser, Ulf und Naumann, Felix: Informationsintegration.
Architekturen und Methoden zur Integration verteilter und heterogener Datenquellen.
dpunkt.verlag, Heidelberg, 2007.ISBN (978-) 3-89864-400-6.

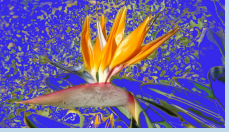


Metadaten, Ontologien



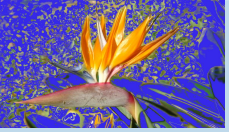
Begriff: Metadaten

- Daten: durch Zustände eines physikalischen Mediums dargestellte Information.
- Metainformationen: Informationen über Informationen:
  Beschreibung der Informationen, meist / zum Teil in einer (externen) Metasprache.
- Bespiele an Hand des E/R-Modells:
  Attribut (kleinste strukturelle Einheit des Modells): Name, Wertevorrat, Kontext, Semantik.
  Dabei:
  Wertevorrat: Wertevorrat im eng. Sinn, Integritätsbed. auf Attributebene.
  Kontext: zu welcher größeren Struktur (E oder R) gehörig.
  Semantik: Erklärung der Bedeutung in der Miniwelt - meist in natürlicher Sprache.
  Entitätsmenge: ...→ freiwille ÜA.



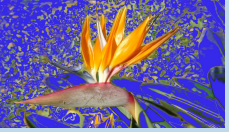
Metadaten (2)

- Metainformationen werden beim Übergang zur Implementierung im DBS (Beschreibung durch SQL) *unvollständig* übernommen.
  Es fehlen:
  Erklärungen der Semantik (gleiche Semantik von Attributen kann in Ausnahmefällen z.B. durch Fremdschlüsseleigenschaft auch im DBS erkannt werden).
  Wertevorrat (natürlicher W. nach Abb. auf vordefinierte Datentypen verloren)
  Kontext der Gesamtdatenbank,
  Charakter von Beziehungen (Teilmengen, Merologien)
- Potentielle Möglichkeiten zur Semantikbeschreibung:
  XML-Schema: Namensräume
  rtf ?
  Nutzung von Ontologien - Ontologiebeschreibungssprachen.



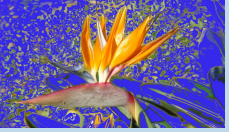
Ontologien

- philosophischer Begriff: το ον - (das) Seiende
- Definition - Informatik:
  - An ontology is an explicit specification of a conceptualization. (Gruber 1993)
    - conceptualization = Modell einer Domäne
    - explicit = ein(ein)deutig
  - An ontology is a **formal**, explicit specification of a **shared** conceptualization. (Breitmann u.a. 2007)
    - formal → maschinenlesbar
    - shared → gemeinsam
- mehrere Ontologien
- Benutzung des Begriffs unscharf.



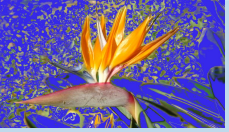
Ontologien in Informatik

- 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.
- Top-Level-Ontologie: Fundamentale Beziehungen (nicht in dieser Vorlesung)
  Domänspezifische O. (Fachterminologie) - Metabeschreibung !.
  Überschneidungen der Gebiete $\rightarrow$ Anpassungen nötig. $\rightarrow$ Ontologiematching



Einteilung der Onologien

- **nach Formalisierungsgrad**
  informal, semi-informal, semi-formal, (streng) formal.
- **nach Allgemeingültigkeit**
  Upper level, domain, task, application.
  Hier Gründe für Ontologiematching:
  Unklare Abrenzung der Begriffe: kontrolliertes Vokabular, Taxonomie, Ontologie
  Überlappende Gebiete - Homonyme, Synonyme, semantische nicht-exakte Gleichheit



Stärke der Begriffe

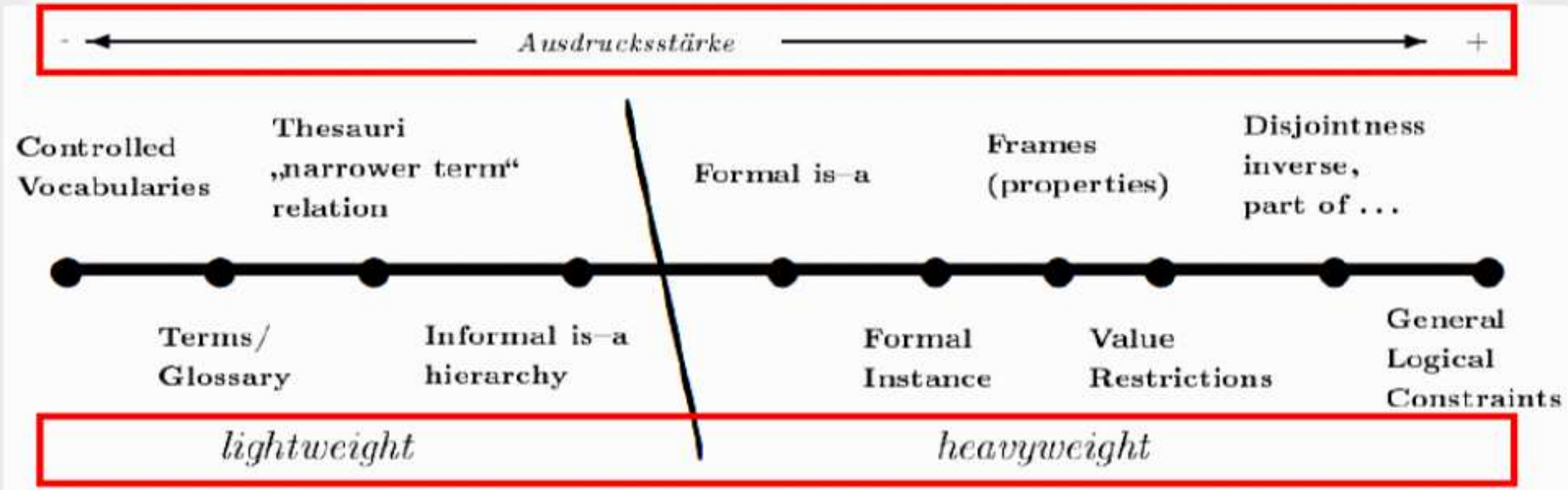
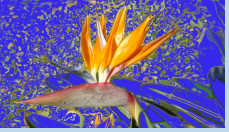


Abbildung in Anlehnung an (Breitmann u. a. 2007, Fig 2.7, Seite 26), Original in (Lassila und McGuinness 2001)

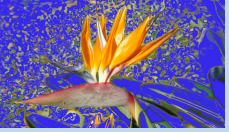


Stärke der Begriffe (2)

lightweight

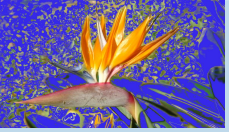
- Kontrolliertes Vokabular = eindeutige Benennungen (Deskriptoren) für jeden Begriff.
- Taxonomoie: Klassifikation, monohierarchisch (z.B. Biologie)
- Thesaurus (zur Dokumentation): kontrolliertes Vokabular: Unterschiedliche Schreibweisen (Photo/Foto), Synonyme bzw. als gleichbedeutend behandelte Quasi-Synonyme, Abkürzungen, Übersetzungen etc.: durch Äquivalenzrelationen miteinander in Beziehung gesetzt.
  Begriffe durch Assoziationsrelationen und hierarchische Relationen vernetzt.
- Ontologien im engeren Sinn - formale Definitionen

heavyweight



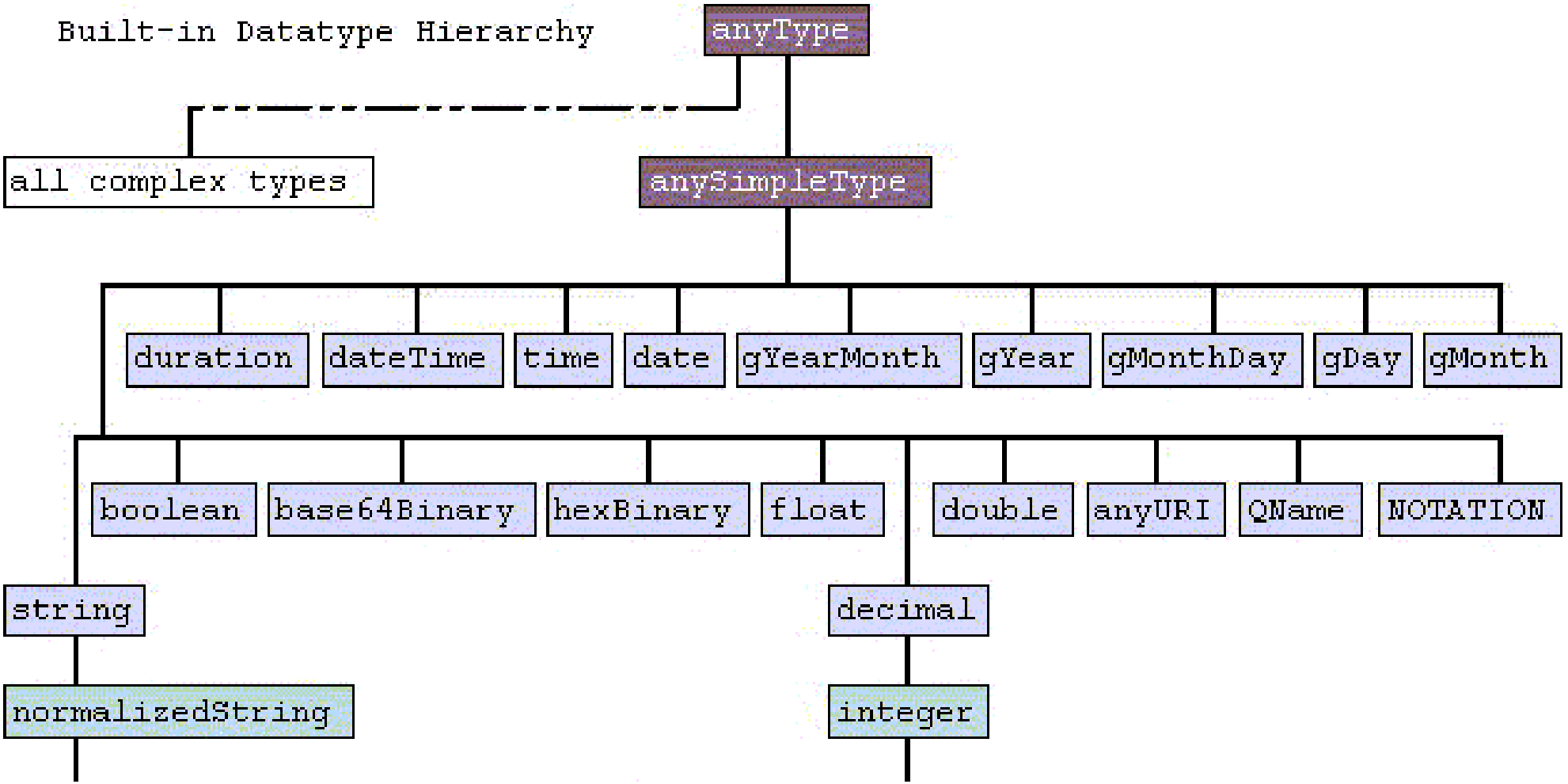
Nutzen der Ontologien

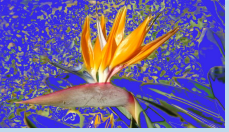
- Einheitliche Begriffswelt (kontrolliertes Vokabular):
  Typ.isches Beispiel: Chemische Nomenklatur:
  Katzengold, Narrengold, fool's gold - Pyrith - Eisen(II)disulfid - FeS_2.
  Übername des Vok. in Daten sichert Vergleichbarkeit von Daten,
  Hilfe bei Überwindung von Heterogenität (z.B. Synonyme
  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 (s.o.)



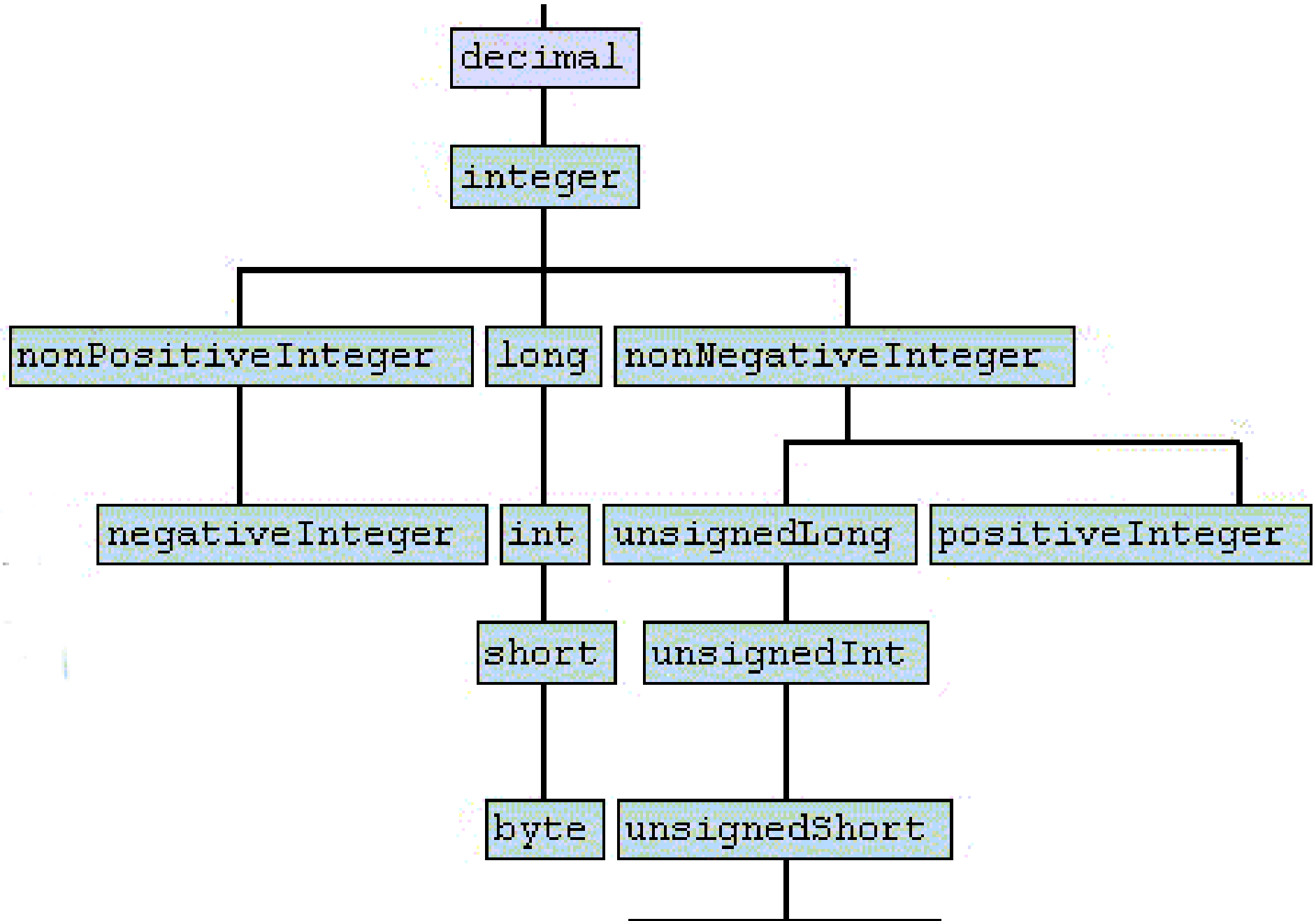
Beispiel - XML-Datentypen

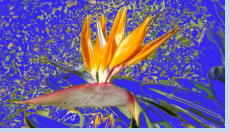
http://www.w3.org/TR/xmlschema-2/





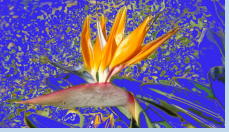
Beispiel - XML-Datentypen (Forts.)





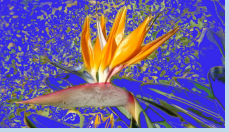
Beispiel - XML-Datentypen (2)

- Lesbare Beschreibung (nach o.g. Quelle)
  3.3.13 integer
  Definition: integer is *derived* from decimal by fixing the value of *fractionDigits* to be 0 and disallowing the trailing decimal point. This results in the standard mathematical concept of the integer numbers. The *value space* of integer is the infinite set ...,-2,-1,0,1,2,.... The *base type* of integer is decimal.
  3.3.13.1 Lexical representation
  integer has a lexical representation consisting of a finite-length sequence of decimal digits ($30_{hex} - 39_{hex}$) with an optional leading sign . If the sign is omitted, "„+“ is assumed. For example: -1, 0, 12678967543233, +100000.
- Lesbare Beschreibung ungenau (z.B. keine vorlaufenden Nullen, welche Vorzeichen $\rightarrow$ exakt + maschinenlesbar: BNF (ÜA).

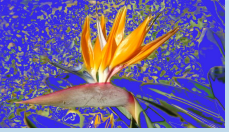


Lit: Leser, Naumann, a.a.O.

- 3 Schritte:
  Erstellung der globalen Ontologie also auch Integration von Ontologien
  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 korrekzte Objekte.
- Prakt. Anwendung /Realisierung bei Bio-DB: ?



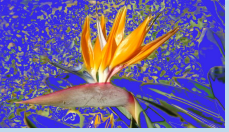
Schemaintegration



Schemaintegration - Definitionen

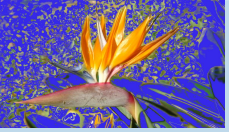
Muster für Integration komplexer Strukturen

- Vorgelegt zwei Schemata **A,B**.
  Gesucht ein Schema **C** und Transformationen t_{AC}, t_{BC} (Schemakonstruktion), d_{AC}, d_{BC} (Datentransformationen) mit folgenden Eigenschaften:
  - **Korrektheit**
  - **Vollständigkeit**
  - **Minimalität** (Erklärung nächste Folie)
- Definition: Schemaintegration
  bezeichnet den Prozeß des Findens von **C** und $t_{AC}, t_{BC}, d_{AC}, d_{BC}$ und
  (wir zählen zur Schemaintegration auch)
  die Anwendung der Transformationen d_{AC}, d_{BC}, mit denen die Daten aus **A,B** in **C** überführt werden.



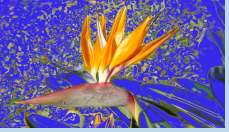
Erklärungen

- **Vollständigkeit:** Das Anwendungsgebiet von **C** umfasst die Anwendungsgebiete der Schemata **A** und **B**.
  Alle Beziehungen zwischen Konzepten in **A** bzw. **B**, können in **C** verlustfrei dargestellt werden. Alle Daten aus **A** bzw. **B** können in **C** dargestellt werden.
- **Korrektheit:** Zu jedem Konzept des Schema **A** (bzw. **B**) existiert ein semantisch gleiches Konzept in **C**.
  Die Beziehungen zwischen Konzepten in **C**, eingeschränkt auf die Daten, die durch Tranformation aus **A** (bzw. **B**) entstanden sind, ist semantisch gleich der Beziehung zwischen den Konzepten in **A** (bzw. **B**).
  Analoge Aussage für Merkmale (Eigenschaften, Attribute)
- **Minimalität:** Aus **C** kann kein Konzept entfernt werden ohne dass die Vollständigkeit verletzt wird. (Das bedeutet insbesondere, dass in **A** und **B** semantisch gleiche Konzepte in **C** nur einmal auftreten.)



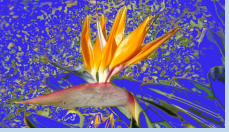
Komponenten der Schemaintegration

- Schemamatching (Finden semantisch gleicher Konzepte in **A** und **B** bzw. in **A** und **C** bzw. in **B** und **C** und der Abbildungen t_{AC}, t_{BC}.
- Schemamapping Finden der Abbildungen d_{AC}, d_{BC}.
- Datentransformation: Anwenden der Abbildungen d_{AC}, d_{BC}.



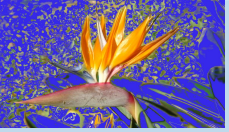
Bemerkungen

- Bisher keine Aussage, wie **C** gefunden wird.
- Aus Vollständigkeit abgeleitet: **Mächtigkeit des Zielmodells** umfasst Mächtigkeit von **A** rsp. **B**
  → Mächtigkeit der Beschreibungssprachen ?
  operationale Vollständigkeit — XML-Schema ausreichend ?



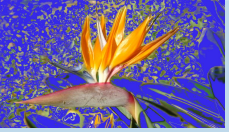
Bemerkung zur Problemgröße

- Ziel: Integration großer Schemata (10^3 Konzepte) erfordert berechnete Integration;
- Widerspruch zwischen Automatisierung und Qualitätsanforderung.
  Erkennung der Semantik aus syntaktischen und strukturellen Merkmalen nur partiell automatisierbar.
  deshalb notwendige Voraussetzung: formale Definitionen und Beschreibungen aller Konzepte (Ontologien) oder
  (wie heute üblich:) Mensch muß die Ergebnisse nachvollziehen, bewerten und korrigieren können.
  Dokumentation der Entsprechungen (welche, wie gewonnen, ...), der Transformationen, ...
  deshalb
  Integrationsalgorithmen müssen lernend sein bzw. korrigierbar sein,
  die Ergebnisse von Auswertungen in weitere Integrationsschritte einbeziehen.
  (re-use von von als *gut* bewerteten Ergebnissen



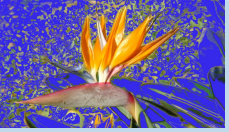
Bemerkungen (3)

- Integration ist gerichtet,
  d.h. die Abbildungen müssen nicht (per se) invertierbar sein,
  Beispiel: OUTER-JOIN-Invertierung:
  $A(a,b) = \{(null, 1)\}, F(c,d) = \{(1,2),(4,5)\}$
  $(A \text{ OUTER-JOIN}_{b=c} F)(a,b,d) = \{(null,1,2),(null,4,5)\}$
  Berechne aus $A \text{ OUTER-JOIN}_{b=c} F$ wieder A !
  Konstruktion der Umkehrabbildung ist neue Aufgabe (Unterschied zu Schemaevolution)



Integrationsschritte

- **Integrationsvorbereitung:** Auswahl der Schemata, bzw. von Teilen davon, Festlegung von Reihenfolgen, anzuwendende Verfahren.
  IV wesentlich für Erfolg, da durch Arbeit des Menschen Semantik eingebracht wird.
- **Schemavergleich:** Ermittlung von Korrespondenzen: semantisch gleiche Elemente, Teilmengenbeziehungen,
  Erkennung von Heterogenitäten zwischen den Schemata: Namenskonflikte der Konzepte (Synonyme, Homonyme), Strukturelle Konflikte (Schlüsselalternativen, Normalformenunterschiede bei rel DB, Position in Ontologie (Buch (Autor (Name,Vorname), Titel, ...) vs. Autor(Name, Vorname, {Buch})).
- **Schemakonstruktion:** Ableitung des neuen Schema durch Vergleich der Korrespondenzen mit den alten Schemata.
  Konstruktion der Abbildungen von den Konzepten jedes alten Schema in des neue Schema.
  Konstruktion von Datenabbildungen (SQL-Befehle oder äquivalent).

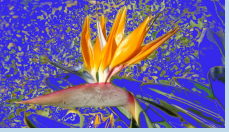


Korrespondenzbasierte Integration

Lit.: Conrad: Föd. DBS. Springer 1997.
Regeln zur Übernahme in das integrierte Schema

- Überhahme: Kategorie ohne Korrespondenz (mit Daten)
- Korespondierende Kategorien: Übernahme, Daten mit OUTER-JOIN.
- Gleiche (in beiden Ausgangsschemata) direkte Beziehungen übernehmen, Daten-JOIN.
- Beziehungen ohne Korrespondenz: Übernahme

Problem: Kategorien meist nicht identisch, sondern überlappend $\rightarrow$ Zersplitterung.

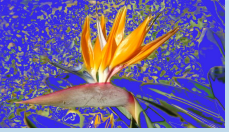


GIM-Modell

Schmitt, Ingo: Schemaintegration für d. Entwurf Föd.DB. Diss. 1998
GIM (Gener. Integrationsmodell) - Matrix und Schemaableitung

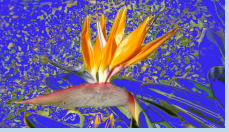
- Spalten: Minimale Zerlegung aller Objekte (aus Ausgangsschemata) in disjunkte Klassen: $\mathcal{A} \setminus \mathcal{B}, \mathcal{A} \cap \mathcal{B}, \mathcal{B} \setminus \mathcal{A}$
- Zeilen: Attribute - homogenisiert
- Felder: Wahrheitswerte: **w** = Attribut ist für Kategorie relevant.

Schemaableitung:
Umordnung der Matrix, so das große recheckige Bereichen mit **w**-Werten entstehen (dabei sind Überlappungen zugelassen.
Breite Rechtecke = Oberklassen, hohe Schmale = Unterklassen
Kritik:
1)Semantische Hauptarbeit: Homogenisierung der Attribute, außerhalb des Modells.
2)Klassenbildung formal, Semantik der Klassen ? K. modelliert evt. keine realen Objekte.

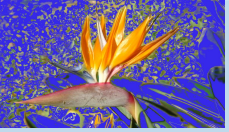


Datentransformation

- Zeitpunkt der DT: materialisierte Integration - sofort
  virtuelle Integration - Konstruktion eines Wrappers.
- Abbildungstypen im Mapping : 1:1, 1:N,N:1, N:M
- Leicht lösbar (auf Grund des Matching weitgehend automatisierbar):
  Wertkorrespondenzen vom Typ 1:1, N:1, 1:N bei einfachen Attributen (rel.DB, Simple-Type bei XML): 1:1-Transformationen (Umrechnungen, ...), funktionale Zusammenfassungen (Konkanation, ...), Extraktion.
- Schwierig (Automatisierung noch im Forschungbereich):
  M:N-Wertkorrespondenzen:
  Buch - Autor (vereinfacht)
  Buch({Autor(Name, Vorname)}, ISBN, Titel) vs.
  Person(Name, Vorname, geschrieb-Buch({(ISBN, Titel)})
  Korrespondenzen ?
- Schwierig: Korrespondenzen über mehrere Konzepte /Konzeptstufen
  Schemaheterogenität

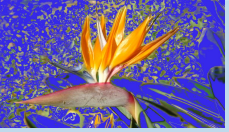


Stabilität



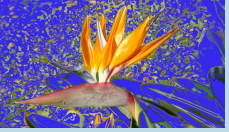
Stabilität

- Motivation: Vielfach (insbes. Schemamatching) stabile (d.h. zeitlich unveränderliche) Zuordnungen gesucht (sicheres Wissen (oder derzeit nicht beforscht)).
- Math. Stabilitätstheorie: z.B. Pendel, Kugel in Schale, Lösung einer Gleichung: Math. Theorie muss adaptiert werden.
- Gesucht: Maß für „in letzter Zeit wenig Änderung“.
- Voraussetzung:
  1) Änderungen eines Objektes o nur zu diskreten (und unregelmäßigen) Zeitpunkten t_i, $i = ..., -3, -2, -1, 0$.
  Dabei ersetzt zur Zeit t_i die i-te Version von o, also o_i den Vorgänger o_{i-1}.
  Genauer: o_{i-1} ist gültig im Zeitintervall $[t_{i-1}, t_i)$
  2) Ähnlichkeitsmaß $s(o_1, O_2)$für Objekte $o \in \mathcal{O}$:
  $\mathcal{O} \times \mathcal{O} \rightarrow [1, 1]$.mit
  $s(x, x) = 1$ für alle $x \in \mathcal{O}$
  $s(x, y) = 1$ falls x oder y nicht existiert (konstante Fortsetzung von s von dem Entstehungszeitpunkt von o in die Vergangenheit).



Variation einer Funktion

- Sei $f : [a, b] \to \mathcal{R}$ eine reellwertige Funktion auf dem (reellen) Intervall $[a, b]$.
  $VAR(f)_{[a,b]} :=$
  $\sup \left\{ \sum_{k=0}^{n-1} \left| f\left(t_{k+1}^{(n)}\right) - f\left(t_k^{(n)}\right) \right| : n \in \mathcal{N}, a \leq t_0^{(n)} < t_1^{(n)} \cdots < t_n^{(n)} \leq b \right\}$
  Das Supremum wird über alle Zerlegungen von $[a, b]$ genommen.
- Satz: Ist $f : [a, b] \to \mathcal{R}$ in den Intervallen $a = [t_0, t_1[, \ , [t_1.t_2[, \ldots [t_{n-1}, t_n = b]$ jeweils monoton, so gilt: $VAR(f)_{[a,b]} = \sum_{k=0}^{n-1} |f(t_{k+1}) - f(t_k)|$.
  Interpretation: $VAR(f)_{[a,b]}$ summiert alle Änderungen auf.
- $$f(t) = \begin{cases} 0 & : \quad t = 0, \\ t \cos \frac{\pi}{2t} & : \quad t \in (0, 1], \end{cases}$$
  $f(t)$ in $[0, 1]$ stetig und $VAR(f)_{[a,b]} = \infty$.
  (Hinweis: Stützstellen 0 und $t_k^{(n)} = 1/(n + 1 - k)$.)



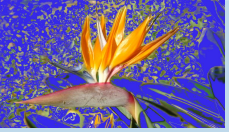
Stabilität - Variation

- Sei o ein Objekt in mehreren Versionen o_i mit Änderungszeitpunkten t_i,
  $s[o](t) = \begin{cases} s(o_i, o_{i-1}) & : \quad t \in [t_i, t_{i+1}, \\ s(o_{-1}, o_0 & : \quad t > t_0, \text{ zu } t_0 \text{ bisher letzte Änderung} \end{cases}$
  Ähnlichkeit zur Vorgängerversion (zu einem belibigen Zeitpunkt -
  ist eine Sprungfunktion: (rechts offene Intervalle konstanter Werte),
  also in $[t_{i-1}, t_i]$ monoton.
- Definition: Stabilität von o : $v(o) := VAR(s[o])_{[-\infty,t]} =$
  $= \sum_{i=-\infty}^{0} |s[o](t_i) - s[o](t_{i-1})|$
  $= \sum_{i=-\infty}^{0} |s(o_i, o_{i-1}) - s(o_{i-1}, o_{i-2})|$

  Summe der Beträge aller Änderungen der Ähnlichkeit zum direkten Vorgänger.
- eine Stabilität „vergißt nicht“.

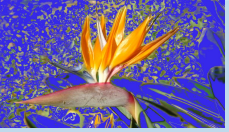
  - Mögliche Lösungen: Abschneiden , Gewichtungen - Dreieck, Exponentialfunktion, ...

  - Einschwingvorgänge in Physik: Exponentialfunktion.
  $v(o) := \sum_{i=-\infty}^{0} e^{(t_i/\bar{t})} \times |s[o](t_i) - s[o](t_{i-1})|$
  Wie weit in Vergangenheit zurückgehen ($\bar{t}$)?



Anwendung Poisson-Verteilung

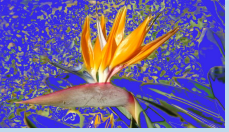
- Kein Beweis - nur Motivation.
- $P_k(\lambda) = \frac{\lambda^k}{k!} e^{-\lambda}$, λ reell
  Wahrscheinlichkeit, dass seltene, zufällige und unabhängige Ereignisse k Mal eintritt, wenn man einen Mittelwert λ des Eintretens kennt.
  Eigenschaft: Mittelwert = Varianz = λ
- Anschauliche Interpretation:
  Ein Feld bestehe aus n Teilfeldern, l Reiskörner werden zufällig auf diese n Teilfelder verteilt. Wie groß ist die Wahrscheinlichkeit, dass ein Teilfeld genau k Körner enthält?
  (Im Mittel kommen also $l/n = \lambda$ Körner auf ein Feld.)
- $\sum_{k=0}^{l} P_k(\lambda)$ ist die Wahrscheinlichkeit, dass das Ereignis Null bis maximal l-mal auftritt.



Stabilität - 3-Sigma-Regel

Wann ist das Auftreten eines Wertes möglicherweise nicht mehr zufällig?

- Drei-Sigma-Regel der Normalverteilung: (σ ... Varianz): Bei einer normalverteilten Zufallsgröße ist die Wahrscheinlichkeit, dass sie im Intervall $[Mittelwert - 3 \times \sigma, Mittelwert + 3 \times \sigma]$ liegt,(ca.) 0,9997.
  Anwendung: Messwerte außerhalb dieses Intervalls sind evt. Ausreißer.
  Achtung: Formale (automatisierte) Anwendung kann zu Fehlern führen!
- Anwendung des Wertes 0,9997 auf das Problem der seltenen Ereignisse: Wie viele Ereignisse auf einem Feld gelten noch als zufällig?

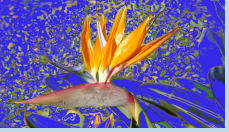


Stabilität - Zahlenbeispiele

- Ontologie: ca. 15000 Konzepte, 15 Änderungen/Ausgabe
  Limit=.9997 (10stellig gerechnet, Werte gerundet)
- Beispiele

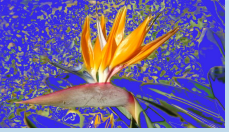
N	15	30	150	375	750	1500	3000
Vers. v	(1)	2	10	25	50	100	200
λ	0,001	0,002	0,01	0,025	0,05	0,1	0,2
P_0	0,9990	,9980	,9901	,9753	,9512	,9048	,81873
P_1	0,0009	,0019	,00990	,0244	,04756	,09048	,16375
$\sum_0^1$	0,9999	,9999	,99995	,9997	,99879	,99532	,98248
P_2	-	-	-	-	,00119	,00452	,01638
$\sum_0^2$	-	-	-	-	,99998	,99984	,99885
P_3	-	-	-	-	-	-	,00109
$\sum_0^3$	-	-	-	-	-	-	,99994

- 25 Versionen ($= i_0$, ca. 2 Jahre): instabil, wenn mehr als 1 Änderungen.



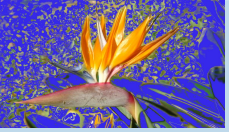
Math. Stabilität

- Idee:Vorgabe einer maximalen Abweichung, welche Objekte zeigen in den letzten Versionen weniger als die Vorgabe Abweichung von der letzen Version.
- Definition Stabilität
  Vorgelegt sei ein Objekt o mit Versionen: $o = \{o_i, i = -\infty, ..., 0\}$ (o_0 ist die letzte, aktuelle Version und eine reelle Zahl $\varepsilon > 0$.
  o heißt *stabil*, wenn gilt $1 - s(o_i, o_0) \leq \varepsilon$ für $i \in (-\infty, 0)$.
  o heißt *seit Version* $\hat{i}$ *stabil*, wenn $1 - s(o_i, o_0) \leq \varepsilon$ für $i \in [\hat{i}, 0)$.
- Asymptotisch stabil: o heißt *asymptotisch stabil*, wenn gilt $e^{-c \times |i|} \times (1 - s(o_i, o_0)) \leq \varepsilon$ für $i \in (-\infty, 0)$.
  Ältere Abweichungen verlieren an Einfluss.
  Wie ist $c > 0$ zu wählen?

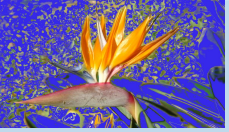


Asymptodische Stabilität- Wahl des c

- Seltene Ereignisse: Zeit X bis zum Eintreten des n^{ten} Ereignisse ist Erlang-verteilt.Eine Zeiteinheit - 25 Versionen - virtuelles $\lambda_v = 0.0375 = 25 \times 15/15000$
- Wahrscheinlichkeit, dass $X < x$: (bei Erlang-Verteilung) $P(X < x) = 1 - e^{-\lambda x} \sum_{i=0}^{n-1} (\lambda x)^i/(i!)$
- Bei $\lambda = \lambda_v$ und $x = 1$, $n = 2$ (eine virtuelle Zeiteinheit: $P(X < x) = 1 - 0.9997$ (Wahrscheinlichkeit, dass in Zeit $x < 1$ eine 2. Änderung kommt.
  Motivation für Wahl der Dämpfung: Was eine virtuelle Zeiteinheit zurückliegt, wird um Faktor $1/e$ gedämpft:
  $e^{i/\bar{i}} \times (1 - s(o_i, o_0)) \leq \varepsilon$ für $i \in (-\infty, 0)$.
- Viele Änderungen pro Ausgabe $\rightarrow \bar{i}$ wird klein.



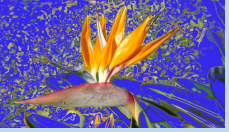
Schemamatching



Matching komplexer Strukturen

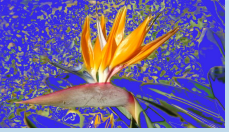
Matching als Vorbedingung zu Integration

- Ähnlichkeit der Probleme bei
  - Ontologiematching,
  - Schemamatching,
  - allg.: Matching komplexer Objekte
- In der Sprache der Objektorientierung:
  (Komplexe) Objektklassen und Beziehungen zwischen diesen.
- Unterschiede:
  Art der Objekte, Darstellung der Komplexität, Art (Typ) der Beziehungen.
- Probleme: kaum Identität; Charakterisierung von Gleichheit / Ähnlichkeit komplexer Strukturen?
  Verschiedene Maße - Vergleichbarkeit?



Schemamatching

- Voraussetzung: Gegeben 2 Quellen mit zugehörigen Metadaten und Instanzdaten Ziel: Finden von semantisch gleichen Konzepten.
- im Bereich der Bio-DB als Besonderheit: vielfach Quellen auf Instanzniveau verbunden /vernetzt.
- 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.

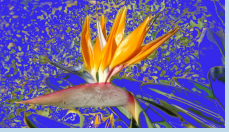


Mißerfolg erwartet

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

auf semantische Ähnlichkeit

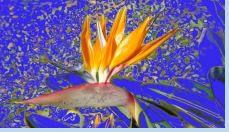
geschlossen werden.
Mit anderen Worten:
Schemamatching ist Forschungsgegenstand. Das Ziel der automatisierten Verfahren ist noch nicht erreicht.
Für jedes Verfahren lassen sich Negativbeispiele finden
$\rightarrow$ Kombination von Verfahren könnte Resultate verbessern. Problem: Wie kombinieren ?



Idee

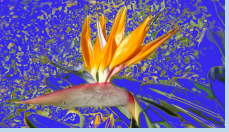
Vergleich zweier Objekte $a = (a_1, \ldots, a_m) \in A$ und $b = (b_1, \ldots, b_n) \in B$

1. Bestimmung der Ähnlichkeitswerte
   - Ähnlichkeitsfunktionen $s(a, b)$ zum Vergleich
   - Verschiedene Funktionen, verschiedene (Attribut-)Vergleiche 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."



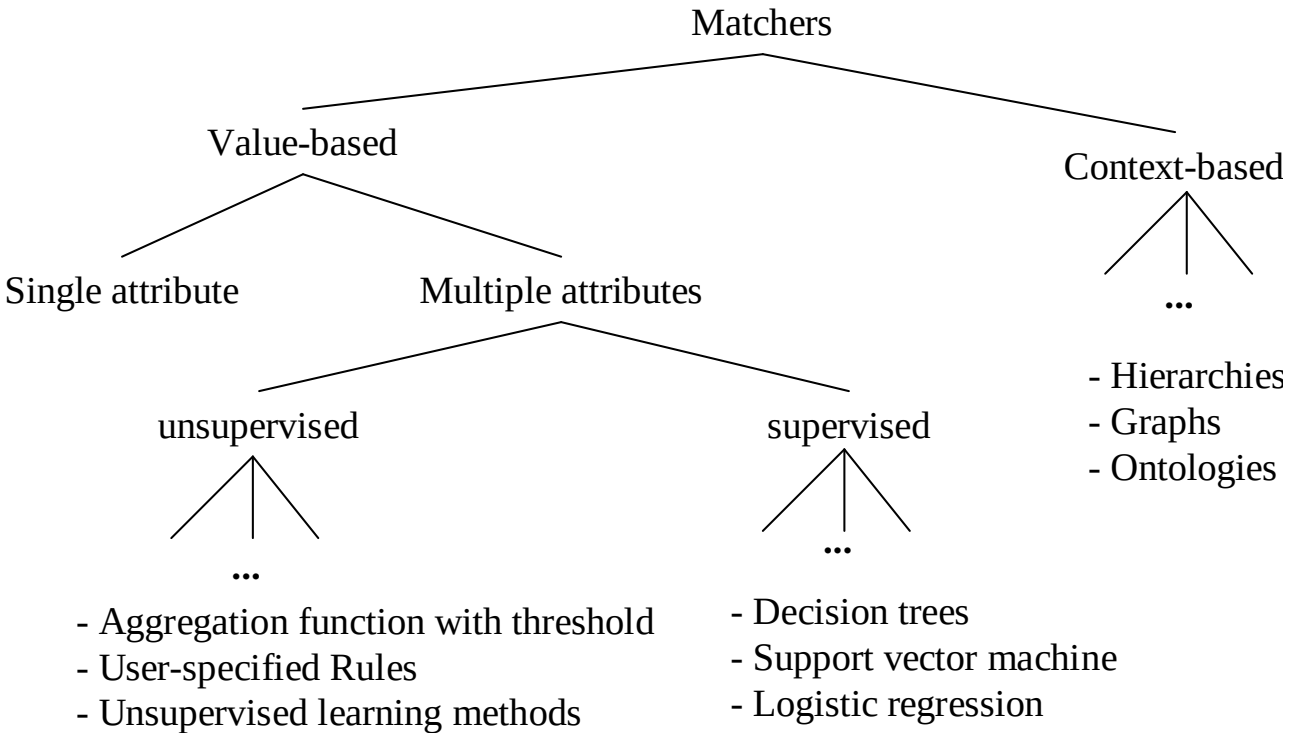
Probleme

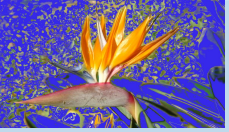
- Sind bottom-up ind
- Finden der /einer **guten Ähnlichkeitsfunktion**
  - toleriert Fehler
  - trennt unterschiedliche Objekte mit hoher Sicherheit
- Regeln und deren (math.) Begründung für Kombination



Ansätze für Object-Matching

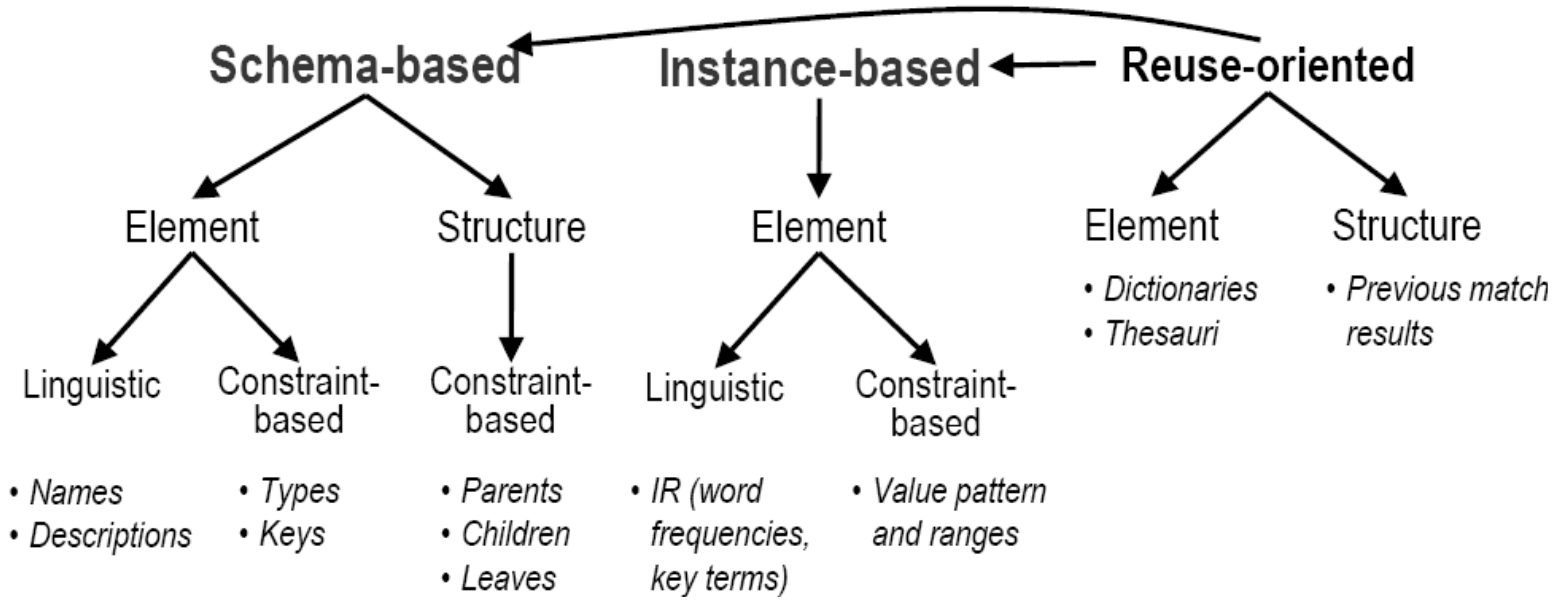
Viele verschiedene automatische Ansätze, die auch kombiniert werden können





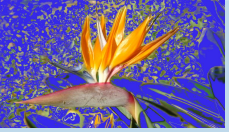
Ansätze für Schema-Matching

Viele verschiedene automatische Ansätze, die auch kombiniert werden können

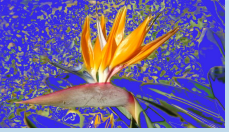


Publikationen:

- 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



Metadatenbasiert

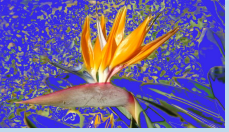


Metadaten: Indikatoren /Berechnungen für Gleichheit

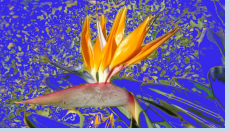
- Namensvergleiche:
  Gleichheit (! Homonyme, bei XML Lösung durch Namensräume)
  Gleichheit nach Normalisierung (Großschreibung, stemming, Übersetzung)
  Hyperonymie (hierarch. Beziehung is-a , Thessausus, Ontologie,Taxonomie)
  Ähnlichkeit
- Strukturvergleiche:
  Cupid (1): Schemata $\rightarrow$ Bäume . 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.



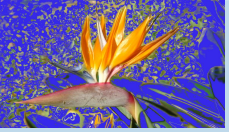
Instanzdatenbasiert



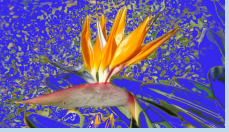
Instanzbasiertes Maching

setzt sets 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.



MOMA (reuse)



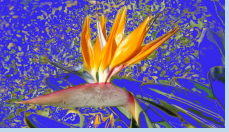
Mapping-Verarbeitung

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

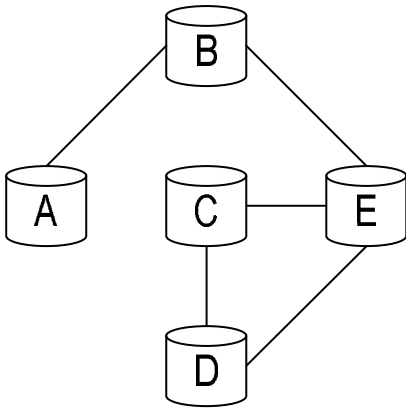
Ziele

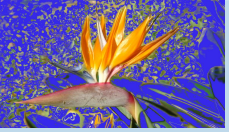
- Wiederverwendung von Match-Ergebnissen zur effizienten Berechnung neuer Match-Ergebnisse
- **Kombination** von Match-Ergebnissen zur Qualitätsverbesserung
- Bestimmung von Match-Ergebnissen, wenn kein geeignetes bf diektes Ähnlichkeitsmaß zur Verfügung steht



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 direktes Ähnlichkeitsmaß: (A,D) mittels (A,B) + (B,E) + (E,D)

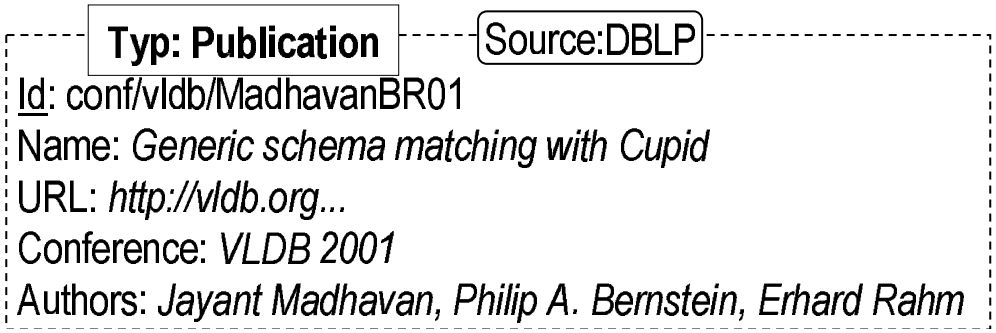


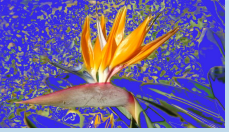


MOMA-Ansatz: Begriffe (1)

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, ...

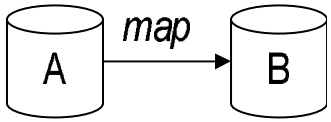




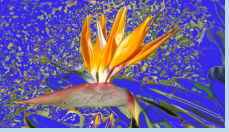
MOMA-Ansatz: Begriffe (2)

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

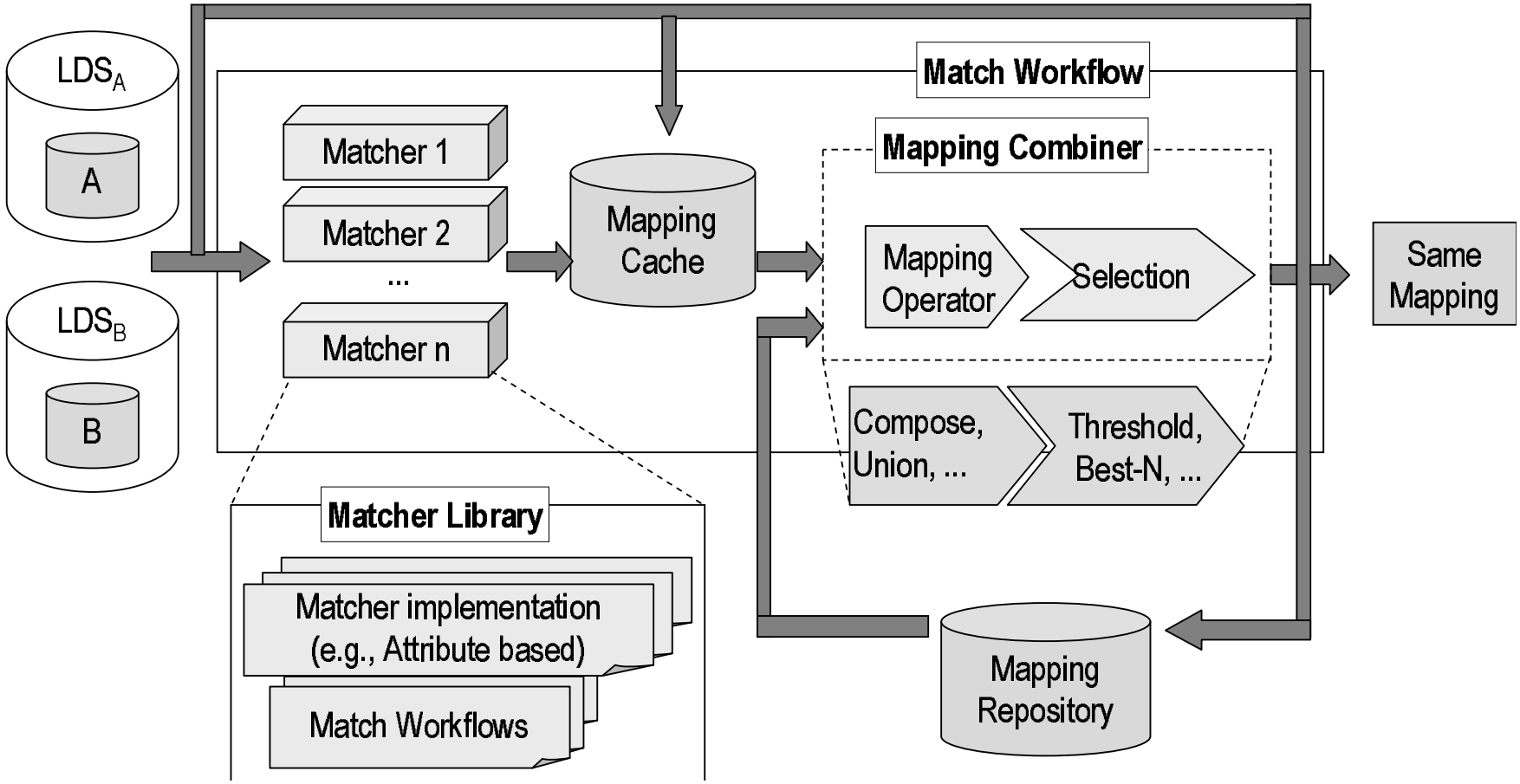


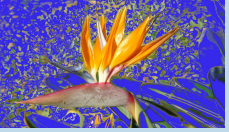
Mapping-Verarbeitung: MOMA-Ansatz

- 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

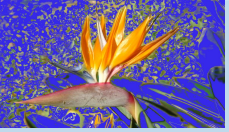
Thor, A., Rahm, E.: MOMA - A Mapping-based Object Matching System. CIDR, 2007





Operatoren: Übersicht (vereinfacht)

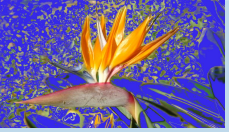
- Attributvergleich: $match(O_1, O_2, f) = map$
  - $\{(a, b, s) | a \in O_1, b \in O_2, s = f(a, b)\}$
  - f ist eine Match-Funktion, die für zwei Objekte den Ähnlichkeitswert s ermittelt.
- Vereingung: $union(map_1, map_2) = map$
  - $\{(a, b, s) | (a, b, s_1) \in map_1 \vee (a, b, s_2) \in map_2\}$
- 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



Wünsche an Ähnlichkeitsmaß

Problem: Welche Eigenschaften gelten allgemein?
kein „Standard“!

- Wertebereich von f: 0-1 (0 nicht ähnlich, 1 mit sehr hoher Sicherheit gleich)
  umgekehrt: Gleichheit $\rightarrow s = 1$
- Hat Ähnlichkeitsmaß die Semantik (oder Eigenschaften) einer Wahrscheinlichkeit (für die semantische Gleichheit der Begriffe) ?
  Mögliche Folgerung: z.B. für den Durchschnittsoperator
  (a, b, f_1) (a, b, f_2)
  Annahme: f_1 und f_2 sind unabhängig - $s_\cap = 1 - (1 - s_1)(1 - s_2)$ Gewichte, Bedingungen?

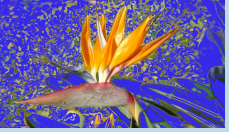


Wünsche (2)

- Beispiele:

$s_1 = s_2$	s	
0,8	0,96	
0,5	0,75	
0,1	0,19	???

  Mehrere unabhängige gute Bewertungen verbessern die Gesamtbewertung - Was machen mehrere Schlechte ?
  $s_\cap = 1 - \sqrt{(1-s_1)(1-s_2)}$ nicht theoretisch begründet.
- Komposition: $(a, x, f_1)\ (x, b, f_2)$? $s = s_1 \times s_2$
  liefert nach subjektivem Gefühl zu schlechte Werte für s.
- Modell ?



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

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

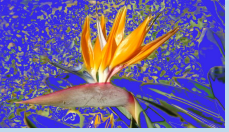
union (Min)		
A	B	s
a1	b1	0.6
a2	b2	0.8
a3	b3	0.6

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

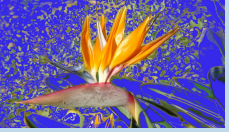


Kombination: Vereinigung / Durchschnitt (2)

- 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%

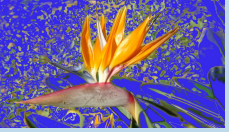
- Fazit
  - Kombination kann Match-Qualität steigern
  - Vereinigung verbessert Recall (evtl. auf Kosten der Precision)
  - Durchschnitt verbessert Precision (evtl. auf Kosten des Recalls)
  - Wahl der Ähnlichkeitsfunktion von Match-Problem abhängig



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 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
    - $Dice = 2 \cdot \frac{s(a,b)}{n(a)+n(b)}$
    - $DiceLeft = \frac{s(a,b)}{n(a)}$, $DiceRight = \frac{s(a,b)}{n(b)}$
    - $DiceMin = \frac{s(a,b)}{min(n(a),n(b))}$

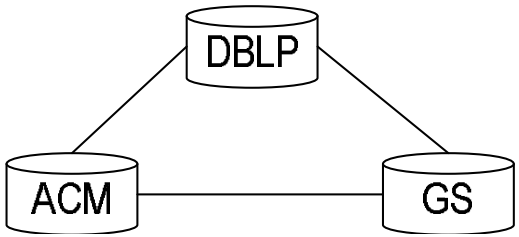
Dabei sei
$s(a, b) =$ Summe der Ähnlichkeitswerte aller Pfade (a, b)
$n(a) =$ Anzahl der Korrespondenzen $(a, x) \in map_1$
$n(b) =$ Anzahl der Korrespondenzen $(x, b) \in map_2$

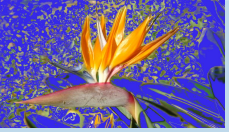


Komposition (2)

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

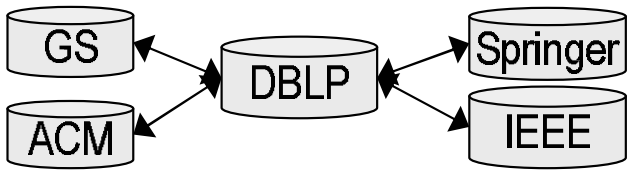
Mapping Compose via	DBLP - GS ACM	DBLP - ACM GS	GS - ACM DBLP
Direkt	81,3%	91,9%	35,3%
Compose	33,9%	63,7%	83,9%
Union	81,3%	91,6%	83,7%



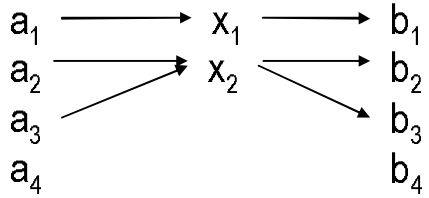


Komposition (3)

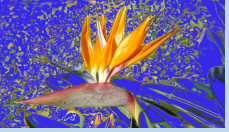
- Fazit
  - Komposition von Mappings ermöglicht effiziente Berechnung neuer Mappings
  - Besonders gut geeignet, falls Hub-Datenquelle vorhanden ist (Sternstruktur)
  - Fehlende Objekte in "mittlerer" Quelle führen zu fehlenden Korrespondenzen (Bsp: $a_4 - b_4$)
  - Komposition kann zu falschen Korrespondenzen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



Hub-Struktur

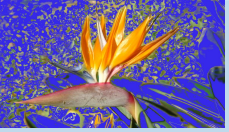


Problemfälle bei Komposition



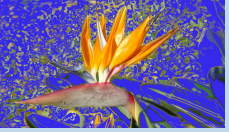
Neighborhood-Matcher: Motivation und Idee

- 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



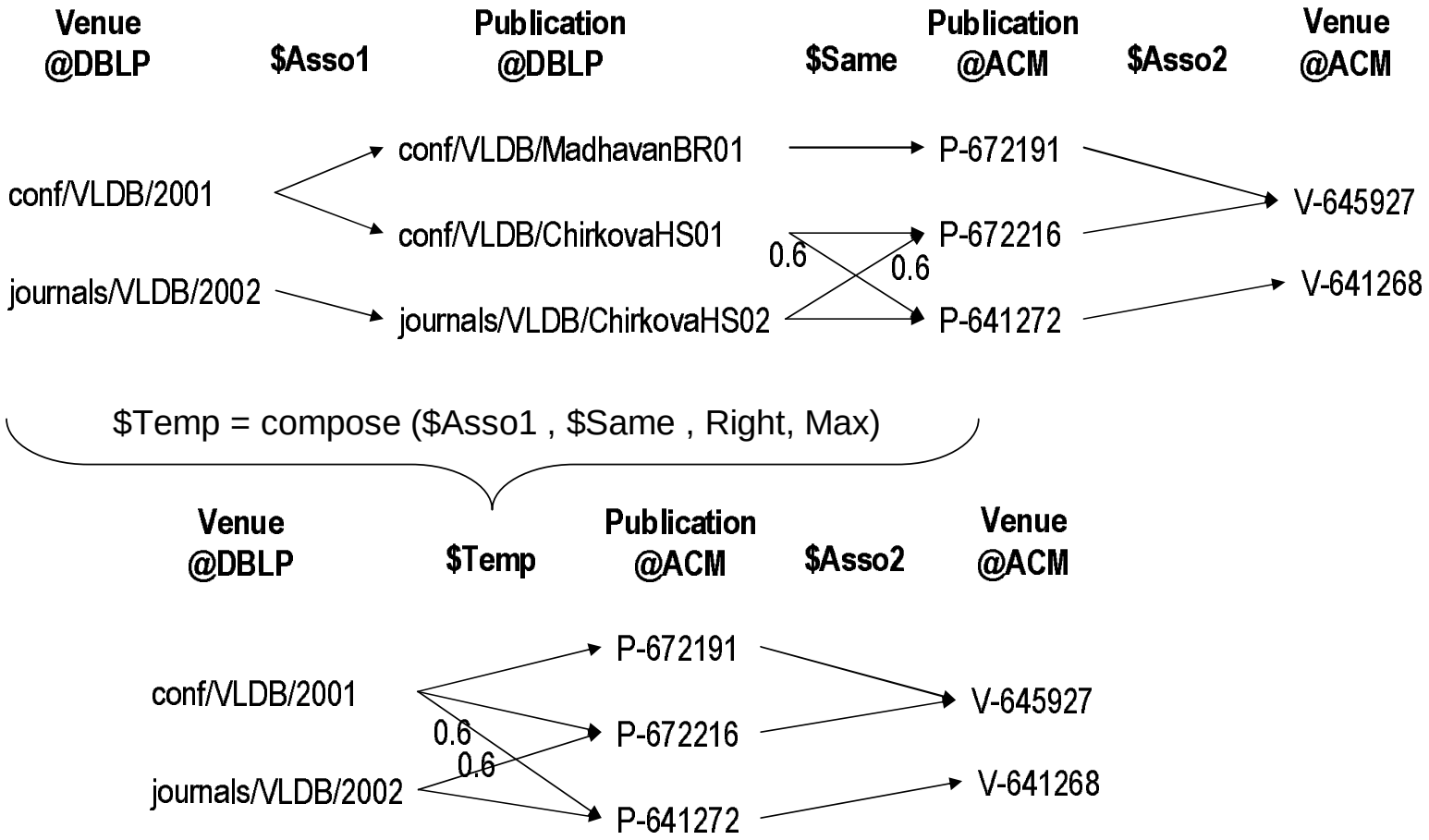
Neighborhood-Matcher: Match-Workflow

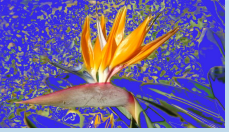
- Verwendung von Assoziations-Mappings
  - Syntax: Gleicher Struktur wie Same-Mappings; fester "Ähnlichkeitswert" = 1
  - Semantik: Korrespondenzen zwischen assoziierten Objekten, z.B. Publikationen - Venue
- Match-Workflow als Kompositon von drei Mappings
  - map1 und map3 sind Assoziations-Mappings; 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



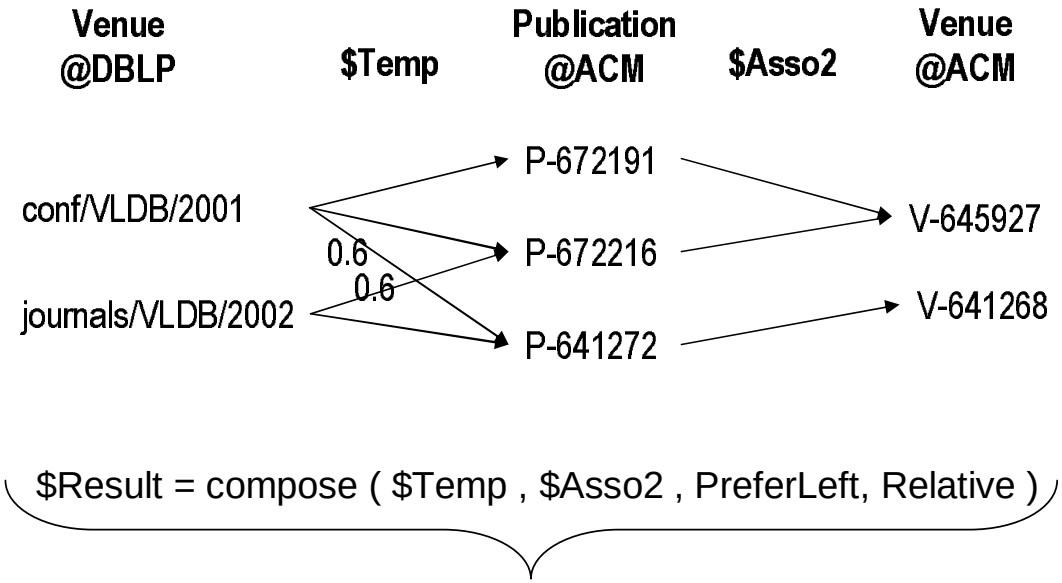
Neighborhood-Matcher: Beispiel (1)

- Ähnlichkeitswerte = 1 (solange nicht anders angegeben)

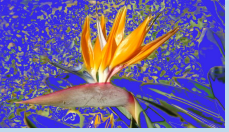




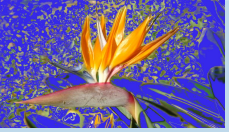
Neighborhood-Matcher: Beispiel (2)



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



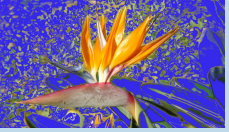
Qualität



Fehlerquellen

- Datenerfassung (Schreibfehler, Falschangaben (keine Wiedererkennung), Platzfüller bei Pflichtfeldern, wenn Wert fehlt, Meßfehler, ...)
- Alterung (Daten nicht aktuell und deshalb falsch)
- Tranformationsfehler, darunter auch falche Behandlung bei Integration !

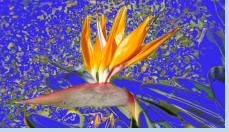
Folgen: Wert der Daten sinkt! Fehler führen zu Folgefehlern.
$\Rightarrow$ Fehlerbereinigung (Normalisierung, fehlende Werte korrekt Darstellen, ...
Diskussion: Fehler vs. Ausreißer $\rightarrow$ bisher keine LÖsung ohne den Experten



Qualitätskriterien - Auswahl

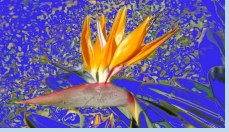
Was ist Qualität ? Meist nur *fitness for use*.

- Techn. Parameter (Verfügbarkeit, Zugriffszeit, Antwortzeit, Datenaktualität; Homogene Darstellung)
- Semantische P.: Metainformationen, Quellenangabe, Mehrwert durch Integration, Vollständigkeit, Fehlerfreiheit)
  Aber auch:
  Objektivität der Ziele, neutrale bzw. objektive Quellenwahl
  Reputation: bisher Quellen durch Experten bewertet, Wertung an Objekte (Attribute, Konzepte gebunden, Nutzerbefragungen.
- Bei Bio-DB haben sich einige (mit manuellem Einsatz) gepflegte Datenquellen etabliert. $\rightarrow$ Qualität durch Expertenwissen.
- Offene Fragen:
  Kann aus Qualität der Quellen und Ähnlichkeitsmaß auf Qualität des Ergebnisschema geschlossen werden?



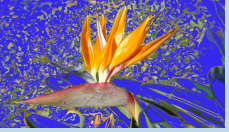
Qualitätsbewertung: Precision und Recall

- 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



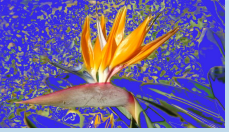
Qualitätsbewertung: Match Ratio und Match Coverage

- 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
- Abschätzung von Precision und Recall durch Match Ratio und Match Coverage



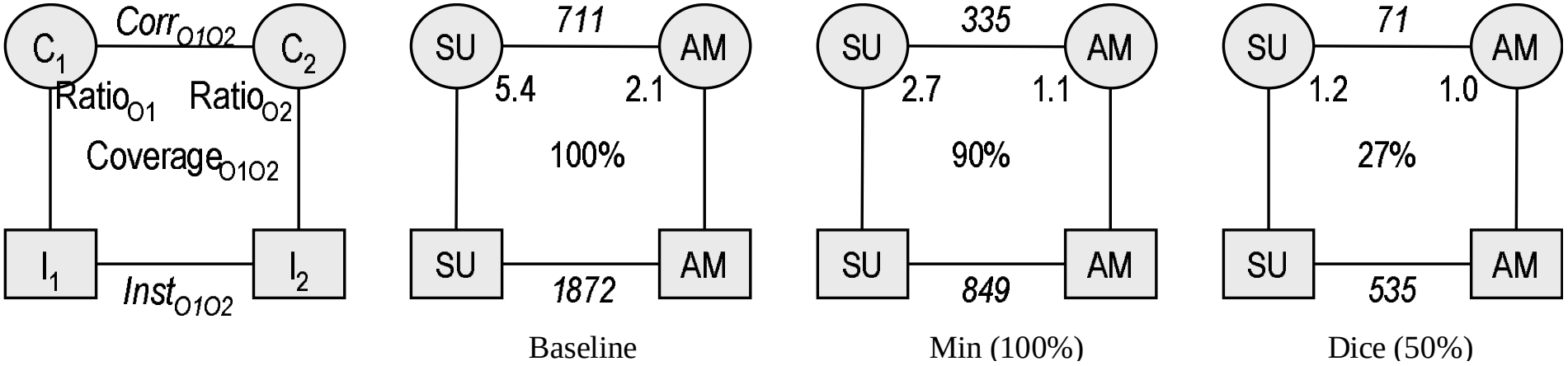
Qualitätsbewertung: Match Ratio und Match Coverage (2)

- 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)
  -

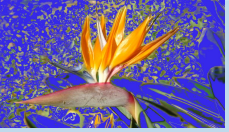


Qualitätsbewertung: Match Ratio und Match Coverage (Beispiel)

- 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



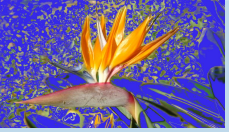
Baseline | Min (100%) | Dice (50%)



Qualität in P2P-Systemenn

- P2P-Systeme: Bewertung der Daten, des Servers, des Bewerters.
  Isolation von schlechten Peers, von schlechten Inhalten,
  Verfahren z.T. widerstandsfähig gegen Manipulation.
  Details: **Vorlesung** D. Sosna: P2P-Systeme und Datenbanken.
- Adaptionen:
  keine Angriffe, aber: verschiedene Lehrmeinungen, verschiedene Ontologien.
  Reputation durch Auswertung des Nutzungsverhaltens z.B. bei linkbasierten Systemen (indirekte Nutzung der Expertenkompenz)

... bisher nicht realisiert !



Zusammenfassung

- Schemaintegration: Ziel der Automatisierung nicht erreicht; Teilautomatisierung
- Ansätze: Metadatenbasiertes bzw instanzdatenbasiertes Mapping. Kombination verschiedener Ansätze erforderlich, Wiederverwendung von als gut erkannten Ergebnissen.
- Qualität: Begriffsbildungen vielfältig, z.T. weich, schwierige Einschätzung, subjektive Bewertung.