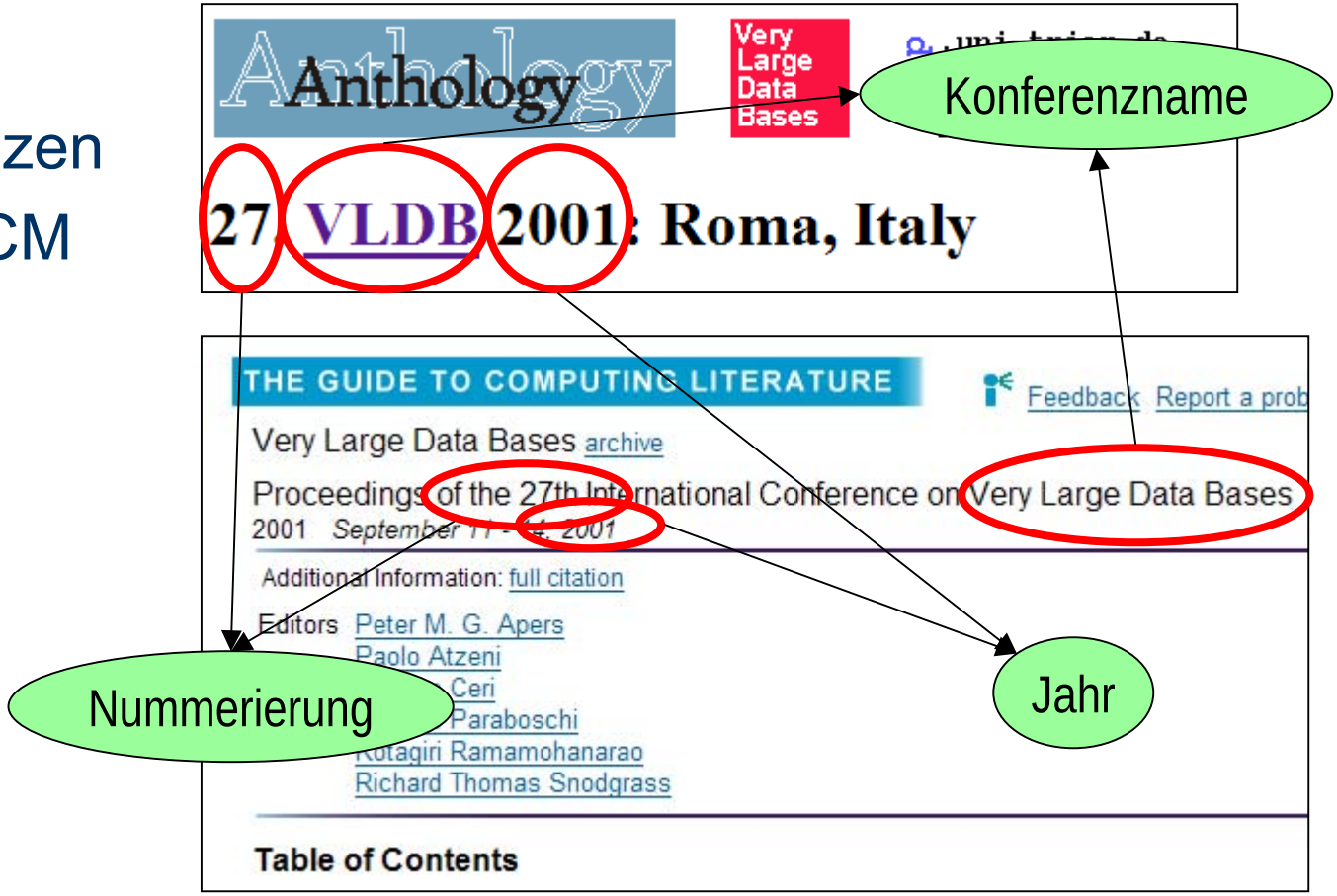
Kontextbasierte Datenintegration mit *iFuice*

Andreas Thor
http://dbs.uni-leipzig.de

Motivation

- Datenintegration: Finden *gleicher* Instanzen in verschiedenen Datenquellen → same-Mappings

- Beispiel:
  - Konferenzen
  - DBLP-ACM



Beispiel: Konferenzen

- Domänenspezifische Matcher
  - z.T. aufwändig zu erstellen
- Matching durch Verwendung weiterer Mappings
  - Verwendung weiterer Mappings
  - Bsp: Publikationen sind einfacher zu matchen

- Umeshwar Dayal, Meichun Hsu, Rivka Ladin:
  Business Process Coordination: State of the Art, Trends, and Ope
  Electronic Edition (link) BibTeX
- Egbert-Jan Sol:
  Ambient Intelligence with the Ubiquitous Network, the Embedded
  Electronic Edition (link) BibTeX
- Philip Wadler:
  Et tu, XML? The downfall of the relational empire (abstract). 15
  Electronic Edition (link) BibTeX
- Pierre-Paul Sondag:
  The Semantic Web Paving the Way to the Knowledge Society. 16
  Electronic Edition (link) BibTeX

Table of Contents

Business Process Coordination: State of the Art, Trends, and Open Issues
Umeshwar Dayal, Meichun Hsu, Rivka Ladin
Pages: 3 - 13
Additional Information: full citation, citings

Ambient Intelligence with the Ubiquitous Network, the Embedded Comput
Egbert-Jan Sol
Page: 14
Additional Information: full citation

Et tu, XML? The downfall of the relational empire (abstract)
Philip Wadler
Page: 15
Additional Information: full citation

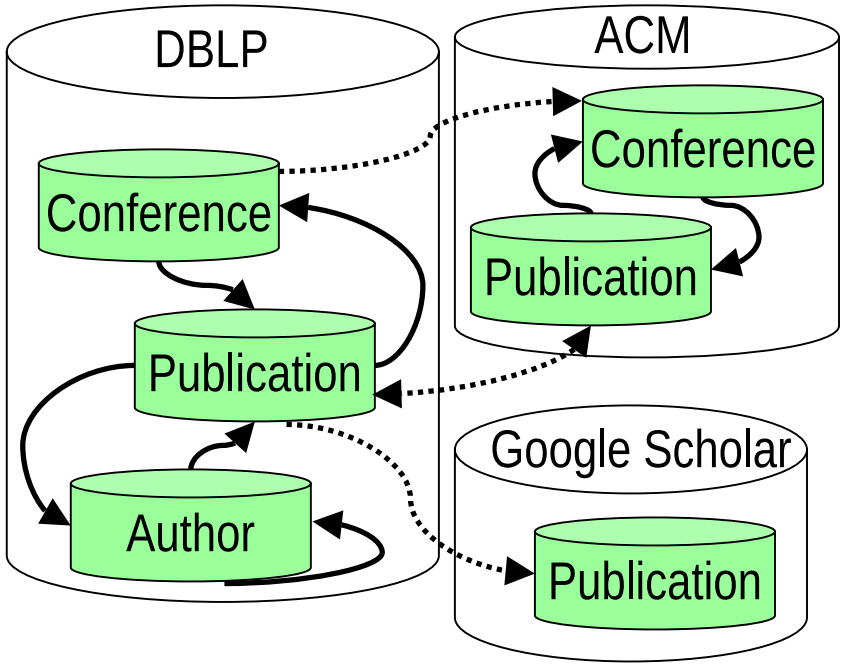
The Semantic Web Paving the Way to the Knowledge Society
Pierre-Paul Sondag
Page: 16
Additional Information: full citation

Agenda

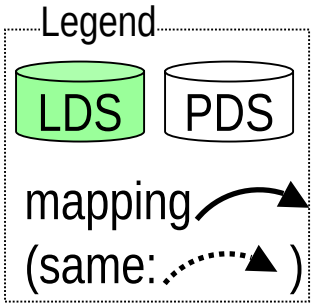
- Motivation
- *iFuice*
- Kontext
- *iFuice*-Erweiterungen
  - Joint Distribution
  - *iFuice*-Matcher
- Beispiel: Neighbourhood-Matcher
- Zusammenfassung & Ausblick

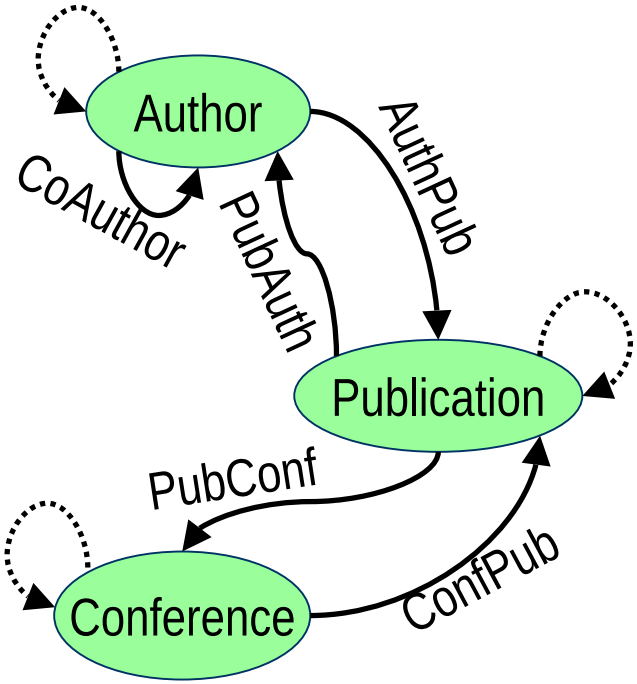
Metadatenmodell

- Logische Datenquelle (LDS) =
  Physische Datenquelle (PDS) + Objekttyp



Source-Mapping-Modell





Domänen-Modell

Begriffe

- Object Instances
  - Menge von Instanzen aus einer LDS
- Mapping Result
  - Menge von Korrespondenzen zwischen Instanzen zweier LDS
- Mapping
  - Funktion: O → MR
- Matcher
  - Funktion: (O, O) → MR
    oder MR → MR

DBLP- Publikationen

(DBLP-Pubs) → (ACM-Pubs)

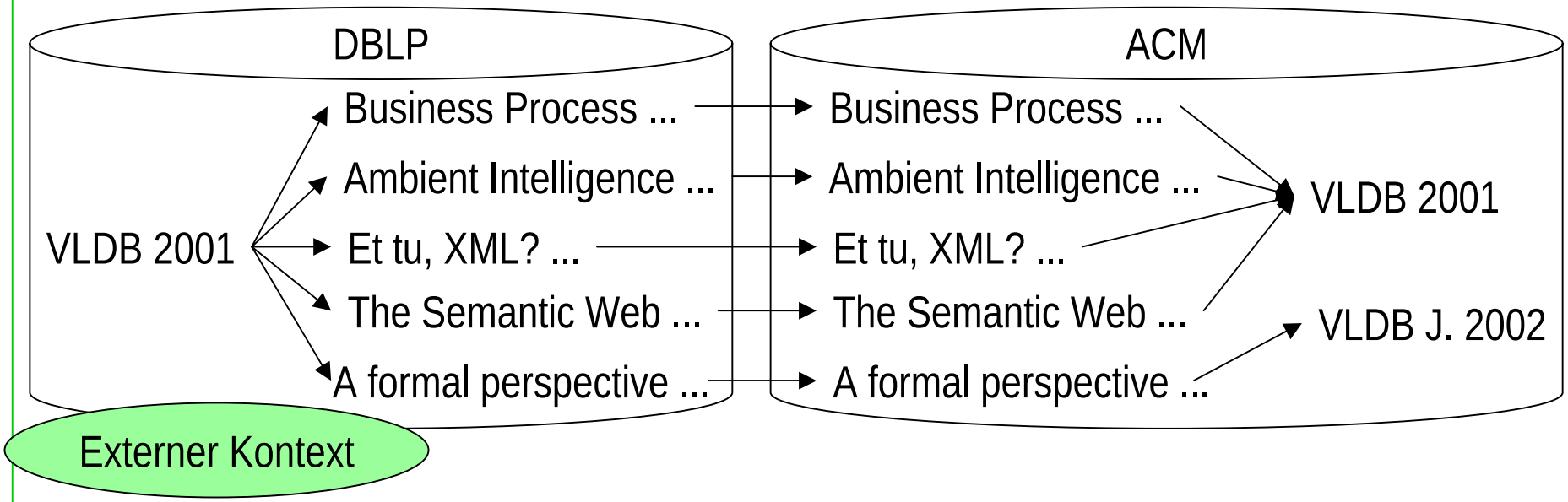
Realisiert als
- Mapping Result
- SQL-Query
- Java-Programm
- (Web Service)
- ...

iFuice-Skripte

- Operatoren für Object Instances
  - **`queryInstances`** (LDS, query)
- Operatoren für Mapping Results
  - **`map`** (O, mapping)
  - **`match`** (O, O, matcher) bzw. **`match`** (MR, matcher)
- Skripte
  - sequentielle Folge von Operatorenaufrufen
  - Schachtelung, z.B. **`map`** (**`queryInstances`** (.. , ..), ..)
  - Variablen für (Zwischen-) Ergebnisse

Beispiel: DBLP-ACM-Konferenzen

- Ziel: Mapping Result DBLP-ACM-Konferenzen



compose =

DBLP	ACM	Gewicht	Verteilung
VLDB 2001	VLDB 2001	4	80%
VLDB 2001	VLDB J. 2002	1	20%

Interner Kontext

Mapping Result

- MR = { (*x*, *y*, *occurences*, *confidence*) }
  - $(x, y) \in O \times O$ Korrespondenz
  - *occurences* $\in N^+$ Anzahl der Auftretens der Korresp.
  - *confidence* $\in [0, 1]$ Ähnlichkeitswert (z.B. Stringähnlichk.)
- Operatoren
  - `union`:
    - Summe der *occurences*, gew. Mittel der *confidence*
  - `compose`:
    - Assoziativität: $(MR_1 \circ MR_2) \circ MR_3 = MR_1 \circ (MR_2 \circ MR_3)$
    - Bestimmung aller compose-Pfade
    - Summe der *occurences*, gew. Mittel der *confidence*

Joint Distribution

- Wie häufig korrespondiert VLDB'01$_{DBLP}$ zu VLDB'01$_{ACM}$ - und wie häufig zu anderen?
- Wie häufig korrespondiert Nicht-VLDB'01$_{DBLP}$ zu VLDB'01$_{ACM}$ - und wie häufig zu anderen?

- $\rightarrow$ Relative Häufigkeit der Korrespondenzen

	y	$\neg y$
x	JD_{11}	JD_{10}
$\neg x$	JD_{01}	JD_{00}

$$JD_{11} = \frac{\sum_{\substack{(x_i, y_i, o_i, c_i) \in MR \\ (x_i = x) \wedge (y_i = y)}} o_i \cdot c_i}{\sum_{\substack{(x_i, y_i, o_i, c_i) \in MR \\ x_i = x}} o_i \cdot c_i}$$

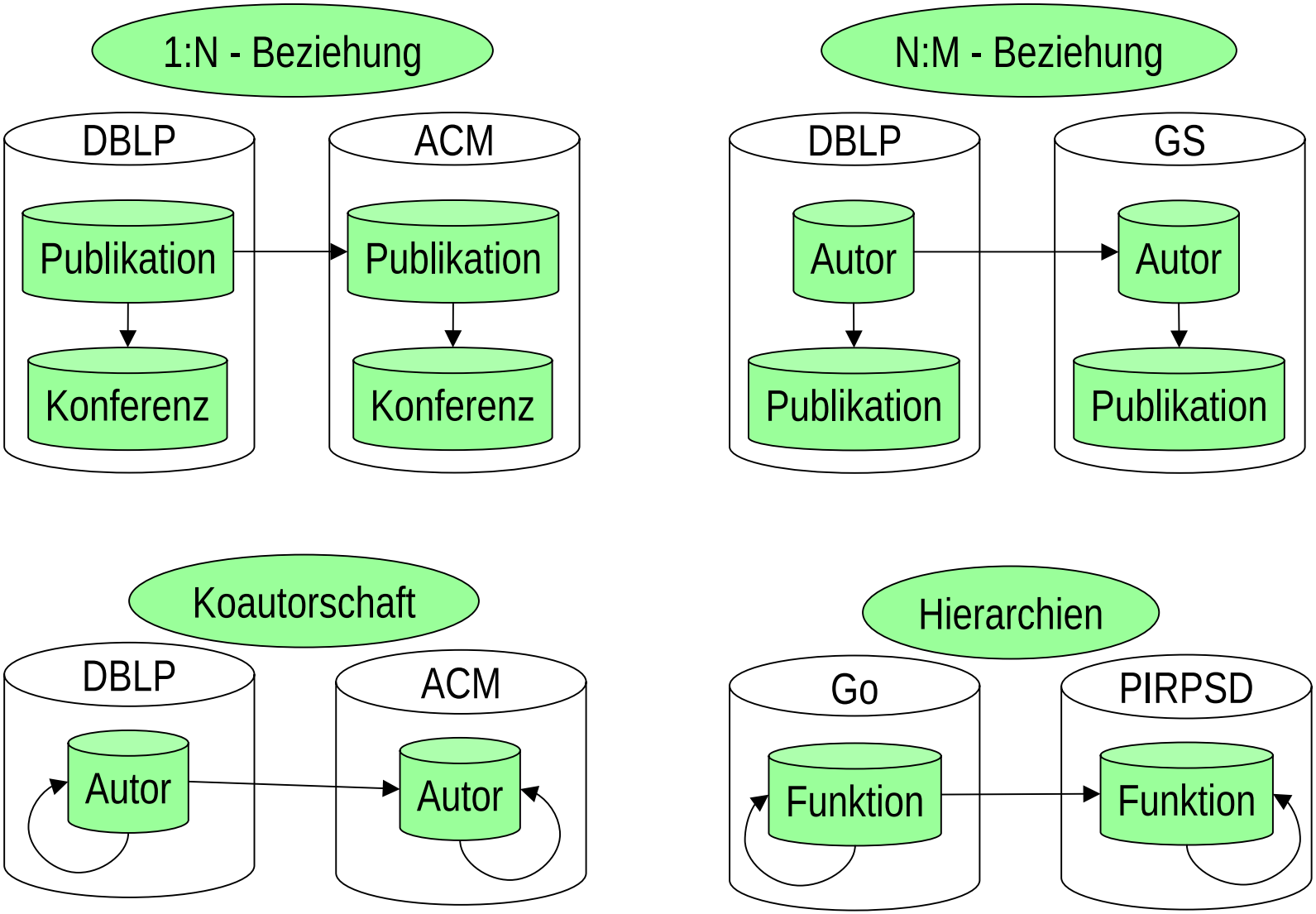
Joint Distribution (Beispiel)

- confidence = 1

DBLP	ACM	Occ.	JD11	JD10	JD01	JD00
VLDB 2001	VLDB 2001	4	80 %	20 %	0 %	100 %
VLDB 2001	VLDB J. 2002	1	20 %	80 %	0 %	100 %
SIGMOD 2001	SIGMOD 2001	7	100 %	0 %	0 %	100 %

DBLP	ACM	Occ.	JD11	JD10	JD01	JD00
VLDB 2001	VLDB 2001	4	80 %	20 %	13 %	87 %
VLDB 2001	VLDB J. 2002	1	20 %	80 %	0 %	100 %
SIGMOD 2001	SIGMOD 2001	7	78 %	22 %	0 %	100 %
SIGMOD 2001	VLDB 2001	2	22 %	78 %	80 %	20%
TODS 2001	TODS 2001	6	100 %	0 %	0 %	100 %

Externer Kontext: Beispiele



Externer Kontext

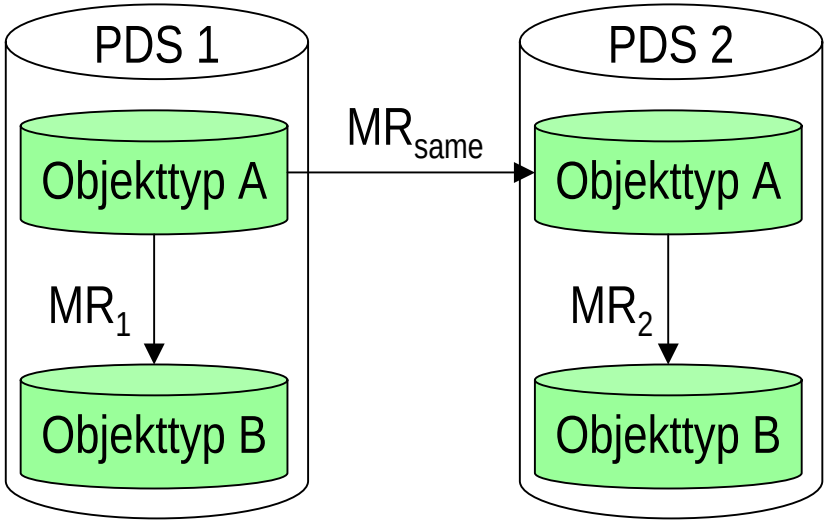
- Abbildung und Behandlung von Kontextmustern in *iFuice*
- Verarbeitung in Unterprogrammen (Skripten)
- Parameter
  - Datenquellen
  - Mappings
- Ergebnis als Rückgabewert

iFuice-Skripte: Erweiterung

- Skalare Funktionen
  - `count`: Anzahl der Objektinstanzen / Korrespondenzen
  - `cardinality`: Kardinalität eines MR
- Kontrollstrukturen
  - IF THEN ELSE
  - WHILE DO
- Unterprogramme
  - Skripte können als Prozedur abgespeichert werden
  - Aufruf über `script`-Operator in anderem Skript
  - Mehrwertige Ergebnisse möglich, z.B. (MR, MR)

Neighbourhood Matcher

- Kandidatenbestimmung mittels compose $(MR_1)^{-1} \circ MR_{same} \circ MR_2$
- Weiterverarbeitung abhängig von Kardinalität
- Beispiel
  - 1:1 : Autor - Homepage
  - 1:N : Konferenz - Publikation
  - N:1 : Publikation - Konferenz
  - N:M : Publikation - Autor



Neighbourhood Matcher: Skript

```
PROCEDURE neighbourhoodMatch ($MRsame, $MR1, $MR2)
  $MRcand := compose(compose (inverse ($MR1), $MRsame), $MR2);
  IF cardinality ($MR1) == %CARD11 THEN
    $MRresult := $MRcand;
  END
  IF cardinality ($MR1) == %CARD1N THEN
    $MRresult := attrMatch ($MRcand);
  END
  IF cardinality ($MR1) == %CARDN1 THEN
    $MRresult := queryMapResult ($MRcand, [JD10]<0.3);
  END
  IF cardinality ($MR1) == %CARDNM THEN
    $MRresult := attrMatch ($MRcand);
    $MRresult := queryMapResult ($MRresult, [JD10]<0.3);
  END
  RETURN $MRresult;
END
```

Neighbourhood Matcher: Aufruf

- Aufruf des Matchers in einem Skript

```
$DBLPConf := queryInstances (Conf@DBLP, "[name]='VLDB'");
$DBLPPubs := traverse ($DBLPConf, DBLP.ConfPub);
$MRsame := map ($DBLPPubs, DBLP2ACMViaGoogle)
$MR1 := map ($DBLPPubs, DBLP.PubConf);
$MR2 := map (range ($MRsame), ACM.PubConf);
$MRConf := script (neighbourhoodMatch, $MRsame, $MR1, $MR2);
```

- Wiederholtes Aufrufen des Matchers
  - Publikation-Map → Konferenz-Map → Publikation-Map
- Pendel für Publikationen und Autoren
  - manuell erstelltes initiales same-Mapping
  - Pendeln zum "crawlen" neuer Publikationen / Autoren

Zusammenfassung und Ausblick

- Nutzung des Kontextes für Datenintegration
- Erweiterung von *iFuice* u.a. um
  - Joint Distribution
  - Kontrollstrukturen und Unterprogramme
- Einsatzmöglichkeiten
  - Definition eigener kontextbasierter Matcher in *iFuice*
  - nicht-sequentielle Ausführung von Mappings / Matchern
- Ausblick
  - Evaluation verschiedener Verfahren in der Biblio-Domäne