



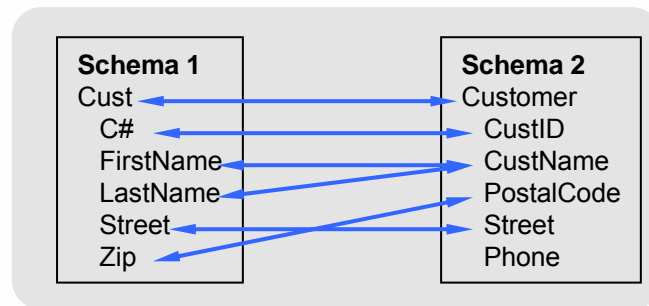
Schema and Ontology Matching with COMA++

David Aumueller, Hong-Hai Do, **Sabine Massmann**, Erhard Rahm

University of Leipzig, <http://dbs.uni-leipzig.de>
Leipzig, November, 29th 2005

Introduction

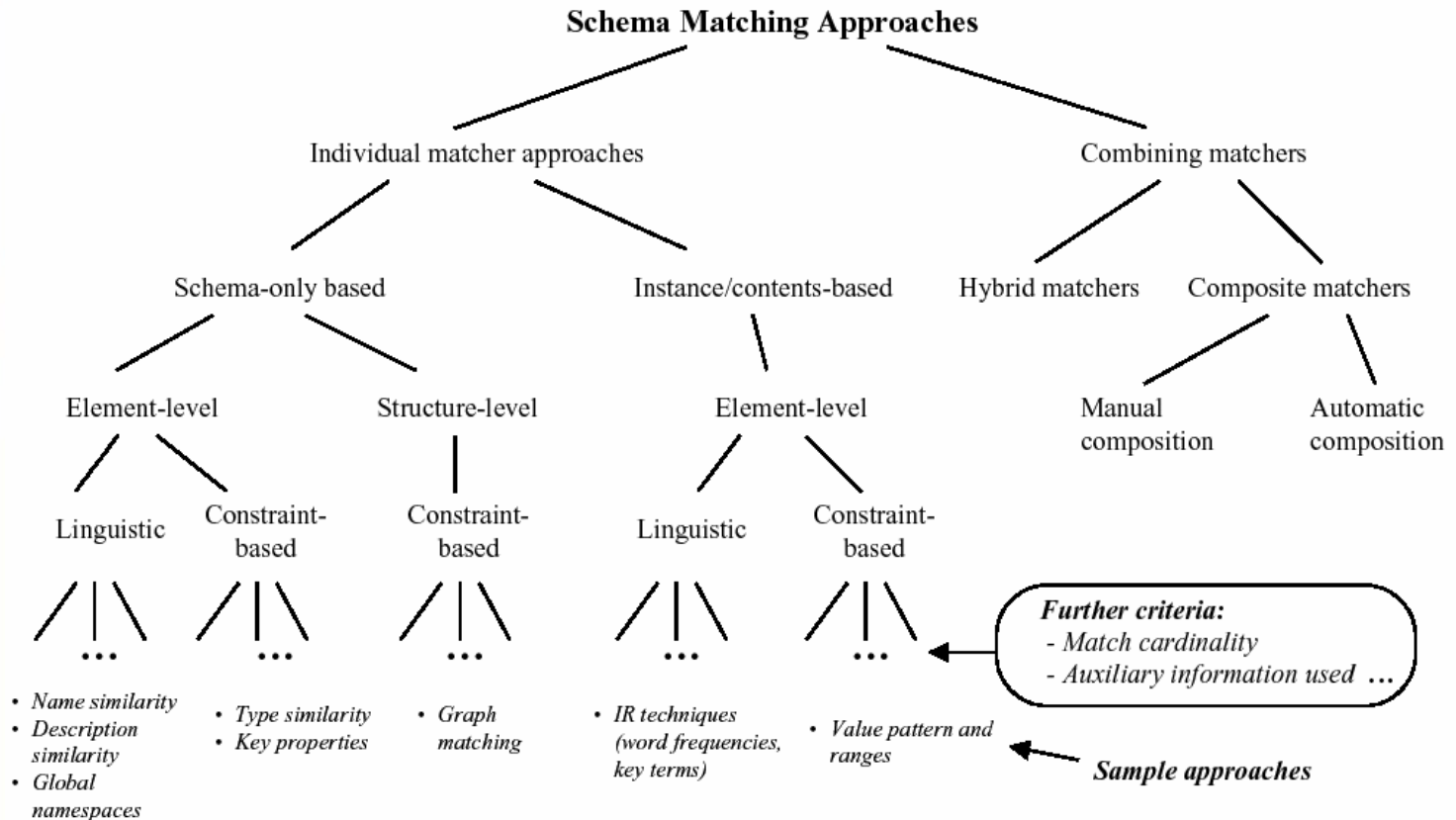
- Schema matching:
 - Input: 2 schemas e.g., DB Schemas, XML message formats
 - Output: Identification of correspondences between elements



- Application domains:
 - Data integration
 - E-business
 - Data warehousing
 - Semantic query processing

Introduction (2)

- Development of many techniques and prototypes to semi-automatically solve the match problem

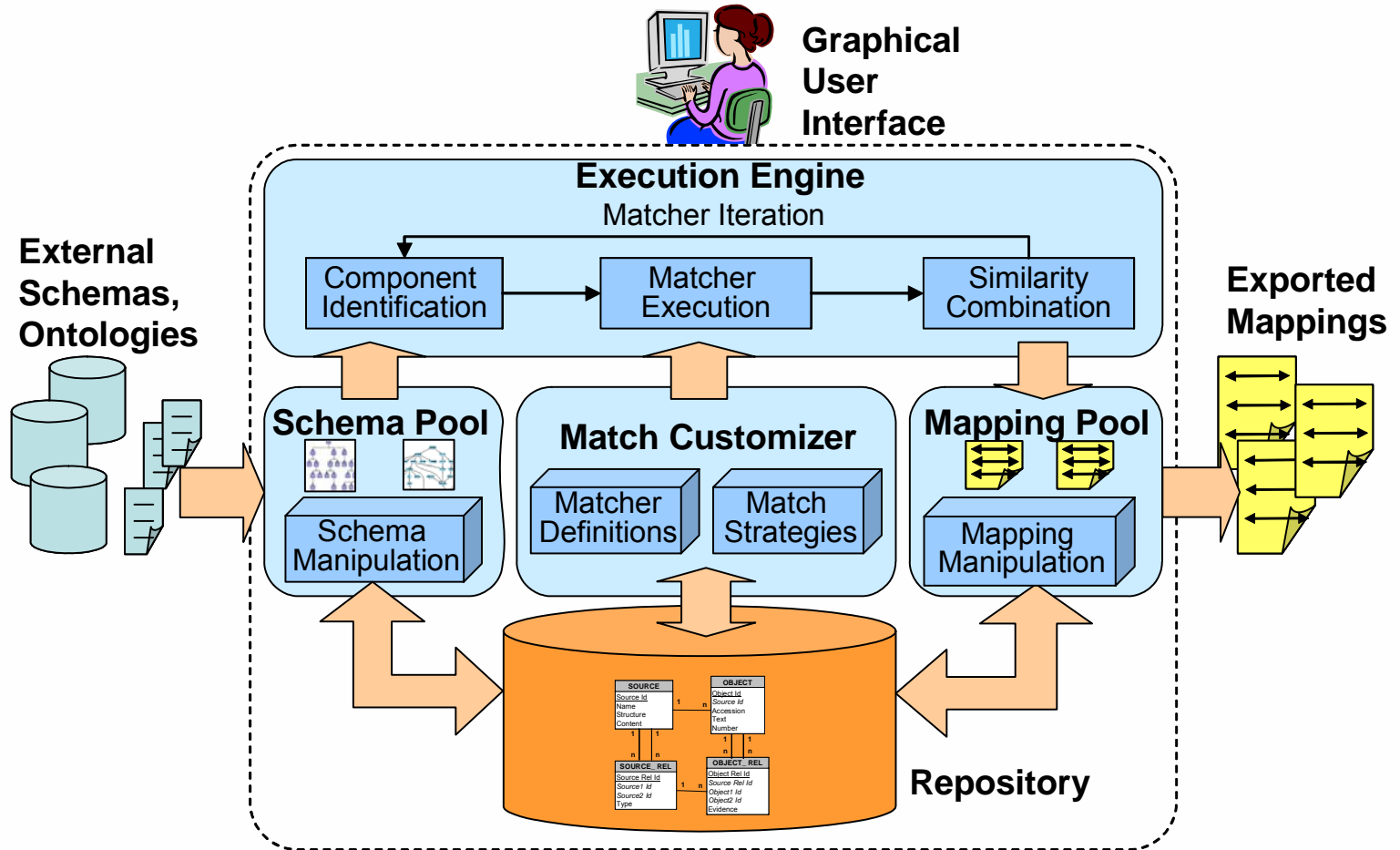


[from: A survey of approaches to automatic schema matching, E.Rahm, P.A. Bernstein, VLDB Journal 10:4 2001]

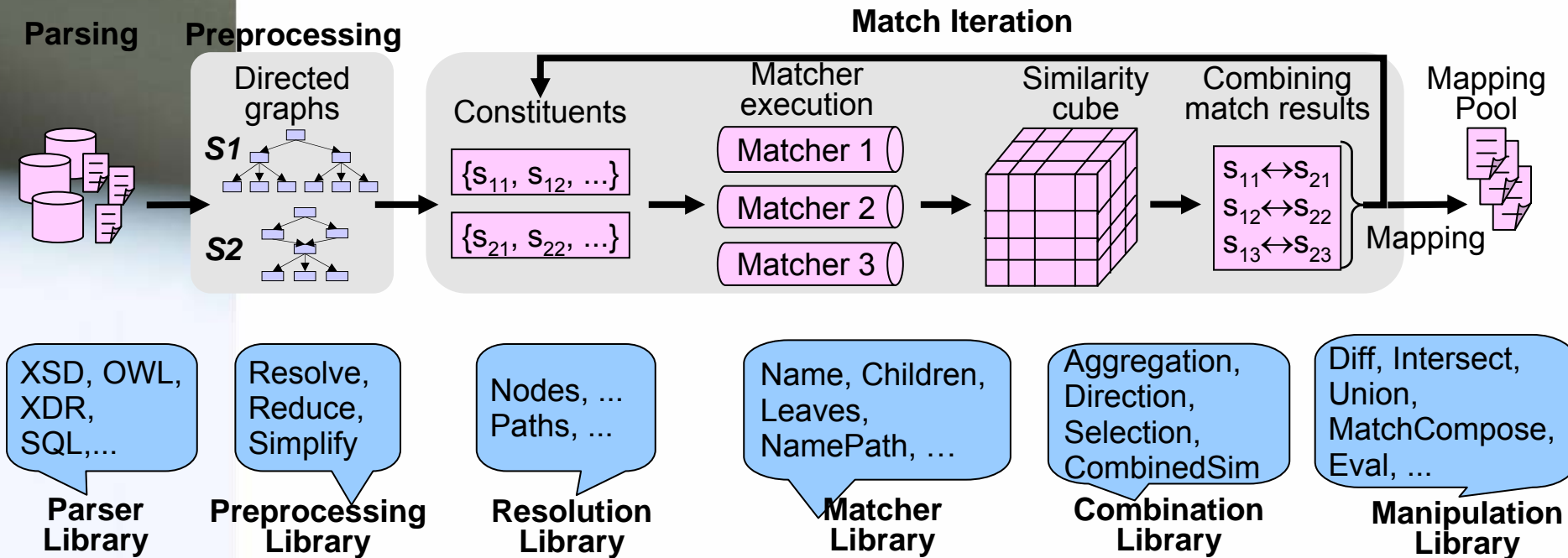
COMA++

- Generic, customizable schema matching tool
- Generic data model to uniformly support schemas and ontologies written in different languages (W3C XSD, XDR, OWL, Relational)
- Repository for uniform management of
 - Schemas, ontologies
 - Match results
 - Auxiliary information
- Flexible combination of matchers to construct new matchers and match strategies
- Optimized implementation of matchers for fast execution times
- New approaches for ontology, reuse-, context-, and fragment-based matching
- Platform for comparative evaluation of matchers and match strategies
- High-level operators for manipulating match results, e.g. compose, merge, and compare mappings
- GUI (Java Swing)

Architecture of COMA++

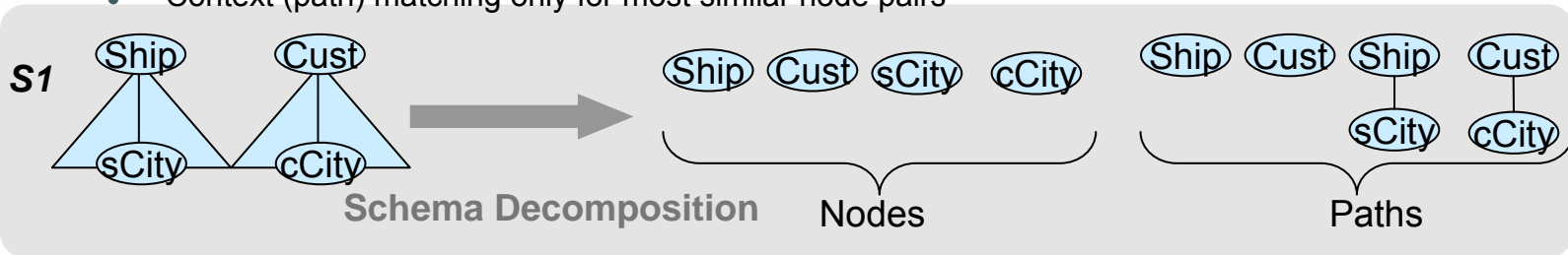


Match processing

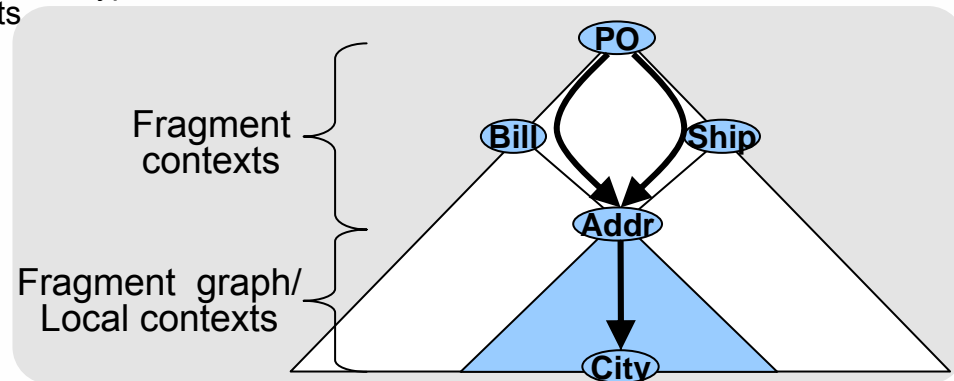


Match-Strategies

- **AllContext:**
Match between all unique contexts, i.e. paths
- **FilteredContext:**
Two-phase matching
 - Node matching
 - Context (path) matching only for most similar node pairs

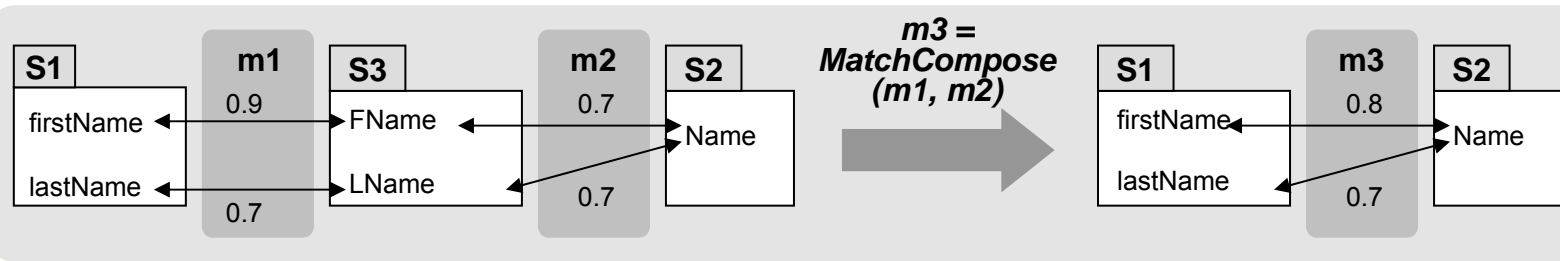
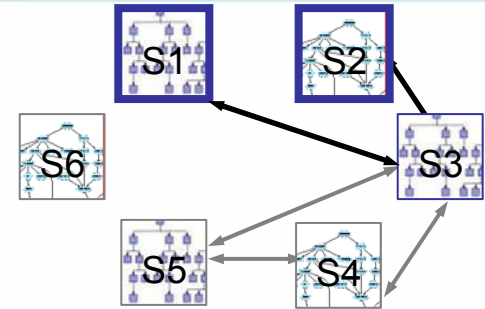


- **Fragment-based** (automatic, step-by-step):
Decomposition of large schemas and matching at fragment level
 - Identify similar fragments of specified type
 - Match between similar fragments

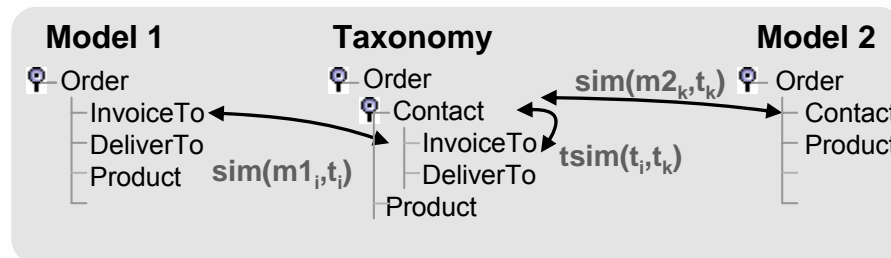


Match-Strategies (2)

- Reuse** (automatic, manual):
 Determination of mapping paths from existing match results to solve a new match tasks



- Taxonomy:**
 Match input schemas to a selected taxonomy and combine match results



Demo

Schema Evolution*

- Frequent need to change schemas
 - Improve original design / remove design errors
 - New application requirements
 - New data source for global schemas ...
- High degree of manual work to deal with database schema evolution
 - Schema redesign
 - Data migration
 - Evolving dependent mappings
 - Adapting dependent applications

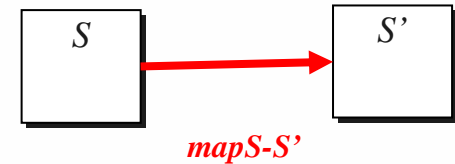
* Model Management Solution Patterns for Schema Evolution
Erhard Rahm, Phil Bernstein, Sergey Melnik

Model Management

- MM operators very promising for schema evolution
 - Automated processing of entire schemas and mappings
 - Compact solution patterns
 - Reuse of previous mappings
 - Low degree of manual interaction
- Previously proposed MM scripts for schema evolution (Vision2000, ER2000, CIDR2003, SIGMOD2003, Diss2004, ICDE2004 tutorial, SIGMOD2005 ...)
 - Based on different sets of operators
 - Different assumptions for mappings and mapping language
 - Early scripts heavily use Match – not fully automatic
 - Later scripts heavily use Compose / Invert of mappings - limitations for composability / invertibility
 - Scripts often address only subset of problems (e.g. only insert or delete propagation)

Mapping Considered Evolution Problems

- **Mapping generation** for changed schema
 - Direct mapping generation (Map)
 - Incremental mapping generation
- **Mapping evolution** due to schema change
 - *Source evolution* (SE)
 - Simple source evolution (SE1)
 - *Change propagation* (SE2)
 - Delete propagation (S-DP)
 - Insert propagation (S-IP)
 - *Target evolution* (TE)
 - Simple target evolution (TE1)
 - *Reverse engineering* (TE2)
 - Delete propagation (T-DP)
 - Insert propagation (T-IP)
- **Schema reintegration**



Simple Source Evolution

1. Direct Match solution

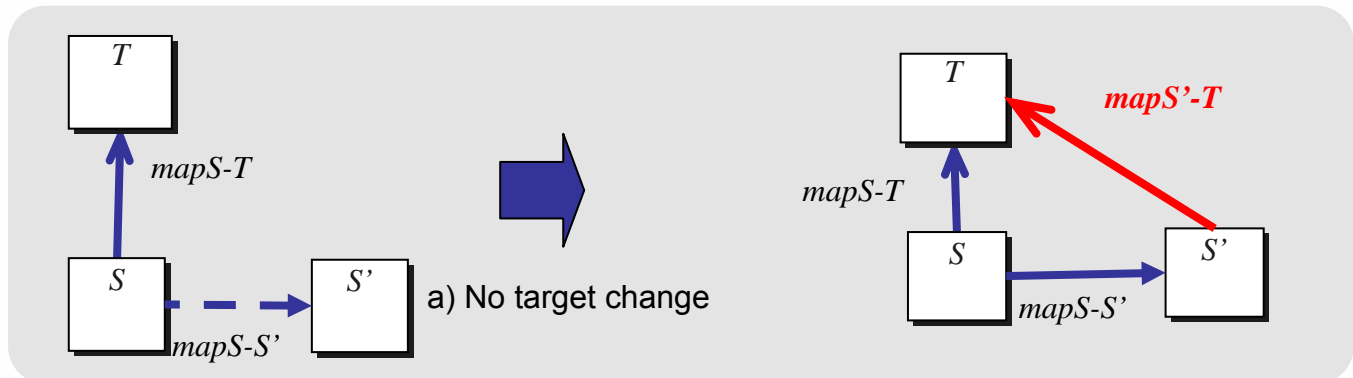
$\text{map}_{S'-T} = \text{Match}(S', T)$

- ☹ Not fully automatic
- ☹ Reuse of existing mappings (?)

2. Compose solution (Example 1)

1. $\text{map}_{S'_S} = \text{embed}(S', S)$ // $\text{map}_{S-S'}$ may not be invertible

2. $\text{map}_{S'-T} = \text{map}_{S'_S} \circ \text{map}_{S-T}$ // reuse of map_{S-T}



- ☺ Simple and fully automatic
- ☹ Result may be incomplete due to added schema elements

Simple Source Evolution (2)

3. Combined solution (Example 2)

1. $\text{mapS}'_S = \text{embed}(S', S)$

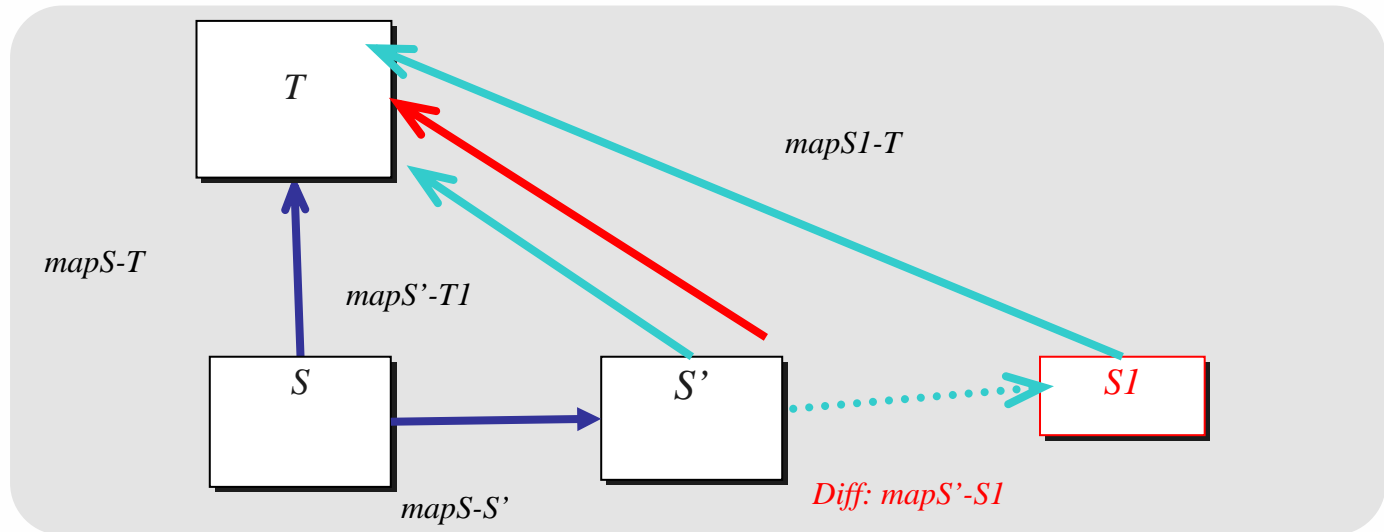
2. $\text{mapS}'_{-T1} = \text{mapS}'_S \circ \text{mapS}_{-T}$ // maximally reuse ex. mapping to T

3. $\langle S1, \text{mapS}'_{-S1} \rangle = \text{Diff}(S', \text{mapS}'_{-T1})$ // S' subset not yet mapped to T

4. $\text{mapS1}_{-T} = \text{Match}(S1, T)$

5. $\text{mapS}'_{-T2} = \text{mapS}'_{-S1} \circ \text{mapS1}_{-T}$

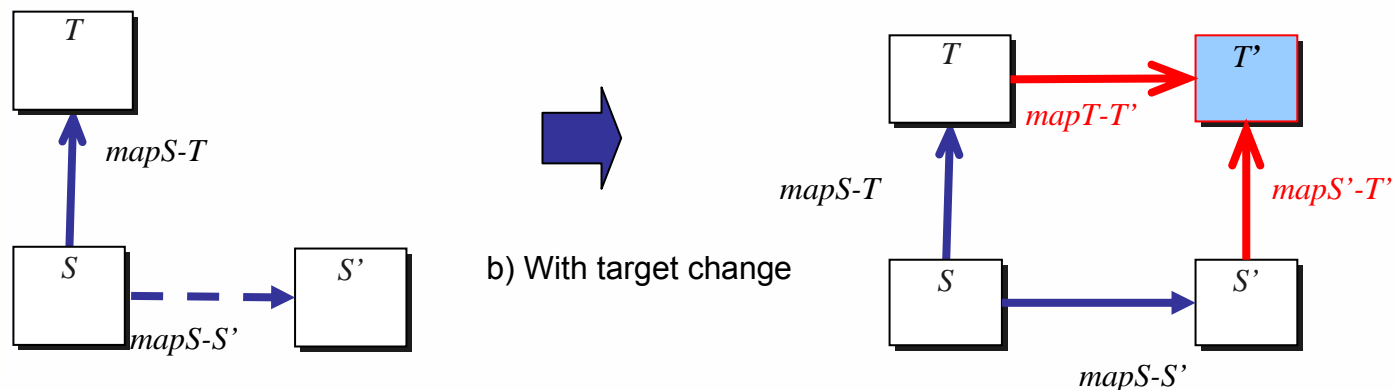
6. $\text{mapS}'_{-T} = \text{mapS}'_{-T1} \oplus \text{mapS}'_{-T2}$ // confluence



- ☺ Added schema elements are as well considered
- ☹ Not fully automatic

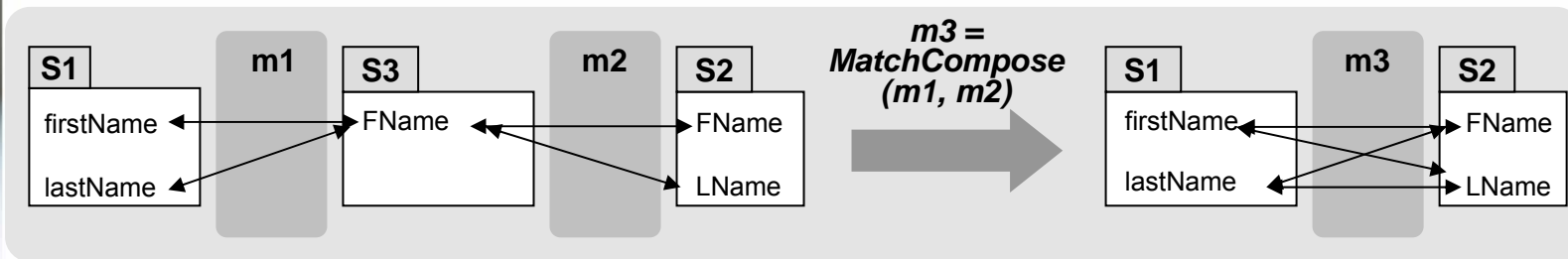
Change Propagation

- $\langle T', \text{map}_{T-T'}, \text{map}_{S'-T'} \rangle = \text{SE2}(S, T, \text{map}_{S-T}, S', \text{map}_{S-S'})$
- Consider invertibility of mappings
- map_{S-T} may not be total, e.g. T may have components unrelated to S (must be retained)
- User should decide about which changes (e.g. deletes) should really propagate / cascade
- Assumption: schemas are represented in same (super) meta-model
- Delete propagation, insert propagation



COMA++

- Simple Source/Target Evolution possible
- Change Propagation not
- Currently assuming 1:1 relations



- No support of key/keyref at the moment

References

- Aumüller, D., Do, H.H., Massmann, S., Rahm, E:
Schema and Ontology Matching with COMA++.
Proc. SIGMOD 2005 (Software Demonstration), Baltimore, June 2005
- Rahm, E., Do H.H., Maßmann, S.:
Matching Large XML Schemas.
Sigmod Record 33(4), December 2004.
- Do, H.H., Melnik, S., Rahm, E.:
Comparison of Schema Matching Evaluations.
Proc. GI-Workshop "Web and Databases", Erfurt, Oct. 2002
- Do, H.H., Rahm, E.:
COMA - A system for flexible combination of schema matching approaches.
Proc. 28th Intl. Conference on Very Large Databases (VLDB), Hongkong,
Aug. 2002
- Rahm, E., Bernstein, P.A.:
A Survey of Approaches to Automatic Schema Matching.
VLDB Journal, Vol. 10, No. 4, Dec. 2001

Thanks.



E-Commerce-Projekt

InnoRegio Musicon Valley:
Untersuchungen zur Entwicklung von
kundenorientierten E-Commerce-Lösungen in
regionaler Kooperation für KMU und Handwerk
im Musikinstrumentenbau

Prof. Dr. Erhard Rahm
Institut für Informatik, Universität Leipzig
<http://dbs.uni-leipzig.de>

Motivation

- E-Commerce ist für den Vertrieb von Musikinstrumenten von zunehmender Bedeutung
- Kunden erwarten weitgehende Unterstützung bei Information und Kauf
- Von einigen Herstellern bereits in unterschiedlichen Entwicklungsstufen eingeleitet
- Aber: oft fehlen noch Wissen und Unterstützung für professionelle Nutzung
- Enge Abstimmung mit den betroffenen Unternehmen und Organisationseinheiten (Fertigung, Vertrieb, etc.) notwendig

Projektskizze

- System zur Unterstützung von E-Commerce-Lösungen für regionale Musikinstrumentenbauer
- Plattform soll Modellcharakter für weitere Unternehmen der Region haben
- Berücksichtigung verschiedener Ausbaustufen
- Anforderungsanalyse
- Berücksichtigung der Besonderheiten der Branche
- Verwaltung/Pflege durch geschulte Mitarbeiter sowie durch externe Partner möglich

Projektpartner

■ Universität Leipzig:

- Recherche und Aufarbeitung vorhandener Lösungen für E-Commerce im Musikinstrumentenbau
- Anforderungsanalyse
- Konzeptionelle Entwicklung der Plattform
- Evaluierung / Verfeinerung



■ Musicon Valley e.V.:

- Bindeglied zwischen beteiligten Unternehmen und Entwicklern
- Marketingstrategien



Projektpartner(2)

■ Informationsagentur VIA GmbH

- Internetagenturen der Region
- Modellhafte Ausarbeitung und Erprobung der Plattform mit ausgewählten Partnerunternehmen
- Externe Unterstützung für Unternehmen
- Schulung der Mitarbeiter der Unternehmen



INFORMATIONSENTUR **VIA** GMBH

■ Unternehmen des Musikinstrumentenbaus

- Involvierung in allen Projektphasen (Anforderungsanalyse, Entwicklung, Evaluierung)



■ Gefördert durch BMBF

Unternehmen des Musikinstrumentenbaus

- Bandonion- und Concertinafabrik GmbH
 - Weltweit bekannt → einer der wenigen Hersteller von Bandonions
 - B2C (Verkauf an Endkunden)
 - Verkauf quasi nur über Mail, selten Fax
 - Geringe Anzahl Instrumente, dafür jedoch mehrere Optionen möglich
- Matri
 - Matri & Friends
 - Präsentation mehrere Instrumentenbauer
 - Geigen, Cellos, Bögen
 - Website → Imagegewinn, Kunden interessieren/informieren
 - Verkauf fast nur über Fax
 - B2B (also Verkauf eher an Händler, weniger Endkunden)
 - Aufbau der Preislisten beibehalten als Orientierung für Händler
→ unterschiedlich für jeden Unternehmer

Aspekte der Branche

- Kunden aus der ganzen Welt
 - Unterstützung mehrerer Sprachen
 - Beachtung rechtlicher Aspekte (AGBs,..) und Kosten (Versandkosten, Zoll/Steuern)
- Aufbau und Pflegen von Kundenkontakten viel über Messe
 - Ankündigung von Messen/Events
- Bestellung noch über Telefon, Fax & Mail; Informationsaustausch zusätzlich über Flyer/Listen
 - Plattform soll informieren & Kommunikation „übernehmen“
- Geschichte, Herkunft & Fertigung wichtig
 - Darstellung auf Plattform
- Qualität (keine Massenware), Bau von Instrumenten fast nur nach Bedarf/Bestellung
 - Konfiguration von Wunschinstrumenten
 - Personalisierung der Preise
- Zielgruppe nur teilweise bewandert im Umgang mit Internet
 - Einfach & übersichtlich halten
 - Viel „Schnickschnack“ von anderen E-Shops überflüssig (z.B Bestseller, Bewertungen durch Kunden)
- Bau (von Instrumenten) dauert meist mehrere Wochen
 - Status der Fertigung *zusätzlich* zum Bestellstatus
- Da Kunden wahrscheinlich zusätzliche Kommunikationsmittel nutzen
 - Admin muss z.B. Bestellung für Kunden aufgeben können, dessen Profildaten ändern, etc.
- Bezahlung zur Zeit meist nur über Vorkasse
 - Unterstützen weiterer Möglichkeiten (Kreditkarte, ePayment ...)

Aktueller Stand

- Anforderungsanalyse
- Auswertung
 - vorhandener Seiten von Musikinstrumentenbauern und Online-Shops
 - Auswertung vorhandener E-Commerce-Systeme
 - rechtlicher Aspekte
- Erstellen bzw. Zusammenstellen benötigter Infos durch die Instrumentenbauer (Fotos, Texte, Informationen)
- Erstellung zweier Plattformen
 - Ursprünglich nur eine geplant
 - benötigt, da beide Unternehmen eigene (unterschiedliche) Ansprüche
 - Verwendung von Open Source Software (xtCommerce, MySQL)
 - Schnittstelle zu DV-Systemen

Workshops

- Voraussichtlich Sommer 2006
- Für die Musikinstrumentenbauer der Region Vogtland
- Präsentation
 - der Plattformen
 - Auswertung des Aufwandes & Nutzen
 - Vor- und Nachteile
- Verwendung der Modellplattformen (ganz oder Module)