Tutorial on
Schema and Ontology Matching

Pavel Shvaiko
pavel@dit.unitn.it

Jérôme Euzenat
Jerome.Euzenat@inrialpes.fr
Goals of the tutorial

- Illustrate the role of schema/ontology matching
- Provide an overview of the basic matching techniques
- Demonstrate the use of basic matching techniques in state of the art systems
- Motivate the future research
Outline

- Matching problem
- Classification of schema-based matching techniques
- Basic techniques
- Matching process
- Review of the matching systems
- Conclusions
Motivations

Match operator

takes two schemas/ontologies, each consisting of a set of discrete entities (e.g., tables, XML elements, classes, properties) as input and determines as output the relationships (e.g., equivalence, subsumption) holding between these entities
Motivations

Two XML schemas

← Equivalence  → Generality  → Disjointness
Motivations

Two relational schemas

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>Tel_No</th>
<th>e-mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Dow</td>
<td>12 Well St, Glasgow</td>
<td>0141-123-4567</td>
<td><a href="mailto:john@aol.com">john@aol.com</a></td>
</tr>
<tr>
<td>Mike O' Neill</td>
<td>37 Achray St, Glasgow</td>
<td>0141-987-6543</td>
<td><a href="mailto:mike@aol.com">mike@aol.com</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FirstName</th>
<th>LastName</th>
<th>Address</th>
<th>Telephone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Karen</td>
<td>Shaw</td>
<td>31 High St, London</td>
<td>0171-456-9876</td>
</tr>
<tr>
<td>Tina</td>
<td>Craig</td>
<td>12 Argyll St, London</td>
<td>0171-664-5138</td>
</tr>
</tbody>
</table>

Equivalence  Generality  Disjointness
Motivations

Two ontologies

- Reference
  - date
  - creator
  - title
  - Book
    - publisher
    - series
    - edition
    - Monograph
    - Proceedings

- Entry
  - year
  - author
  - title
  - Book
    - Conference
    - Proceedings

Equivalence

Generality

Disjointness
Schema matching vs. Ontology matching

Differences:

- Schemas often do not provide explicit semantics for their data
  - Relational schemas provide no generalization

- Ontologies are logical systems that constrain the meaning
  - Ontology definitions as a set of logical axioms
Schema matching vs. Ontology matching

Commonalities:

- Schemas and ontologies provide a vocabulary of terms that describes a domain of interest.
- Schemas and ontologies constrain the meaning of terms used in the vocabulary.

Techniques developed for both problems are of mutual benefit.
Statement of the problem

Scope

- Reducing heterogeneity can be performed in 2 steps
  - Determine the alignment (*matching*)
  - Process the alignment (merging, transforming, etc.)

- When do we match?
  - Design time
  - Run time
Statement of the problem

Mapping element $M$ is a 5-uple: $\langle id, e, e', R, n \rangle$

- $id$ is a unique identifier of the given mapping element
- $e$ and $e'$ are entities (e.g., XML elements, classes)
- $R$ is a relation (e.g., equivalence ($=$); more general ($\supseteq$); disjointness ($\perp$))
- $n$ is a confidence measure in some mathematical structure (typically in the $[0,1]$ range)
Alignment ($A$) is a set of mapping elements depending on the two schema/ontologies with some multiplicity: 1-1, 1-*, etc. and some other properties (complete).
Statement of the problem

Matching process

$O$ $r$ (e.g., thesauri)

$A$ $p$ (e.g., weights)

Matching process

$A'$
Application domains

- **Traditional**
  - Schema integration
  - Data warehouses
  - Mediator generation

- **Emergent**
  - P2P databases
  - Agent communication
  - Web services integration
Application domains

Schema integration: catalog matching
Application domains

Schema integration: catalog matching

- In order for a private company to participate in the marketplace (e.g., eBay), it has to determine correspondences between entries of its catalogs and entries of a common catalog of a marketplace.

- Once the correspondences between two schemas have been determined, the next step is to generate query expressions that automatically translate data instances of these catalogs under an integrated catalog.

- Having aligned the catalogs, users of a marketplace have a unified access to the products which are on sale.
Application domains

P2P databases

- Peers are autonomous
  - They appear and disappear on the network
  - They use different terminology

- Matching (on-the-fly)
  - Determine the relationships between peer schemas
  - Use these relationships for query answering
  - An assumption that all peers rely on one global schema, as in data integration, can not be made, because the global schema might need to be updated any time the system evolves
Application domains

Agent communication

Matching

Generating

Message

T
Application domains

Web services integration

Functional level:

Process level:

WSDL

BPEL4WS
Application domains

Web services integration

- Matching

Discovery goal  \[\text{solution}\]  Web Service

\[\begin{array}{c}
\text{inputs} \\
\text{preconditions} \\
\hline
\text{outputs} \\
\text{effects}
\end{array}\]  \[\begin{array}{c}
\subseteq \\
\subseteq \\
\hline
\text{inputs} \\
\text{preconditions} \\
\text{outputs} \\
\text{effects}
\end{array}\]

- Executing the alignment
  
  Generate a mediator able to transform the output of the first service in order to be input to the second one
Outline

- Matching problem
- Classification of schema-based matching techniques
  - Basic techniques
  - Matching process
  - Review of the matching systems
- Conclusions
Matching dimensions

- **Input dimensions**
  - Underlying data models (e.g., XML, OWL)
  - **Schema-level** vs. Instance-level

- **Process dimensions**
  - Approximate vs. Exact
  - Interpretation of the input

- **Output dimensions**
  - Cardinality (e.g., 1:1, 1:m)
  - Equivalence vs. Diverse relations (e.g., subsumption)
  - Graded vs. Absolute confidence
Classification of schema-based techniques

Three layers

- **The upper layer**
  - Granularity of match
  - Interpretation of the input information

- **The middle layer** represents classes of elementary (basic) matching techniques

- **The lower layer** is based on the kind of input which is used by elementary matching techniques
Classification of schema-based techniques

Element-level

Syntactic

- String-based
  - Name similarity
  - Description similarity
  - Global namespaces

- Language-based
  - Tokenization
  - Lemmatization
  - Morphological analysis
  - Elimination

- Linguistic resource
  - Lexicons
  - Thesauri

- Constraint-based
  - Type similarity
  - Key properties

- Alignment reuse
  - Entire schema/ontology
  - Fragments

- Upper level (logic-based) ontologies
  - SUMO, DOLCE

- Graph-based
  - Graph matching
  - Paths
  - Children
  - Leaves

- Taxonomy-based
  - Taxonomic structure

- Repository of structures
  - Structure’s metadata

- Model-based
  - Propositional SAT
  - DL-based

Structure-level

Syntactic

- External

- String-based

- Language-based

- Linguistic resource

- Constraint-based

- Alignment reuse

- Upper level (logic-based) ontologies

- Graph-based

- Taxonomy-based

- Repository of structures

- Model-based

External

Semantic

Relational

Structural

Terminological

Linguistic

Internal

Granularity / Input Interpretation Layer

Basic Techniques Layer

Kind of Input Layer

ESWC'05 – 29.05.2005 – p. 24/71
Outline

- Matching problem
- Classification of schema-based matching techniques
- Basic techniques
- Matching process
- Review of the matching systems
- Conclusions
Basic techniques

... techniques from the following systems have been taken into consideration:

- Anchor-PROMPT
- Artemis
- COMA, COMA++
- Cupid
- NOM, QOM, FOAM
- OLA
- SF, Rondo
- CtxMatch, S-Match
Element-level techniques

String-based (e.g., COMA, SF, S-Match, OLA)

- **Prefix**
  - It takes as input two strings and checks whether the first string starts with the second one
  - net = network; but also hot = hotel

- **Suffix**
  - It takes as input two strings and checks whether the first string ends with the second one
  - phone = telephone; but also word = sword
Element-level techniques

String-based (e.g., S-Match, OLA, Anchor-Prompt)

- **Edit distance**
  - It takes as input two strings and calculates the number of *insertions*, *deletions*, and *substitutions* of characters required to transform one string into another, normalized by $\max(\text{length}(\text{string1}), \text{length}(\text{string2}))$.
  - $\text{EditDistance}(\text{NKN}, \text{Nikon}) = 0.4$
Element-level techniques

**String-based** (e.g., COMA, S-Match)

- **N-gram**
  - It takes as input two strings and calculates the number of the same n-grams (i.e., sequences of $n$ characters) between them
  - *trigram(3)* for the string *nikon* are *nik, iko, kon*
Element-level techniques

Language-based (e.g., COMA, Cupid, S-Match, OLA)

- **Tokenization**
  - Names are parsed into tokens by recognizing punctuation, cases
  - Hands-Free_Kits → ⟨ hands, free, kits ⟩

- **Lemmatization**
  - Tokens are morphologically analyzed in order to find all their possible basic forms
  - Kits → Kit
Element-level techniques

Language-based (e.g., Cupid, S-Match)

- **Elimination**
  - Tokens that are articles, prepositions, conjunctions, and so on, are marked to be discarded
  - a, the, by, type of
Element-level techniques

Constraint-based (e.g., OLA, COMA)

- **Datatype comparison**
  - `integer < real`
  - `{a, c, g, t}[1 − 10] < {a, c, g, u, t} +`

- **Multiplicity comparison**
  - `[1 1] < [0 10]`
Element-level techniques

Linguistic resources (e.g., Artemis, S-Match, OLA)

- Sense-based: WordNet
  
  Relations between schema/ontology entities can be computed in terms of lexical relationships
Element-level techniques

Linguistic resources (e.g., Artemis, S-Match)

- Sense-based: WordNet
  - $A \sqsubset B$ if $A$ is a hyponym or meronym of $B$
  - Brand $\sqsubset$ Name
  - $A \sqsupset B$ if $A$ is a hypernym or holonym of $B$
  - Europe $\sqsupset$ Greece
  - $A = B$ if they are synonyms
  - Quantity = Amount
  - $A \perp B$ if they are antonyms or the siblings in the part of hierarchy
  - Microprocessors $\perp$ PC_Board
Element-level techniques

**Linguistic resources** (e.g., S-Match)

- **Sense-based: WordNet hierarchy distance**
  - These return the equivalence relation if the distance between two input senses in the WordNet hierarchy is less than a given threshold
  - $\text{red} = \text{pink}$

![Diagram]

- chromatic color
  - red
  - pink
Element-level techniques

Linguistic resources (e.g., S-Match)

- Gloss-based: WordNet gloss comparison
  - The number of the same words occurring in both input glosses increases the similarity value. The equivalence relation is returned if the resulting similarity value exceeds a given threshold
  - Maltese dog is a breed of toy dogs having a long straight silky white coat
  - Afghan hound is a tall graceful breed of hound with a long silky coat
Element-level techniques

Linguistic resources (e.g., Cupid, COMA)

- Specific thesauri
  - These usually store specific domain knowledge
  - \( PO = \text{Purchase Order} \)
  - \( \text{uom} = \text{UnitOfMeasure} \)
  - \( \text{line} = \text{item} \)
Element-level techniques

Alignment reuse (e.g., COMA, COMA++, OLA)

- Entire schemas
- Schema fragments

...we need to match schema/ontology $o'$ and $o''$, given the alignments between $o$ and $o'$, and between $o$ and $o''$ from the external resource, storing previous match operations results
Structure-level techniques

Taxonomy-based (Anchor-Prompt, NOM, QOM)

... schemas/ontologies are viewed as graph-like structures containing terms and their inter-relationships

- **Bounded path matching**
  - These take two paths with links between classes defined by the hierarchical relations, compare terms and their positions along these paths, and identify similar terms

- **Super(sub)-concepts rules**
  - If super-concepts are the same, the actual concepts are similar to each other
Structure-level techniques

Taxonomy-based

**Upward cotopic distance**
Measures the ratio of common superclasses.

\[
\delta(c, c') = 1 - \frac{|UC(c, H) \cap UC(c', H)|}{|UC(c, H) \cup UC(c', H)|}
\]

where \( UC(c, H) = \{c' \in H; c \leq c'\} \) is the set of superclasses of \( c \).

\[
\begin{align*}
\delta(a, a) &= 1 - 1 = 0 \\
\delta(a, e) &= 1 - 3/5 = .4 \\
\delta(a, f) &= 1 - 2/5 = .6 \\
\delta(d, a) &= 1 - 3/8 \approx .625 \\
\delta(b, c) &= 1 - 5/7 \approx .286 \\
\delta(c, d) &= 1 - 4/8 = .5 \\
\delta(a, b) &= 1 - 3/8 \approx .625
\end{align*}
\]
Structure-level techniques

Graph-based (e.g., Cupid, COMA)

- Children
  - Two non-leaf schema elements are structurally similar if their immediate children sets are highly similar

- Leaves
  - Two non-leaf schema elements are structurally similar if their leaf sets are highly similar, even if their immediate children are not
Structure-level techniques

Graph-based (e.g., Cupid, COMA)

Leaves
Structure-level techniques

Graph-based (e.g., SF, OLA)

- Iterative fix point computation
  - If two nodes from two schemas/ontologies are similar, their neighbors might also be somehow similar
**Structure-level techniques**

**Graph-based** (e.g., SF, OLA)

- Iterative fix point computation

\[
\sigma_C(c, c') = 0.6 \cdot \frac{1}{\max(|A(c)|, |A(c')|)} \cdot \sum_{\langle a, a' \rangle \in \text{match}(A(c), A(c'))} \sigma_A(a, a') + 0.4 \cdot \sigma(N(c), N(c'))
\]

\[
\sigma_A(a, a') = 0.6 \cdot \sigma_C(\text{domain}(a), \text{domain}(a')) + 0.4 \cdot \sigma(N(a), N(a'))
\]
Structure-level techniques

**Graph-based** (e.g., SF, OLA)

- Iterative fix point computation

\[
\sigma_C(c, c') = 6 \cdot \frac{1}{\max(|A(c)|, |A(c')|)} \sum_{(a, a') \in \text{match}(A(c), A(c'))} \sigma_A(a, a') + 4 \cdot \sigma(N(c), N(c'))
\]

\[
\sigma_A(a, a') = 6 \cdot \sigma_C(\text{domain}(a), \text{domain}(a')) + 4 \cdot \sigma(N(a), N(a'))
\]
Structure-level techniques

Graph-based (e.g., SF, OLA)

Iterative fix point computation

\[
\sigma_C(c, c') = 0.6 \frac{1}{\max(|A(c)|, |A(c')|)} \sum_{\langle a, a' \rangle \in \text{match}(A(c), A(c'))} \sigma_A(a, a') + 0.4 \sigma(N(c), N(c'))
\]

\[
\sigma_A(a, a') = 0.6 \sigma_C(\text{domain}(a), \text{domain}(a')) + 0.4 \sigma(N(a), N(a'))
\]

<table>
<thead>
<tr>
<th></th>
<th>(C_1)</th>
<th>(p)</th>
<th>(C_2)</th>
<th>(q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C'_1)</td>
<td>.57</td>
<td>.47</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(p')</td>
<td>.64</td>
<td>.27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C'_2)</td>
<td>.51</td>
<td>.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(q')</td>
<td>.38</td>
<td>.58</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Structure-level techniques

**Graph-based** (e.g., SF, OLA)

- Iterative fix point computation

$$
\sigma_C(c, c') = \frac{1}{\max(|A(c)|, |A(c')|)} \cdot \sum_{(a, a') \in \text{match}(A(c), A(c'))} \sigma_A(a, a') + 4.\sigma(N(c), N(c'))
$$

$$
\sigma_A(a, a') = 6.\sigma_C(\text{domain}(a), \text{domain}(a')) + 4.\sigma(N(a), N(a'))
$$

<table>
<thead>
<tr>
<th></th>
<th>$C_1$</th>
<th>$p$</th>
<th>$C_2$</th>
<th>$q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1'$</td>
<td>.54</td>
<td></td>
<td>.4</td>
<td></td>
</tr>
<tr>
<td>$p'$</td>
<td>.62</td>
<td></td>
<td>.39</td>
<td></td>
</tr>
<tr>
<td>$C_2'$</td>
<td>.43</td>
<td>.59</td>
<td></td>
<td>.54</td>
</tr>
<tr>
<td>$q'$</td>
<td>.44</td>
<td></td>
<td>.54</td>
<td></td>
</tr>
</tbody>
</table>
Structure-level techniques

Graph-based (e.g., SF, OLA)

- Iterative fix point computation

\[
\sigma_C(c, c') = 0.6 \cdot \frac{1}{\max(|A(c)|, |A(c')|)} \cdot \sum_{(a, a') \in \text{match}(A(c), A(c'))} \sigma_A(a, a') + 0.4 \cdot \sigma(N(c), N(c'))
\]

\[
\sigma_A(a, a') = 0.6 \cdot \sigma_C(\text{domain}(a), \text{domain}(a')) + 0.4 \cdot \sigma(N(a), N(a'))
\]

<table>
<thead>
<tr>
<th></th>
<th>( C_1 )</th>
<th>( p )</th>
<th>( C_2 )</th>
<th>( q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C'_1 )</td>
<td>.53</td>
<td>.47</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p' )</td>
<td>.67</td>
<td></td>
<td>.34</td>
<td></td>
</tr>
<tr>
<td>( C'_2 )</td>
<td>.46</td>
<td>.56</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( q' )</td>
<td>.4</td>
<td>.52</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Threshold reached: no .1 variation
Structure-level techniques

Model-based (e.g., CtxMatch, S-Match)

- Propositional satisfiability (SAT)
  - Decompose the graph (tree) matching problem into the set of node matching problems
  - Translate each node matching problem, namely pairs of nodes with possible relations between them, into a propositional formula
  - Check the propositional formula for validity
Structure-level techniques

**Model-based** (e.g., CtxMatch, S-Match)

- Propositional satisfiability (SAT)

\[ Axioms \rightarrow \text{rel}(context_1, context_2) \]

![Diagram showing axioms and contexts](image)
Structure-level techniques

Model-based

Description Logics (DL)-based

- micro-company = company
  - \( \sqsubseteq 5 \) employee

- SME = firm
  - \( \sqsubseteq 10 \) associate

company = firm ; associate \( \sqsubseteq \) employee

micro-company \( \sqsubseteq \) SME
Outline

- Matching problem
- Classification of schema-based matching techniques
- Basic techniques
- **Matching process**
- Review of the matching systems
- Conclusions
Matching process

Architectural perspective: Sequential (hybrid) (e.g., Cupid, Artemis)
Matching process

Architectural perspective: Parallel (composite) (e.g., COMA, QOM)
Matching process

Architectural perspective: Parallel (composite) (e.g., COMA, QOM)

Aggregation (e.g., Min, Max, Weighted, Average)
Matching process

User-centric perspective

- **Alignments as solutions** (e.g., Rondo, OLA)
  These consider the matching problem as an optimization problem and the alignment is a solution to it

- **Alignments as theorems** (e.g., S-Match)
  These rely on semantics and require the alignment to satisfy it

- **Alignments as likeness clues** (e.g., Cupid)
  These produce only reasonable indications to a user for selecting the alignment
Matching process

Selecting the final alignment

- Ranking strategies
  - Thresholds
  - MaxDelta

- Cardinalities
  - 1-1; 1-*; *-*

- Directionality
  - $O \rightarrow O'$; $O' \rightarrow O$ (SmallLarge, LargeSmall)
  - $O \rightarrow O'$ and $O' \rightarrow O$ (Both)
Outline

- Matching problem
- Classification of schema-based matching techniques
- Basic techniques
- Matching process
- Review of the matching systems
- Conclusions
Review of the matching systems

Some state of the art systems

- **Cupid** (Microsoft Research, USA)
- **FOAM/QOM** (University of Karlsruhe, Germany)
- **OLA** (INRIA Rhône-Alpes/Université de Montréal, France/Canada)
- **S-Match** (University of Trento, Italy)
- ...
Review of the matching systems

Cupid

- Schema-based
- Computes *similarity coefficients* in the [0,1] range
- Performs *linguistic and structure* matching
- Sequential system
- Alignments as likeness clues
Review of the matching systems

Cupid

Schema 1
Linguistic Matching
Thesaurus
Schema 2
Lsim
Structure Matching
Ssim
Weighting
Alignment
Generating alignment
Wsim
Review of the matching systems

OLA

- Schema- and Instance-based
- Computes dissimilarities + extracts alignments (equivalences in the [0,1] range)
- Based on terminological (including linguistic) and structural (internal and relational) distances
- Neither sequential nor parallel
- Alignments as solutions (to an optimization problem)
Review of the matching systems

QOM/FOAM

- Schema- and Instance-based
- Computes similarities + extracts alignments (equivalences in the [0,1] range)
- Based on terminological (including linguistic) and structural (internal and relational) distances
- Parallel with elaborated aggregation
- Alignments as likeness clues
Review of the matching systems

OLA

Create distance equations

Alignment extraction

Iterative equation resolution
Review of the matching systems

OLA

Create distance equations

Alignment extraction

Iterative equation resolution

ESWC'05 – 29.05.2005 – p. 59/71
Review of the matching systems

S-Match

- Schema-based
- Computes equivalence (\(=\)); more general (\(\supseteq\)); less general (\(\subsetneq\)); disjointness (\(\perp\))
- Analyzes the meaning (concepts, not labels) which is codified in the elements and the structures of schemas/ontologies
- Sequential system with a "composition" at the element level
- Alignments as theorems
Review of the matching systems

S-Match

Input Schemas

Translator

Preprocessing

PTrees

Match Manager

Oracles

Element Level Semantic Matchers

SAT Solvers (JSAT, SAT4J)

Alignment
## Review of the matching systems

### Analytical comparison

<table>
<thead>
<tr>
<th></th>
<th>SF</th>
<th>Artemis</th>
<th>Cupid</th>
<th>COMA</th>
<th>NOM</th>
<th>Anchor-PROMPT</th>
<th>OLA</th>
<th>S-Match</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Element-level</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntactic</td>
<td>string-based (2);</td>
<td>domain compatibility;</td>
<td>language-based (2);</td>
<td>string-based (4);</td>
<td>string-based (1);</td>
<td>string-based (3);</td>
<td>string-based (6);</td>
<td></td>
</tr>
<tr>
<td></td>
<td>data types;</td>
<td>language-based (1);</td>
<td>data types;</td>
<td>language-based (1);</td>
<td>domains and</td>
<td>data types;</td>
<td>language-based (2);</td>
<td></td>
</tr>
<tr>
<td></td>
<td>key properties</td>
<td>key properties</td>
<td></td>
<td>data types</td>
<td>ranges</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>External</td>
<td>common thesaurus (CT);</td>
<td>synonyms, broader terms, related terms</td>
<td>auxiliary thesaurus (synonyms, hypernyms, hyponyms, abbreviations); alignment reuse (2)</td>
<td>application-specific vocabulary</td>
<td>-</td>
<td>WordNet(1)</td>
<td>WordNet: sense-based (2), gloss-based (6)</td>
<td></td>
</tr>
<tr>
<td><strong>Structure-level</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntactic</td>
<td>iterative fix-point computation</td>
<td>matching of neighbors via CT</td>
<td>tree matching weighted by leaves</td>
<td>DAG (tree) matching with a bias towards leaf or children structures (2); paths</td>
<td>matching of neighbors (2); taxonomic structure (4)</td>
<td>bounded paths matching (arbitrary links); bounded paths matching (processing is-e links separately)</td>
<td>iterative fix-point computation; matching of neighbors; taxonomic structure</td>
<td></td>
</tr>
<tr>
<td>Semantic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Outline

- Matching problem
- Classification of schema-based matching techniques
- Basic techniques
- Matching process
- Review of the matching systems
- Conclusions
Conclusions

Summary

- We have discussed the schema/ontology matching problem and its application domains
- We have provided classificatory elements for approaching schema/ontology matching techniques
- We have presented a number of basic matching techniques as well as different strategies for building the matching process
- We have reviewed and compared (analytically) some existing matching systems
Conclusions

Uses of classifications

- They provide a common conceptual basis, and hence, can be used for comparing (analytically) different existing schema/ontology matching systems.
- They can help in designing a new matching system, or an elementary matcher, taking advantages of state of the art solutions.
- They can help in designing systematic benchmarks, e.g., by discarding features one by one from schemas/ontologies, namely, what class of basic techniques deals with what feature.
Conclusions

Research Challenges

- **Industry-strength** schema/ontology matching
- Scalability
- Interactive approaches
- **Infrastructures** (e.g., Rondo, Chimaera)
  - Representing the alignment
  - Executing the alignment
  - Explaining the alignment
Conclusions

Research Challenges

- Matching web services at the process level
- Lightweight ontology matching and emerging semantics
- Automatic partial alignment
Conclusions

Research Challenges

Evaluation

- Testbed environment
  - Series of tests, each with a pre-defined problem
  - Real-world case studies

- More accurate evaluation measures

- Adequacy task / measure

- Testing methodology which is able to estimate quality of the alignment between schemas/ontologies with thousands of entities
Questions?
Acknowledgments

We thank all the participants of the Heterogeneity workpackage of the Knowledge Web network of excellence. In particular, we are grateful to T.-L. Bach, J. Barrasa, P. Bouquet, J. Bo, R. Dieng-Kuntz, M. Ehrig, E. Franconi, R. García Castro, F. Giunchiglia, M. Hauswirth, P. Hitzler, M. Jarrar, M. Krötzsch, R. Lara, D. Maynard, A. Napoli, L. Serafini, G. Stamou, H. Stuckenschmidt, Y. Sure, S. Tessaris, P. Traverso, P. Valchev, S. van Acker, M. Yatskevich, and I. Zaihrayeu for their support and insightful comments.
Thank You
for your attention and interest!
The ESWC’05 Tutorial on
Schema and Ontology Matching

BIBLIOGRAPHY

Pavel Shvaiko\(^1\) and Jérôme Euzenat\(^2\)

\(^1\) University of Trento, Povo, Trento, Italy, pavel@dit.unitn.it
\(^2\) INRIA, Rhône-Alpes, France, Jerome.Euzenat@inrialpes.fr

1 Surveys

Good surveys through the recent years are provided in [18, 24, 36, 40, 44, 47, 49].
Major contributions of the last decades were presented in [2, 26, 27, 43].

2 Schema-based matching systems

<table>
<thead>
<tr>
<th>Name</th>
<th>Publications</th>
<th>Project web-site</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artemis</td>
<td>[3, 4, 9]</td>
<td></td>
</tr>
<tr>
<td>CtxMatch</td>
<td>[7, 8]</td>
<td></td>
</tr>
<tr>
<td>Cupid</td>
<td>[29]</td>
<td></td>
</tr>
<tr>
<td>Similarity Flooding (SF)</td>
<td>[32, 33]</td>
<td><a href="http://www-db.stanford.edu/~melnik/mm/sfa/">http://www-db.stanford.edu/~melnik/mm/sfa/</a></td>
</tr>
</tbody>
</table>

3 Infrastructures

<table>
<thead>
<tr>
<th>Name</th>
<th>Publications</th>
<th>Project web-site</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protoplasm</td>
<td>[5]</td>
<td>-</td>
</tr>
</tbody>
</table>

4 Further Readings

– Instance-based Matching: [10, 13, 25];
– **Languages for the alignment representation:** [6, 42];
– **Executing the Alignment:** [28, 31, 48, 50];
– **Explaining the Alignment:** [10, 45];
– **Evaluation:** [12, 17, 46].

**Acknowledgments:** This work has been partly supported by the European Knowledge Web network of excellence (IST-2004-507482).

**References**


