A survey of approaches to **automatic schema matching**

E. Rahm, and P. A. Bernstein.

The VLDB Journal, 10(3): 334-350, 2001.

Presented by Joel So (j2so@cs.uwaterloo.ca)

Topics to be covered

- Motivation: match application domains
- Generic match operator
- Generic match taxonomy
- Combining matchers
- 7 surveyed prototypes
- 5 related prototypes
- Conclusions

Motivation: application domains

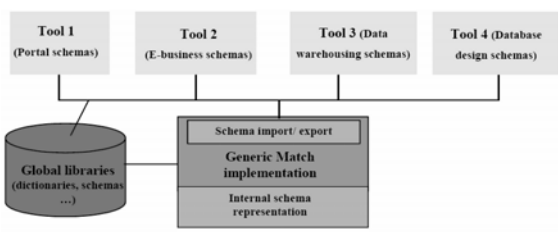
- Schema integration
  - Developing global view over set of independently developed schemas
- Data warehousing
  - Transforming data from source format to warehouse format
- E-commerce / B2B integration
  - Transforming between message types and trading partner formats
- Semantic query processing
  - Mapping user-specified query concepts to database schema elements

Generic match operator

- **IN:** 2 *schemas* S1 and S2
- **OUT:** *match result* – mapping of elements in S1 and S2
- **Schema:** set of elements connected by some structure
- **Match result:** set of mapping elements
- **Mapping element:** specification of elements in S1 that map to elements in S2 and a *mapping expression* specifying how they are related...
  - {elements in S1} ~= {elements in S2} : (mapping expression)

Generic match operator (cont'd)

- A generic matcher architecture:



Taxonomy: instance vs. schema

- Schema-level matchers
  - Consider schema information, not instance data
  - Mapping expressions on schema element name, description, type, constraints, structure, etc.
- Instance-level matchers
  - Characterize (using linguistic- or constraint-based techniques) contents of schema elements, mapping expressions on characterizations
  - Useful for semi-structured data, absence of schema information

Taxonomy: element vs. structure

- Element-level matchers (instance- or schema-level)
  - Match schema elements (attributes, fields/columns) in isolation *without* considering relative parent- or sub-structure
- Structure-level matchers (schema-level)
  - Match schema structures (sub-trees, tables), i.e. combinations of elements that form structures
  - Can have full or partial structural matches

Taxonomy: linguistic vs. constraint

- Linguistic matchers (element-level)
  - Consider semantic similarities in element names, descriptions, instance values
  - Examine equality, canonical extraction, synonyms, hypernyms, common forms, user-provided matches
- Constraint-based matchers (element- or structure-level)
  - Consider similarities in constraint information (cardinalities, relationships, data types, value constraints, etc.)

Taxonomy: matching cardinality

- Matching cardinality (1:1, 1:n, n:1, n:m) describe how (many) elements in S1 are mapped to elements in S2
- **Global cardinality:** defined across mapping elements
- **Local cardinality:** defined for an individual mapping element
- Cardinality may differ w.r.t. to structure-level match vs. element-level match perspectives

Taxonomy: auxiliary information

- Additional information (beyond schemas S1 and S2) used by match operator
- E.g., dictionaries, global schemas, previous mappings, user input, namespaces, etc.

Combining matchers

- Hybrid matchers
  - Integrate multiple matching criteria
  - Individual matchers synchronously contribute to final match result
- Composite matchers
  - Aggregate multiple matching criteria
  - Individual matchers output match results independently
  - Match results serially/subsequently combined automatically or manually (external to matchers)

7 surveyed prototypes

1. SemInt
2. LSD
3. SKAT
4. TransScm
5. DIKE
6. ARTEMIS & MOMIS
7. Cupid

SemInt (Northwestern Univ.)

- Supports up to 15 constraint-based and 5 content-based matching criteria
- Determines *match signature* and considers Euclidean distance between signatures
- Uses neural networks
- element-level matching, constraint-based schema-level matching, constraint-based instance-level matching, 1:1 global cardinality, hybrid matcher

LSD (Univ. of Washington)

- Multi-strategy machine-learning approach
- Training phase, matching phase
- Automatic (trained) composition of match results
- Highly extensible; incorporates user-supplied, domain-specific constraints
- element- and structure-level matching, linguistic-based schema-level matching, constraint-based structure-level matching, linguistic- and constraint-based instance-level matching,1:1 global cardinality, training results and domain-specific constraints as auxiliary input, composite matcher

SKAT (Stanford Univ.)

- Rule-based, semi-automatic
- First-order logic rules express match and mismatch relationships
- Intended for ontology matching, matching based heavily on “is-a” relationships
- element- and structure-level matching, linguistic- and constraint-based schema-level matching, constraint-based structure-level matching, 1:1 and n:1 global cardinality, general matching rules as auxiliary input, hybrid matcher

TransScm (Tel Aviv Univ.)

- Automatic data translation between schema instances
- Schemas internally represented as labeled graphs
- Rule-based matchers
- element- and structure-level matching, linguistic- and constraint-based schema-level matching, constraint-based structure-level matching, linguistic- and constraint-based instance-level matching,1:1 global cardinality, hybrid matcher

DIKE (Univ. of {Reggio Calabria, Calabria})

- System to automatically determine synonym, homonym, is-a, and hypernym relationships between objects in E-R schemas
- Uses schema matching techniques to determine similarities between objects
- element- and structure-level matching, linguistic- and constraint-based schema-level matching, constraint-based structure-level matching, 1:1 global cardinality, synonyms and inclusion definitions as auxiliary input, hybrid matcher

ARTEMIS (Univ. of {Milano, Brescia}) & MOMIS (Univ. of Modena & Reggio Emilia)

- ARTEMIS clusters schema attributes based on “affinities” (determined using schema matching techniques)
- MOMIS is a database mediator, must integrate independently developed schemas into virtual global schema
- element- and structure-level matching, linguistic- and constraint-based schema-level matching, constraint-based structure-level matching, 1:1 global cardinality, thesauri as auxiliary input, hybrid matcher

Cupid (Microsoft Research)

- Generic schema matcher, prototype applied to XML and relational schemas
- 3-phase algorithm:
  - Linguistic processing
  - Structural processing
  - Evaluate weighted mean of similarity coefficients and determine mapping
- element- and structure-level matching, linguistic- and constraint-based schema-level matching, constraint-based structure-level matching, 1:1 and n:1 global cardinality, thesauri and glossaries as auxiliary input, hybrid matcher

5 related prototypes

1. Clio
   - Semi-automatic schema matching
   - Schema Readers and Correspondence Engine (CE)
   - CE uses mapping knowledge-base and user input (via GUI tool)
2. Similarity flooding
   - Graph matching algorithm applied to schema matching
   - Convert schemas into directed labeled graphs, exploit structural similarities between resulting graphs
3. Delta
   - Exploits (exclusively) textual/semantic similarities it attribute metadata
   - Converts attribute metadata into simple text string, matchings found by way of text search

5 related prototypes (cont’d)

4. Tess
   - Focuses on mapping between schema evolution, therefore, high degree of similarity between matched schemas can be assumed
   - Identifies top-level match candidates, then recursively matches sub-structure(s)
5. Tree matching
   - Algorithm for finding mappings between two labeled trees (purely structural)
   - Can be applied to schema matching (obviously)
   - Simple linguistic rules can be incorporated using “rename” transformation operator

Conclusions

- Schema matching is a basic problem in many database applications
- Schema matching taxonomy:
  - Schema- and instance-level
  - Element- and structure-level
  - Linguistic- and constraint-based
  - Matching cardinality, Auxiliary information
- Hybrid and composite matchers
- Wishes from the authors: continued study of schema matching as an independent/generic problem (agnostic of domain/application), quantitative comparison of approaches (performance, accuracy)