A Survey of Approaches to Automatic Schema Matching

Name: Samer Samarah
Number: 3740535
Email: SSamarah@Site.Uottawa.Ca

This Presentation based on the following paper:

Presentation Outlines

- Definition of Schema Matching.
- Schema Matching problem.
- Applications of Schema Matching.
- Schema matching approaches.
- Personal contribution.
- Conclusion.
Schema Matching: Definition

- Schema Matching: is the process of finding semantic correspondence between elements of two schemas.
- Schema Matching is achieved through \textit{Match} operation.
- Match Operator is a function that takes two schemas as input and returns a mapping between those two schemas as output, called the match result.

\[ \text{Match}(S_1, S_2) \Rightarrow \text{Match Result.} \]
The Match operator

\[ \text{Match}(S_1, S_2) \rightarrow \text{Match Result} \]

- The schema (either S1 or S2) defined to be a set of elements connected by some structure (ER model, OO model,..).
- The Match Result is a set of mapping elements, each of which indicate that certain elements of S1 are mapped to certain elements of S2, expressed by \((\cong)\).
- A mapping expressions, which specifies how the S1 and S2 elements are related, may be associated with the mapping elements.
The Match Operator

Example:

Suppose S1 and S2 are two schemas contain the following elements:

<table>
<thead>
<tr>
<th>S1 Elements</th>
<th>S2 Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cust (Relation)</td>
<td>Customer (Relation)</td>
</tr>
<tr>
<td>C#</td>
<td>CustID</td>
</tr>
<tr>
<td>CName</td>
<td>Company</td>
</tr>
<tr>
<td>FirstName</td>
<td>Contact</td>
</tr>
<tr>
<td>LastName</td>
<td>Phone</td>
</tr>
</tbody>
</table>

\[
\text{Match} \ (S1, S2) \Rightarrow \\
\{
\text{Cust.C#} \equiv \text{Customer.CustID}, \\
\text{Cust.CName} \equiv \text{Customer.Company}, \\
(\text{Cust.FirstName, Cust.LastName}) \equiv \text{Customer.Contact}
\}
\]

With the following mapping expressions:

Cust.C# = Customer.CustID
Cust.CName = Customer.Company
Concatenate (Cust.FirstName, Cust.LastName) = Customer.Contact
Schema Matching Problem

- Schema Matching currently performed manually which makes it:
  - Error-prone.
  - Tedious.
  - Time-consuming.

- So, the solution is to automate the match function, but
  - There is no mathematical model that capture the matching process.
  - Application dependent.
Schema Matching Applications

- Schema Integration.
- Data Warehouses.
- E-commerce.
- Semantic Query Processing.
- Data Integration system
1-Schema Integration

- Is the process of constricting a global view from a set of independently developed schema.

- Schema integration achieved through identifying the interschema relationships (applying schema matching) then unify these matched elements.
Example:

S1
{SName, ...}
S2
{FName, LName, ...}
Sn

Match + Unify

Sg
{Name, ...}

Determining the correspondence between SName and FName, Lname achieved through the match operation.
2-Data Warehouses

- DWH is a decision support database that is extracted from a set of data sources.
- DWH and sources represent data in different format (i.e. relations or XML versus multidimensional view)
- Constructing DWH require transform data from its format into DWH format.
- match operation can be used to identify those elements in the sources that are represented in the DWH, according to this mapping appropriate transformation can be designed.
3- E-commerce

- Trading partners exchange messages that describes business transactions.
- Each partners uses its own message format (EDI, XML,..) or different message schema.
- In order to exchange messages, there is a need to translate the messages to the format required by different partners (Matching problem).
4- Semantic Query Processing

- A run time scenarios where the user specifies the output of the query (in terms of concepts familiar to him, which may be not the same concepts presented in the DB {the Select Clause}), and the system figures out how to produce that output (i.e. the from and where clauses in SQL).

- The match operation is used to determent the mapping between the user concepts and the DB concepts).
4-Semantic Query Processing (Example)

Example

All employees earn more than 2000$ → Applying Match → Mapping elements → Producing New Query in SQL format: 
{Select FName, LName From Employee Where Salary >2000} → SQL Q → Output

Employee(FName,LName,Salary)
5- Data Integration System

- The major component of data integration system is the source description.
- Source description maps the sources schema to the mediated schema.
- Match operation applied in order to specify this mapping.
Generic Match Architecture

- Schemas to be matched are represented in a uniform internal representation.
- Importer translates input schemas from their native representation into the internal representation.
- Exporter translates the match result produced by the match from the internal representation into the representation required by each tool.
Generic Match Architecture

- Tool 1 (Portal Schemas)
- Tool 2 (E-business Schemas)
- Tool 3 (DWH Schemas)
- Tool 4 (DB Schemas)

Global Libraries (dictionaries, schemas,..)

Schema import / export

Generic Match Implementation

Internal Schema Representation
Classification of schema matching approaches

- **Instance Vs schema**: consider instance data or schema information.
- **Element Vs structure**: matching performed for individual schema element (attribute), or for combinations of elements (structure).
- **Language Vs constraint**: use linguistic information (names and textual description) or constraint information (key, relationship)
- **Matching cardinality**: the overall match result may relate one or more elements of one schema to one or more elements of the other (1:1, 1:n, m:n).
- **Auxiliary information**: the use of auxiliary information (dictionaries, previous matching results, user input,..)
Classification of schema matching approaches

Schema Matching Approaches

- Individual Matcher
  - Schema Based
    - Element Level: Linguistic Constraint
    - Structure Level: Constraint
  - Instance Based
    - Element Level: Constraint
- Combine Matcher
- Hybrid Matcher
- Composite Matcher
  - Manual
  - Automatic
Schema level matcher

- Consider only schema information, not instance data, such as:
  - Name
  - Description
  - Data Type
  - Relationship (is-a, part-of)
  - Constraints
  - Schema structure

- Multiple match candidates could be founded, each of which assigned with a similarity degree.
Granularity of match

- *Element level matching:* for each element in the first schema, determines the matching elements in the second schema.

- *Structural level matching:* matching combinations of elements that appear together in a structure.
  
  Could be fully match or partial match.
## Granularity of match

### Example

<table>
<thead>
<tr>
<th>S1 Elements</th>
<th>S2 Elements</th>
<th>Match Granularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>CustomerAddress</td>
<td>Address $\cong$ CustomerAddress (S.L full Match)</td>
</tr>
<tr>
<td>Street</td>
<td>City</td>
<td>Street $\cong$ City (E.L)</td>
</tr>
<tr>
<td>State</td>
<td>USState</td>
<td>State $\cong$ USState (E.L)</td>
</tr>
<tr>
<td>ZIP</td>
<td>PostalCode</td>
<td>Zip $\cong$ PostalCode (E.L)</td>
</tr>
<tr>
<td>AccountOwner</td>
<td>Customer</td>
<td>AccountOwner $\cong$ Customer (S.L partial Match)</td>
</tr>
<tr>
<td>Name</td>
<td>CName</td>
<td>Name $\cong$ CName (E.L)</td>
</tr>
<tr>
<td>Address</td>
<td>CAddress</td>
<td>Address $\cong$ CAddress (E.L)</td>
</tr>
<tr>
<td>Birthdate</td>
<td>CPhone</td>
<td></td>
</tr>
<tr>
<td>TaxExempt</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Linguistic approaches

- Linguistic matchers use names and text (word or sentence) to find semantically similar schema elements:
  - Name Matching
  - Description matching
Name Matching

- Name based matching matches schema elements with equal names or similar names.
  - Equality of names.
  - Equality of canonical name representation after stemming and preprocessing.
  - Equality of synonyms.
  - Equality of hypernyms (X is a hypernym of Y if Y is a kind of X).
  - Similarity based on common substring, edit distance, soundex.
  - User-provided name matches.
- Thesauri or dictionary should be exploited.
# Name Matching

Example: two schema S1, S2 represent two automobile suppliers

<table>
<thead>
<tr>
<th>S1 Elements</th>
<th>S2 Elements</th>
<th>Matching Based on</th>
</tr>
</thead>
<tbody>
<tr>
<td>CarID</td>
<td>TruckID</td>
<td>Car $\cong$ Truck (Hypernyms)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Car is an automobile and truck is an automobile.</td>
</tr>
<tr>
<td>Brand</td>
<td>Make</td>
<td>Brand $\cong$ Make (Synonyms)</td>
</tr>
<tr>
<td>Price</td>
<td>Price</td>
<td>Price $\cong$ Price (Equality of Names)</td>
</tr>
<tr>
<td>SoldTO</td>
<td>Sold2</td>
<td>SoldTo $\cong$ Sold2 (Soundex)</td>
</tr>
<tr>
<td>CAddress</td>
<td>CustomerAddress</td>
<td>CAddress $\cong$ CustomerAddress (Preprocessing)</td>
</tr>
</tbody>
</table>
Constraint-based approaches

- Exploit constraints information associated with the input schemas to determine the similarity of schema elements.
  - Data types and domain constraints
  - Key characteristics (primary, unique)
  - Relationship cardinality
  - Structural constraints such as foreign key (used by structural matches approaches)
### Constraint-based approaches

#### S1 Elements | S2 Elements | Matching
--- | --- | ---
Employee | Personal | 
**EmpNo** {int, PK} | **Pno** {int, PK} | Born \(\cong\) Birthdate {Type}
**EmpName** {String} | **Pname** {string} | Pno \(\cong\) EmpNo | DeptNo {Key}
Pname \(\cong\) DeptName {Type}
Pname \(\cong\) EmpName {Type}
Dept \(\cong\) DeptName {Type}
Dept \(\cong\) EmpName {Type}

DeptNo {int, ref dep} | Dept {String} | S2.Personal {Pno, Pname, Dept, born} \(\cong\) Select S1.Employee.EmpNo,
| | | S1.Employee.EmpName,
| | | S1.Department.DeptName,
| | | S1.Employee.BirthDate
salary {single} | Born {date} | From S1.Employee, S1.Department
BirthDate {date} | | Where (S1.Employee.DeptNo = S1.Department.DeptNo)

### Note:
Structural Matching

Several match candidate could results so this approach could be sued to limit the number of candidate.
Description Matching

- Based on linguistic evaluation for the comment associated with schema elements.

Example:

S1: empn // employee name
S2: name // name of employee

NL Understanding technology

Empn ≈ name
Reusing Schema and mapping information

- This approach supports and exploits the reuse of common schema components and previously determined mapping.
- Useful when matching applied to different but similar schemas to the same destination schemas.
Reusing Schema and mapping information

EX:

Schema S1
Purchase order
  Product
  BillTo
    Name
    Address
  ShipTo
    Name
    Address
  Contact
  ContactPhone

Schema S2
Purchase order
  Product
  BillTo
    Name
    Address
  ShipTo
    Name
    Address
  Contact
    Name
    Address

Schema S
  POrder
    Article
    Payee
    BillAddress
    Recipient
    ShipAddress

Goal: Mapping S1 to S
Matching result between S2 and S are previously determent and can be reused to map S1 to S
Match cardinality

- Global cardinality: how many mapping elements $S_1$ or $(S_2)$ elements can participate in the matching results.
- Local cardinality: how many elements in $S_1$ match how many elements in $S_2$ within individual mapping element.
- Most of approaches restricted to 1:1 local and 1:1 or 1:n global cardinality.
# Match cardinality

<table>
<thead>
<tr>
<th>Local Match Cardinality</th>
<th>S1 elements</th>
<th>S2 elements</th>
<th>Matching Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1:1 element level</td>
<td>Price</td>
<td>Amount</td>
<td>Amount = Price</td>
</tr>
<tr>
<td>2 n:1 element level</td>
<td>Price, Tax</td>
<td>Cost</td>
<td>Cost = Price* (1 + Tax/100)</td>
</tr>
<tr>
<td>3 1:n element level</td>
<td>Name</td>
<td>FirstName, LastName</td>
<td>FirstName, LastName = Extract(Name,..)</td>
</tr>
<tr>
<td>4 n:1 structure level (n:m element level)</td>
<td>B.Title, B.PuNo, P.Name</td>
<td>A.Book, A.Publisher</td>
<td>A.Book, A.Publisher = Select B.Title, P.Name From B.P Where B.PuNo = P.PuNo</td>
</tr>
</tbody>
</table>

Example:

Price has 1:n Global Cardinality
Instance level approaches

- Consider data contents.
- Useful when schema information limited (or no schema at all).
- Enhance schema matching by considering elements whose instances are more similar.
  - Linguistic approach based on IR techniques for text elements.
  - Constraint based such as value range and average for numeric element.
Instance level approach

Example:

<table>
<thead>
<tr>
<th>EmpNo</th>
<th>Dept</th>
<th>SSN</th>
<th>Works for</th>
</tr>
</thead>
<tbody>
<tr>
<td>234</td>
<td>Marketing</td>
<td>230</td>
<td>Accounting</td>
</tr>
<tr>
<td>235</td>
<td>Accounting</td>
<td>229</td>
<td>Marketing</td>
</tr>
<tr>
<td>236</td>
<td>Marketing</td>
<td>228</td>
<td>Marketing</td>
</tr>
</tbody>
</table>

{Dept ≈ works for} (based on “Marketing” Frequency)
{EmpNo ≈ SSN} (based on value range)
Combining matchers

- Combine several approaches to achieve good match candidates.
  - Hybrid Matcher: combine several matching approaches to determine match candidate based on multiple criteria (name, type constraints).
    - More effective (poor candidates filtered out early)
    - Better performance (reducing number of pass over the schema).
  - Composite matcher: combines the results of several independently executed matchers, including hybrid matchers.
    - More flexible than hybrid matchers, (allow us to select from set of matchers).
    - The combination of results could be automatic or manually.)
References

- DES system (Datalog Educational System) available at: http://www.fdi.ucm.es/professor/fernan/DES/.
- AnHai Doan, Pedro Domigos, Alon Levy, “Learning Source Description for Data Integration”, University of Washington, Seattle, WA 98195