Overview

- U-MAP is the first system that uses usage information buried in database query logs to tackle data integration and exchange challenges. U-MAP generates correspondences between the attributes of the source and target schemas, and then complex mapping rules to transform data records from one schema to another using these query logs.

Illustrative Example - Bookstores

(a) X-Schema

(b) Y-Schema

Basic Approach

1. Generating correspondences: Match schemas to generate the set of attribute correspondences (using any of a variety of techniques).
2. Constructing logical relations: Find all meaningful associations between schema attributes – using the chase algorithm.
3. Creating mappings: For each pair of source and target logical relations with corresponding attributes, create a new mapping.
4. Optimizing mappings: Eliminate redundant mappings, merge selected mappings together, and so on to ultimately get a “higher quality” set of mappings.
5. Compiling mappings: Generate an executable script (e.g., SQL queries) to transform source data records into target data records.

Mapping Example

<table>
<thead>
<tr>
<th>Mapping</th>
<th>X_Order_Line_LR</th>
<th>Y_Order_Line_YR</th>
</tr>
</thead>
<tbody>
<tr>
<td>contenu</td>
<td>X_Order_Line</td>
<td>Y_Order_Line</td>
</tr>
<tr>
<td>relation</td>
<td>X_Address</td>
<td>Y_Address</td>
</tr>
<tr>
<td>source</td>
<td>X_Bid</td>
<td>Y_Bid</td>
</tr>
<tr>
<td>logical relation</td>
<td>X_Address</td>
<td>Y_Address</td>
</tr>
<tr>
<td>target</td>
<td>X_Address</td>
<td>Y_Address</td>
</tr>
<tr>
<td>X_Address</td>
<td>X_Address</td>
<td>Y_Address</td>
</tr>
</tbody>
</table>

The U-MAP Approach

- Build upon existing approaches for schema matching and mapping.
- Analyze the query log to find guidance on how to handle some of the key unresolved issues in the existing approaches.

Generating Correspondences

- **Problem:**
  The sources of evidence currently used for schema matching (e.g., name similarity, structure similarity, data similarity, etc) are not always available or sufficiently reliable.
- **Solution:**
  Introduce a new source of evidence: usage info from query logs
  Collect co-occurrence statistics for pairs of attributes in the query log, and match the resulting co-occurrence graphs across the source and target.
  Track and compare the usage of attributes with aggregate functions and in join predicates

Managing IS-A Relationships

- **Problem:**
  Current tools cannot detect and handle overlapping sibling relations (subclasses of the same super class).
  Each sibling relation is mapped separately leading to duplicate records in the target.
- **Solution:**
  Detecting candidate sibling relations
  Check for shared attributes
  Detecting overlapping sibling relations
  Check for queries looking for the overlap
  Merging overlapping sibling relations
  Outer Join the two relations and merge their attributes

Aggressive Chase

- **Problem:**
  Current tools can miss interesting attribute associations because the chase operation is limited to the forward direction.
  Example: the association between the order line and the author name in the Y-Schema will be missed.
- **Solution:**
  Discover all meaningful pairs of opposite references by analyzing the query log.
  Allow chasing in the reverse direction as guided by the discovered pairs in the query log.

Resolving Correspondence Conflicts

- **Problem:**
  Current tools cannot automatically resolve the potential correspondence conflicts across the attributes of the source and target logical relations – during mapping generation.
  Existing solutions vary from to generating all possible resolutions to requesting the user to manually resolve the conflicts one attribute at a time.
- **Solution:**
  Treat the problem as a schema matching problem (across the two logical relations).
  Build upon the usage-based schema matching approach
  Collect the statistics separately for each version of the same attribute present in the logical relation.
  E.g., three versions of ad_city can represent billing, shipping, and customer cities.
  Match attribute contexts across the logical relation and the queries in the query log to achieve the separation in statistics collection.
  Group attributes to limit the number of possible resolutions.

U-MAP System

- Supported by the Qatar Computing Research Institute and the National Science Foundation under Grants No. IIS-0916614 and IIS-0811954.