Mapping Patterns [2]

A mapping pattern is a quadruple $(C, S, M, O)$, where $C$ is a conceptual schema, $S$ is a database schema, $M$ is a set of mappings, and $O$ is an ontology. In such mapping pattern, the pair $(C, S)$ puts into correspondence a conceptual representation to one of its (many) admissible (i.e., formally sound) database schemata. Such variants are due to differences in the applied methodology, efficiency and performance optimizations, and database space consumption. The database schema ontology ontology $O$ [3] is the OWL 2 QL encoding (hence, not lossless) of $C$, and the set $M$ of database schema mappings provides the link between $S$ and $O$.

Some Basic (Schema-driven) Patterns

<table>
<thead>
<tr>
<th>E-R Diagram</th>
<th>DB Schema</th>
<th>Mapping</th>
<th>Ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Diagram" /></td>
<td><img src="image2.png" alt="Diagram" /></td>
<td><img src="image3.png" alt="Diagram" /></td>
<td><img src="image4.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

**Mapping Patterns**

**The Algorithm**

In this analysis, we check how many mappings in the analyzed scenarios (Cordis and NPD) can be explained through mapping patterns.

**Coverage Analysis**

<table>
<thead>
<tr>
<th>Pattern</th>
<th># usages</th>
<th># mappings</th>
<th>Pattern</th>
<th># usages</th>
<th># mappings</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE</td>
<td>13</td>
<td>60</td>
<td>SE</td>
<td>61</td>
<td>454</td>
</tr>
<tr>
<td>SR</td>
<td>3</td>
<td>3</td>
<td>SRm</td>
<td>74</td>
<td>74</td>
</tr>
<tr>
<td>SRR</td>
<td>1</td>
<td>16</td>
<td>SRR</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>SH</td>
<td>3</td>
<td>132</td>
<td>SH</td>
<td>3</td>
<td>132</td>
</tr>
</tbody>
</table>

Covered Mappings: 89 out of 120
Covered Mappings: 672 out of 1173

**Mismatches Analysis**

We compare the classification returned by ADaMAP to a classification manually verified by a human expert. The table below refers to Cordis. Similar results were obtained for NPD.

**Conclusion and Future Work**

**Contributions**

- We have introduced ADaMAP, an algorithmic technique that extracts semantics from a relational data source, by automatically identifying how mapping patterns are applied to fragments of its schema.
- The patterns identified by ADaMAP provide a solid basis that can be manually improved by human experts.
- The validation of ADaMAP in two real-world case studies confirms that the identified patterns by-and-large agree with those detected by a human expert.

**Future Work**

- ADaMAP comes with some limitations that should be tackled:
  - For a given relational schema there are in general many possible combinations of mapping patterns that are, in principle, equally valid.
  - While ADaMAP only returns the "most typical" one.
  - ADaMAP ignores data, however "data-driven patterns" [2] are also important.
  - Especially in those scenarios where the DB schema is poorly structured or denormalized.

**References**