Conceptual schema matching with the Ontology Mapping Language: requirements and evaluation

- Marian de Vries / Thorsten Reitz
- AGILE workshop 2009
Overview

• About conceptual schema matching
• Requirements for a mapping language
• What is OML
• First tests
Data harmonisation

• Two kinds of issues: at general level, and at instance level

• At general level:
  Data model (conceptual schema)
  Spatial reference system
  Level-of-detail, scale / resolution
  (Data format)
  Terminology, semantics (meaning)
  Metadata profile
  Portrayal

• At instance level:
  Edge matching
  Solving conflation (doubles, etc.)
  Other data quality issues
Data harmonisation

• Two kinds of issues: at general level, and at instance level

• At general level:
  - Data model (conceptual schema)
  - Spatial reference system
  - Level-of-detail, scale / resolution
  - (Data format)
  - Terminology, semantics (meaning)
  - Metadata profile
  - Portrayal

• At instance level:
  - Edge matching
  - Solving conflation (doubles, etc.)
  - Other data quality issues
Transforming existing geodata to new schema, e.g. INSPIRE

Steps are

• Specifying (or re-using) target data model / schema
• Reverse-engineering (often) data model of existing data
• Definition of mapping rules between schema of source data and target schema
• ‘Execution’ of mapping rules in actual data transformation process
The diagram illustrates the HUMBOLDT Framework, which comprises a Technical Process and Target Definition.

**Technical Process**
- Source
- Processing
- Target

**Target Definition**
- Decision-based specifications
- Technical specifications
- National Data Sources on the same theme, e.g. parcels
- Applications based on different themes
- HUMBOLDT scenarios

The framework connects sources through processing steps to a target, with decision-based and technical specifications influencing the outcomes.

**INSPIRE**

The diagram also highlights the INSPIRE initiative, emphasizing the importance of national data sources and the potential for applications and scenarios.
On-the-fly, at server, in client, ...

- Several options for when, where and how of actual data transformation
  - On-the-fly during data retrieval
    - in mediator service?
    - in client?
  - Or beforehand by data provider
    - as migration step (physical copy)
    - or by configuration of e.g. WFS
- Different data formats, software platforms
[Conceptual Schema] Transformation

**Goal:**
- **Application expert** is able to specify the mapping
- **HUMBOLDT Software** is able to perform the processing automatically based on the mapping specification

**Processing chain**
(edge matching, coordinate transformation, etc.)
### Matching tables, spreadsheets

- **Matching table (input domain expert)**

<table>
<thead>
<tr>
<th>Feature class: er:RoadLink</th>
<th>Feature class: de:Ver01_L</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>er:id</code>/<code>er:permanentId</code></td>
<td>de:OB</td>
</tr>
<tr>
<td><code>er:RoadName</code></td>
<td>If de:OBJART in (3102, 3101)</td>
</tr>
<tr>
<td><code>er:roadName1</code></td>
<td>de:GN (when ‘NNMN’ then ‘Null or no value’)</td>
</tr>
<tr>
<td><code>er:roadName2</code></td>
<td>de:KN (when ‘NNMN’ then ‘Null or no value’)</td>
</tr>
<tr>
<td><code>er:alternativeName</code></td>
<td>de:ZN (If not null)</td>
</tr>
<tr>
<td><code>er:RoadSurface</code>/<code>er:roadSurface</code></td>
<td>If de:OBJART =3102 and de:BEF!=1000 then ‘unpaved’</td>
</tr>
</tbody>
</table>
Encoding schema mappings

• Must be a better way than spreadsheets or tables
• ETL tools, but for professionals
• HUMBOLDT: open source tools for data harmonisation,
• One focus point is: developing tool for conceptual schema matching
  (HALE, see Session 6 tomorrow)
• Schema mappings are then used in other components of the HUMBOLDT software for actual data transformation

• → we needed a language to encode the schema mappings
Requirements for mapping language

• Open, non-proprietary
• Generic, at conceptual level, not bound to specific implementation
• Declarative (preferably), because that fits activity by domain experts of Model 2 Model mapping
• Support for all needed aspects / types of schema translation = expressiveness
• Complete and unambiguous: no extra information is needed at runtime to do the actual data transformation
‘Must have’

• Rename

• Filter: definition of subsets based on attribute values or spatial operations (intersect, buffer, etc.)

• (Simple and complex) Functions
  - Concatenate, multiply by 10, covert from yards to meters, …
  - Geometric conversions (e.g. MultiCurve -> Curve)
  - Etc.

• Easy way to recode attribute values (form of reclassification)

• Set default values
Schema translation: lessons learnt

• Previous work, e.g. GiMoDiG, see Lehto 2007
• Our own tests
  1. Filtering: conditional statements applied to source data to filter features (extract sub-sets)
  2. Reclassification of attribute values
  3. Renaming of feature classes or attributes
  4. Merge / split of features
  5. Change attribute order
  6. ...
Candidate languages considered

- OWL (Web Ontology Language)
- QVT implementation: ATL (Eclipse context)
- UML-T (mdWFS research)
- (SPARQL-Construct)
- SWRL

- But none of these fulfilled even part of the requirements
- …
OML = Ontology Mapping Language

• From EU projects SEKT, DIP and Knowledge Web

• Consists of 2 parts
  Align (which is used in some tools)
  Extension to Align, called OML

• OML is not a standard

• There are papers and other documentation, e.g.
OML: cells with entity-entity mappings

• See oml_v0.jpg
• Example mappings (plain XML) waterBW2inspire.xml
Steps

• Testing of OML’s strengths and weaknesses
  1. The expressiveness = can OML handle the most common schema translation situations (Lehto, our own tests) (mapping time)
  2. The completeness and unambiguity -> does OML provide all necessary information for the actual data transformation (runtime)

• Make a Java API for OML

• Use OML API in the Conceptual Schema Transformer components of the HUMBOLDT software
First evaluation: expressiveness

• We can express in OML all schema translation situations from previous
  “data transformation with XSLT” tests
• (= tests with German, Austrian and Swiss River and Road data sets)
• With a little tweak for the ‘missing value’/ set default case

• But we have to add / extend also
  A standardized list of function names and semantics
  More precise way to specify parameters to functions
  Clearer way to specify aggregation, intersection, and other set operations
Classifications of river width

### Source Schema BY and BW

<table>
<thead>
<tr>
<th>Attribute „BRG“, Enumeration</th>
<th>Code</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>&lt; 3 m</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>3 - 6 m</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>6 - 12 m*</td>
</tr>
</tbody>
</table>

### Source Schema VA

<table>
<thead>
<tr>
<th>Attribute „BRG“, Enumeration</th>
<th>Code</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>&lt; 5 m</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>5 - 20 m</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>&gt; 20 m</td>
</tr>
</tbody>
</table>

### Target Schema

<table>
<thead>
<tr>
<th>Attribute „WidthLowerRange“</th>
<th>Range value, &gt;</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Attribute „WidthUpperRange“</th>
<th>Range value, &gt;</th>
</tr>
</thead>
</table>
First evaluation: runtime

• We can parse the OML examples and derive data transformation operations from the mappings

• With until now 2 exceptions (test with XML/GML input)
  
  Namespaces of data sets are not in the OML mapping -> but needed in Conceptual Schema Transformer software

At conceptual level (OML) not distinction 2 kinds of Properties, as is the case in XML (element and attribute)
Future work, outlook

• Add configuration possibility of implementation-specific details (in case of XML/GML: namespaces, element/attribute)
• Development of the CST (Conceptual Schema Transformer) components
• Integration with existing libraries (GeoTools most probably): re-use of functions
• More tests, especially with INSPIRE target schemas
• Serialization from HALE (HUMBOLDT Alignment Editor)