HUMBOLDT
www.esdi-humboldt.eu

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

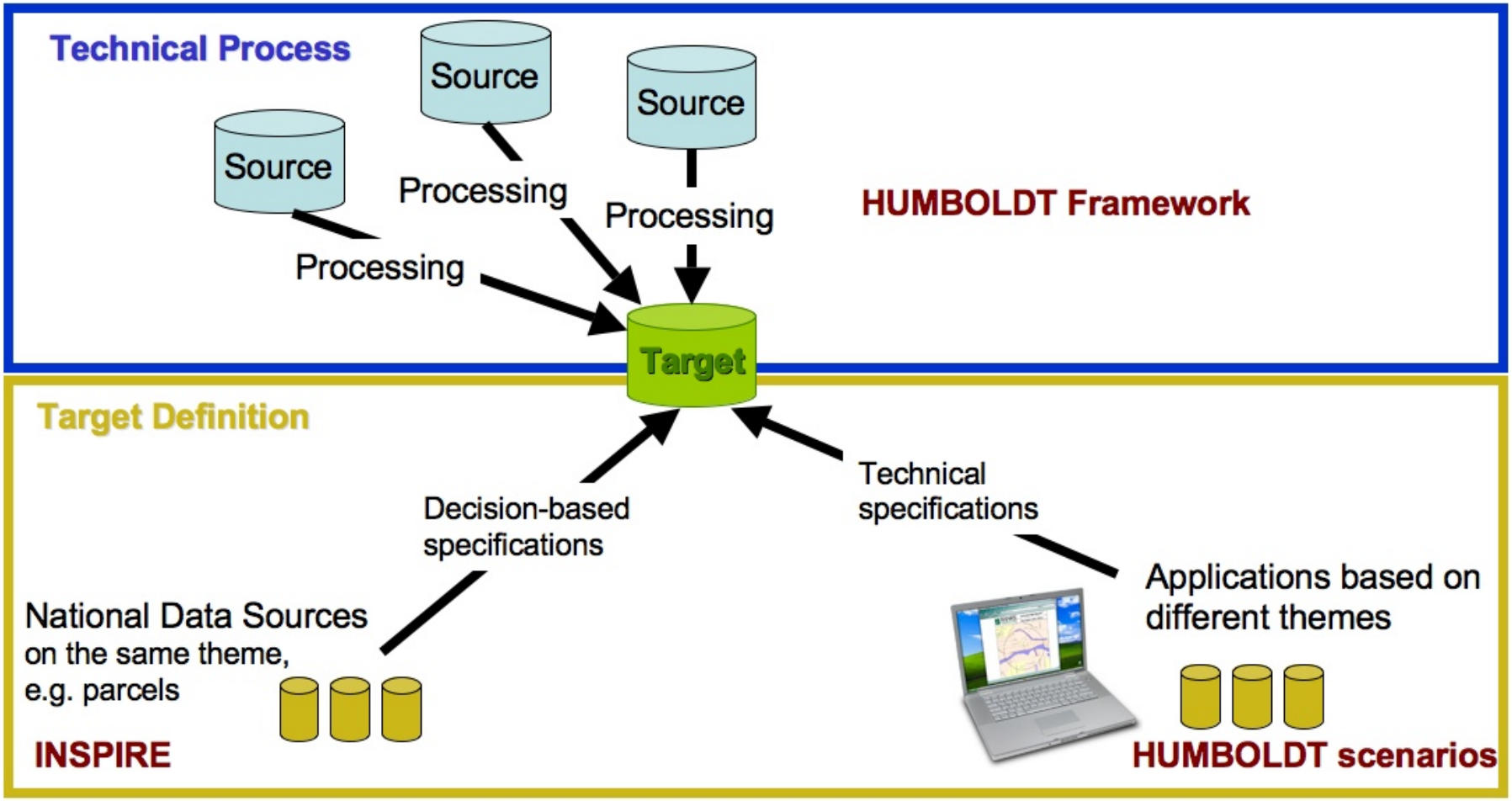
Data harmonisation

- 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

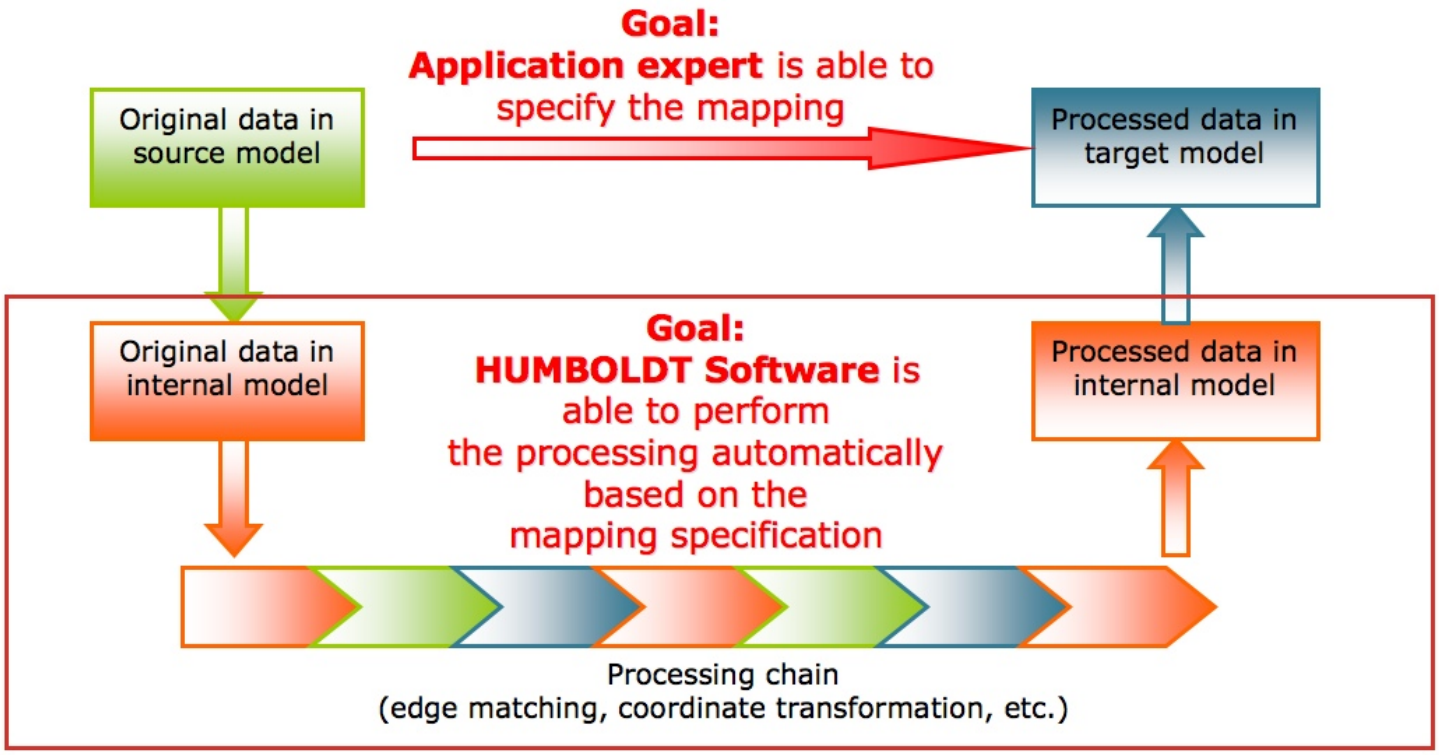
- Specifying (or re-using) target data model / schema
- Reverse-engineering (often) data model of existing data
- Definition of mapping rules between schema of source data and target schema
- ‘Execution’ of mapping rules in actual data transformation process



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



Matching tables, spreadsheets

- Matching table (input domain expert)

Feature class: er:RoadLink	Feature class: de:Ver01_L
er:id/er:permanentId	de:OB
er:RoadName er:roadName1 er:roadName2 er:alternativeName	If de:OBJART in (3102, 3101) de:GN (when 'NNNN' then 'Null or no value') de:KN (when 'NNNN' then 'Null or no value') de:ZN (If not null)
er:RoadSurface/er:roadSurface	If de:OBJART =3102 and de:BEF!=1000 then 'unpaved'

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.

Euzenat, J., F. Scharffe, et al. (2007). D2.2.10: Expressive alignment language and implementation. Project deliverable 2.2.10, Knowledge Web NoE (FP6-507482), 2007.

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	
Attribute „BRG“, Enumeration	
Code	**Value**
3	< 3 m
6	3 - 6 m
12	6 - 12 m*

Source Schema VA	
Attribute „BRG“, Enumeration	
Code	**Value**
3	< 5 m
2	5 - 20 m
1	> 20 m

Target Schema
Attribute „WidthLowerRange“
Range value, >

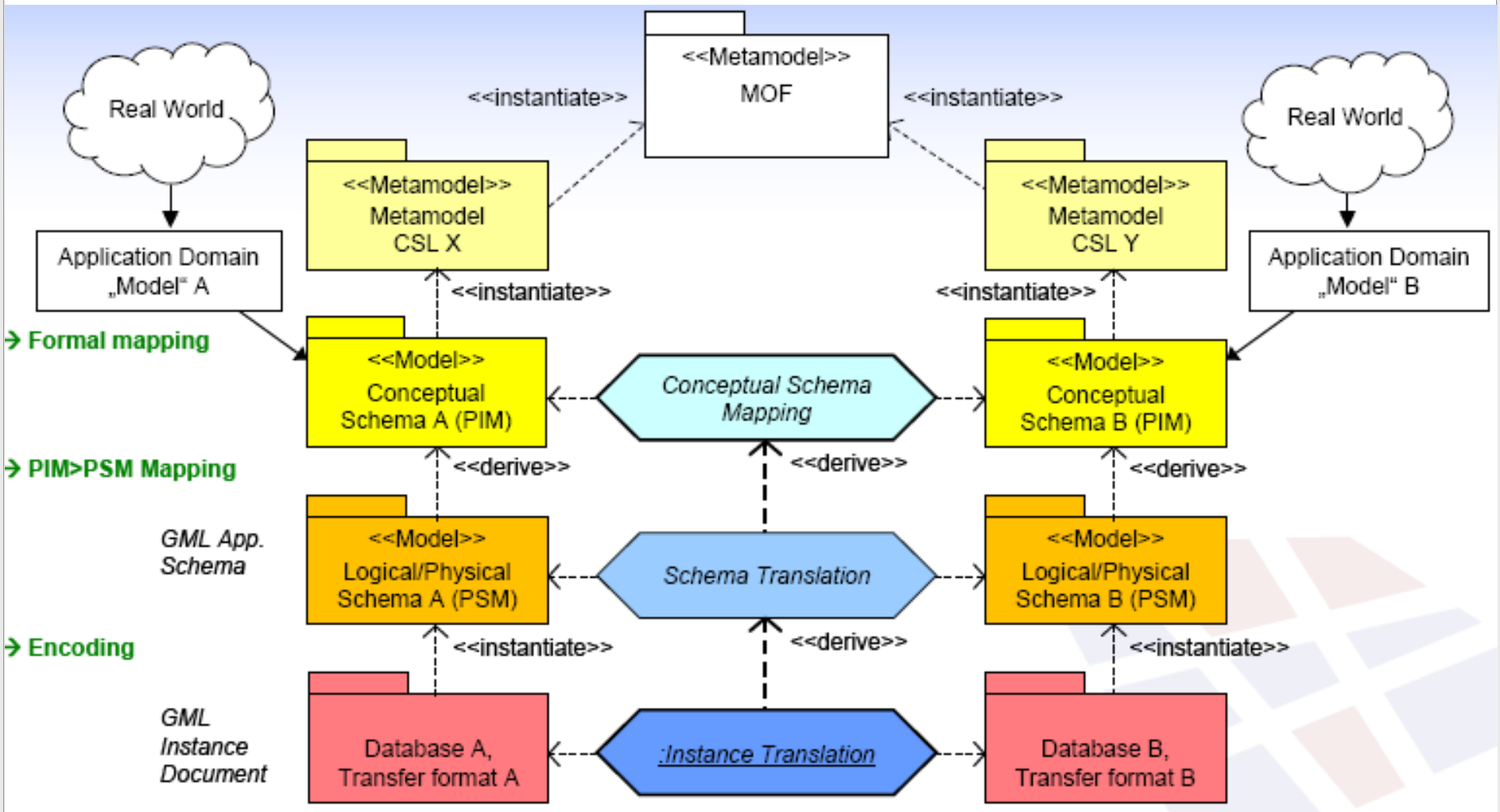
Target Schema
Attribute „WidthUpperRange“
Range value, >

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)