Challenges in Geospatial Data Harmonisation: Examples and Approaches from the HUMBOLDT project

AGILE Workshop 2009

Astrid Fichtinger, Eva Klien, Christine Giger
Overview

- The HUMBOLDT Project
- Data harmonisation
- Data harmonisation challenges - examples from HUMBOLDT
- Approaches and lessons learnt for schema transformation
The HUMBOLDT Project
The HUMBOLDT Project (IP in FP6, 10/2006 to 09/2010)

(Some of the) Objectives

- Overcome barriers in multidisciplinary and cross-border usage of geoinformation
- Common tools and services to support transformation needs
- Integration of these tools and services into existing SDI
- Support for INSPIRE (tools for data harmonisation) and GMES (theme-specific services in scenarios)

Users’ perspective on the SDI

- From a data-centric view to a usage centric view
HUMBOLDT – Vision

**Formulation of the demand**

- Format
- Spatial reference
- Graphical form
- Object generation
- Language
- Object semantics
- Expected quality

Usage Domain A

Provision of information in the language area / semantics of the application domain

- Information directly useable

Quality information / information about the level of goal achievement

Demand-oriented information
HUMBOLDT – Vision

Formulation of the demand

Provision of information in the language area / semantics of the application domain

Format
Spatial reference
Graphical form
Object generation

Language
Object semantics
Expected quality

“creating the possibility to combine data from heterogeneous sources into integrated, consistent and unambiguous information products, in a way that is of no concern to the end-user” (A 3.5-D1)

Demand-oriented information

Data Domain X

Domain X

Usage Domain A
HUMBOLDT Technological Framework

→ Supporting SDI enablement by providing the functionality for covering the data harmonisation process as a whole
→ Offering the possibility to make use of single functionalities as part of existing SDI

- Collection of tools and services for harmonisation (and use) of spatial data with a level of automation as high as possible

- Provision as Open Source Software
Data Harmonisation
What is data harmonisation?

INSPIRE

“process of developing a common set of data product specifications in a way that allows the provision of access to spatial data through spatial data services in a representation that allows for combining it with other harmonised data in a coherent way.”

NOTE: This includes agreements about coordinate reference systems, classification systems, application schemas, etc.

HUMBOLDT

“creating the possibility to combine data from heterogeneous sources into integrated, consistent and unambiguous information products, in a way that is of no concern to the end-user” (A 3.5-D1)
Data harmonisation processes

**Technical Process**

Source → Processing → Source → Processing → Target

**HUMBOLDT Framework**

**Target Definition**

National Data Sources on the same theme, e.g. parcels

- Decision-based specifications
- Technical specifications
- Applications based on different themes

**INSPIRE**

© 2008 Humboldt Consortium  http://www.esdi-humboldt.eu
HUMBOLDT:

+ computational models (process models), their constraints and parameters as 21st component

INSPIRE data harmonisation components

(A) INSPIRE Principles
(B) Terminology
(C) Reference model

(D) Rules for application Schemas and feature catalogues
(E) Spatial and temporal aspects
(F) Multi-lingual text and cultural adaptibility

(G) Coordinate referencing and units model
(H) Object referencing modelling
(I) Data translation model/guidelines

(J) Portrayal model
(K) Identifier Management
(L) Registers and registries

(M) Metadata
(N) Maintenance
(O) Quality

(P) Data Transfer
(Q) Consistency between data
(R) Multiple representations

(S) Data capturing
(T) Conformance

(INSPIRE 2008)
Results of state-of-the-art analysis

- Existing tools do not support all mappings and steps in a harmonisation workflow
- Existing tools often do not work service-oriented or cannot be integrated easily in a service-oriented architecture or framework
- Existing tools do not fully support testing whether a model or data set is consistent and correct
Results of state-of-the-art analysis

The top five requirements are:

- Problems with different data formats ➔ provide interoperable access to heterogeneous data sources
- Problems with different data models ➔ provide solutions for data model harmonisation
- Problems due to missing / inconsistent / outdated metadata ➔ provide solutions to search for and possibly capture / publish metadata
- Problems with the meaning of objects, i.e. semantics ➔ provide solutions like application domain dictionaries and thesauri
- Problems with different coordinate reference systems ➔ provide Coordinate Transformation Service(s)
Data model harmonisation
(schema mapping)

OMG (Object Management Group):
- Architecture for open distributed systems (e.g. used by OGC)
- Basis for state-of-the-art, mainstream IT: Model-Driven Architecture (MDA)
  - PIM: Platform Independent Model
  - PSM: Platform Specific Model
  - UML: Unified Modeling Language
  - MOF: Meta Object Facility (basis of UML and other specialised description languages)
  - MDA used as basis for interface standardization in ISO/TC211, CEN/TC287, OGC
Technology Framework for Data Harmonisation
Goal: Provide functionalities for covering the data harmonisation process as a whole

Executed technical processes

- Source
- Schema Transf.
- Coord. Transf.
- Language
- Target

Analysis of sources

Input for transformation (mapping rules)

Creation of target

User knowledge
Functionalities for covering the data harmonization process as a whole

- Capturing domain knowledge / application-specific knowledge for supporting the specification of transformations and the definition of the information product (Model Editor, Alignment Editor, …)
  - Defining the information product (target schema, SRS, spatial extent, …) to which the processed data needs to be transformed
  - Identification of transformation needs as part of the overall processing of an information request
  - Enhancing the formalization of the transformation between two data schemas

- Handling of transformation needs as part of the overall processing of an information request

- Enhancement of the automation of the data harmonization processes (depends on the possibility of capturing the required knowledge and the availability of this knowledge in the system)
1. Defining the information product (target schema, SRS, spatial extent, …) to which the processed data needs to be transformed
The HUMBOLDT GeoModel Editor

Developed to **support the process of data specification**

- **Main goals:**
  - Easy-to-use for application experts
  - Collecting all required information on the data
  - Providing formal representation of the information in order to feed it into a harmonisation process

- **Implemented** on a framework (Eclipse), which is (in itself) based on the MDA

- **Usage of a “spatial UML”** (HUMBOLDT modelling language)
  - Producing/providing a graphical and a textual representation of the data model
  - **Basic data model** containing basic spatial data types
  - Support of vertical mapping: serialisation possible to transfer standards or other representation (e.g. XMI, GML, ISO19131, ...)
2. Identification of transformation needs as part of the overall processing of an information request

Source
GML

? 

Target
GML

Analysis of sources

User knowledge
3. Enhancing the formalization of the transformation between two data schemas

- Source
- Schema Transf.
- Target

Executed technical processes

- Analysis of sources
- User knowledge
- Input for transformation (mapping rules)
The HUMBOLDT Alignment Editor (HALE)

- HALE is a tool for geodomain application experts and GI experts, especially data custodians.
- It provides means to create automatically executable horizontal mappings on the conceptual schema level and in addition in some aspects on the logical schema level.
- It provides the means to ensure the validity of a mapping on the base of the source and target schema as well as on the base of the source and a reference data set.
- It uses schemas created in the HUMBOLDT Editor.
- It does not execute those mappings.
The mapping language

- Requirements
  - Expressive enough (must support transformation operations like renaming, restructuring, reclassification, geometric & topological functions, etc.)
  - Implementation-neutral
  - Actual mapping code can be derived from it
  - Fitting the model-driven approach

- Candidate: Ontology Mapping Language (OML)

→ cf. presentation by Thorsten Reitz and Marian de Vries
The HUMBOLDT Conceptual Schema Transformer

- Processing Service (WPS) interface
- Transformation of source data from source schema to target schema
- Loads source data (GML)
- Loads mapping rules defined in OML
- Transformation processing chain is composed using the HUMBOLDT Workflow Service and orchestrated by the HUMBOLDT Mediator Service
Further HUMBOLDT Transformation Services

- Edge Matching Service
- Coordinate Transformation Service
- Language Transformation Service (planned)
- Quality Measurement in Transformation (planned)
- Multiple Representation Merging Service (planned)
4. Handling of transformation needs as part of the overall processing of an information request

**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**
(schema translation, edge matching, coordinate transformation, etc.)
Data harmonisation challenges -
Real world examples from HUMBOLDT Scenarios
Data used in the testing

Topographic vector data (streets and rivers) from

- Germany: Baden-Wuerttemberg (BW)
- and Bavaria (BY)
- Austria: Vorarlberg (VA)
- Switzerland (CH)

Data sources:
© Bayerische Vermessungsverwaltung
© Landesamt für Geoinformation und Landentwicklung Baden-Württemberg
© Land Vorarlberg
© swisstopo
Definition of the target data model

Simple requirements:

- Vector data to allow analysis (no routing for street data necessary)
- Limited set of attributes, e.g. for streets:
  - Name(s)
  - Functional road type (highway, local street, residential street, cycling path, …)
  - Road “importance” (international street, regional relevance, …)
  - Road surface (paved/unpaved)
- Reusing elements of the INSPIRE Data Specifications
Executing a “real” harmonisation process: lessons learned

- Analysis of sources
- User knowledge

Executed technical processes

Source → Schema translation → Coord. Transf. → ... → Target
Analysis of street data sources

Similar data (roads and watercourses) from similar organisations (national/regional mapping agencies)

Germany: data from DLM (ATKIS) Bavaria (BY)

data from DLM (ATKIS) Baden Wuerttemberg (BW)

Austria: data from regional GIS System (VOGIS) (VA)

Switzerland: data from Swiss Mapping Agency (Vector25) (CH)
Analysis of street data - Austria

Austria

- ArcInfo Export .e00
- Not documented (codelists unavailable)
- Purpose is unknown (extract from Austrian landscape model)
- Split into different files according to administrative aspects
  (“municipal” and “higher-class” streets)
Analysis of street data - Germany

Germany

- ArcView Shapefiles
- Well documented data model (ATKIS), poorly documented export
- Purpose – general topographic data
- Split into different files according to geometry
  - “simple streets”, “complex geometry streets”, “distinguished lanes”, “centrelines”
Analysis of street data - Switzerland

Switzerland
- ArcView Shapefiles
- Well documented (exact file definition)
- Purpose – map production
- One file (only lines, from highway to hiking path)
Street features in detail (Austria)

What is a feature?

- “A collection of line segments having the same street name”

Attributes

- Names
- Category (highway, country road, ...)
- Type (single carriageway, dual carriageway, roundabout, ...)

Notes:

- Missing attribute values
- Missing geometry elements
Street features in detail (Germany)

What is a feature?
- A street feature is a continuous collection of line segments differentiated by their functional classes or their international relevance.
- Any other attributes value changes distinguish the street segments

Notes:
- The exported files combine “street segments” and “complex street”, the relation is destroyed – highly redundant data
Street features in detail (Germany 2)

Attributes
- Names
- Width
- Additional carriageways
- Function
- Relevance for national/international traffic
- …

Notes:
- The exports (encodings) differ in BaWü and Bavaria – in combination of files/attribute names/ …
  although the same data model is valid for both states
Street features in detail (Switzerland)

What is a feature?
- A street feature is a collection of continuous line segments; from node to node

Attributes
- Names – mostly blank, except for cartographic aspects
- Category (highway, country Road, .., hiking path)

Notes:
- The only purpose of the data is map production
Street data - different geometries

Complex streets (Austria)
Segments (Switzerland)
Segments and combinations of segments (Germany)

=> Split or merge operations necessary
Street data - different classification “aspects”

Austria
- categories (Autobahn, regional streets, …)
- specific construction (ramp, parking lot, circular, …)

Germany
- administration (Autobahn, regional streets, different local streets, …)
- function (pedestrians, car traffic)
- importance / relevance (local, international)

Switzerland
- cartographic aspect (Autobahn, 1st class road, 2nd class, …, footpath, mountain path, …)

Logic rules to derive other classification aspects (e.g. “Autobahn” -> “paved surface”, “way” and no “footpath” and no “bicycle” -> “agricultural way”)

Dealing with uncertainty necessary: who decides?
Street data - different classification Systems

Street category

Austria
- 5 Classes from Highway to private Road

Germany
- 7 Classes (5 for roads and 2 for ways)

Switzerland
- 36 classes (including cartographic “help lines” e.g. connection of footpaths)

Reclassification rules needed

Dealing with uncertainty
Approaches and lessons learnt for schema transformation
Test Environment: Definition of mapping rules using XSLT

Matching Table as input:

<table>
<thead>
<tr>
<th>Feature class: er:RoadLink</th>
<th>Feature class: de:Ver01_L</th>
</tr>
</thead>
<tbody>
<tr>
<td>**er:id/**er:permanentId</td>
<td>de:OB</td>
</tr>
<tr>
<td><strong>er:RoadName</strong></td>
<td>If de:OBJART in (3102, 3101)</td>
</tr>
<tr>
<td><strong>er:roadName1</strong></td>
<td>de:GN (when ‘NNMN’ then ‘Null or no value’)</td>
</tr>
<tr>
<td><strong>er:roadName2</strong></td>
<td>de:KN (when ‘NNMN’ then ‘Null or no value’)</td>
</tr>
<tr>
<td><strong>er:alternativeName</strong></td>
<td>de:ZN (If not null)</td>
</tr>
<tr>
<td>**er:RoadSurface/**er:roadSurface</td>
<td>If de:OBJART =3102 and de:BEF!=1000 then ‘unpaved’</td>
</tr>
</tbody>
</table>
Test Environment: Definition of mapping rules using XSLT-Formalisation

```xml
<xsl:template match="streetBY:streets_Bavaria">
    <xsl:element name="gml:featureMember">
        <xsl:element name="er:RoadLink">
            <xsl:attribute name="fid">
                <xsl:value-of select="@fid"/>
            </xsl:attribute>
            ....
            <xsl:if test="streetBY:OBJART = '3102' or streetBY:OBJART = '3101'">
                <xsl:element name="er:RoadName">
                    <xsl:element name="er:roadName1">
                        <xsl:choose>
                            <xsl:when test="streetBY:GN = 'NNNN'">Null or No
```
Test Environment: Definition of mapping rules using XSLT

xslt-functionality:
- "filter" elements to apply matches in tree (<template match=""/>)
- "selections" to read values (<value-of select=""/>)
- "conditions" (if-then-else, choice-when-otherwise)
- "loop" over elements (<for-each select=""/>)
- "create" new elements and attributes (<element name=""/>)

Problems:
- Spliting and combining Features/Elements not trivial ("jumping in XML-tree")
- Dealing with Ids on-the-fly
- Dealing with external values (current date, actual user, ...) on-the-fly
- No geometric operations
- Maintenance
- Long, complex stylesheets
  - Namespace problems
  - Tag renaming
  - Types
Lessons learnt from Test Environment (Schema translation for Street Datasets based on xslt)

- “Domain knowledge” is the key to harmonisation
- Formalisation of knowledge must be simplified
  (for the description of source and target as well as for the description of the mapping)
- Direct XSLT production is painful/error prune
- Pure XSLT (esp. XSLT 1.0) is too limited

Conceptual Schema Mapping with the Ontology Mapping Language
(rypto presentation of Thorsten Reitz and Marian de Vries)
HUMBOLDT Open Source Software (in development status)
http://community.esdi-humboldt.eu/

HUMBOLDT Model Editor
- For the creation of UML application schemas

HUMBOLDT Alignment Editor
- Definition of conceptual schema transformation

Workflow Service
- Service Composition for handling transformation needs

Mediator Service
- Execution of transformation chains

Context Service
- Definition of transformation products

Transformation Service
- Exposed as Web Processing Services (e.g. CTS, Edge Matching)
Thank you for your attention!

→ HUMBOLDT Workshop incl. tool demonstration at INSPIRE Conference, Rotterdam, 15 June 2009