Approaches & Languages for Schema Transformation

Findings of HUMBOLDT & follow-up Activities

INSPIRE KEN Workshop on Schema Transformation
Paris, France, 08.10.2013

Thorsten Reitz
Esri R&D Center Zurich AG,
data harmonisation panel
The data harmonisation panel

INSPIRE KEN & EuroSDR Workshop in Paris

If you'd like to get an excellent overview of available software and approaches for transforming data to INSPIRE formats, a good opportunity is coming up: The INSPIRE KEN (Knowledge Exchange Network) and EuroSDR are organizing a workshop about schema transformation tools and methods in the premises of the ENSG – Marine-La-Vallée (near Paris) – France.

Citing from the workshop's information and registration page, “NMAs, as other data producers, will have to make their data compliant with INSPIRE interoperability implementing Rules. During [the] next years, this compliance will mainly be achieved through schema and data transformation. The objectives of the workshop is to make a state-of-play about (existing or projected) schema transformation tools, to help NMAs to assess these tools and to help them to choose the most appropriate and possibly, to provide background to disseminate knowledge about schema transformation at national level.”

The workshop is scheduled from Tuesday 8th October 2013 09:00 – Wednesday 9th October 2013 16:00.

This is the draft lineup of presentations of the workshop:

**Tuesday Morning**

<table>
<thead>
<tr>
<th>Time</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:00 – 09:10</td>
<td>Welcome and introduction</td>
</tr>
<tr>
<td>09:10 – 09:35</td>
<td>A Study about schema transformation services</td>
</tr>
<tr>
<td>09:35 – 10:00</td>
<td>Approaches &amp; Languages for Schema Transformation: Findings of HUMBOLDT &amp; follow-up Activities</td>
</tr>
<tr>
<td>10:00 – 10:25</td>
<td>From production data base to INSPIRE data: potential methods</td>
</tr>
<tr>
<td>10:25 – 10:45</td>
<td>Pause</td>
</tr>
</tbody>
</table>
Content

- What is the problem with Schema Transformation?
- Requirements
- A Classification of Approaches
  - Declarative, Procedural
  - Schema-driven, Instance-driven
  - Visual, Textual
  - Conceptual, Logical, Physical
- Comparison & Conclusion
Schema Transformation Languages are Programming Languages...

- A Schema Transformation project typically includes the following:
  - Design (matching tables, UML)
  - Development (Coding Xquery, building a Pipes-and-Filters Graph)
  - Debugging (analysing ST behaviour)
  - Testing (Validation & Acceptance Testing)
  - Documentation (hopefully!)

- Iterate through these activities for updated/new datasets, new schemas...
... in a high-complexity environment!
REQUIREMENTS ON SCHEMA TRANSFORMATION (ST) IN INSPIRE
Requirements for ST approaches (I)

- Expressivity
  - Six Levels defined in INSPIRE Schema Transformation Pilot [Beare 2010]
    - 1 – Renaming classes and attributes
    - 2 – Simple attribute derivation
    - 3 – Aggregating input records
    - 4 – Complex derivation and dynamic type selection
    - 5 – Deriving values based on multiple features
    - 6 – Conflation and model generalisation
Requirements for ST approaches (II)

- Verbosity
  - How many statements are needed to define a mapping?
  - Impacts maintenance, effectiveness

- Transparency and Traceability
  - What happens in the transformation engine?
  - Can I debug it?

- Tools/Implementations
  - Performance, Reliability, Usability, ...
Specific Requirements In INSPIRE

- Spatial harmonisation
  - Edge Matching, Reference Systems, Multiple Representations
- Structure
  - References, Aggregates & Composites, Inheritance
- Quality
  - Metadata, geometric quality, classification accuracy, ...
- Semantics
  - Code Lists, Feature Types
CLASSIFICATION AND COMPARISON OF APPROACHES
Classification by Paradigm

- Declarative Approaches
  - Describe the logic of a computation without describing its control flow.
  - Leave optimization and actual execution order to the runtime engine.
  - Examples: XSLT, EDOAL/gOML

- Procedural Approaches
  - Describe a computation by giving its control flow through a series of functions.
  - Examples: Python GeoProcessing Tool, FME

- Other Paradigms (Rule-based, Functional, Object-oriented, Aspect-Oriented, Agent-based...)

Thorsten Reitz – INSPIRE KEN Workshop, Paris, 08.10.2013
Classification by Abstraction Level

Horizontal Transformations

Vertical Transformations

Metamodel A
OWL

Conceptual Schema A
OWL Ontology

Logical Schema A
GML App. Schema

Instance Data A
GML

Conceptual Schema Alignment

Metamodel B
UML

Conceptual Schema B
UML Model

Logical Schema B
GML App. Schema

Instance Data B
GML

Transformation Definition

<<instatiates>>

<<derived>>

<<extends>>

<<derived>>

<<instatiates>>

<<derived>>

<<instatiates>>
Classification by Drivers

- **Instance-driven Approaches**
  - Use information available in features (instances) to infer correct transformation functions to use
  - Examples: FCA-Merge, XSLT, FME

- **Schema-driven Approaches**
  - Use conceptual schema and define relations based on the elements of the schema
  - Examples: PROMT, COMA++ (Protégé), EDOAL/gOML

- **Combined Approaches**
  - Use information available on instance level to enhance the quality of an automatically derived mapping on the extension level
Classification by Representation

Graphical

Textual

```xml
(:pragmas function <f:function xmlns:f="urn:annotations.id.bea.com" visibility="protected" kind="library" isPrimary="false">)

declare function f1:is_access_allowed_with_response_attributes($x1 as xsd:string?, $x2 as xsd:string?, $x3 as xsd:string?) as xs:query { 
  let $result := f1:is_access_allowed_with_response_attributes("RTLApp/datacontrol/orderview", 
    { "totalorderamount" }, 
    (fn:string(fn:round($x2/TotalOrderAmount)))) return 
  if ($result[1] eq "true") then 
    let $class_index := fn:index-of($result, "class") return 
    if (fn:empty($class_index)) then 
      fn:fail() 
    else 
    let $class_values := fn:subsequence($result, $class_index[1]) return 
    if (fn:empty(fn:index-of($class_values, "ALESResponse"))) then 
      if (fn:empty(fn:index-of($class_values, "A"))) then 
        fn:fail() 
      else 
        fn:fail() 
  }
```
## Classification

<table>
<thead>
<tr>
<th>Name</th>
<th>Abstraction</th>
<th>Schema Language</th>
<th>Paradigm(s)</th>
<th>Primary Driver</th>
<th>Primary Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSLT</td>
<td>Logical</td>
<td>XSD</td>
<td>Declarative+</td>
<td>Instance</td>
<td>Textual</td>
</tr>
<tr>
<td>XQuery</td>
<td>Logical</td>
<td>XSD</td>
<td>Procedural</td>
<td>Instance</td>
<td>Textual</td>
</tr>
<tr>
<td>EDOAL/OML</td>
<td>Conceptual</td>
<td>OWL+*</td>
<td>Declarative</td>
<td>Schema</td>
<td>Graphical</td>
</tr>
<tr>
<td>RIF</td>
<td>Conceptual</td>
<td>*</td>
<td>Declarative</td>
<td>Instance</td>
<td>Textual</td>
</tr>
<tr>
<td>QVT/ATL</td>
<td>Conceptual</td>
<td>UML</td>
<td>Both</td>
<td>Schema</td>
<td>Combined</td>
</tr>
<tr>
<td>UML-T</td>
<td>Conceptual</td>
<td>UML</td>
<td>Declarative</td>
<td>Schema</td>
<td>Graphical</td>
</tr>
<tr>
<td>GeoKettle</td>
<td>Logical</td>
<td>Internal</td>
<td>Procedural</td>
<td>Instance</td>
<td>Graphical</td>
</tr>
<tr>
<td>Talend</td>
<td>Logical</td>
<td>Internal</td>
<td>Procedural</td>
<td>Instance</td>
<td>Graphical</td>
</tr>
<tr>
<td>FME</td>
<td>Logical</td>
<td>Internal</td>
<td>Procedural</td>
<td>Instance</td>
<td>Graphical</td>
</tr>
</tbody>
</table>

* Independent of a specific logical schema
## Comparison

<table>
<thead>
<tr>
<th>Name</th>
<th>Tools</th>
<th>Domain</th>
<th>Expressivity</th>
<th>Verbosity</th>
<th>Maturity</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSLT</td>
<td>10+</td>
<td>Universal + Spatial</td>
<td>80%</td>
<td>11145</td>
<td>4</td>
</tr>
<tr>
<td>XQuery</td>
<td>10+</td>
<td>Universal + Spatial</td>
<td>96%</td>
<td>392</td>
<td>4</td>
</tr>
<tr>
<td>EDOAL/gOML</td>
<td>2</td>
<td>Universal + Spatial</td>
<td>92%</td>
<td>91</td>
<td>3</td>
</tr>
<tr>
<td>RIF</td>
<td>3</td>
<td>Universal</td>
<td>80%</td>
<td>4145</td>
<td>2</td>
</tr>
<tr>
<td>QVT/ATL</td>
<td>4+</td>
<td>Universal</td>
<td>76%</td>
<td>117</td>
<td>3</td>
</tr>
<tr>
<td>UML-T</td>
<td>1</td>
<td>Universal + Spatial</td>
<td>60%</td>
<td>-.-</td>
<td>1</td>
</tr>
<tr>
<td>GeoKettle</td>
<td>1</td>
<td>Spatial</td>
<td>96%</td>
<td>-.-</td>
<td>3</td>
</tr>
<tr>
<td>Talend</td>
<td>1</td>
<td>Universal + Spatial</td>
<td>92%</td>
<td>-.-</td>
<td>3</td>
</tr>
<tr>
<td>FME</td>
<td>1</td>
<td>Spatial</td>
<td>100%</td>
<td>205</td>
<td>3</td>
</tr>
</tbody>
</table>

1. Research conducted in 2010-2011.
2. Coverage of functions listed in [Beare 2010], Appendix 2
3. Number of statements needed for a set of example projects
4. no. of implementations, user base, iterations, functionality
5. generated XSLT/RIF

Thorsten Reitz – INSPIRE KEN Workshop, Paris, 08.10.2013
CONCLUSION & DISCUSSION
Conclusion (I) – What makes ST hard?

- **Language Expressiveness & Complexity**
  - Most approaches are Turing complete – you can build a First Person Shooter using them if you want to

- **Schema complexity & data volume, variance...**

- **Tooling**
  - Debugging/Inspection
  - Limited collaboration

- **Unconnected Documentation**

- **No maintainability or collaboration features**

Thorsten Reitz – INSPIRE KEN Workshop, Paris, 08.10.2013
Conclusion (II) - Selecting approaches

- How well does an approach cover all phases in my ST project?
  - If only some steps are covered, which issues are introduced through combination of multiple approaches?

- How well can I maintain my ST project?
  - As with any software, a project outcome will lose value quickly if it cannot be maintained.

- How can I document my ST project?
  - How do I explain to others what tricks I used to get the approach to do what I wanted?
Discussion & Questions

- Are we happy with the standards and de-facto standards?
  - Limited evidence of re-use and exchange of Schema Transformation projects

- Disagree with any of the classification/comparison items? Contact me!
Happy end

- Read the data harmonisation panel to learn more: http://blog.dhpanel.eu

- Contact the presenter: tr@xdsi.eu / treitz@esri.com