Schema transformation of Administrative Data with GeoKettle
1. INTRODUCTION
1. INTRODUCTION

- **ETL definition**: An ETL tool is a type of software generally used to populate databases or data warehouses from heterogeneous data sources.

- **GeoKettle**: GeoKettle is an open source ETL developed in Java by Spatialistics (license GNU Lesser General Public : LGPL).

- **It can be downloaded from the web site** (V2.5 actually available): http://www.spatialytics.org/fr/projets/geokettle/
1. INTRODUCTION

- **Generalities:**

- **GeoKettle** is a “spatially-enabled” version of the generic ETL tool Kettle (Pentaho Data Integration).

- GeoKettle benefits from geospatial capabilities from Open Source libraries like JTS, GeoTools, deegree, OGR and, via a plugin, Sextante.

- This tool requires a Java Runtime Environment (JRE) version 5 or newer.

- It can be installed in Windows, Linux or Macintosh systems.
2. GEOKETTLE FUNCTIONALITIES
2. GEOKETTLE FUNCTIONALITIES

1. INTRODUCTION
2. GEOKETTLE FUNCTIONALITIES
   - Vocabulary:
   - Steps
   - Transformation example
   - Extraction depuis fichier Shapefile
   - Branchement conditionnel
   - Calcul
   - Calcul 2
   - Alimentation fichier
   - Hops
   - Job example

3. TEST OF SCHEMA TRANSFORMATION
4. CONCLUSION
2. GEOKETTLE FUNCTIONALITIES

- **Record**
  - Transformations and jobs are recorded in **XML format** files.
    - File ‘.ktr’ for the transformations,
    - File ‘.kjb’ for the jobs.

  This is a specific format including required steps and their parameters.

- **Variables** can be defined to reuse easily existing transformations with other data:
  - Other data: same structure but different date or area,
  - Variable: name of the file.
2. GEOKETTLE FUNCTIONALITIES

- **Data schema**

  - GeoKettle reads data sources and produced transformed data.
  - GeoKettle reads automatically the data schemas.
  - Rows can be previewed with a field table or a cartographic view.
2. GEOKETTLE FUNCTIONALITIES

Steps

The transformation steps are classified into categories:

- extract and load data, change the format, ...

Steps:
- Input
- Output
- Transform
- Flow
- Scripting
- Lookup
- Joins
- Statistics
- ...

- transform data
2. GEOKETTLE FUNCTIONALITIES

- **Steps : INPUT and OUTPUT categories for files**

<table>
<thead>
<tr>
<th>File types</th>
<th>INPUT</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>CSV S3</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>CSW catalog</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>ESRI</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Excel</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Fixed file</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>fXbase</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>GML</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Kettle</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>KML</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>LDAP</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>LDIF</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Mondrian</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>OGR</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Property file</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>RSS</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Salesforce</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Shapefile</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>SOS</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Text file</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>WFS</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Xbase (DBF)</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>XML</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

- **GML geometry only**

- **MapInfo**
  - GML 2
  - DXF
## 2. GEOKETTLE FUNCTIONALITIES

### Steps: database table types for INPUT, OUTPUT and UPDATE

<table>
<thead>
<tr>
<th>Vocabulary</th>
<th>Database types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record</td>
<td>Apache Derby</td>
</tr>
<tr>
<td>Data schema</td>
<td>AS/400</td>
</tr>
<tr>
<td>Steps</td>
<td>Borland Interbase</td>
</tr>
<tr>
<td>File types</td>
<td>dBase III, IV ou V</td>
</tr>
<tr>
<td>Database types</td>
<td>ExtenDB</td>
</tr>
<tr>
<td>Transformation steps</td>
<td>Firebird SQL</td>
</tr>
<tr>
<td>Generics</td>
<td>Generic database</td>
</tr>
<tr>
<td>Steps used for schema: examples</td>
<td>Greenplum</td>
</tr>
<tr>
<td>Oracle</td>
<td>Gupta SQL Base</td>
</tr>
<tr>
<td>Oracle RDB</td>
<td>H2</td>
</tr>
<tr>
<td>Remady Action Request System</td>
<td>Hypersonic</td>
</tr>
<tr>
<td>SAP R/3 System</td>
<td>IBM DB2</td>
</tr>
<tr>
<td>SQLite</td>
<td>Infobright</td>
</tr>
<tr>
<td>Sybase</td>
<td>Informix</td>
</tr>
<tr>
<td>SybaseIQ</td>
<td>Ingres</td>
</tr>
<tr>
<td>SybaseIQ</td>
<td>Intersystems Cache</td>
</tr>
<tr>
<td>Teradata</td>
<td>KingbaseES</td>
</tr>
<tr>
<td>UniVerse database</td>
<td>LucidDB</td>
</tr>
<tr>
<td>Vertica</td>
<td>MaxDB (SAP DB)</td>
</tr>
</tbody>
</table>
2. GEOKETTLE FUNCTIONALITIES

- **GeoKettle steps allow various transformations** like data transfer, spatial analysis, CRS transformation, filter, script execution, schema transformation...

- **For schema transformation, steps are particularly used to:**
  - select, modify (name, type) or remove fields
  - aggregate values
  - calculate new fields
  - filter rows (conditions)
  - join rows (rows sorted with the key)
  - map values
  - replace field value
  - split a field
  - switch according to conditions
  - …
2. GEOKETTLE FUNCTIONALITIES

- Example:

✓ select, modify or remove fields

- Rename:

- Modify the field type:
  - integer → string,
  - date → string...
2. GEOKETTLE FUNCTIONALITIES

**Example:**

- Default values can be defined:

  ![Diagram of Inspire structure](image)

  - And there are many calculation types:

- calculate new fields

- And there are many calculation types:
2. GEOKETTLE FUNCTIONALITIES

- **Examples:**
  - ✔ **filter rows**

  According to one or more **conditions** on fields and values, rows are distributes in two flows.
  → If the condition is fulfilled (**true**), rows are sent to a next step.
  → If the result is **false**, rows are sent to another one.

- ✔ **switch depending on values**

  Rows are distributes in the following steps depending on field values.
  There are so many output flows as defined values.
2. GEOKETTLE FUNCTIONALITIES

- Example:
  - join rows
    - Two data sets are input.
    - The common keys are identified in the tables.
    - Rows must be sorted by their key before a joining.
    - All the fields are joined in output data.

Steps used for schema : examples
2. GEOKETTLE FUNCTIONALITIES

- **Example**: ✓ map value

Source values are replaced by target values.

<table>
<thead>
<tr>
<th>IGN boundaries type</th>
<th>Inspire National Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>source values</td>
<td>target values</td>
</tr>
<tr>
<td>frontière</td>
<td>1stOrder, 2ndOrder, 3rdOrder, 4thOrder, 5thOrder</td>
</tr>
<tr>
<td>limite de région</td>
<td>2ndOrder, 3rdOrder, 4thOrder, 5thOrder</td>
</tr>
<tr>
<td>limite de département</td>
<td>3rdOrder, 4thOrder, 5thOrder</td>
</tr>
<tr>
<td>limite d'arrondissement</td>
<td>4thOrder, 5thOrder</td>
</tr>
<tr>
<td>limite de commune</td>
<td>5thOrder</td>
</tr>
</tbody>
</table>
2. GEOKETTLE FUNCTIONALITIES

- **Example:**
  - Split a field

If several values are delimited by a separator in a field, the field can be split.
2. GEOKETTLE FUNCTIONALITIES

- **Example:**
  - **aggregate values**

This step is used to aggregate informations in a field. For example, with the Inspire AU, it’s necessary to find all the municipalities included in the upper level unit.

This step allows to group the municipalities by the code of the upper level units and to list all the municipalities for each upper code.

<table>
<thead>
<tr>
<th>code</th>
<th>lower level unit list</th>
</tr>
</thead>
<tbody>
<tr>
<td>code 1</td>
<td>municipality1, municipality2, municipality5, municipality6, ...</td>
</tr>
<tr>
<td>code 2</td>
<td>municipality3, municipality4, municipality10, municipality12, ...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

You can aggregate values, or add numerical values, or calculate an average, ...
3. TEST OF SCHEMA TRANSFORMATION
3. TEST OF SCHEMA TRANSFORMATION

- **Context**
  - The schema transformation I have tested relates to administrative data.
  - My purpose was to transform internal data to Inspire structured data.
  - Input: BDUniGE / Administrative theme (large scale topographic data of IGN, like BDTopo®)
    - Administrative data extracted from a french unit ‘département’ (Vendée)

- **Output**: Inspire Administrative Units
3. TEST OF SCHEMA TRANSFORMATION

- **Demonstration:**

  → **Use of Geokettle to build the Inspire Identifier ‘InspireId’**

```plaintext
IGN feature
...
CLEABSOLUTE
...

Inspire feature
...
InspireId
...
```

4. CONCLUSION
3. TEST OF SCHEMA

1. INTRODUCTION
2. GEOKETTLE FUNCTIONALITIES
3. TEST OF SCHEMA TRANSFORMATION
   - BDUini GE specifications
   - Context
   - Demonstration
   - BDUini GE specifications
   - Inspire specifications
   - Transformations
   - Difficulties
4. CONCLUSION
3. TEST OF SCHEMA TRANSFORMATION

1. INTRODUCTION

2. GEFKETTE

- Inspire specifications
3. TEST OF SCHEMA TRANSFORMATION

1. INTRODUCTION
2. GEOKETTLE FUNCTIONALITIES
3. TEST OF SCHEMA TRANSFORMATION
   - Context
   - Demonstration
   - Inspire specifications
   - BDUni GE specifications
   - Transformations
   - Difficulties
4. CONCLUSION

... Transformations done with GeoKettle...
3. TEST OF SCHEMA TRANSFORMATION

1. INTRODUCTION
2. GEOKETTLE FUNCTIONALITIES

1) Transformation to ‘AdministrativeBoundary’
3. TEST OF SCHEMA TRANSFORMATION

- Transformation to ‘AdministrativeBoundary’
3. TEST OF SCHEMA TRANSFORMATION

2) Transformation to determine the ‘Boundary’ link

It’s to prepare the transformation to ‘AdministrativeUnits’. There is no direct link between units and boundaries in IGN structure.
3. TEST OF SCHEMA TRANSFORMATION

- Transformation to determine the ‘Boundary’ link

**Building of the ‘boundary’ link for the administrativeUnit**

**INPUT**: shapefile
- BDUnit: administrative units

**OUTPUT**: text file
- 'boundary' link for each administrativeUnit

**Process Steps**:
1. Extraction of the administrative structure flux
2. Calcul 2, Ajouter lignes X ML
3. Trilignes
4. Jointure comparaison lignes
5. Rows before join
6. Trilignes 2
7. Trilignes 3
8. Jointure comparaison lignes 2
9. Sort et group by administrative unit inspired
   Aggregate all administrative boundary inspired in the new ‘boundary’ field
10. Agregation shapes, Alignement vertical

**Join to find the right administrative unit**

**Join to find the left administrative unit**
3. TEST OF SCHEMA TRANSFORMATION

1. INTRODUCTION
2. GEOKETTLE FUNCTIONALITIES
3. TEST OF SCHEMA TRANSFORMATION

3) Transformation to ‘AdministrativeUnits’
3. TEST OF SCHEMA TRANSFORMATION

1. INTRODUCTION
2. GEOKETTLE FUNCTIONALITIES
3. TEST OF SCHEMA TRANSFORMATION

- Transformation to ‘AdministrativeUnits’
3. TEST OF SCHEMA TRANSFORMATION

Job for schema transformation of administrative data

- Job including 3 transformations to transform BDUni administrative data to Inspire Administrative Units:
  - Administrative Boundary transformation
  - AdministrativeUnit transformation
  - Waiting for the 'boundary link' file
  - Attente aposition fichier
  - Building of the 'boundary' link
  - Constituit d'un en boundary
3. TEST OF SCHEMA TRANSFORMATION

- **Encountered difficulties**
  - The input data for AU is not close to Inspire schema. It was difficult to implement the associations.
  - Using an ETL tool requires a good knowledge of the available steps and what they can do.
  - Find the various components, constitute various fields and values, and establish links, is relatively easy with GeoKettle. Steps allow to transform data schema.
  - For this particular test, the highest difficulty is to produce Inspire structure with complex attributes (data types) in output. Data types can’t be defined.
  - → GeoKettle is so not able to produce a gml file with a so complex structure today (with the existing output steps).
4. CONCLUSION
Advantages of GeoKettle use for schema transformation

- It’s a very intuitive tool, simple to use.
- It’s powerful and performant.
- The diversity of the provided steps is important.
- It reads automatically the schema from the data.
- Available steps are adapted to the schema transformation.

Drawbacks of GeoKettle use for schema transformation

- Jobs and transformations are stored in xml files. The code of the transformations is not generated. So the transformations can’t be executed from another platform.
- The Inspire complex structure can’t be specified and output (voidValueReason, inspireId, geographicalName, ...).
- There is no help in the software, and step description is very light in the documentation. However, you can get support in the forum.
Thank you for your attention.