

Adding new metadata schemas to GeoNetwork

Andrea Carboni

FAO

Summary

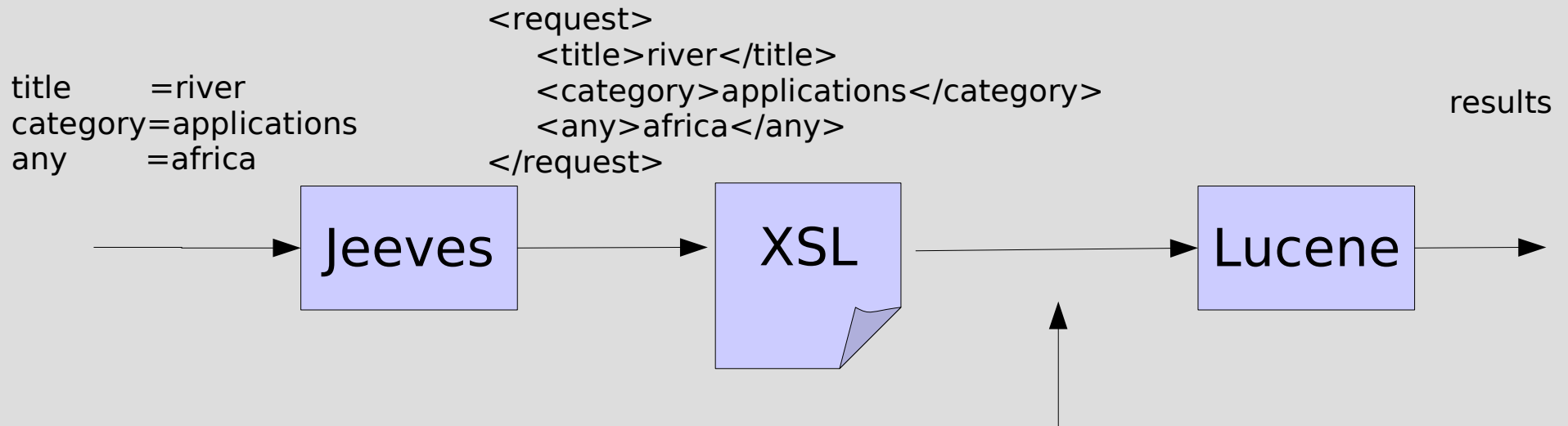
- Search engine's overview
- Schema's management
- Changes to Java code
- Schema's folder structure
 - Indexing fields
 - Stylesheets and labels
- Stylesheets used to view and edit
- CSW support

Searching with Lucene

Overview

- Search parameters are taken from search form and converted into XML
- A stylesheet converts this XML into an XML query tree
- The tree is mapped into a Lucene's query
- The query is executed

Search process



```
<request>
  <title>river</title>
  <category>applications</category>
  <any>africa</any>
</request>
```

results

Jeeves

XSL

Lucene

```
<BooleanQuery>
  <BooleanClause required="true" prohibited="false">
    <TermQuery fld="any" txt="africa"/>
  </BooleanClause>
  <BooleanClause required="true" prohibited="false">
    <TermQuery fld="title" txt="river"/>
  </BooleanClause>
  <BooleanClause required="true" prohibited="false">
    <TermQuery fld="category" txt="applications"/>
  </BooleanClause>
</BooleanQuery>
```

Search stylesheet

- Located at */xml/search/lucene.xsl
- Input: a flat XML containing search params
- Output: a search query in tree form

```
<BooleanQuery>  
  <BooleanClause>...</BooleanClause>  
  <BooleanClause>...</BooleanClause>  
  ...  
</BooleanQuery>
```

- Used to search fields stored by the schema's indexing stylesheet

Boolean Clauses

- Attributes (values can be **true** or **false**)
 - required : the clause must be satisfied
 - prohibited : the clause must not be satisfied
- Content: One of the supported query types or a BooleanQuery
- Mapping to logic conditions

Required	Prohibited	Bool operator
false	false	or
false	true	not (*)
true	false	and
true	true	(meaningless)

* Not operator cannot be used alone

Query types

- All types have 2 attributes:
 - fld : field name stored into the index
 - txt : text string to match
- Term : matches exactly a word
`<TermQuery fld="any" txt="africa"/>`

Query types (2)

- Wildcard : matches words with a common pattern (slow). Symbols:
 - * : matches 0..n characters
 - ? : matches exactly 1 character

```
<WildcardQuery fld="any" txt="river*"/>
```
- Prefix : same as above. Matches the beginning of a word. Does not need *

```
<PrefixQuery fld="any" txt="river"/>
```

Query types (3)

- Fuzzy : lax match. Useful to match plurals or bad typed words (slow).

sim : similarity ranges from 0.0 (very lax) to 1.0 (exact match)

```
<FuzzyQuery fld="title" txt= "rivers"  
sim="0.8"/>
```

- Activated by "accuracy" control in advanced search

Query types (4)

- Phrase : matches a sequence of words.
Works when the indexed field is tokenized

```
<PhraseQuery>
```

```
  <Term fld="title" text="rivers"/>
```

```
  <Term fld="title" text="africa"/>
```

```
</PhraseQuery>
```

- Activated when search criteria is surrounded by quotes

Query types (5)

- Range : matches a range of values. The match is lexicographic **not** numeric

```
<RangeQuery fld="westBL"  
  lowerTxt="{ $westBL - 1 }"  
  upperTxt="{ $westBL + 1 }"  
  inclusive="true"/>
```

- Used only when searching by date range (dates are yyyy-mm-ddThh:mm:ss)

Query example

```
<BooleanQuery>  
  <BooleanClause required="true" prohibited="false">  
    <TermQuery fld="protocol" txt="{/request/protocol}"/>  
  </BooleanClause>  
</BooleanQuery>
```

Query example (2)

```
<BooleanQuery>
  <BooleanClause required="true" prohibited="false">
    <BooleanQuery>
      <BooleanClause required="false" prohibited="false">
        <WildcardQuery fld="protocol" txt="ogc:wms-*-get-
          map"/>
      </BooleanClause>

      <BooleanClause required="false" prohibited="false">
        <WildcardQuery fld="protocol" txt="ogc:wms-*-get-
          capabilities"/>
      </BooleanClause>
    </BooleanQuery>
  </BooleanClause>
</BooleanQuery>
```

Schema management

Schema management

- Metadata schemas are located in */xml/schemas (one folder each)
- Schemas are dynamically loaded at runtime
- Each schema has an XSD file(s) to drive the editor
- To add 'my-schema' to GeoNetwork
 - create 'my-schema' folder in */xml/schemas and fill it with proper content
 - Adjust Java code and visualization stylesheets

Changes to Java code

Changes to Java code

Areas of intervention:

- GN : DataManager
- GN/MEF : Importer and GAST : MefLib
- CSW : SearchController
- Maybe GN : SchemaLoader if the XSD schema is complex

Changes to Java code (2)

- `DataManager.autodetectSchema(...)`

Add autodetection of the new schema

- `Importer.handleInfo(...)` and `MefLib.addMetadata(...)`

```
boolean iso139 = schema.equals("iso19139");
```

```
if (!dcore && !fgdc && !iso115 && !iso139)
```

```
    throw new Exception("Unknown schema format :  
        "+schema);
```

Changes to Java code (3)

- SearchController.retrieveMetadata

```
if (schema.equals("fgdc-std") || schema.equals("dublin-  
core") || schema.equals("iso19115"))
```

```
    if (outSchema != OutputSchema.OGC_CORE)
```

```
        return null;
```

Schema's folder structure

Folder Overview

- Metadata schema in XML schema format
- Schema suggestions for the editor (xml)
- Thumbnails (extract, set, unset) (xsl)
- Uuid related (extract, set) (xsl)
- Field indexing (xsl)
- Update of fixed information (xsl)
- Conversion stylesheet for OAI-PMH
- Localized labels for metadata elements

schema-suggestions.xsl

- Is a flat XML file containing **field** elements
- It is used by the editor during an element's creation to create useful optional children

```
<field name="gmd:CI_OnlineResource">  
  <suggest name="gmd:protocol"/>  
  <suggest name="gmd:name"/>  
  <suggest name="gmd:description"/>  
</suggest>
```

extract-thumbnails.xsl

- Returns thumbnail filenames present into the metadata
 - input : metadata
 - output example:

```
<thumbnail>  
  <large>large.png</large>  
  <small>small.png</small>  
</thumbnail>
```
- Both large and small are optional.
- The filename refers to a file into the metadata's public folder

set-thumbnail.xsl

- Sets **one** thumbnail into a metadata (small or large)

- input : an XML tree

```
<root>
  <env>
    <file>large.png</file>
    <ext>png</ext>
    <type>thumbnail|large_thumbnail</type>
  </env>
  <METADATA>...</METADATA>
</root>
```

- output : input metadata with thumbnail set

unset-thumbnail.xsl

- Removes information about a thumbnail from a metadata (small or large)
 - input : an XML tree

```
<root>
  <env>
    <type>thumbnail|large_thumbnail</type>
  </env>
  <METADATA>...</METADATA>
</root>
```
 - output : input metadata with thumbnail info removed

extract-uuid.xsl

- Extracts the uuid from a metadata element
 - input : metadata
 - output : an element containing the uuid
`<uuid>...</uuid>`
- If the metadata has no uuid (like fgdc-std) the element must be empty

set-uuid.xsl

- Sets the uuid inside a metadata (if there is a place for it)
 - input : an XML tree

```
<root>
  <env>
    <uuid>...</uuid>
  </env>
  <METADATA>...</METADATA>
</root>
```
 - output : input metadata with uuid set

index-fields.xsl

- Specifies which fields must be indexed and in which way
 - input : metadata
 - output : a list of Field elements to index

```
<Field name="title" string="{string(.)}"  
store="true" index="true" token="true"/>
```
- Attributes meaning
 - store : field is stored into the index
 - index : field must be indexed
 - token : field is a phrase to tokenize

update-fixed-info.xsl

- Updates information that is fixed (like the format) or system generated (changedate)
 - input : an XML tree

```
<root>
  <env>
    <id>...</id>
    <uuid>...</uuid>
    <changeDate>...</changeDate>
    <siteURL>...</siteURL>
  </env>
  <METADATA>...</METADATA>
</root>
```

- output : input metadata with information set

labels.xml

- Flat XML file of **element** elements located in `*/xml/schemas/my-schema/loc/XX`.
 - name attribute : Fully qualified name of the element. Must be declared in the root element
 - label : Localized label of the element
 - description : General description used for tooltips
 - condition : A generic string (like 'mandatory') that is shown in red on tooltips. Different than the mandatory information inside the schema!

labels.xml (2)

- Limitation: parent is not taken into account
- Example:

```
<element name="gmd:title">  
  <label>Title</label>  
  <description>name by which the cited resource is  
    known</description>  
  <condition>mandatory</condition>  
</element>
```

codelists.xml

- Flat XML file of **codelist** elements located in `*/xml/schemas/my-schema/loc/XX`.
 - name attribute : Fully qualified name of the codelist. Must be declared in the root element
- Each codelist contains a sequence of **entry** elements. Each entry contains:
 - code : Code stored inside the metadata
 - label : Localized label shown on video
 - description : Used for tooltips

codelists.xml (2)

```
<codelist name="gmd:CI_DateTypeCode">
  <entry>
    <code>creation</code>
    <label>Creation</label>
    <description>Date identifies when the resource was
      brought into existence</description>
  </entry>
  <entry>
    <code>publication</code>
    <label>Publication</label>
    <description>Date identifies when the resource was
      issued</description>
  </entry>
</codelist>
```

Stylesheets used to view and edit metadata

Schema's stylesheet

- Create */xsl/metadata-my-schema.xsl
- The stylesheet must generate HTML code to view or edit the metadata
 - view : nested tables
 - edit : form with nested tables
- The stylesheet must match all relevant elements and render them. The match mode must be 'my-schema'
- Input parameters: **schema** and **edit**

Schema's stylesheet (2)

- Rendering facilities
 - complexElement : draws a box and recurse
 - simpleElement : draws a label/textfield
 - Both accept the current element and the **schema** and **edit** parameters
- Variable \$currTab is always defined and can be 'simple', 'advanced' or other. Indicates the currently selected tab on the left.

Schema's stylesheet (3)

- There must be a template for the simple view
- There must be a template for the brief view
 - <metadata>
 - <title>
 - <abstract>
 - <keyword>
 - <link type="url|download|wms">
 - <image type="unknown|thumbnail|overview">

Schema's stylesheet (4)

- `<geoBox>`
 - `<westBL>`
 - `<eastBL>`
 - `<southBL>`
 - `<northBL>`
- `<geonet:info>` (copied from input metadata)

Example

```
<xsl:template mode="fgdc-std" match="onlink">  
  <xsl:param name="schema"/>  
  <xsl:param name="edit"/>
```

```
<xsl:apply-templates mode="simpleElement" select=".">  
  <xsl:with-param name="schema" select="$schema"/>  
  <xsl:with-param name="edit" select="$edit"/>  
  <xsl:with-param name="text">  
    <a href="{.}"><xsl:value-of select="."/></a>  
  </xsl:with-param>  
</xsl:apply-templates>  
</xsl:template>
```

Changes to existing XSL

- metadata-edit.xsl

- Should be changed only if the schema supports thumbnails (line 212)

```
<!-- thumbnails -->
```

```
<xsl:if test="string(geonet:info/schema)='iso19115'  
or string(geonet:info/schema)='iso19139'">
```

- metadata.xsl

- The dispatching to the new schema must be added at the beginning, in template:

```
<xsl:template mode="elementEP" match="*|@*">
```

Changes to existing XSL (2)

- metadata.xsl : dispatching example

```
<xsl:when test="$schema='iso19139'">
  <xsl:apply-templates mode="iso19139" select="." >
    <xsl:with-param name="schema"
      select="$schema"/>
    <xsl:with-param name="edit"  select="$edit"/>
  </xsl:apply-templates>
</xsl:when>
```

Changes to existing XSL (3)

- metadata-utils.xsl

- Add inclusion of new stylesheet

```
<xsl:include href="metadata-my-schema.xsl"/>
```

- Add link to brief view

```
<xsl:template match="*" mode="brief">
```

```
...
```

```
<!-- ISO 19139 -->
```

```
<xsl:when test="$schema='iso19139'">
```

```
  <xsl:call-template name="iso19139Brief"/>
```

```
</xsl:when>
```

Changes to existing XSL (4)

- metadata-utils.xsl

- Add new schema to autodetect code

```
<xsl:template mode="schema" match="*">
  <xsl:choose>
    ...
    <xsl:when test="name(.)='Metadata'">iso19115
      </xsl:when>
    <xsl:when test="local-name(.)='MD_Metadata'">
      iso19139</xsl:when>
    ...
  </xsl:choose>
</xsl:template>
```

Changes to config.xml

- Add labels inclusion in config.xml inside the **geonet/default/gui** element

```
<xml name="iso19139" base="xml/schemas/  
iso19139/loc" file="labels.xml" />
```

```
<xml name="iso19139" base="xml/schemas/  
iso19139/loc" file="codelists.xml" />
```

CSW support

CSW support

- Add 'my-schema' folder in */xml/csw/schemas with the following stylesheets:
 - ogc-brief.xsl, ogc-summary.xsl, ocf-full.xsl
 - input: metadata
 - output: BriefRecord, SummaryRecord or Record in CSW format

That's all