GeoNetwork Plug-in Schemas

Simon Pigot
CSIRO (Marine and Atmospheric Research)

Simon.Pigot@csiro.au

01/12/10

Overview

- Background
    - What/Where is it?
    - How to deploy it?
- Adding a profile of ISO19115 to GeoNetwork
- Adding a schema to GeoNetwork

How to deploy?

- Deploy using the installer (easy)
  - Download installer from
    `http://geonetwork.globaldial.com/testdownloads`
  - Schema and profile plugins are available as optional installer packs or they can be installed from the URL above

Select the packs you want to install:

Note: greyed out packs are required.

☐	Core	298.82 MB
☒	ISO19115/19139 ANZLIC Metadata Schema Version 1.1	11.34 MB
☒	ISO19115/19139 MCP (Marine Community Profile) Metadata Schema Version 1.5 - Experi...	2.14 MB
☐	ISO19115/19139 MCP (Marine Community Profile) Metadata Schema Version 1.4	2.04 MB
☒	ANZMETA Metadata Schema Version 1.3	99.39 KB

Description

The profile of ISO19115/19139 for Marine users - this version 1.5 is 1.4 plus extensions for taxonomic metadata (Taxonomic Extent). This version is completely compatible with 1.4 if you don't use the taxonomic elements. Taxonomic elements may change after submission to the Marine Community Profile governance committee in November/December 2010

Total space Required: 312.4 MB

Available space: > 2 GB

(Made with IzPack - http://www.izforge.com/)

Previous Next Quit

Adding a Metadata ISO Profile

- Overview
  - Schema directory layout
  - Schema for a profile – example Marine Community Profile (MCP)
  - Implementation Principles – use ISO19139 implementation as basis, only change what we need: applies to codelists, help, schemas, schematrons
  - Presentation layer
  - Adding Schematrons
  - Using the metadata schema services in the 'Administration' menu

Schema Directory - I

- GeoNetwork stores the basic information about a built-in metadata schema in `web/geonetwork/xml/schemas`
- GeoNetwork stores plugin schemas in directory config'd in `web/geonetwork/WEB-INF/config.xml`
- Schemas that are profiles of ISO19115/19139 should be named `iso19139.<namespace_prefix>`, where `namespace_prefix` is the one used in the schema
    - for the Marine Community Profile this is

      `iso19139.mcp`

Schema Directory - II

- Apart from the schema definitions, the schema directory provides the 'handles' that GeoNetwork needs to interface with any metadata schema:
  - 'handle' for the official schema definitions (XSDs) – always `schema.xsd`
  - get/set XSLTs that are used as 'handles' for metadata elements needed by GeoNetwork eg. `extract-uuid.xsl`
  - mappings from metadata elements to Lucene fields in `index-fields.xsl`

Schema Directory Overview - I

convert	Directory of XSLTs to convert from this schema to others (rifcs, oai-dc)
loc	Directory of localized information: labels, codelists
present	Presentations xslts (main presentation xslt must be metadata-<schemaname>.xsl)
process	XSLTs for processing metadata in the editor
sample-data	Sample data for this schema
schema	Directory containing the official XSDs of the metadata schema (eg. ISO19139 + profile)
templates	Templates for this schema

Schema Directory Overview - II

extract-gml.xsl	XSLT used to extract geometry (eg. bounding boxes) from metadata for spatial indexing
extract-thumbnails.xsl	XSLT to extract thumbnails/graphicOverview/browsegraphic from metadata
extract-uuid.xsl	XSLT to extract uuid of metadata from record (eg. gmd:fileIdentifier)
schema.xsd	GeoNetwork's standard way of referencing the official XSDs
schema-autodetect.xml	Information used when identifying and detecting records that use this schema
schematron*.xsl	XSLTs generated from schematron rules – for doing content based and conditional validation tasks
index-fields.xsl	XSLT that maps elements from this metadata to indexed Lucene fields

Schema Directory Overview - III

update-fixed-info.xsl	XSLT that hard codes specified metadata elements
schema-substitutes.xml	Used to restrict choices that might be offered by the editor eg. gco:CharacterString
schema-suggestions.xml	Complex metadata elements (ie. Elements with children) that should be automatically expanded by the editor
oasis-catalog.xml	Oasis catalog maps for this schema

Defining the Profile Schema

- Basic idea is to leave ISO19139 schema untouched
- Choose a namespace prefix and a URI for profile eg. `mcp` and `http://bluenet3.antcrc.utas.edu.au/mcp`
- Use extension and restriction mechanisms from XML schema definition language
    - Define new elements in profile namespace
    - Substitute extended/restricted elements for their originals
- Extend codelists and/or add new codelists for your profile
- eg. in: `web/geonetwork/xml/schemas/iso19139.mcp/schema`
    - `extensions/mcpExtensions.xsd`
    - `resources/Codelist/gmxCodelists.xml`

Building the Profile Directory - I

Working from `schemaPlugins` directory checked out from repository:

1. Copy `iso19139.anzlic` schema directory to `iso19139.<prefix>` directory eg `iso19139.mcp`
2. Place extension schema in schema directory eg. `schema/extensions/mcpExtensions.xsd`
3. Create a `schema.xsd` which includes the extension schema
4. Modify set/extract XSLTs to include profile namespace and cope with new elements eg. `set-uuid.xsl`
5. Add profile namespace to `index-fields.xsl` and index any fields in Lucene that you want to search

Building the Profile Directory - II

Working from `schemaPlugins` (continued):

6. Modify `update-fixed-info.xsl` to include profile namespace and set profile specific values
7. Update scripts in `convert` to include profile namespace and handle profile specific elements
8. Remove all iso19139 labels from `loc/en/labels.xml` and add labels, help and suggestions for profile specific elements
9. Modify `schema-substitutes.xml` and `schema-suggestions.xml` to include new elements
10. Include `schema-autodetect.xml` info (see next slide)
11. Include presentation xslt (see later slides)

Schema Auto-detection

- Used to be hardcoded in Java
- Records from a schema can be detected using elements in the record (eg. `gmd:metadataStandardName`) or on the root element itself or by finding any element with a particular namespace
- Example: iso19139.anzlic (uses metadata element and value method):

```
<elements>
    <gmd:metadataStandardName>
      <gco:CharacterString>ANZLIC Metadata Profile: An Australian/New
Zealand Profile of AS/NZS ISO 19115:2005, Geographic information -
Metadata</gco:CharacterString>
    </gmd:metadataStandardName>
    <gmd:metadataStandardVersion>
      <gco:CharacterString>1.1</gco:CharacterString>
    </gmd:metadataStandardVersion>
  </elements>
```

Schema Auto-detection

- Example: anzmeta (detect using root element):

```
<elements type="root">
   <anzmeta/>
</elements>
```

Building the Presentation XSLT

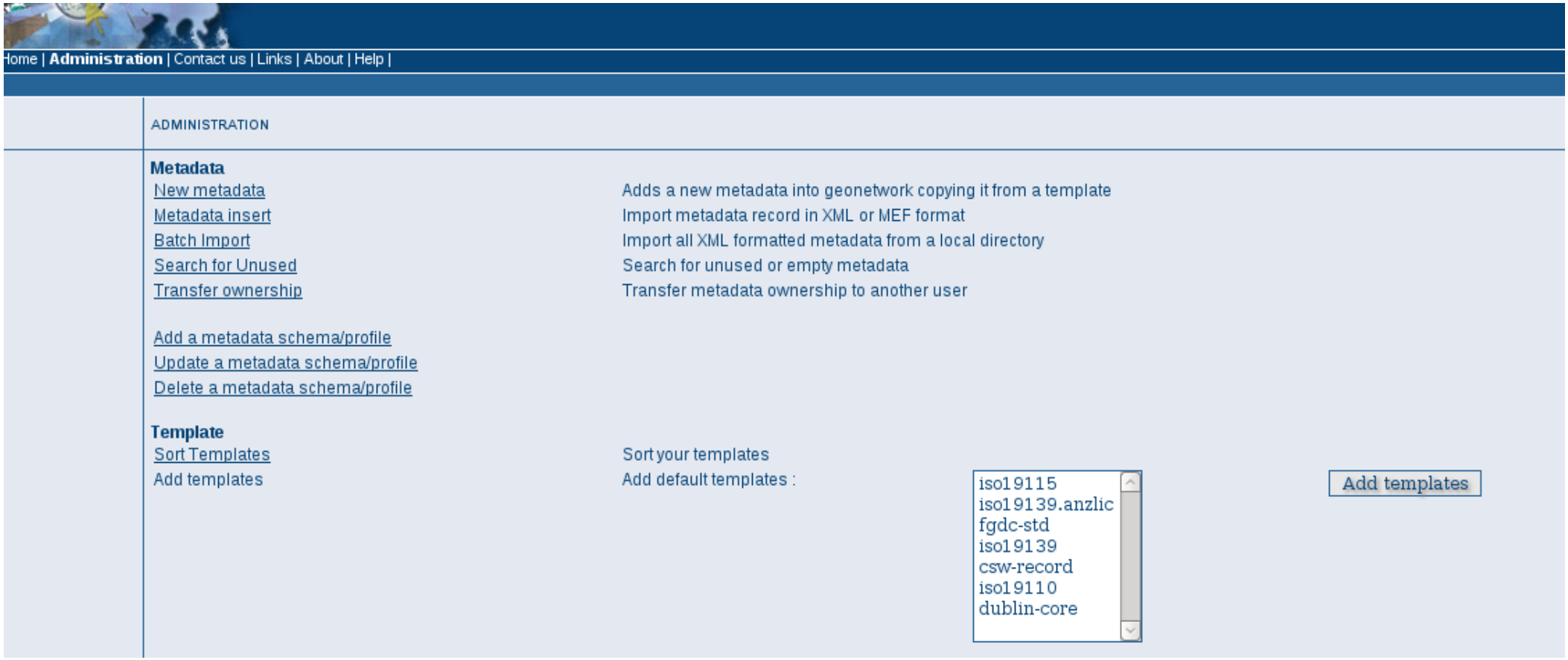
- Idea is to use basic ISO19139 presentation in place as our profile is based on ISO19139 anyway
- Working in `present` subdirectory
- Create main presentation XSLT using same naming convention as schema directory – metadata-iso19139.<profile_prefix>.xsl eg. For MCP this would be
  – `present/metadata-iso19139.mcp.xsl`
- In this XSLT:
    - Templates match on profile elements
    - Override existing elements by checking schema (eg. `gmd:keyword`) – eg. with extended codelists
    - All templates process in iso19139 mode

Building the Presentation XSLT

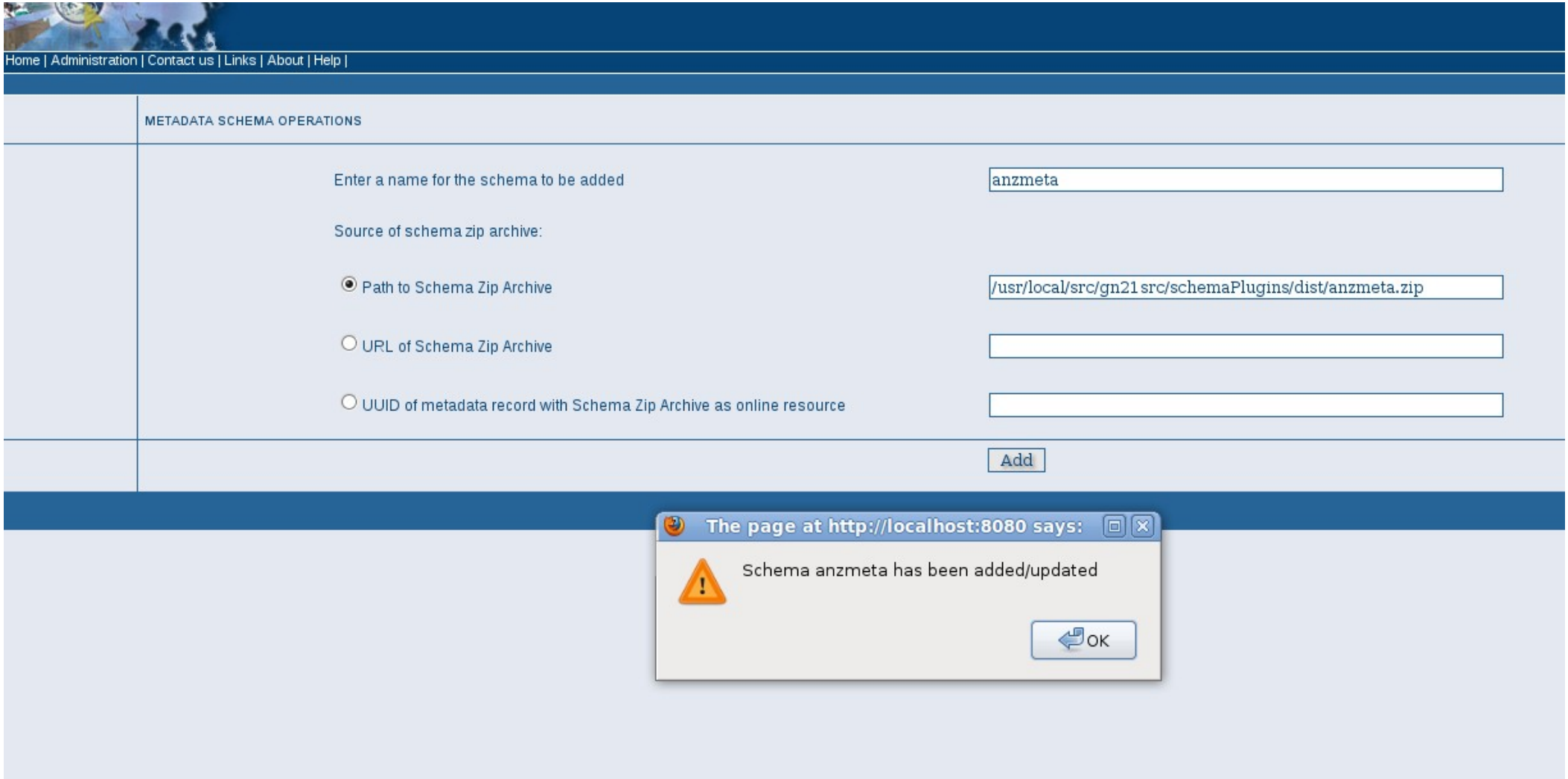
- In this XSLT (continued):
  - Must have 'main' template that is called when processing elements in this schema. eg. for anzmeta:
  - `<xsl:template match="metadata-anzmeta" name="metadata-anzmeta">`
  - Must have 'CompleteTab' to display metadata tabs in viewer/editor. eg. For anzmeta:
  - `<xsl:template match="anzmetaCompleteTab”>`
  - Must have 'Brief' template to produce format-neutral summary of metadata record for search results. eg. For anzmeta:
  - `<xsl:template match=”anzmetaBrief”>`

Other Tasks

- Define brief, summary and full presentation XSLTs for CSW OGC (csw:Record schema) and IsoRecord format:
    - Working in `present/csw`:
        - Copy iso19139 XSLTs, add profile namespace and templates for any additional or profile specific elements to be presented by the CSW service



Part of the administration menu – showing metadata schema Operations – add, update and delete (update and delete may not be available)



Adding the anzmeta schema – schemas can be zip archives on server disk, they can also be read from a URL and read from an onlineresource attached to a metadata record

What's left to be done?

- Do more checks on the schema before we plug it in:
  - Compile xslt stylesheets
  - Parse and validate XML config files
- Separate the XSDs from the editor ie. Make the editor work with XSDs that are not on the local disk
- Finish support for validating metadata schemas that use a DTD rather than an XSD (eg. anzmeta)
- Add a Class to Jeeves to accept http proxy details from GeoNetwork so they can be used by Jeeves in the oasis resolver and schemaLocation aware validation methods