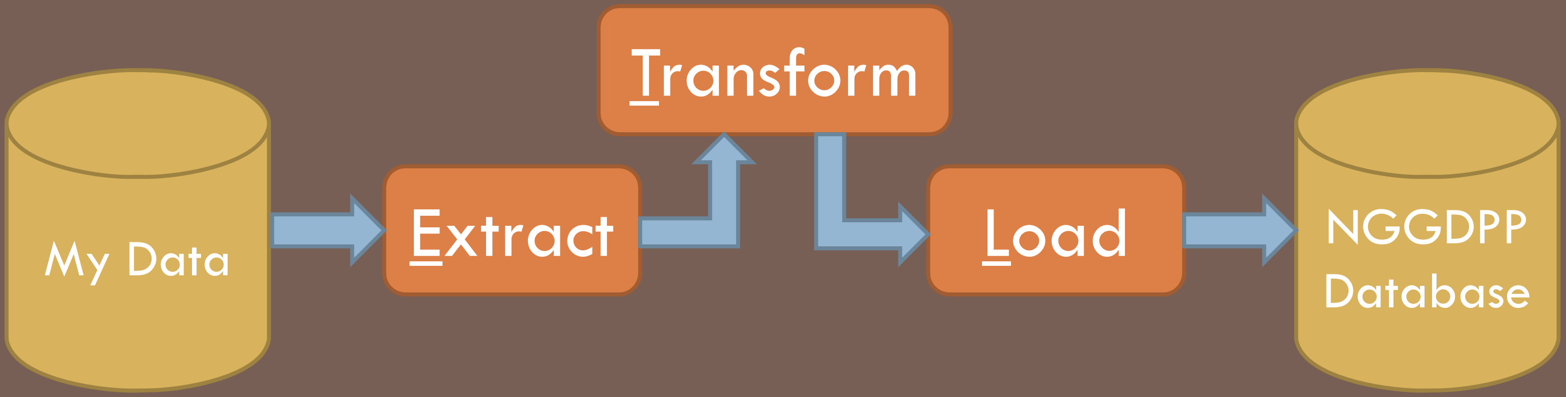


NGGDPP XML ETL

ALPHABET SOUP

BLOOMINGTON, JULY 2009

Wolfgang Grunberg, Steve Richard, Ryan Clark
Arizona Geological Survey

Overview

- AZGS metadata efforts
- The NGGDPP XML metadata format
- Why use XML format?
- XML Extract-Transform-Load (ETL) process
  - Export metadata records
  - Transform metadata records
    - Choosing the right tool
    - Creating the transformation rules
    - Executing the transformation rules
  - Load NGGDPP metadata records

What we do at AZGS

- We set up our own standards compliant Metadata Catalog Service
- Our source metadata is stored in Access, ArcSDE, Excel, etc. repositories.
- We want to serve metadata to the public in the ISO 19115/19139 metadata standard through the OpenGIS compliant CSW protocol for metadata catalog services.
- Serve metadata through deegree Java Framework with PostgreSQL or Oracle RDBMS backend.
- CSW clients in development:
  - ArcGIS Desktop
  - Geonetwork product
  - CatalogConnector application
  - GEON Portal

Review NGGDPP Metadata Format

- CSV - Comma Separated Values
  - Text file with comma, tab, pipe (|), etc. separated values.
- XML - Extensible Markup Language
  - Text files with values in custom mark-up elements (tags)

NGGDPP_sample_metadata.csv

```
collectionID,title,alternateTitle,abstract,dataType,
supplementalInformation,coordinates,alternateGeometry,
onlineResource,browseGraphic,date,datasetReferenceData,
verticalExtent¶
123456789,Primary Title,Alternate Title,A description.,
Rock CoreSupplemental information,,"-108,47",
"T2S, R3W Section 14, Northwest Quarter",
http://my.collection.gov,
http://my.collection.gov/item/graphic.png,
1939-1945,2008-12-31,"m,35.4,0"¶
```

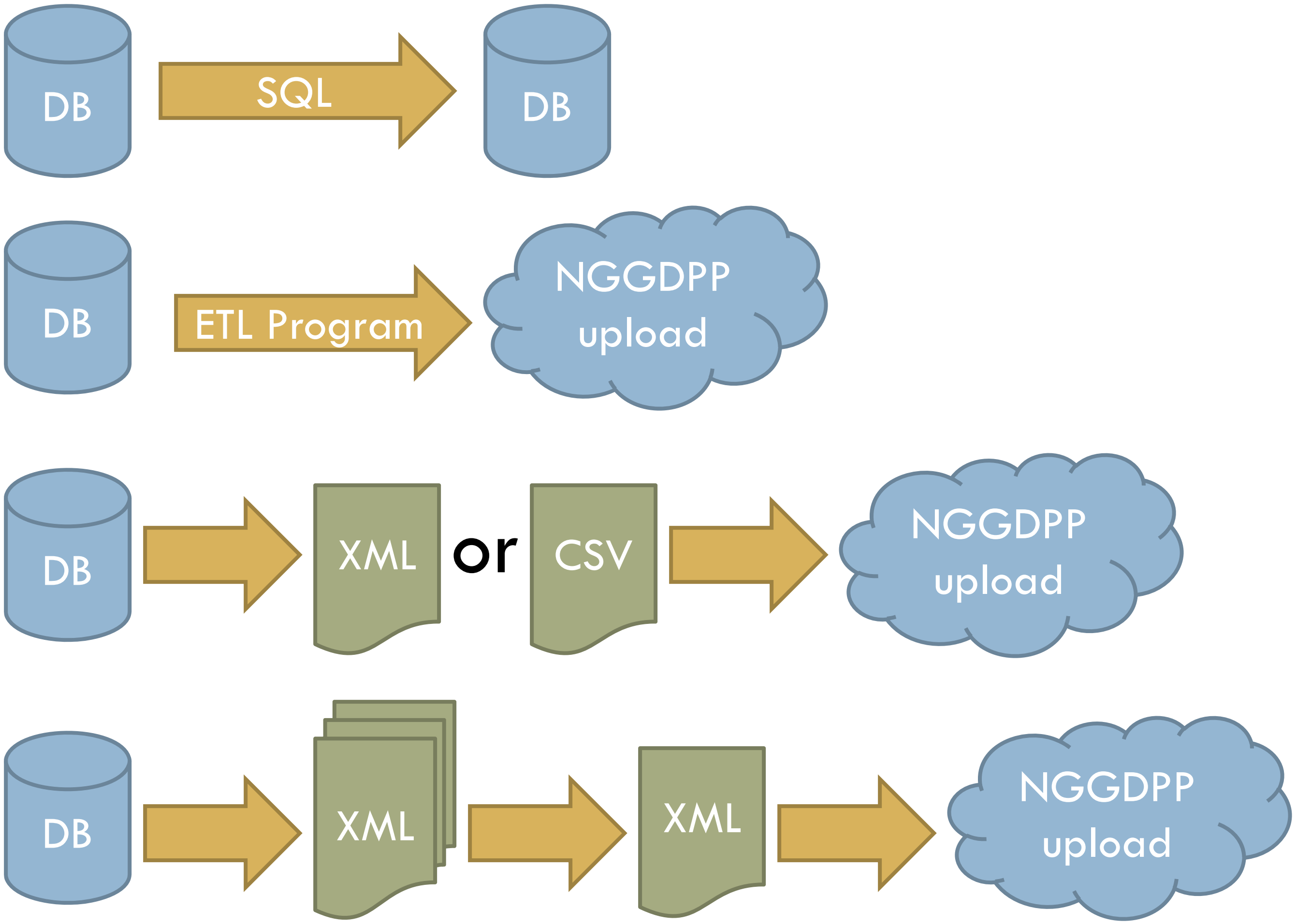
NGGDPP_sample_metadata.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<samples xmlns:xsi=“http://www.w3.org/2001/XMLSchema-instance”
         xsi:noNamespaceSchemaLocation="NGGDPPSampleMetadata-1.1_public.xsd">
    <sample>
        <collectionID>123456789</collectionID>
        <title>Primary Title</title>
        <alternateTitle>
            <title>Alternate Title1</title>
            <title>Alternate Title2</title>
        </alternateTitle>
        <abstract><![CDATA[A description.]]></abstract>
        <dataType>Rock Core</dataType>
        <supplementalInformation>
            <info>Supplemental information</info>
        </supplementalInformation>
        <coordinates>-108,47</coordinates>
        <alternateGeometry>T2S, R3W Section 14, Northwest Quarter</alternateGeometry>
        <onlineResource>
            <resourceURL>http://my.collection.gov</resourceURL>
            <resourceURL>http://my.collection.gov/item/detail?id=####</resourceURL>
        </onlineResource>
        <browseGraphic>
            <resourceURL>http://my.collection.gov/item/graphic.png</resourceURL>
            <resourceURL>ftp://my.collection.gov/item/graphic.gif</resourceURL>
        </browseGraphic>
        <dates>
            <date>1939-1945</date>
            <date>20081231</date>
        </dates>
        <datasetReferenceDate>2008-12-31</datasetReferenceDate>
        <verticalExtent>m,35.4,0</verticalExtent>
    </sample>
</samples>
```

Why use XML instead of CSV?

- The XML formatted NGGDPP metadata allows multiple values for the same field in one record such as: `alternateTitle`, `dataType` (?), `date`, `resourceURL`, `onlineResource`
- We already use XML metadata (ISO 19115/19139) to expose AZGS metadata through an interoperable CSW web service – OpenGIS' Catalogue Service.
- NGGDPP's XML file was easy to produce given our source DB, processing pipeline, and expertise.

ETL Workflow Examples



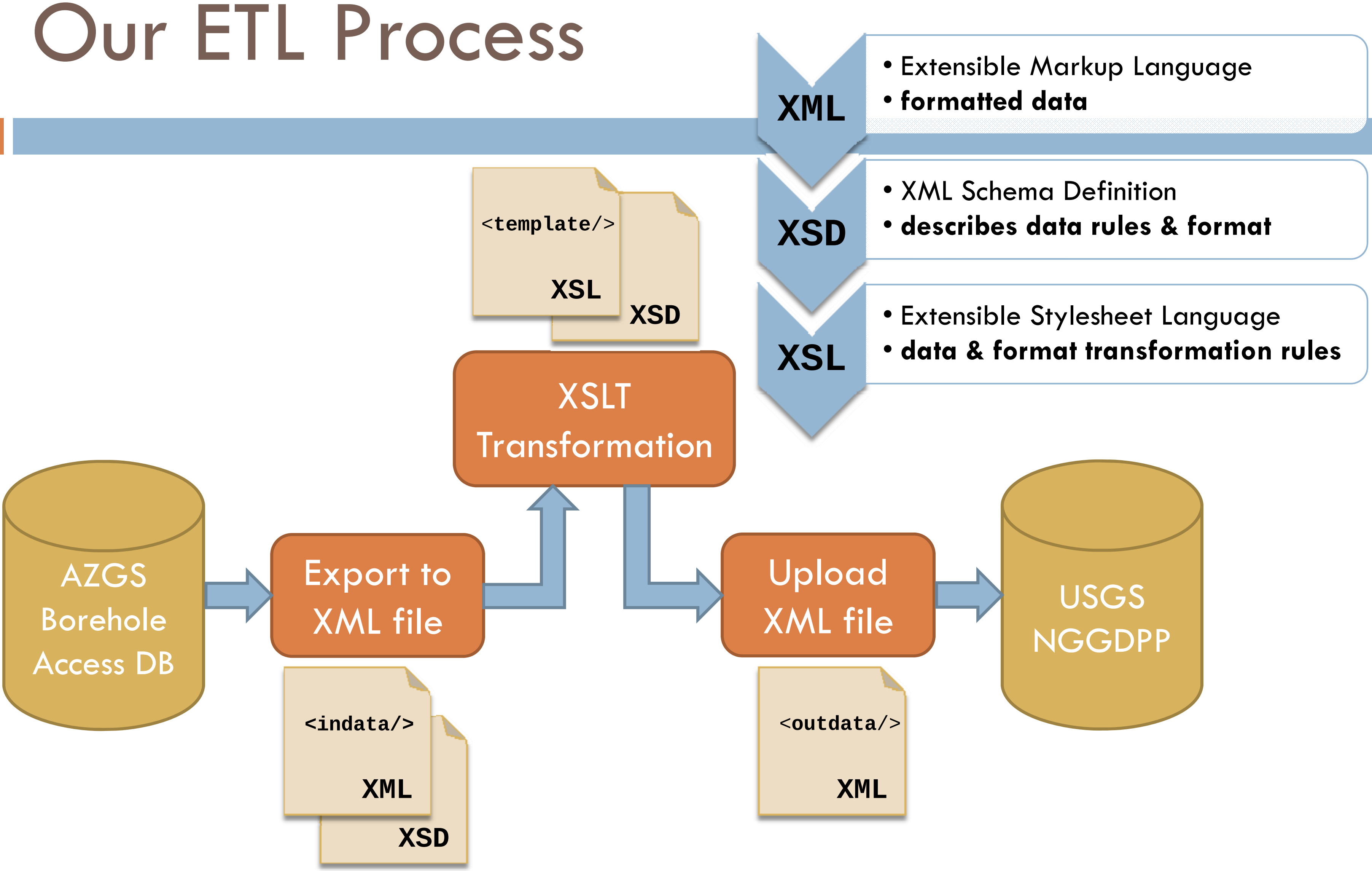
Extract, Transform, Load (ETL)

- Looked at various ETL software:
  - Open Source ETL:
    - **Talend** – powerful; limited XML tools; automatization is not free
    - **Apatar** – optimized for eBusiness services (Salesforce, QuickBooks); proprietary automatization
    - **Scriptella** – promising; no graphical mapping
  - XML oriented ETL:
    - **XMLSpy/MapForce** – nice graphical mapping; supports several automatization options, weak RDBMS connection
    - **StylusStudio/DataDirect** – strong RDBMS connection, generates Xquery too; very expensive automatization license
    - **oXygen** – difficulties with validating OGC XML schemas

ETL continued

- Easiest and most options available to just transform from XML to XML
  - Transformation scripting with XSLT, XQuery, SAX etc.
  - Many coding and parsing tools available for above languages
  - Affordable and good visual mapping tools
- Chose XMLSpy/MapForce as a compromise solution
  - Visual mapping tool generates license-free XSLT1, XSLT2, Java, C#, and C++ transformation code
  - Built-in XML schema validation and generation tools

Our ETL Process



XML
- Extensible Markup Language
- **formatted data**

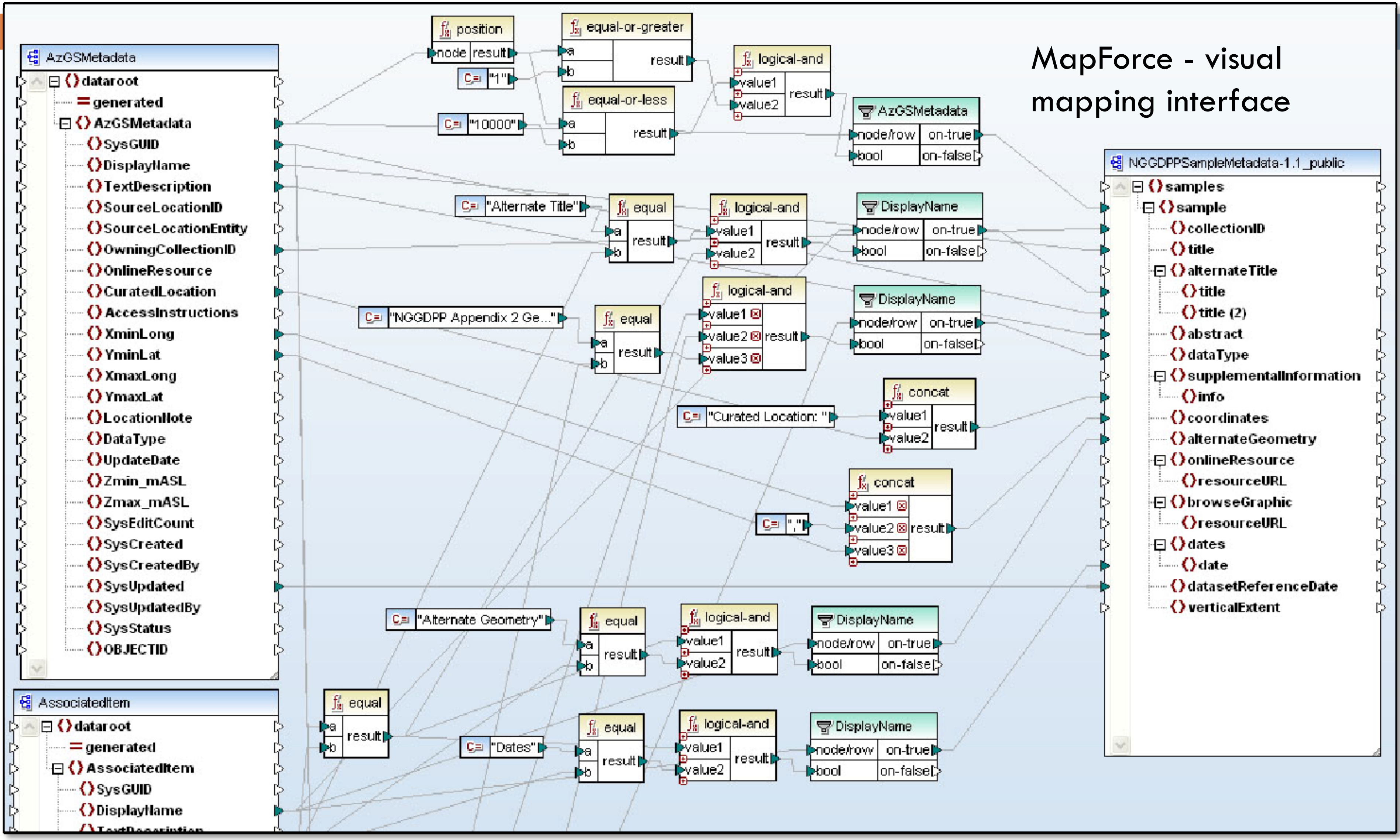
XSD
- XML Schema Definition
- **describes data rules & format**

XSL
- Extensible Stylesheet Language
- **data & format transformation rules**

Export – XML

- Onetime export and prototyping
  - Used MS Access 2007 to export DB tables to XML plus Schema Definition (XSD) files
    - Virtual MS Access tables can be links to external tables in other RDBMS such as MS SQL Server, PostgreSQL, etc.
    - Generates pretty good XSD which is needed for XSLT mapping
- Automated export
  - Still working on it …

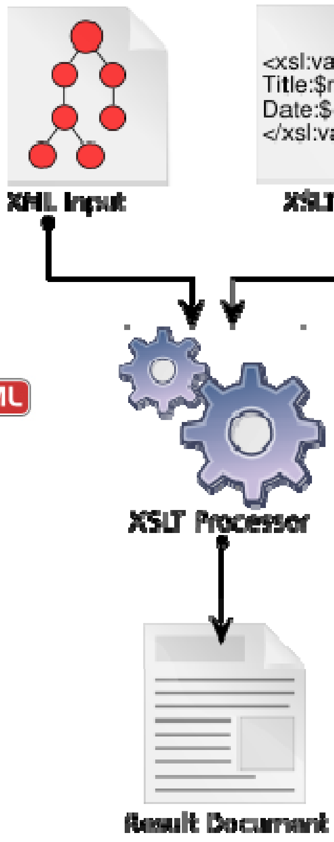
Transform – XML to XML mapping



MapForce - visual mapping interface

Transform – XSLT script

- Using XSLT1 transformation language because:
  - Easy to learn, debug, and validate with XMLSpy
  - Many XSLT1 processing software options
  - We use MSXSL.EXE **Microsoft**
    - a shell for the Microsoft XML Core Services (MSXML)
    - MSXML 3.0 to 6.0 comes with Windows OS
    - very fast compared to the free AltovaXML parser ALTOVA XML
- Disadvantages
  - XSLT1 and XSLT2 must load entire input XML tree into memory!
    - Only tested with 12,000 records so far
    - May want to switch to SAX – a stream parser.
  - Does not connect to DBs
  - Requires additional programming to automate ETL



```
<xsl:value-of>
Title:$name
Date:$curdat
</xsl:value-o
```

ETL - Load

- NGGDPP
  - Upload NGGDPP formatted XML metadata file to http://my.usgs.gov/csc/nggdpp/upload
  - XML and CSV records are fed into the NGGDPP DB
  - Magic happens! (Thanks Sky and team)

ETL Automatization/Synchronization

- □ How to automate the synchronization of the in-house DB with public catalog services?
- □ Not yet an issue with NGGDPP
- □ CSW Metadata Catalog
  - Currently use XSLT to generate a CSW metadata insert transaction
  - We use Python python™ script to execute XSLT and submit metadata insert to CSW web service.
  - Next, we want to automate DB extraction, XSLT or SAX transformation, and metadata catalog insert & update through Python.

Any questions?

Wolfgang Grunberg

Arizona Geological Survey

wgrunberg@azgs.az.gov

http://lab.usgin.org