Interoperability Data Challenge

Reading or Writing XML
Make Data Available
in a standard format and data model to enable
system integration and information exchange

IC 61968 CIM
- Maintained by IEC TC57, WG14
- Based on CIM data model
- Covers distribution & transmission
- Usually RDF XML messages

MultiSpeak
- Developed by National Rural Electric Cooperative Association (NRECA)
- Covers electric distribution utilities
- Uses XML messages
Option #1

- Leverage industry XML tools such as XSLT and XQuery
  - Users must know these complex technologies.
  - Needs XML experts to maintain
Option #2

- Custom code
  - Users must know development languages
  - Need experts to maintain
XML/GML: Strategy Options

Option #3

- Make it Easy(er) – ETL Tool
  - Minimal understanding of XML required
  - Focus on your domain expertise

Esri Data Interoperability Extension
Spatial ETL

FME Desktop
FME Workbench
Interoperability using an ETL Tool

- Separate format issues (RDF & XML) from data modeling issues (GIS to CIM Schema Mapping)
  - Map between an Electric GIS data model and CIM / Multispeak using an ETL tool
  - Create XML data exchange messages for CIM/MultiSpeak using an ETL tool
- Avoids developing custom code
- Easier to maintain
- UML data model from EA
- CIMTool for validation
- An “electric” model rather than a GIS model
- Complex data model
- Data can be transferred using RDF XML

```xml
<rdf:Description rdf:about="#cn2">
  <cim:ConnectivityNode.MemberOf_EquipmentContainer rdf:resource="#substation1"/>
  <cim:IdentifiedObject.name>cn2</cim:IdentifiedObject.name>
</rdf:Description>
```
Challenges

- Working with XML requires solving:
  - the format problem **AND**
  - the data model problem
Span the gap between the GIS and XML data representations using a Spatial ETL tool
Key challenges for XML interoperability

- GIS extract
- Schema Mapping
- Topology
- Create XML
Challenge #1: GIS Extract

**GIS extract**
- Requires a network trace or circuit query in the GIS
- Probably requires network validation
- May require extract of assets from a separate database
- Incremental changes (i.e. circuit reconfiguration) from a checkpoint or different alternatives
Challenge #2: Schema Mapping

- **Schema Mapping**

  what we have

  ![Diagram](image)

  *what we want*

  You have to know where you’re going
  i.e. DMS, OMS, SCADA…

---

Schema mapping can be one of the most time consuming tasks in an interoperability project
<table>
<thead>
<tr>
<th>Level</th>
<th>Style</th>
<th>Weight</th>
<th>Colour</th>
<th>Group</th>
<th>Geometry</th>
<th>GIS</th>
<th>Geometry</th>
<th>Domain 1</th>
<th>Domain Value 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>0</td>
<td>0</td>
<td>21</td>
<td>PRIMARY</td>
<td>shape</td>
<td>row</td>
<td>area</td>
<td>row_type</td>
<td>Railway</td>
</tr>
<tr>
<td>36</td>
<td>0</td>
<td>0</td>
<td>21</td>
<td>OPTIONAL</td>
<td>text</td>
<td>row</td>
<td>annotation</td>
<td>row_type</td>
<td>Railway</td>
</tr>
<tr>
<td>37</td>
<td>0</td>
<td>0</td>
<td>37</td>
<td>PRIMARY</td>
<td>shape</td>
<td>lot</td>
<td>area</td>
<td>dist_lot_type</td>
<td>Lot</td>
</tr>
<tr>
<td>37</td>
<td>0</td>
<td>0</td>
<td>37</td>
<td>OPTIONAL</td>
<td>text</td>
<td>lot</td>
<td>annotation</td>
<td>dist_lot_type</td>
<td>Lot</td>
</tr>
<tr>
<td>38</td>
<td>0</td>
<td>0</td>
<td>53</td>
<td>PRIMARY</td>
<td>shape</td>
<td>row</td>
<td>area</td>
<td>row_type</td>
<td>Hydro</td>
</tr>
<tr>
<td>38</td>
<td>0</td>
<td>0</td>
<td>53</td>
<td>OPTIONAL</td>
<td>text</td>
<td>row</td>
<td>annotation</td>
<td>row_type</td>
<td>Hydro</td>
</tr>
<tr>
<td>39</td>
<td>0</td>
<td>0</td>
<td>69</td>
<td>PRIMARY</td>
<td>shape</td>
<td>row</td>
<td>area</td>
<td>row_type</td>
<td>BC Gas</td>
</tr>
<tr>
<td>39</td>
<td>0</td>
<td>0</td>
<td>69</td>
<td>OPTIONAL</td>
<td>text</td>
<td>row</td>
<td>annotation</td>
<td>row_type</td>
<td>BC Gas</td>
</tr>
<tr>
<td>56</td>
<td>0</td>
<td>0</td>
<td>85</td>
<td>PRIMARY</td>
<td>shape</td>
<td>easement</td>
<td>area</td>
<td>easement_type</td>
<td>Telephone</td>
</tr>
<tr>
<td>56</td>
<td>0</td>
<td>0</td>
<td>85</td>
<td>OPTIONAL</td>
<td>text</td>
<td>easement</td>
<td>annotation</td>
<td>easement_type</td>
<td>Telephone</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>PRIMARY</td>
<td>cell</td>
<td>corner</td>
<td>location</td>
<td>corner_type</td>
<td>S&amp;PD Tie Point</td>
</tr>
<tr>
<td>57</td>
<td>0</td>
<td>0</td>
<td>101</td>
<td>OPTIONAL</td>
<td>text</td>
<td>corner</td>
<td>annotation</td>
<td>corner_type</td>
<td>S&amp;PD Tie Point</td>
</tr>
<tr>
<td>58</td>
<td>0</td>
<td>0</td>
<td>117</td>
<td>PRIMARY</td>
<td>shape</td>
<td>easement</td>
<td>area</td>
<td>easement_type</td>
<td>Other</td>
</tr>
<tr>
<td>58</td>
<td>0</td>
<td>0</td>
<td>117</td>
<td>OPTIONAL</td>
<td>text</td>
<td>easement</td>
<td>annotation</td>
<td>easement_type</td>
<td>Other</td>
</tr>
</tbody>
</table>
What is Schema Mapping?

Mapping from source to destination for:

- Attribute Names
- Attribute Values (domains or enumerated lists)
- Attribute Types
- Feature Type Names
- Geometry Change
- Dataset Change

Schema mapping is easy(er) in an ETL tool

Very easy to maintain and make changes if data models or specifications change
Challenge #3: Topology

- **Topology**
  - Addition of new topological entities
    - Terminals
    - Connectivity Nodes
    - Transformer windings
    - Split complex edges
    - Others...
GIS Transformer object

GIS Transformers are a single point entity: Transformer
CIM Topology Example: Transformer

- **CIM Transformer is multiple objects**
  - Transformer
  - Windings (primary & secondary)
  - Terminals
  - Connectivity Nodes
Challenge #4: Create XML

- **Create XML**
  - Create XML templates
  - Construct XML hierarchy – if it applies
CIM Distribution is a moving specification

Need a flexible approach to creating the XML when the specifications change
XML Expert Not Required!
Writing CIM XML
XML Writer Transformers

**XMLTemplatator**
- Convert FME feature attributes to XML

**XMLFormatter**
- Formats and beautifies XML documents

**XMLValidator**
- Ensures XML is valid
XML Template approach

- **Template approach:**
  - Steal an example and reuse it

- **Sample XML:**
  - Source of XML templates

```xml
<?xml version="1.0" encoding="utf-8"?>
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <cim:BaseVoltage rdf:ID="BaseVoltage_0.120">
    <cim:IdentifiedObject.name>BaseVoltage_0.120</cim:IdentifiedObject.name>
    <cim:BaseVoltage.nominalVoltage>0.12</cim:BaseVoltage.nominalVoltage>
  </cim:BaseVoltage>

  <cim:VoltageLevel rdf:ID="VoltageLevel_0.120">
    <cim:IdentifiedObject.name>VoltageLevel_0.120</cim:IdentifiedObject.name>
    <cim:VoltageLevel.BaseVoltage rdf:resource="#BaseVoltage_0.120"/>
    <cim:VoltageLevel.lowVoltageLimit>0.114</cim:VoltageLevel.lowVoltageLimit>
    <cim:VoltageLevel.highVoltageLimit>0.126</cim:VoltageLevel.highVoltageLimit>
  </cim:VoltageLevel>
</rdf:RDF>
```
XML Template approach

- Sample XML

```xml
<cim:BaseVoltage rdf:ID="BaseVoltage_0.120">
  <cim:IdentifiedObject.name>BaseVoltage_0.120</cim:IdentifiedObject.name>
  <cim:BaseVoltage.nominalVoltage>0.12</cim:BaseVoltage.nominalVoltage>
</cim:BaseVoltage>

- Replace attribute values with Xquery statements

```xml
<cim:BaseVoltage rdf:ID="{fme:get-attribute("_cim:ID")}">
  <cim:IdentifiedObject.name>{fme:get-attribute("_cim:BaseVoltage")}</cim:IdentifiedObject.name>
  <cim:BaseVoltage.nominalVoltage>{fme:get-attribute("_cim:nominalVoltage")}</cim:BaseVoltage.nominalVoltage>
</cim:BaseVoltage>
XML Template approach

- Replace attribute values with Xquery statements

```xml
<cim:BaseVoltage rdf:ID="{fme:get-attribute("_cim:ID")}">
  <cim:IdentifiedObject.name>{fme:get-attribute("_cim:BaseVoltage")}
  </cim:IdentifiedObject.name>
  <cim:BaseVoltage.nominalVoltage>{fme:get-attribute("_cim:nominalVoltage")}
  </cim:BaseVoltage.nominalVoltage>
</cim:BaseVoltage>
```

- Make sure you have a good GUI to help you...
XML Template approach

- Template approach to writing XML works well for:
  - Complex hierarchical XML
    - EU INSPIRE GML
  - Simple Relational or RDF XML
    - NEIM
    - MultiSpeak
    - CIM

Example CIM Output

```xml
<!-- Base Voltage -->
<rdf:Description rdf:about="#BaseVoltage_0.120">
  <cim:BaseVoltage.nominalVoltage>0.120</cim:BaseVoltage.nominalVoltage>
  <cim:IdentifiedObject.name>BaseVoltage_0.120</cim:IdentifiedObject.name>
  <rdf:type rdf:resource="cim:#BaseVoltage"></rdf:type>
</rdf:Description><!-- Base Voltage -->
<rdf:Description rdf:about="#BaseVoltage_0.720">
  <cim:BaseVoltage.nominalVoltage>0.720</cim:BaseVoltage.nominalVoltage>
  <cim:IdentifiedObject.name>BaseVoltage_0.720</cim:IdentifiedObject.name>
  <rdf:type rdf:resource="cim:#BaseVoltage"></rdf:type>
</rdf:Description>
```
Data Consumer Challenge

Reading XML
In Summary...

- ETL Tool such as **Esri Data Interoperability Extension** or **FME Desktop** can simplify your XML woes:
  - Separate format issues (RDF XML) from data modeling issues (GIS to CIM Schema Mapping)
  - Covers all aspects of the ETL problem – format, schema mapping, topology
  - Template approach – reduces need to understand XML syntax

- Reduce maintenance of the ETL process
Question & Answer
Thank You!

- For more information, contact:
  - **Info:**
    - [Mark.Stoakes@safe.com](mailto:Mark.Stoakes@safe.com)
    - (604)501-9985 ext. 211
  - **Support:**
    - [support@safe.com](mailto:support@safe.com)
    - (604)501-9985 ext. 278
Useful references

- CIM Standards Overview And Its Role in the Utility Enterprise: CIM Users Group: Prague:Terry Saxton
- System Interfaces For Distribution:
  - Management Part 13 CIM RDF Model, I.E.C.
  - Part 11: Common Information Model (CIM) Extensions for Distribution