RiseClipse: why Working at the Model Level is Better for Validating Data Conforming to IEC Standards

Dominique Marcadet
CentraleSupélec & Laboratoire de Recherche en Informatique
Context

• In the context of smart grids, more and more data is produced and exchanged
• Standards are needed to allow for this exchange of data
• These standards must specify the syntax **and** the semantic of data
• XML is often used for the syntactic level, the use of UML for the semantic level is more and more common
• Interoperability tests are conducted
Common Information Model

• Initially developed by EPRI, now a series of standards under the IEC

• Common definition for power system components for use in the EMS API
  • Extended to assets, customer billing, electricity markets...

• Theses standards use (a small part of) UML as an ontology language: the CIM is an UML model
Exchange of CIM data

• A CIM model (e.g. the description of an electric network) is a graph which must be serialized in order to be exchanged

• IEC 61970-501 defines a mapping from the CIM UML model to an RDF Schema

• IEC 61970-552 defines an XML serialization of CIM data based on RDF (the CIM RDF XML format)
Serialization of CIM data

```xml
< CIM::Line rdf:ID="l"/>

< CIM::ACLineSegment rdf:ID="acl">
  < CIM::Equipment.EquipmentContainer rdf:resource="l"/>
</ CIM::ACLineSegment/>

< CIM::Terminal rdf:ID="t1">
  < CIM::Terminal.ConductingEquipment rdf:resource="acl"/>
  < CIM::Terminal.ConnectivityNode rdf:resource="cn"/>
</ CIM::Terminal/>

< CIM::ConnectivityNode rdf:ID="cn"/>
```
Validation of CIM data

• At the syntactic level using XML tools
  • Well-formed
  • Valid (conformed to the schema)

• At the semantic level using XML tools
  • Cardinalities and type of object for the association end present in the serialized data

• Not possible with standards XML tools
  • Cardinalities and type of object for the other association end
  • Constraints on values
  • Constraints depending on values
From software engineering to IEC standards

• Model Driven Engineering is the (not so) new approach for building (software) systems
  • Models help to manage complexity
  • Modeling languages can be defined for a specific domain

• This approach used dedicated tools (validation of models, transformation of models...) based on standards (OMG is the main actor)

• These tools can be used outside software engineering if the needed pieces are presents
OMG modeling layers

<table>
<thead>
<tr>
<th>Layer</th>
<th>Description</th>
<th>Levels</th>
<th>Category</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>M3</td>
<td>MOF</td>
<td>Meta-language</td>
<td>Meta-class</td>
<td></td>
</tr>
<tr>
<td>M2</td>
<td>MOF models (e.g. UML)</td>
<td>Languages</td>
<td>Class</td>
<td></td>
</tr>
<tr>
<td>M1</td>
<td>UML models (e.g. CIM)</td>
<td>Models</td>
<td>Line</td>
<td></td>
</tr>
<tr>
<td>M0</td>
<td>Real things (e.g. power system)</td>
<td>Systems</td>
<td>The real line</td>
<td></td>
</tr>
</tbody>
</table>

- Object Constraint Language is used for specifying constraints (invariants) on models
- It can be used on models (M1) defined using a known (i.e. defined with MOF) language (M2)
- It can be used on systems (M0) modeled with UML
Eclipse Modeling Framework

• Ecore is an implementation of MOF inside Eclipse
• From an Ecore model, EMF generate Java code for manipulating data conforming to the model and for (de-)serializing these data in XML
• There is an operational implementation of OCL on top of EMF
• CimClipse was built using these technologies
  • We adapted the (de-)serialization to be compatible with CIM RDF XML
  • We added later CIM difference files, profiles…
CimClipse layers

<table>
<thead>
<tr>
<th>M3</th>
<th>Ecore</th>
<th>Meta-Language</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>M2</td>
<td>CIM</td>
<td>Languages</td>
<td>Line</td>
</tr>
<tr>
<td>M1</td>
<td>CIM models</td>
<td>Models</td>
<td>&lt;cim:Line rdf:id=&quot;l&quot;/&gt;</td>
</tr>
<tr>
<td>M0</td>
<td>Power systems</td>
<td>Data</td>
<td>The real line</td>
</tr>
</tbody>
</table>

- OCL is used to check CIM models (M1) defined using the CIM language (M2) which is itself defined using the Ecore meta-language (M3)
- CIM has been promoted to a Domain Specific Modeling Language (DSML)
Substation Configuration Language

- SCL is one of the IEC 61850 series of standards
- XML Schema is used to define the language, UML is used to illustrate the definitions
RiseClipse

• We added explicit links in the model for navigation purpose
• We also made all links bidirectional
• Our tool became agnostic with respect to (meta-)models
• And rename it RiseClipse
Future work

- Some features of CimClipse have still to be ported to RiseClipse
- Other standards (COSEC)
- We are investigating model transformations
  - Evolutions of CIM
  - CIM-IEC 61850 harmonization effort
- We have to finalize an agreement between CentraleSupélec and EDF to be able to release RiseClipse as open source
Thanks!

Questions?
Backup
OWL vs OCL

• OWL:
  
  ```xml
  <rdf:Description rdf:nodeID="..." >
    <owl:minCardinality rdf:datatype="int">1</owl:minCardinality>
    <owl:onProperty
      rdf:resource="Terminal.ConnectivityNode"/>
    <rdf:type rdf:resource="Restriction"/>
  </rdf:Description>
  ```

• OCL:
  
  ```ocl
  context Terminal inv:
  self.ConnectivityNode <> null
  ```
CimTool vs OCL

- CimTool:
  ```
  problem("Isolated node")
  <-(?n rdf:type ConnectivityNode)
  countLessThan(2 * Terminal.ConnectivityNode ?n)
  ```

- OCL:
  ```
  context ConnectivityNode inv "Isolated node":
  self.Terminals->size() >= 2
  ```
Inheritance

• OCL:

context ACLineSegment inv:
   self.EquipmentContainer.oclIsTypeOf(Line)

context Equipment inv:
   not self.oclIsTypeOf(ACLineSegment) implies
   self.EquipmentContainer.oclIsTypeOf(Bay)
   xor
   self.EquipmentContainer.oclIsTypeOf(VoltageLevel)
Smart grids domains