The Semantically Enabled Smart Grid

Grid-Interop 2009, Semantic Models
Wednesday, November 18, 2009

Andrew Crapo (crapo@research.ge.com)
GE Global Research, Niskayuna, NY

Xiaofeng Wang (wang.Xiaofeng@ge.com)
GE Energy, Melbourne, FL

John Lizzi (lizzi@research.ge.com)
GE Global Research, Niskayuna, NY

Ron Larson (ron.larson@ge.com)
GE Energy, Melbourne, FL

Copyright General Electric, 2009
Outline

1. Challenges
2. Semantic Technology as Enabler
3. The Foundation of Semantic Models
4. Semantic Models Compared to...
5. Semantic Application Design Language (SADL)
Challenges

Interoperability
- System ↔ System, System ↔ Humans
- Between Standards (Hundreds of Them!)

Efficiency
- Implementing and Maintaining Evolving System
- Managing Distributed, Intelligent Systems to Achieve Business Purpose
Smart Grid Use Case Example: Demand Response

Demand Response Example Diagram

- Price Signal
- Meter & Bill
- Reduce Usage
- Subscription
Semantics As Enabler: Example

Core distribution model:
- **Meter**
  - `customerID` (range string)
  - `reading` (range `Reading`)
    - `value` (range `Amount`)
    - `time` (range `DateTime`)

Extension model:
- **SmartMeter** is a type of **Meter**
  - `price` (range `Price`)
    - `value` (range `Amount`)
    - `startTime` (range `DateTime`)
    - `endTime` (range `DateTime`)

Measurement model:
- **Amount**
  - `value` (range float)
  - `unit` (range `Unit`)

Billing model:
- **Invoice**
  - `lineItem` (range `EnergyCost`)
    - `pricePoint` (range `Price`)
      - `consumption` (range `Usage`)
        - `value` (range `Amount`)
        - `cost` (range `Amount`)
  - `lineItem` (range `EnergyCost`)
    - `pricePoint` (range `Price`)
      - `consumption` (range `Usage`)
        - `value` (range `Amount`)
        - `cost` (range `Amount`)

Key: → is imported by

- **Semantic Models Can Be Modular, Extensible**
- **Reasoners/Classifiers Can Handle Logical Implications**
- **Rules Can Do Conversions, Calculations, Business Logic**

Copyright General Electric, 2009
Models

1) Capture our understanding of the World
2) Enable creativity and problem-solving
3) Enable creation of artifacts in the World
What’s In a Model?

- Classes (generalizations of things)
- Properties (generalizations of attributes, relationships)
- Instances of things (individuals)
- Statements about the above (instances of properties)
How are Models Captured? (that is, outside our heads)

<table>
<thead>
<tr>
<th>Natural language</th>
<th>flexible, often ambiguous; ambiguity allows modeling process to start</th>
</tr>
</thead>
<tbody>
<tr>
<td>PowerPoint</td>
<td>even more ambiguous, often misleading, not computational</td>
</tr>
<tr>
<td>Databases</td>
<td>mostly capture property values; abstractions (ontological commitments) of the model are implicit in the DB schema</td>
</tr>
<tr>
<td>Code (e.g., Java)</td>
<td>often a limited extraction, context and hierarchy of concepts implicit, often focuses on the procedure/behavior (script)</td>
</tr>
<tr>
<td>XML documents</td>
<td>data is self describing, achieves syntactic interoperability (but what do the tags mean?)</td>
</tr>
<tr>
<td>RDF/RDFS</td>
<td>weak ontology language but achieves both syntactic and semantic interoperability</td>
</tr>
<tr>
<td>OWL</td>
<td>stronger ontology language, enables much richer definitions via restrictions and much more powerful reasoning</td>
</tr>
<tr>
<td>Rules</td>
<td>can provide bridge between declarative and procedural; should be built on strong semantics (ontology)</td>
</tr>
</tbody>
</table>
An Example of an XML Schema (XSD)

```xml
<xs:complexType name="MeterReading">
  <xs:annotation>
    <xs:documentation>Used to convey quantities that are measured by a meter.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="MeterAsset" type="m:MeterAsset"/>
    <xs:element name="IntervalBlocks" type="m:IntervalBlock" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Readings" type="m:Reading" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="startTime" type="m:AbsoluteDateTime" use="optional">
    <xs:annotation>
      <xs:documentation>Start time of the data items contained within the MeterReading</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="endTime" type="m:AbsoluteDateTime" use="optional">
    <xs:annotation>
      <xs:documentation>End time of the data items contained within the MeterReading</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

Copyright General Electric, 2009
Example: Breaker in RDF (from CIM)

```xml
<rdfs:Class rdf:about="http://iec.ch/TC57/2001/CIM-schema-cim10#Switch">
  <rdfs:label xml:lang="en">Switch</rdfs:label>
  <rdfs:comment>A generic device designed to close, or open, or both, one or more electric circuits. The typeName attribute may be used to indicate that the database switch does not represent a corresponding real device but has been introduced for modeling purposes only.</rdfs:comment>
  <cims:profile>Nerc</cims:profile>
</rdfs:Class>

<rdfs:Class rdf:about="http://iec.ch/TC57/2001/CIM-schema-cim10#Breaker">
  <rdfs:label xml:lang="en">Breaker</rdfs:label>
  <rdfs:comment>A mechanical switching device capable of making, carrying, and breaking currents under normal circuit conditions and also making, carrying for a specified time, and breaking currents under specified abnormal circuit conditions e.g. those of short circuit. The typeName is the type of breaker, e.g., oil, air blast, vacuum, SF6.</rdfs:comment>
  <cims:profile>Nerc</cims:profile>
  <rdfs:subClassOf rdf:resource="http://iec.ch/TC57/2001/CIM-schema-cim10#Switch"/>
</rdfs:Class>
```

Copyright General Electric, 2009

Grid-Interop 2009
CIM RDF Model in Protégé-Owl
Observation 1: Models are Graphs

What's In a Model?

- **Classes** (generalizations of things)
- **Properties** (generalizations of attributes, relationships)
- **Instances** of things (individuals)
- Statements about the above (instances of properties)

```
Mammal
  ^
  v
Dog
  ^
  v
Lassie
```

```
Lassie ownedBy Jeff
```

```
Mammal subClassOf Dog
Dog instanceOf Property
```

Copyright General Electric, 2009
Observation 1: Models are Graphs

What's In a Model?

- **Classes** (generalizations of things)
- **Properties** (generalizations of attributes, relationships)
- **Instances** of things (individuals)
- **Statements** about the above (instances of properties)

Diagram:
- Mammal
- Dog
- Lassie
- Jeff
- Property

Arrows indicate relationships:
- **subClassOf**
- **instanceOf**
- **ownedBy**
Observation 2: Schemas and Semantic Models are Fundamentally Different

• Schema Structures are Domain-Specific
  – Metadata is for Domain
  – Useful for Document/Message Definition

• Semantic Model Structures are Domain-Independent
  – For OWL/RDF, Structure is the RDF Triple
  – Analogous to Grammar in Language

• The Tags of a Schema Structure May Be Defined In a Semantic Model

Semantics and Schemas are Synergistic
Figure 1. The GridWise Architecture Council’s eight-layer stack provides a context for determining Smart Grid interoperability requirements and defining exchanges of information.

from “NIST Framework and Roadmap for Smart Grid Interoperability Standards”
New IT capabilities required for smart grid, understood by both sides; domain knowledge essential for success, understood by subject matter experts. Need shared models, understood by IT professionals.
Semantic Application Design Language (SADL)

- Controlled English, unambiguous mapping to OWL
- Rule expressed in “formula syntax” using model concepts, translated to SWRL or Jena Rules
- Eclipse-based IDE provides

**Designed to enable semantic model comprehension and creation by subject matter experts**
The SADL IDE

Shape is a top-level class. 
area describes Shape has values of type float.

Circle is a type of Shape. 
radius describes Circle has values of type float.

Rule AreaOfCircle
  given
  x is any Circle
  if
  then
  area of x = (radius of x ^ 2) * 3.14159 .

Rectangle is a type of Shape.
height describes Rectangle has values of type float.
width describes Rectangle has values of type float.

Rule AreaOfRectangle
  given
  x is any Rectangle
  if
  then
  area of x = height of x * width of x .

MyCircle is a Circle, has radius 3.5.
MyRect is a Rectangle, has height 3.5, has width 4.5.
Authoring Helps

Real-time error checking

Templates and completion proposals
Model Lifecycle Support

Version control w/ differencing via Eclipse (CVS, SVN, etc.)

Debugging, querying, model composition and pedigree

Testing/regression testing
ConductingEquipment is a type of Equipment, described by fromConnect with values of type Terminal, described by toConnect with values of type Terminal, described by isolationCompliance with a single value of type boolean.

```java
package com.ge.grc.gridex;
import java.util.List;

public class ConductingEquipment extends Equipment {
    private List<Terminal> fromConnects;
    private List<Terminal> toConnects;
    private boolean isolationCompliance;

    public boolean addToConnects(Terminal toConnect) {
        if (!toConnects.contains(toConnect)) {
            toConnects.add(toConnect);
            return true;
        }
        return false;
    }

    public boolean removeToConnects(Terminal toConnect) {
        return toConnects.remove(toConnect);
    }

    public void setToConnects(List<Terminal> toConnects) {
        this.toConnects = toConnects;
    }

    public List<Terminal> getToConnects() {
        return toConnects;
    }

    public boolean addFromConnects(Terminal fromConnect) {
        if (!fromConnects.contains(fromConnect)) {
            fromConnects.add(fromConnect);
            return true;
        }
        return false;
    }

    public boolean removeFromConnects(Terminal fromConnect) {
        return fromConnects.remove(fromConnect);
    }

    public void setFromConnects(List<Terminal> fromConnects) {
        this.fromConnects = fromConnects;
    }

    public List<Terminal> getFromConnects() {
        return fromConnects;
    }

    public boolean isIsolationCompliance() {
        return isolationCompliance;
    }
}
```
Domain Rule: SADL vs. Jena Rule

uri "http://sadl.imp/GridInteropExample".

import "file://SelectedCim.sadl" as SelectedCim.

// Desired relationship of Breaker to Disconnector on each side:
//   Disconnector --toConnect--> Terminal --connectivityNode--> ConnectivityNode --terminal--> Terminal--fromConnect--> Breaker --toConnect--> Terminal --connectivityNode --> ConnectivityNode --terminal--> Terminal --fromConnect--> Disconnector

Rule BreakerIsolationConforms
given b is a Breaker
if e1 is toConnect of connectivityNode of terminal of fromConnect of b
e2 is fromConnect of terminal of connectivityNode of toConnect of b
e1 is a Disconnector
e2 is a Disconnector
then isolationCompliance of b is true.

Rule BreakerIsolationFromViolation
given b is a Breaker
if e is toConnect of connectivityNode of terminal of fromConnect of b
e is not a Disconnector
then isolationCompliance of b is false.

Rule BreakerIsolationToViolation
given b is a Breaker
if e is fromConnect of terminal of connectivityNode of toConnect of b
e is not a Disconnector
then isolationCompliance of b is false.
SADL is NOT competing with existing standards

It is a TOOL to make semantic modeling standards (OWL + rules) accessible to domain experts
SADL builds on Open Source and is Open Source

1. The Eclipse environment, more particularly Eclipse 3.2.2 or later, see http://eclipse.org.
2. The Eclipse IDE Meta-tooling Platform (IMP, from IBM's Watson Research Center, see http://eclipse.org/imp)
3. The Jena Semantic Web Framework (HP Labs, see http://jena.sourceforge.net/)
4. The OWL API, which is similar to Jena but supports SWRL (University of Manchester, see http://owlapi.sourceforge.net/)
5. The Pellet OWL DL reasoner (Clark & Parsia, see http://pellet.owldl.com/).
6. SADL Open Source Version (see http://sadl.sourceforge.net/)
Summary

• Smart Grid Requires Interoperability at Multiple Levels

• Semantic Technology Is An Important Enabler of Interoperability
  – Open, Shared Information Model
  – Reduced Cost, Increased Flexibility

• SADL Makes Semantic Technology More Accessible

For information contact A. W. Crapo: crapo@research.ge.com