Transforming, Testing & Explaining Smart Grid Interoperability Models
Some Smart Grid Standards

Abstract Model – Shared Concepts, Fragments???

Grid Operations & Planning

Service Provider

C12.19

CIM 61970

CIM 61968

Multispeak

ICCP

DER (solar...)

ISA88

ISA95

61850-420 DER (solar...)

61850

IEEE c37.239 comfede

IEEE 1815 dnp3

Transmission & Distribution

Substations

Crews

Protection

Protection

Independent Generation

Controllable Generation

Controllable Load

Buildings

Custom

Meters

WS Calendar

EMIX

CIM 62325

OpenADR

OASIS energy interop

61850-410 hydro

61400-25-2 wind

61850-420 DER (solar...)

naspi

IEEE 1815 dnp3

IEEE c37.239 comfede

ISO 16484 BACnet

ASHRAE SPC201 FSGiM

CEA 709 LonTalk

62351-7 comm net and system mgnt

61850-420

DER

(energy)

CIM 61968

Abstract Model – Shared Concepts, Fragments???

Grid Operations & Planning

Service Provider

C12.19

CIM 61970

CIM 61968

Multispeak

ICCP

DER (solar...)

ISA88

ISA95

61850-420 DER (solar...)

61850

IEEE c37.239 comfede

IEEE 1815 dnp3

Transmission & Distribution

Substations

Crews

Protection

Protection

Independent Generation

Controllable Generation

Controllable Load

Buildings

Custom

Meters

WS Calendar

EMIX

CIM 62325

OpenADR

OASIS energy interop

61850-410 hydro

61400-25-2 wind

61850-420 DER (solar...)

naspi

IEEE 1815 dnp3

IEEE c37.239 comfede

ISO 16484 BACnet

ASHRAE SPC201 FSGiM

CEA 709 LonTalk

62351-7 comm net and system mgnt

61850-420

DER

(energy)
Work presented today

Testing the integrity and design of smart grid standards (designed in UML)

- Transform to OWL
- Do integrity testing
- Facilitate harmonization with adjacent standards

Testing conformance to the standards by implementations

- Build reference data sets
- Explain the values of the data sets
Semantic Testbed

Testing the integrity of a standard

Standards

Text

UML

XSD

Model Transformation

Semantic representation
Web Ontology Language (OWL)

Model Verification Testing
UML to OWL SPARQLMotion script

Converts XMI to RDF triples. SXML mapping reads the XML tree and generates instances against OWL class definitions for UML constructs.

Converts UML-based RDF triples to OWL classes, properties and instances.

Iterates over packages selected by the user and generates an ontology for each: Copies relevant triples from the conversion; constructs import triples to identify dependent ontologies that were generated from other packages or coming from external sources.
Custom Query Manager Editor

Model Name: FSGIM
Model Version: ppr1
Query Classification: Class

Query Name: --- Please select an option ---

Query Prefix:
- Classes defined but never referred to in a relation
- Classes that share substantially the same properties
- Display the UML package hierarchy for the class containing "string"
- Find all classes in "sourcenamespace" that refer to a class or datatype in "destinationnamespace"
- Find all classes that refer to a class or datatype in "namespace"
- Identify all external classes pointing to FSGIM classes

Query:
- Navigate up superclasses to find the ultimate parent class, for all classes containing a given string
- Superclasses that have no properties

Result Display: ○ Truncated ○ Complete

Execute Query  Save Query  Clear Results
Benefits

- Exhaustively searches a standard to find errors that might escape human detection
  - Orphan definitions (defined but never used)
  - Opportunities for model refactoring (similar classes)
  - Disallowed changes to imported standards
  - Redundant classes and properties
  - Non-standard data type definitions
After Verification Testing

- **Model Healing**
  - Recommendations to correct errors
  - Automatic error correction for native OWL specifications

- **Conformance Testing**
  - Does a particular implementation properly represent the information according to the standard?
  - Generation of reference data sets

- **Standards Harmonization**
  - Checking for missing information
    - Information present in one standard but not in another
    - Mapping among different ways of modeling the same information
Goals

To test the integrity and design of smart grid standards (designed in UML)

– Transform to OWL
– Do integrity testing
– Facilitate harmonization with adjacent standards

To test conformance to the standards by implementations

– Build reference data sets
– Explain the values of the data sets
A Power Aggregation Example
Input Datasets

Dataset #1 – System definition
- Declare all the entities
  - Circuits, ComponentElements, Routers
- Declare ConnectionPoints, connect entities to Circuits
- Specify RouterConnectionPoint properties
- Define the EM hierarchy, and specify which entities are managed by which EM
- Specify Generator properties

Dataset #2 – Power readings
- Provide all measured power and energy data
Statistics

- 79 rules so far
- 1739 triples inferred (so far) for this power aggregation reference data set
The Testing & Explanation Environment
Missing Data
A Power Aggregation Rule

**DemandRuleset**

- name = DemandRuleset
- nameType = Standard Rulesets
- nameTypeAuthority = ASHRAE 201 Standard

**notes**

Calculate "metered demand":
- For each direct subordinate that is an instance of the ElectricMeter Class where only
  1. direct subordinate instances of the Load Class (including instances of the CurtailableLoad Class) and/or
  2. direct subordinate instances of the EM Class where EM.hasElectricalGenerators is False are connected to its output ConnectionPoint and where ElectricMeter.powerReading is available, sum ElectricMeter.powerReading.

Calculate "demand from unmetered loads":
- For each direct subordinate that is an instance of the Load Class (including instances of the CurtailableLoad Class) that is not included in "metered demand" and where Load.actualDemand is available, sum Load.actualDemand.

Calculate "demand from energy managers":
- For each direct subordinate that is an instance of the EM Class that is not included in "metered demand" and where EMPresentData.presentAggregateDemand is available, sum the subordinate EMPresentData.presentAggregateDemand.aggregateQuantity.

Calculate final result:
- Sum "metered demand", "demand from unmetered loads", and "demand from energy managers."
A Power Aggregation Example
Explaining Aggregate Demand Calculations

Resource Form
Name: example12.5:DemandFromUnmeteredLoadsFloor2Circuit4

Annotations
Other Properties
explain:createdBy
standard_aggregations---collections_rulesets_and_aggregations
explain:hasCircuit
example12.5:Circuit4
explain:hasCollection
example12.5:CollectionFloor2
explain:hasComputedAggregation
37.0
explain:hasEMPPresentData
example12.5:EMPPresentDataFloor2Circuit4
explain:hasLoads
example12.5:Lighting2
dexample12.5:Server
dexample12.5:Unmanaged2
explain:supportsComponentElement
dexample12.5:Floor2
rdf:type
explain:AggregateDemand

Resource Form
Name: example12.5:CollectionFloor2

Annotations
Other Properties
explain:createdBy
energy_manager_component---device_and_model_components:EMPPresentData
explain:supportsComponentElement
dexample12.5:Floor2
device---device_and_model_components:hasMRID
dexample12.5:CollectionFloor2
device---device_and_model_components:hasName
dCollectionFloor2
device---device_and_model_components:hasNameType
dStandard Collections
device---device_and_model_components:hasNameTypeAuthority
dASHRAE203 Standard
device---device_and_model_components:hasPartOfRef
dexample12.5:DeviceCollectionFloor2
device---device_and_model_components:hasPartRef
dexample12.5:HVAC2
dexample12.5:Lighting2
dexample12.5:PVArray
dexample12.5:Server
dexample12.5:Submeter
dexample12.5:Unmanaged2
Benefits

- Reference data sets are useful in traditional conformance testing exercises

- Natural language rules are codified

- Explanation mechanism can help implementers understand how the rules within a standard should function, using either reference datasets or their own datasets
Summary

- Using semantic representations
  - expose inconsistencies in new standards
  - codify natural language rules
  - use automated reasoning for instantiation and explanation