Using SPARQL/OWL for Validation of Smart Grid Standards

Semantic Harmonization of Smart Grid Concepts

Surbhi Dangi, Steven Ray, Ralph Hodgson*, Gokhan Soydan*, Ankur Oberai*

Carnegie Mellon University Silicon Valley, *TopQuadrant

NASA Research Park, Building 23 - Moffett Field, CA 94043

surbhi.dangi@sv.cmu.du, steve.ray@sv.cmu.edu, rhodgson@topquadrant.com, gsoydan@topquadrant.com, ankur@topquadrant.com

Keywords: OWL representation, SPARQL, Smart Grid, FSGIM, Ontology, RDF

Abstract

This project concerns the application of semantic modeling techniques to demonstrate how the smart grid community can manage the large challenge of harmonizing definitions and relations among the several hundred standards that currently constitute the suite of smart grid standards under consideration or development. Our research captures definitions, relations and constraints in a machine-computable form (using ontology modeling and query languages RDF, OWL, SPIN and SPARQL) for one of these standards that addresses energy usage information. We have deployed the transformed model and a web application to query the model on the cloud for use by the smart grid community. The tool provides automated reasoning to uncover contradictions and inconsistencies in the standard, collect model statistics and metrics and eventually, align terms amongst different standards. We have found that obscure errors that had escaped notice during UML modeling were more easily discovered with this approach.

1. INTRODUCTION

While modern information technologies have transformed much of the economy, the electricity sector has not yet embraced and implemented these technologies. "Smart grid", as defined by the Illinois Smart Grid Initiative [1], "*refers to the modernization of the electric system through the integration of new information-age technologies, new strategic public policies, and allows for new uses of the electric grid, both in operations and through new customer side applications, that extract the benefits of more efficient operation, more efficient use of grid assets, and more cost-effective expansion of the electric grid.*"

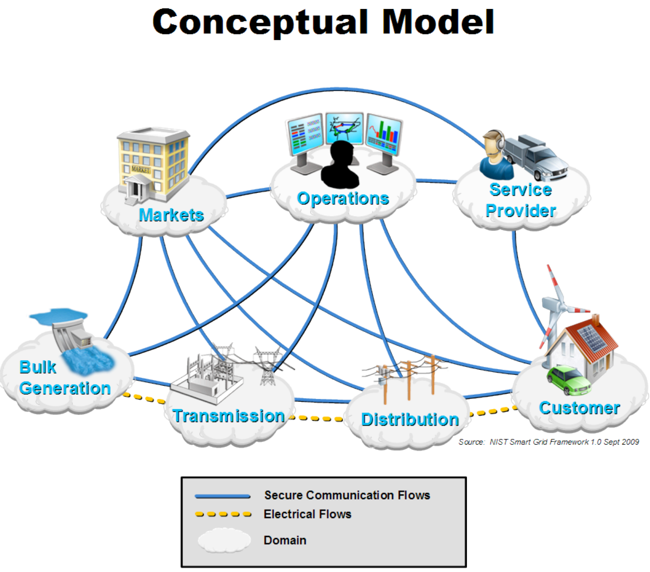


Figure 1: Conceptual Smart grid model

As described on the smart grid collaboration site [2], "*By integrating an end-to-end, advanced communications infrastructure into the electric power system, a smart grid can provide consumers near real-time information on their energy use, support pricing that reflects changes in supply and demand, and enable smart appliances and devices to help consumers avoid higher energy bills.*"

Our project concerns the application of semantic modeling techniques to demonstrate how the smart grid community can manage the large challenge of harmonizing definitions and relations among the 200+ standards that currently constitute the suite of smart grid standards under consideration or development.

2. APPLICATION AREA

Smart grid standardization is different from other standards activities because smart grid standards must be designed for interoperability among the seven contexts shown in Fig. 1

and involves three distinct industry sectors - IT, energy and telecom, each with their existing value chains and standards. This study focuses on the validation of one of these standards - the ASHRAE [3] /NEMA [4] SPC201 standard for exchanging energy information within a facility. This standard centers around the definition of the Facility Smart Grid Information Model (FSGIM) [5].

The FSGIM model is being developed to represent the customer domain of the electricity model in response to the requirements of PAP-17[1] in SGIP [6] initiated by NIST [7]. The FSGIM model is comprised of four central components to represent energy information

1. Load
2. Meter
3. Generator
4. Energy manager

Four modeling teams, operating somewhat independently, initially developed the four FSGIM model components. Therefore, the potential need for harmonization among these components is high. In addition, because the model components intentionally import portions of other smart grid standards, notably OASIS eMIX (Energy Market Information Exchange) [8], NAESB Energy Usage Information [9], IEC 61850-7 [10], OASIS WS-Calendar (WSCPIM) [11] and WXXM (Weather) [12], it is important to harmonize with those definitions as well.

The process of developing a standard is largely a manual one. With four modeling teams developing the four core component classes, there is a possibility of inconsistent vocabularies and data-type declarations arising from disparate reference standards that can affect the integrity of the standard. The payback for reducing these kinds of errors and inconsistencies is great.

3. APPROACH

There are other techniques in use for converting UML to OWL, such as using the QVT approach [14] [15]. Conversely, some groups have defined mappings in the reverse direction in order to use the UML methodology and tools to create, edit and view OWL ontologies [16]. We believe the approach used in this paper i.e., SPARQL-based scripts are at least as powerful as the QVT method. A stepwise description of the process is detailed in this section.

3.1. Model Transformation

As might be expected, converting an information model of any kind from one representation into another can be a daunting task. In our case, conversion from a UML model to an OWL representation is a promising approach since OWL is a richer representation and can thus capture all the relevant UML concepts. (It is important to note that this claim is specifically talking about the information model portion of the UML methodology, not the UML process model, swimlanes, etc.). Many of the advantages and challenges of migrating the representation to OWL are also described in [14]. To accomplish the conversion, a scripting language called SPARQLMotion [17] was used to design the data flow from an XMI [18] representation created as an export from the Enterprise Architect™ UML modeling tool to UML-schema-based RDF [19] triples using SXML[2] [20], then mapping those RDF triples to well-structured OWL classes and properties using SPIN [21] and finally resulting in a set of Turtle [22] files suitable for loading into an ontology editor such as TopBraid Composer™.

3.2. Development of the SPARQL Query Editor

The SPARQL Query Editor is a web-based application that provides an interactive user interface to query the FSGIM model. The web application can be accessed at http://fsgim.sv.cmu.edu. The SPARQL Query Editor gives domain experts, who may or may not be ontology experts, a simple mechanism to query and validate vocabularies and data-types in the standard. The SPARQL Query Editor, an instance of TopBraid Live™ and the transformed FSGIM model are deployed on the Amazon Cloud to enable both faster query response times and to accommodate future needs of querying multiple standards using the web client.

3.3. Model Validation

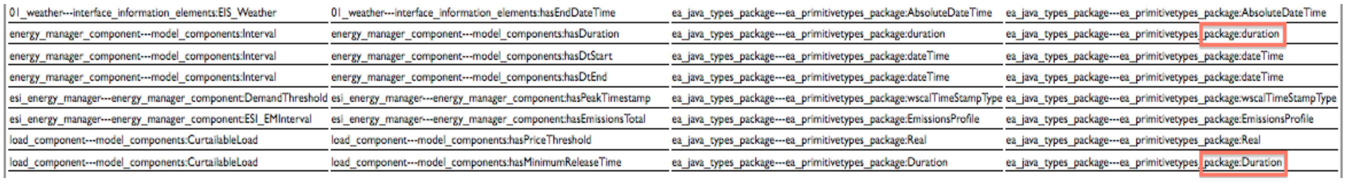
All of the activities in this task were carried out by means of SPARQL queries.

3.3.1. Uncovering non-standard use of datatypes and units of measure

---

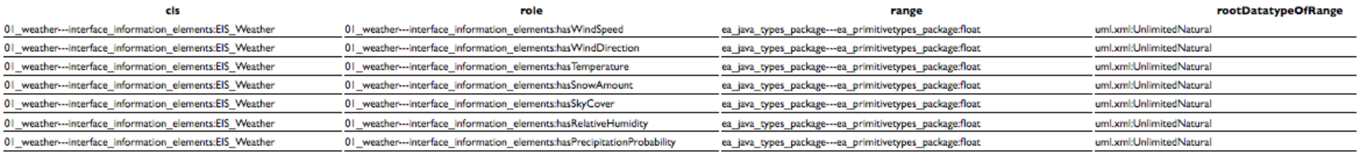
[1] PAP-17 - A Priority Action Plan maps the applications of a standard to use cases. Specifically, PAP-17 covers the requirements to develop a standard to model the flow of electrical load information between control systems and end use devices found in single- and multi-family homes, commercial and institutional buildings, and industrial facilities. (http://collaborate.nist.gov/twiki-sggrid/bin/view/SmartGrid/PAP17FacilitySmartGridInformationStandard)

[2] SXML - Semantic XML is a XML to RDF/OWL mapping capability (http://composing-the-semantic-web.blogspot.com/2007/11/xmap-mapping-arbitrary-xml-documents-to.html)



01_weather---interface_information_elements:EIS_Weather	01_weather---interface_information_elements:hasEndDateTime	ea_java_types_package---ea_primitivetypes_package:AbsoluteDateTime	ea_java_types_package---ea_primitivetypes_package:AbsoluteDateTime
energy_manager_component---model_components:Interval	energy_manager_component---model_components:hasDuration	ea_java_types_package---ea_primitivetypes_package:duration	ea_java_types_package---ea_primitivetypes_package:duration
energy_manager_component---model_components:Interval	energy_manager_component---model_components:hasDtStart	ea_java_types_package---ea_primitivetypes_package:dateTime	ea_java_types_package---ea_primitivetypes_package:dateTime
energy_manager_component---model_components:Interval	energy_manager_component---model_components:hasDtEnd	ea_java_types_package---ea_primitivetypes_package:dateTime	ea_java_types_package---ea_primitivetypes_package:dateTime
esi_energy_manager---energy_manager_component:DemandThreshold	esi_energy_manager---energy_manager_component:hasPeakTimestamp	ea_java_types_package---ea_primitivetypes_package:wscalTimeStampType	ea_java_types_package---ea_primitivetypes_package:wscalTimeStampType
esi_energy_manager---energy_manager_component:ESI_EMInterval	esi_energy_manager---energy_manager_component:hasEmissionsTotal	ea_java_types_package---ea_primitivetypes_package:EmissionsProfile	ea_java_types_package---ea_primitivetypes_package:EmissionsProfile
load_component---model_components:CurtailableLoad	load_component---model_components:hasPriceThreshold	ea_java_types_package---ea_primitivetypes_package:Real	ea_java_types_package---ea_primitivetypes_package:Real
load_component---model_components:CurtailableLoad	load_component---model_components:hasMinimumReleaseTime	ea_java_types_package---ea_primitivetypes_package:Duration	ea_java_types_package---ea_primitivetypes_package:Duration

Figure 2: Partial result snapshot for the query to uncover data-types not inheriting from standard definitions or common primitive types



cls	role	range	rootDatatypeOfRange
01_weather---interface_information_elements:EIS_Weather	01_weather---interface_information_elements:hasWindSpeed	ea_java_types_package---ea_primitivetypes_package:float	uml.xml:UnlimitedNatural
01_weather---interface_information_elements:EIS_Weather	01_weather---interface_information_elements:hasWindDirection	ea_java_types_package---ea_primitivetypes_package:float	uml.xml:UnlimitedNatural
01_weather---interface_information_elements:EIS_Weather	01_weather---interface_information_elements:hasTemperature	ea_java_types_package---ea_primitivetypes_package:float	uml.xml:UnlimitedNatural
01_weather---interface_information_elements:EIS_Weather	01_weather---interface_information_elements:hasSnowAmount	ea_java_types_package---ea_primitivetypes_package:float	uml.xml:UnlimitedNatural
01_weather---interface_information_elements:EIS_Weather	01_weather---interface_information_elements:hasSkyCover	ea_java_types_package---ea_primitivetypes_package:float	uml.xml:UnlimitedNatural
01_weather---interface_information_elements:EIS_Weather	01_weather---interface_information_elements:hasRelativeHumidity	ea_java_types_package---ea_primitivetypes_package:float	uml.xml:UnlimitedNatural
01_weather---interface_information_elements:EIS_Weather	01_weather---interface_information_elements:hasPrecipitationProbability	ea_java_types_package---ea_primitivetypes_package:float	uml.xml:UnlimitedNatural

Figure 3: Partial result snapshot for the query to uncover data-types that **do** inherit from standard definitions or common primitive types

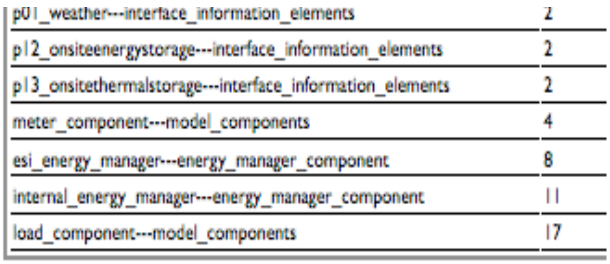
We wrote queries to check for consistent use of terminology, consistent inheritance of datatypes (inheriting from appropriate OMG and W3C standards, for example), and proper importing of classes from other smart grid standards. Figure 2 (Query #1) shows a SPARQL query and part of its results, where we search for all data type declarations in FSGIM that do not explicitly inherit from either the UML or W3C primitive data type specifications. This query was written because use of a common definition for datatypes such as "string" "integer" "real" as well as dimensioned types such as "duration" "timestamp" and even "price" would greatly improve coherence and interoperability of messages. ISO 8601 [23], the International standard for representation of dates and times, defines 6 levels of granularity for the date and time formats that may be referenced by standards. However, ISO 8601 does not support references to a URI. Therefore, there is not yet an easy way to inherit from a rigorous and machine-readable definition of time-related quantity types. One promising candidate for such a definition is the QUDT ontology [24], but it is not yet a recognized standard. In contrast, Figure 3 (Query #9) shows a similar query where the declared datatypes do inherit from UML or W3C.

3.3.2. Model statistics and metrics

Statistics about models such as counts of the class definitions that a component refers to and the number of redundant classes in a model enable us to find refactoring opportunities. To give the users insight into the number of distinct concepts in the four core component packages, we wrote a query (Query #12) to give us a count of the number of classes. The results from Figure 4 allow us to think about components that refer to an abnormally large number of class definitions. Queries (Query #3) that expose classes that are defined but never used in a relation give us additional suggestions for simplifying the model. Components such as PowerQuality and EnergyEmissions from Figure 5 may be ideal candidates in the refactoring process. In addition, we construct a query (Query #4) to expose classes that share substantially the same properties.

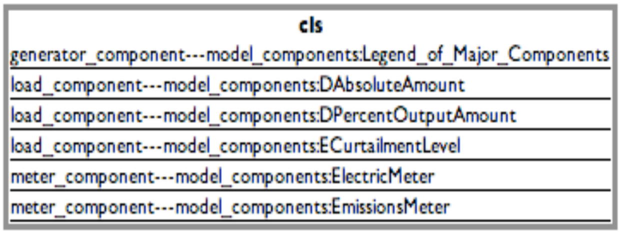
4. CONCLUSION

In working with SPARQL queries over the transformed FSGIM model, we have found that obscure errors that had escaped notice during UML modeling were more easily



p01_weather---interface_information_elements	2
p12_onsiteenergystorage---interface_information_elements	2
p13_onsitethermalstorage---interface_information_elements	2
meter_component---model_components	4
esi_energy_manager---energy_manager_component	8
internal_energy_manager---energy_manager_component	11
load_component---model_components	17

Figure 4: Partial result snapshot for the query to list the number of core component classes in the FSGIM model



cls
generator_component---model_components:Legend_of_Major_Components
load_component---model_components:DAbsoluteAmount
load_component---model_components:DPercentOutputAmount
load_component---model_components:ECurtailmentLevel
meter_component---model_components:ElectricMeter
meter_component---model_components:EmissionsMeter

Figure 5: Partial result snapshot for the query to list classes that are defined but never referred to in a relation

discovered. Some were simple typographic errors such as datatypes of “duration” and “Duration” coexisting in the model. Other anomalies required navigation through the inheritance chains within the model to uncover, such as the existence of superclasses and subclasses both declaring largely similar properties, leading to the possibility of refactoring the model. In the end, the kinds of errors or design anomalies that can be discovered is limited only by the ability of a query designer to ferret out the problem. The advantage is that an automated reasoner can be thorough in examining the entire model automatically, in contrast to a laborious manual validation of a UML model.

5. FUTURE WORK

To date, a systematic, rigorous approach to managing the definitions and relations between concepts among all the standards continues to elude the SGIP community. In the future, we intend to take the principles demonstrated in our work to date and apply them more broadly to show their general applicability across all of the smart grid content standards (i.e. at the so-called semantic level of the GWAC stack). We will continue to support the development and modifications to the FSGIM, but hope to also expand our scope to examine some of the “adjacent” standards, such as the NAESB model supporting SGIP PAP 10, EI/EMIX/WS-Calendar, IEC 61850 and SEP 2.0. Our intent is to show the same kinds of benefits as were demonstrated among the four components of the FSGIM. In addition, we plan to research the applicability of inferencing over sets of these models to determine the value of automated recommendation of improvements to the design of the models, and to aid in the harmonization of collections of standards.

References

[1] *Empowering Consumers - Through a Modern Electric Grid*, Summary Report, Illinois Smart Grid Initiative, April 2009, http://www.cnt.org/repository/ISGI.SummaryReport.pdf

[2] NIST Smart Grid Collaboration Wiki – Smart Grid Interoperability Panel Site – https://collaborate.nist.gov/twiki-sggrid/bin/view/SmartGrid/WebHome

[3] ASHRAE – American Society of Heating, Refrigerating Air-conditioning Engineering, http://www.ashrae.org/home/

[4] NEMA – National Electrical Manufacturers Association, http://www.nema.org

[5] FSGIM – ASHRAE Standard Project Committee 201 (SPC 201) Facility Smart Grid Information Model, http://spc201.ashraepcs.org/

[6] SGIP – Smart Grid Interoperability Panel, http://collaborate.nist.gov/twiki-sggrid/bin/view/SmartGrid/WebHome

[7] NIST – National Institute of Standards and Technology, http://www.nist.gov/smartgrid/

[8] Energy Market Information Exchange (EMIX) Version 1.0. 11 January 2012, OASIS Committee Specification 02. http://docs.oasis-open.org/emix/emix/v1.0/cs02/emix-v1.0-cs02.html

[9] North American Energy Standards Board – Energy Usage Information Model, http://www.naesb.org/pdf4/naesb_energy_usage_information_model.pdf

[10] IEC/TR 61850-1 Ed.2. Communication networks and systems in substations - Part 1: Introduction and overview. Forecast Publication Date: 2013-03-04

[11] WS-Calendar Version 1.0. 17 June 2011, OASIS Committee Specification Draft 04 / Public Review Draft 03. http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/csprd03/ws-calendar-spec-v1.0-csprd03.pdf

[12] WXXM – Weather Exchange Model, https://wiki.ucar.edu/display/NNEWD/WXXM

[14] Andrew Crapo, Katrina Griffith, Ankesh Khandelwal, John Lizzi, Abha Moitra, Xiaofeng Wang, *Overcoming Challenges Using the CIM as a Semantic Model for Energy Applications*, Grid Interop Forum 2010, (http://www.pointview.com/data/files/3/2433/1730.pdf)

[15] Ivan Kurtev, *State of the Art of QVT: A Model Transformation Language Standard*, Third International Symposium, AGTIVE 2007, Kassel, Germany, October 10-12, 2007.

[16] Sara Brockmans, Raphael Volz, Andreas Eberhart, and Peter Loffler, *Visual Modeling of OWL DL Ontologies Using UML*, ISWC 2004, LNCS 3298, pp. 198–213, Springer-Verlag Berlin Heidelberg 2004.

[17] Holger Knublauch, SPARQLMotion Version 2.1.0, February 01, 2010, http://sparqlmotion.org/

[18] XMI 2.0, an adopted standard of the OMG, http://www.omg.org

[19] RDF – Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/rdf-concepts/

[20] Oleg Kiselyov, SXML Specification Revision 2.1, ACM SIGPLAN Notices v.37, June 2002 p52-58

[21] SPIN – SPARQL Inferencing Notation: Overview and Motivation, W3C Member Submission, 22 February 2011, http://www.w3.org/Submission/spin-overview/

[22] Turtle – Terse RDF Triple Language, W3C Team Submission, 28 March 2011, http://www.w3.org/TeamSubmission/turtle/

[23] ISO 8601:2004(E), "Data elements and interchange formats -- Information interchange -- Representation of dates and times," http://dotat.at/tmp/ISO_8601-2004_E.pdf

[24] QUDT – Quantities, Units, Dimensions and Data Types in OWL and XML, http://www.qudt.org

Acknowledgement

The work reported on in this paper was generously supported by Grant 60NANB11D170 from the National Institute of Standards and technology.

Biography

Surbhi Dangi is a Researcher at Carnegie Mellon University Silicon Valley, where she is exploring the use of current web-based voting technologies (audio-only and multimedia voting machines and web-based support of vote-by-mail ballots) to improve the voting experience for individuals with disabilities. In her research on information interoperability, Miss Dangi was responsible for the development of a web-based tool to achieve the high-level goal of ensuring interoperability amongst the various smart grid standards. Miss Dangi holds a Masters degree in Software Engineering from Carnegie Mellon University Silicon Valley and specialized in application development on the Siebel eCRM platform for AT&T at IBM for 2 years prior to Carnegie Mellon.

Steven Ray is a Distinguished Research Fellow at Carnegie Mellon University Silicon Valley, where he researches information interoperability and standards in application domains including the smart electrical grid, electronic business, supply chains, and manufacturing. Dr. Ray has a 27 year track record of initiating and leading technical R&D projects at the National Institute of Standards and Technology. For the past decade, he was responsible for the management of a $10-13M/year division of 60 staff and visiting researchers dedicated to the solution of national problems related to measurements and standards supporting systems interoperation in the manufacturing sector. He led the establishment of "TIMA—Technologies for the Integration of Manufacturing Applications" on assignment to the NIST Advanced Technology Program. He served as Chairman of the IGES/PDES Organization that coordinated the U.S. participation in the STEP product data standard (STEP: Standard for the Exchange of Product Model Data, ISO 10303). Dr. Ray has twice been awarded the Department of Commerce Bronze Medal. Dr. Ray holds a Ph.D. in Physics from Princeton University.

Ralph Hodgson is a co-founder and the CTO of TopQuadrant, Inc., a US-based company that specializes in semantic technology consulting, training, and tools. Dr. Hodgson combines expertise in semantic technologies with over 25 years of experience in business application development and deployment, consulting, software development, and strategic planning. Prior to starting TopQuadrant in 2001, he held executive consulting positions at IBM Global Services where he was a founding member of the Object Technology Practice. Prior to IBM, he was European Technology Director, founder, and Managing Director of Interactive Development Environments, which was an international CASE tools vendor. Between 1996 and 2000, he organized ACM OOPSLA work- shops on "System Envisioning." He is the creator of the oeGOV ontologies and an author of the VAEM, VOAG and QUDT ontologies. As a consultant, he has worked with NASA on Ontologies, Schemas and Vocabularies for Space Systems Interoperability.

Gokhan Soydan is a semantic solution developer in TopQuadrant, Inc. Mr. Soydan has been working in TopQuadrant for 6 years specializing as a Java and SPARQL developer. He has designed and developed many applications using widely-known TopQuadrant tools and technologies such as TopBraid Suite – a platform to develop, integrate and deploy semantic technologies, SPIN (SPARQL Inferencing Notation) – a SPARQL based inferencing language and SPARQLMotion – a SPARQL based scripting language. He was one of the main developers of TopBraid Composer contributing towards numerous features. He was also one of the main designers and implementers of a CCTS electronic message builder project for the Netherlands Ministry of Justice. He has also been involved in various semantic projects for NASA as a part of TopQuadrant.

Ankur Oberai is a Semantic Solutions Developer at TopQuadrant Inc. Mr Oberai joined TopQuadrant in 2012 and works on the TopBraid Suite of products. At TopQuadrant, he designed a web-based testing framework for the new product line in the TopBraid Suite. He has previously worked on web mash up's and payment-system integration. He holds a Masters degree in Computer Science from the University of Georgia where his topic of research was "Optimizing SPARQL Queries on Multi-core Processors".