# Combining Multiple CIM-Based Models for OpenADR Interface Standardization

Bruce Bartell [#1], Gerald Gray [*2], Justin Berger [3]

[#] *Xtensible Solutions*
*6312 S. Fiddlers Green Circle, Suite 210E*
*Greenwood Village, CO 80111*
[1] bbartell@xtensible.net

[*] *Guiding Principle Consulting*
*PO Box 381*
*Laingsburg, MI 48848 USA*
[2] gerald.gray@guiding-principle.com

*Justin Berger*
*3464 South Eagle Street*
*Aurora, CO 80014*
[3] j.david.berger@gmail.com

***Abstract*. Establishing repeatable and consistent methods for leveraging multiple standard information models as a basis for standardizing interfaces between systems is essential to the success of Smart Grid initiatives. While the CIM working groups (WG 14 and WG 13) have begun promoting a concept of CIM Profiles, which are essentially restricted subsets of the larger CIM UML model, there has not been a clear method or toolset available to represent CIM Profiles in UML, combine multiple models for a specific implementation, and consistently generate integration artifacts, such as XML Schema (XSD), from the CIM Profiles. The CIM EA add-in to Enterprise Architect (EA) can be used to both create CIM Profiles as UML models in EA and subsequently generate integration artifacts from them. This paper describes how the OpenADR Task force used CIM EA to generate standardized XML Schema by combining five UML models: the IEC CIM, the IRC Wholesale Information Model for Demand Response, OpenADR Extensions, SEP2.0 – Direct Load Control, and SEP2.0 – Pricing Data.**

## I. Introduction

The OpenADR Task Force defines the systems requirements, principles, best practices, and services required for standardizing control and pricing signals for Demand Response (DR) and Distributed Energy Resources (DER) as part of the Smart Grid implementation. The Task Force was charged with developing OpenADR Service Definitions based on the principles and requirements defined in the OpenADR 1.0 System Requirements Specification. Additionally, the Task Force needed to easily create and maintain CIM Profiles based on five different UML models and iteratively generate OpenADR XML Schemas as the Service Definitions matured.

### *A. CIM Profiles*

While both IEC WG 13 and WG 14 mention "CIM Profiles" in various draft documents, a well-defined specification for a CIM Profile and the rules associated with a Profile have not yet been formalized. While CIM Profile formalization is expected in the near future, the Task Force needed an exact specification for a CIM Profile to move forward with a CIM Profile-based approach for OpenADR. As applied to OpenADR, a CIM Profile:

- is defined in UML
- is a restricted subset of the CIM UML[1]
- is self-contained
- contains traceability to the original model
- preserves the inheritance structures of the original model

### *B. Specifying UML to XML Schema Transformation*

In addition to an exact specification of a CIM Profile, the OpenADR Task Force needed a repeatable process for transforming a CIM Profile in UML to an XML Schema. The Task Force agreed that the *IEC 62361-100 Naming and Design Rules for CIM Profiles to XML Schema Mapping* [1] document provided the explicit rule set for UML to XSD transformation. This transformation needed to be programmatically repeatable to ensure consistent XSD generation as the CIM and CIM Profiles for OpenADR evolved.

## II. Creating the OpenADR Model

### *A. Leveraging Existing Reference Models*

[1] See Reference [3] for a more formal definition

To create the OpenADR UML model, the Task Force used EA's XMI Import feature to import individual UML models that provided the information concepts required by OpenADR. The Task Force used five reference models:

- IEC CIM
- IRC Wholesale Information Model for Demand Response
- OpenADR Extensions
- SEP2.0 – Direct Load Control
- SEP2.0 – Pricing Data

To simplify the project structure, the Task Force imported the five models into a single root package named "Reference Models."

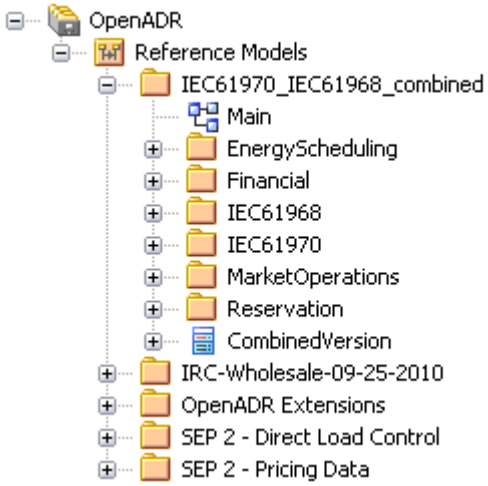



Figure 1: The Project Browser view in EA, after importing all of the reference models.

### B. *Creating the OpenADR CIM Profiles*

Over the course of multiple months, a small team of core editors defined the OpenADR Profiles. The Task Force used CIM EA to create CIM Profiles as UML models directly in EA. Using CIM EA provided an easy way to select the classes and attributes required in the OpenADR Standard Definitions. As the editors revised and extended definitions, the OpenADR team used CIM EA to update the CIM Profile to reflect the changes.

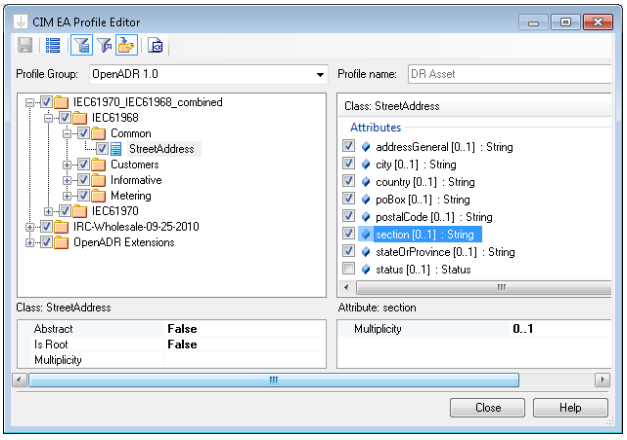
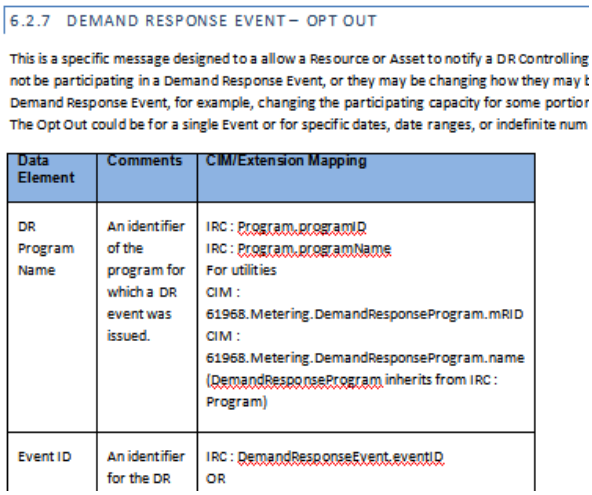



Figure 2: CIM EA's Profile Editor interface [4]



6.2.7 DEMAND RESPONSE EVENT – OPT OUT

This is a specific message designed to a allow a Resource or Asset to notify a DR Controlling not be participating in a Demand Response Event, or they may be changing how they may b Demand Response Event, for example, changing the participating capacity for some portion The Opt Out could be for a single Event or for specific dates, date ranges, or indefinite num

| Data Element | Comments | CIM/Extension Mapping |
|---|---|---|
| DR Program Name | An identifier of the program for which a DR event was issued. | IRC : Program.programID<br>IRC : Program.programName<br>For utilities<br>CIM : 61968.Metering.DemandResponseProgram.mRID<br>CIM : 61968.Metering.DemandResponseProgram.name<br>(DemandResponseProgram inherits from IRC : Program) |
| Event ID | An identifier for the DR | IRC : DemandResponseEvent.eventID<br>OR |

Figure 3: An example of the Profile definitions, available in full in [2]

### C. *Generating XSD*

CIM EA fully supports the 62361-100 specification for XSD generation. Using the OpenADR Profiles as input, the team used CIM EA's Generate Artifacts capability, shown in Figure 4, to create the OpenADR XML Schemas.

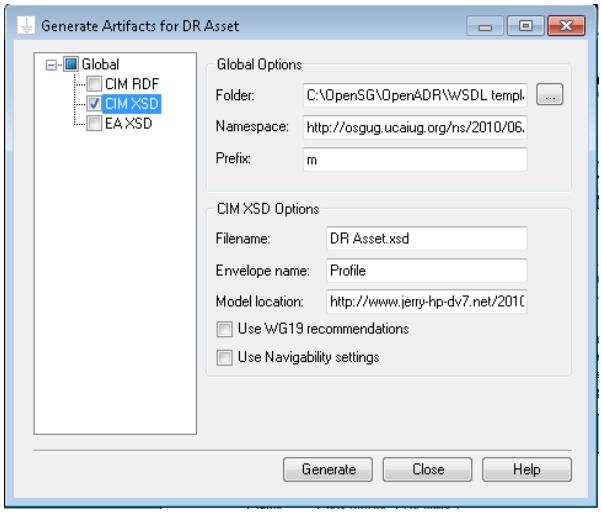



Figure 4: CIM EA's Generate Artifacts interface [4]

The resulting XSD is fully compliant with 62361-100 specification, making it structurally interoperable with any system capable of reading a 62361-100 XSD and specifically an OpenADR XSD.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xml:base="http://zigbee.com/sep" xmlns:xs
"http://zigbee.com/sep" targetNamespace="http://zigb
"2.0.03">
  <xs:simpleType name="ChargeKind">
    <xs:annotation>
      <xs:documentation>Kind of charge.</xs:document
      <xs:documentation xml:lang="en" cimp:category=
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="auxiliaryCharge">
        <xs:annotation>
          <xs:documentation>Any other charge which i
charge or charge for another service such as street
          <xs:documentation xml:lang="en" cimp:categ
        </xs:annotation>
```

Figure 5: An example XSD generated with CIM EA [4].

## III. Profile and Artifact Maintenance

The process of creating and improving models and schemas is highly iterative. During the course of the project, the OpenADR Task Force maintained a Microsoft Word document that defined the OpenADR Standard Definitions acting as the blueprints for the Profiles. The team also kept a revision log with the Word document.

OpenADR team members used CIM EA to update the UML model throughout the course of the project. The team used Subversion for version control on the UML model to allow for multiple editors. This was possible in part through the version control support offered by EA. Since Profiles in CIM EA are completely defined in UML, this version control applied to both the Profiles and their resultant artifacts.

## IV. Conclusion

The repeatable methodology supported by CIM EA provided the OpenADR Task Force with the ability to combine multiple reference models into a CIM Profile for OpenADR Service Definitions. The model-driven approach to CIM Profiles in UML allowed the team to iteratively extend and tune models and schemas quickly as OpenADR requirements were identified and agreed upon. As the OpenADR standard matures, the team will continue to use CIM EA to easily create new versions of the OpenADR CIM Profile or create additional CIM Profiles and generate new releases of implementation artifacts, with the added advantages of backward compatibility and ease of maintenance.

## Acknowledgment

The following individuals and their companies have contributed to and/or provided support for the Profile-driven standardization of Smart Grid interfaces:

- Ed Koch, Akuacom
- John Mani, Commverge
- Phil Davis, Schneider Electric
- Shawn Hu, Xtensible Solutions

The OpenADR Task Force wishes to thank all of the contributors to OpenADR, especially the above-mentioned individuals and companies for their support of this important endeavour, as it sets a key foundation for an interoperable Smart Grid.

## References

[1] *Naming and Design Rules for CIM Profiles to XML Schema Mapping*, IEC 62361-100, 2010.
[2] *OPENADR 1.0 SERVICE DEFINITION – COMMON*, OpenADR 2010
[3] Xtensible Solutions, "Profiles and Profile Groups," unpublished.
[4] CIMEA version 1.3.5. Denver, Colorado: Xtensible Solutions, 2010. Available: CIMEA.org