CIM as a Framework for Smart Grid Interoperability
Part 1

CIM Users Group
Erlangen, Germany
Terry Saxton
Part 1
- What is the interoperability problem?
- Role of CIM in achieving interoperability in the Smart Grid
- Key architectural concepts – three layer architecture/framework
- Design time modeling approach
- Work flow from semantic model to message/file assembly using CIM
- Layer 1 - CIM UML information model and contents
  - Who Manages the CIM UML Model?
    - TC 57 Organization and Formal Liaisons
  - Example: Substation model using CIM
  - Demo of UML modelling Tool – Sparx EA

Part 2
- Layer 2 - Profiles for defining system interfaces
  - IEC 61970 network model exchange
  - IEC 61968 message payloads for system integration
- Layer 3 - Implementation syntax of instance data
  - CIM expressed in XML and RDF Schema
- Value of an Enterprise Semantic Model (ESM) based on the CIM
- Case studies
- Where to get more CIM information

Break
What is the Problem that created the need for Interoperability in the Smart Grid?

- Deregulation of the power industry, and
- Increasing difficulty in building new capacity, complicating today’s power grid and pushing the grid nearer to operating limits

Consequence: Power grid operations have become dramatically more dependent on complex computer-based analytically-intensive operating practices

- Smart Grid is a transformative movement
  - expanding the number and variety of active participants
  - expanding the types of energy sources
  - expanding the kinds of active business processes

  - Result: significant increase in the operational complexity of the system
Interoperability is about Data Exchange – Lots of Data!

• Exchanged between systems/applications from many different suppliers within an enterprise (horizontal)

• Let’s look at where we are today and where we are headed
The NIST Smart Grid Conceptual Model
Sample of Systems/Communications Paths Comprising the Smart Grid Illustrates Complexity and Need for Interoperability
Interoperability is about Data Exchange – Lots of Data!

- Exchanged between systems/applications from many different suppliers within an enterprise (horizontal)
- Received from the field from many different sources (vertical/hierarchical)
Asset Health - Data Aggregation Challenge

Data

- Labs
- Test Results
- Inspection
- Work Order
- Online Monitors
- Operational Data

Systems

- TOA
  - Field Service Management
- Doble Database
  - Asset Health Data
- Maximo Cascade
  - Asset Catalog Data
- SCADA Historian

Owner/User

- Asset Engineer
  - Review
  - Replacement Planning
  - Maintenance Planning
  - Long term planning

The OT Divide

- Operations Manager
  - Alarms
  - Customers
  - Short Term planning

Challenging to bring the data together
Interoperability is about Data Exchange – Lots of Data!

- Exchanged between systems/applications from many different suppliers within an enterprise (horizontal)
- Received from the field from many different sources (vertical/hierarchical)
- Exchanged between utilities (DSO-DSO and DSO-TSO)
- Distributed Energy Resources (DER) and MicroGrids
- Distributed Intelligence in Field Devices and the Internet of Things (IoT)
  - Sheer numbers of interconnections
- The list goes on and on
That’s the Problem – What’s the Solution?

• Solution requirements
  – Requires information transfer between systems that were designed independently
  – More important: The hundreds of transfers involved in complex processes must all be informationally compatible with one another in order to achieve the required end result
  – All of this is only practical with intensive computer-based automation dependent on
    • Multiple systems at multiple locations owned and operated by multiple parties cooperating within defined business scenarios
    • Requires information produced by System A and System B to be consumable by System C in order to produce information for System D in time frames that may not tolerate errors or human intervention
Example - Let’s focus on distribution ops and outage management
Example: Non-Smart System Integration – Results in Multiple Unique Point-Point Data Exchanges and Mappings

CIS – Customer Information System
OMS – Outage Management System
DMS – Distribution Management System
WMS – Work Management System
GIS – Geographic Information System
TT – Trouble Ticket
OR – Outage Report
WO – Work Order
CR – Customer Report
Role of CIM in Smart Grid Architecture

• Here’s where the CIM comes into the solution
• CIM standards aim to simplify integration of components and expand options for supply of components by standardizing information exchanges
  – Reduce complexity with clear consistent *semantic modeling* across the enterprise
  – Data sources: achieve a clear picture of data mastership in the enterprise
  – Data consumers: make ‘data of record’ available on demand to qualified users
• CIM employs a *canonical data model* (CDM) strategy for standardizing interfaces in the power system operations and planning domain
What is a Canonical Data Model?

A common language, like use of English in International IEC standards
A common vocabulary or set of semantics for creating understanding
The CIM is the Basis for a Common Systems Language for Utilities

• The same dictionary is used for multiple forms of human communication:
  - Letters
  - Phone calls
  - Conversations
  - Emails
  - Etc.

• In similar manner, the same CIM is used for multiple forms of computer communication:
  - XML
  - RDF
  - OWL
  - DDL
  - Etc.
Using A Semantic Model Simplifies & Scales Up The Mapping Process

• What is a Semantic Model?
  – The key ingredients that make up a semantic model are a vocabulary of basic terms, a precise specification of what those terms mean and how they relate to each other.

• How does one define a Semantic Model?
  – Before making mappings, a model (or an ontology) of a given business domain is defined.
    • The model is expressed in a knowledge representation language – CIM uses UML
    • The model contains business concepts, relationships between them and a set of rules

• The Semantic Model is then used to provide a common language for exchanging information between systems that each have a different ways of representing data internally

• This is sometimes referred to as Canonical Data Model which is mapped to by each system
Using A Semantic Model Simplifies & Scales Up The Mapping Process

• How is a Semantic Model used in practice?
  – By organizing the knowledge in a discrete layer for use by adapters (i.e., data converters/mappers)
  – In this way, semantic models enable communication between computer systems that is independent of the individual system technologies, information architectures and applications
The CIM Semantic Model Provides a Semantic Layer in an Enterprise Architecture

Composite Applications

Business Intelligence

Web Services

Integration Bus

ETL

CIM-based Common Language Adapters

Generic Services

Apps.

Semantic Model Metadata

DW
Example: System Integration with the CIM - in Step With Business Needs

CIS – Customer Information System
OMS – Outage Management System
DMS – Distribution Management System
WMS – Work Management System
GIS – Geographic Information System

TT – Trouble Ticket
OR – Outage Report
WO – Work Order
CR – Customer Report
Warning: Don’t be fooled by external appearances
You need the CIM ingredient to avoid adding to ‘Integration Anarchy’

Integration anarchy is a chaos of:
(1) duplicated logic,
(2) duplicated data,
(3) duplicated effort,
(4) newly acquired integration difficulties,
(5) lack of ability to easily create new application functionality from services, and
(6) lack of ability to support business processes with applications

Integration anarchy will result in higher costs and an inflexible, brittle Smart Grid System of Systems

Data Integration Anarchy!

Without common semantics, Point-to-Point Integration will continue at the data level
Decoupled Information Exchange

Publisher

Publishers:
One Application Connector:
Obtains Data From Application And/Or Database
Transforms Data (if necessary) to the “Common Language” (a Canonical Data Model)
Puts Data Into Message Template
 Publishes The Message (Fires & forgets)

Subscribers:
Several Application Adapters Receive The Same Message
Each Adapter:
Parses Message, Pulling Out Data Needed By Application
Transforms Data (if necessary) to Local Application Format
Passes Data To Local Application And/Or Database
Through Most Appropriate Means

Message Type Instance: ChangedNetworkDataSet (Expressed In Common Language)
Now let’s look under the hood of the CIM standards as applied to smart distribution application integration
The IEC Common Information Model (CIM) - What Is It?

• A set of standards to enable system integration and information exchange based on a common information model
• Enables integration of applications/systems
  – Provides a common information model (semantics)
  – Basis for defining CDMs (profiles) for each information exchange and associated message/file schemas for all messages/files exchanged between systems
• Enables data access to enterprise data warehouse in a standard way
  – Common language to navigate and access complex data structures in any database
  – Inspiration for logical data schemas (e.g., for an operational data store)
• Enables exchange of power system network models between utilities
What is the CIM?

• It’s more than a set of standards – it’s a solution!
• UML model can be customized or tailored to fit specific utility data requirements
  – Private extensions
  – Merging and harmonizing with other standard models
  – Subset can be published as a standard from a different SDO
    • Example: NAESB Open Field Message Bus (OpenFMB) based on combined CIM and other standard object models
• New interface profiles can defined for information exchange
• New design artifacts (e.g., XSDs) can be generated with same tools used to develop the CIM standards
• A key differentiator: The CIM standards are based on a Unified Modeling Language (UML) based information model representing real-world objects and information entities exchanged within the value chain of the electric power industry
  – Not tied to a particular application’s view of the world
    • But permits same model to be used by all applications to facilitate information sharing between applications
  – Developed and standardized by IEC using Sparx Enterprise Architect modeling tools
We Need An Organizing Framework

• Let’s break it down
• The CIM standards include an information model expressed in UML
• Profiles for specifying a subset of the CIM classes and attributes for a specific business context at a specific system interface or system interaction
• Implementation models
  – Use of XML to create serialized files and messages
    • RDF Schema-based standards for power system model exchange
    • XML Schema-based standards for information message payloads
  – Could include ETL based on CIM for data base access
    • DDLs for data tables
• But how do we organize these pieces?
GWAC* Stack and the CIM Standards

*GWAC – GridWise Architecture Council
**CIM Layered Architecture**

**Information and Semantic Models**

**CIM UML**

- **Information Model**
  - Generalized model of all utility objects and their relationships
  - Application independent, but defines all concepts needed for any application

**Profiles**

- **Contextual layer restricts information model**
  - Specifies which part of CIM is used for given profile
  - Mandatory and optional
  - Restrictions
  - But cannot add to information model

**Message Syntax**

- **Message syntax describes format for instance data**
  - Serialization of instance data
  - Can re-label elements
  - Change associations to define single structure for message payloads
  - Mappings to various technologies can be defined

---

* Based on UN/CEFACT CCTS (Core Component Technical Specification)
Let's consider how these layered CIM standards are applied to solve the system interoperability problem. First step is to link business process data exchanges discovered by creating activity diagrams and sequence diagrams and linking with a common semantic model (such as the CIM model).

- Example sequence diagram
Global methodological framework
inspired from UN/Cefact CCTS
(Core Component Technical Specification) standard

1. Exchanged Data analysis

2. Information Model or Core Components (CIM)

3. Contextual Model or Business Information Entity

4. Message Conceptual Model or Message Assembly
   (Exchanged at app interfaces)

5. Implementation Message Model or Syntax Binding

Business Process Study:
Ex: outage Management

Validation

XML Exchanged Data

DMS → OMS

CIM extensions

Xtensible Solutions
An IND Technologies Company
CIM Design Time Methodology

- Another Example
  - Need to exchange an Energy Transaction between System A and System B
  - Illustrates in more detail the design artifacts generated at each step
  - Important to note all design is done in UML using Sparx Enterprise Architect with choice of CIM Tools
  - Final step is to automatically generate XML or RDF schemas from the UML
From Information Model to Syntactic Model

Ex: Energy Transaction

UML World

XML Syntactic World

Abstract Model

Information/Semantic Model

Context/Profiles

Message Assembly

Message Syntax
Information/Semantic Model Expressed in UML (Unified Modeling Language) Notation

**Class Name** usually describes things in the real world

**Class Attributes** describe significant aspects about the thing

This **Specialization** indicates that an “EnergyTransaction” is a type of “Document.” “EnergyTransaction” inherits all of the attributes from Document

**Aggregation** is a variant of Association and indicates a class is a collection or container of other classes, but if the container is destroyed, its contents are not.

**Associations** connect classes and are assigned a role that describes the relationship
From Information Model to Syntactic Model

UML World

XML Syntactic World

Abstract Model

Information/Semantic Model

Context/Profiles
Context/Profiles

Various tools available to create Profiles

String Length changed to exactly 6

String Length changed to max of 4

Only “code” attribute retained

Association inherited from parent Document class, cardinalities changed to “1”
From Information Model to Syntactic Model

UML World

Information Semantic Model

Message Assembly

XML Syntactic World

Abstract Model

Context/Profiles

EnergyTransaction

Verbs/NonTypeMessage
Message Assembly
From Information Model to Syntactic Model

Abstract Model

Information/Semantic Model

Context/Profiles

UML World

Message Assembly

XML Syntactic World

Message Syntax

<?xml version="1.0" encoding="UTF-8"?>
<xsd:element name="MT_EnergyTransaction">
  <xsd:sequence>
    <xsd:element name="EnergyTransaction">
      <xsd:sequence>
        <xsd:element name="Name"/>
        <xsd:element name="Type"/>
      </xsd:sequence>
    </xsd:element>
  </xsd:sequence>
</xsd:element>
To Summarize

- The CIM **information model** standard expressed in UML is used is the source of the semantics needed for a particular exchange.
- A **Profile** specifies the restricted subset of the CIM classes and attributes for specific business context.
  - This is the CDM (Canonical Data Model) for a particular information exchange.
- An **Implementation Technology**, such as XML, is used to create the schema for serializing the instance data as files or messages, resulting in:
  - Standards for power system models
  - Standards for information message payloads
- The good news --- most of the power system models and message schemas needed by a utility are already defined as IEC standards:
  - 61970 series: Power system models for operations and planning (T&D)
  - 61968 series: Message schemas for enterprise integration (T&D)
Let’s look at each layer of the CIM standards

**Information and Semantic Models**

**CIM UML**
- Information Model
  - Generalized model of all utility objects and their relationships
  - Application independent, but defines all concepts needed for any application

**Context**

**Profiles**
- Contextual layer restricts information model
  - Specifies which part of CIM is used for given profile
  - Mandatory and optional
  - Restrictions
  - But cannot add to information model

**Message Syntax**

**Message/File Format**
- (XSD, RDF Schema, OWL)
- Message syntax describes format for instance data
  - Can re-label elements
  - Change associations to define single structure for message payloads
  - Mappings to various technologies can be defined
Foundational Relationships Of The CIM

- **PowerSystemResource**: Electrical Network Role Used For Planning, Operations, etc.
- **Asset**: Physical Plant Filling A Role Such As A Transformer, Pole, etc.
- **Location**: Where To Find Something By GPS, Address, Electronically, etc.
- **Organisation**: Entities Performing Roles Such As Operations, Tax Authority
- **Person**: People Performing Roles Such Dispatcher, Field Operator, etc.
- **Customer**: Industrial, Commercial, & Residential Which Can Have Multiple Accounts
- **Document**: Information Containers Such As Trouble Ticket, Work Orders, etc.
Who Manages the CIM UML Model? - TC 57 Organization and Formal Liaisons

Legend
- CIM-based
- 61850-based
IEC TC57 CIM Packages

- **WG13** Operations/Planning
  - Network Models
  - PackageDependencies
    - IEC61970
      - from TC57CIM

- **WG14** Operations/Planning
  - Assets and Back Office
  - System Integration
  - PackageDependencies
    - IEC61968
      - from TC57CIM

- **WG16** Deregulated
  - Market
  - Communications
  - PackageDependencies
    - IEC62325
      - from TC57CIM
Concepts: Generalization/Inheritance

- **Equipment**: Specialization of PowerSystem Resource
- **ConductingEquipment**: Specialization of Equipment
- **Switch**: Specialization of Conducting Equipment
- **ProtectedSwitch**: Specialization of Switch
- **Breaker**: Specialization of ProtectedSwitch
Naming and Container Hierarchy Part 1
Naming and Equipment Hierarchy Part 2
Names

class Names

  IdentifiedObject
  + aliasName: String [0..1]
  + mRID: String [0..1]
  + name: String [0..1]

  + IdentifiedObject 1
  + Names 0..* 

Name
  + name: String [0..1] 0..*

  + Names 1
  + NameType 0..*

NameType
  + description: String [0..1] 0..*
  + name: String [0..1]

  + NameTypes 0..1
  + NameTypeAuthority

NameTypeAuthority
  + description: String [0..1]
  + name: String [0..1]
Connectivity and Topology Model
Converting a Circuit to CIM Objects

• Example to show how voltage levels, current transformers, power transformers and generators are modelled
• Circuit contains a single generating source, load, line and busbar. The circuit also contains two power transformers resulting in three voltage levels of 17kV, 33kV and 132kV

Example Circuit as a Single Line Diagram

- EnergyConsumer
- Breaker
- SynchronousMachine
- GeneratingUnit
- Breaker
- BusbarSection
- Current measurement represented by Measurement connected to Terminal
Transformer Class Diagram
CIM Release 15

ConductingEquipment with associations to types of TransformerEnds for electrical connectivity.

Winding terminal for balanced transformer model network connection.

TransformerTank added for distribution transformer windings so each phase winding could be modeled.

Winding terminal for unbalanced transformer model network connection.

Previously included in Winding class.
Balanced Transformer Model

Contains legacy attributes for resistance, reactance, conductance, susceptance,

For backward compatibility, can consider as optional
Transformer 17-33 is represented as four CIM objects plus optional objects. Connections from the transformer to the network are made directly from the PowerTransformer via association to PowerTransformerEnd.
Unbalanced Distribution Transformer with Multiple Tanks Instance Example
Example Circuit with Full CIM Mappings

- Maps to
  - 17 CIM classes
  - 45 CIM objects
- Could be extended further with addition of objects for
  - control areas
  - equipment owners
  - measurement units
  - generation and load curves
  - asset data
WG14 CIM Packages - 61968
How The CIM Handles Location For Logical Devices And/Or The Physical Asset Performing The Device’s Role
Types Of Document Relationship Inherited By All Assets
Activity Records

class tmpActivityRecord
IdentifiedObject
InfCommon::
Person

IdentifiedObject
Common::
Document

+createdDateTime: DateTime [0..1]
+reason: String [0..1]
+severity: String [0..1]
+status: Status [0..1]
+type: String [0..1]

IdentifiedObject
Assets::Asset

+Assets: Asset [0..*]
+ActivityRecords 0..*

IdentifiedObject
InfCommon::
ScheduledEvent

+ActivityRecords 0..*
+Assets [0..*]
+Organisations 0..*

IdentifiedObject
InfWork::
WorkStatusEntry

+ActivityRecords [0..*]
+Assets 0..*
+Organisations 0..*

IdentifiedObject
InfOperations::
PSREvent

+ActivityRecords 0..*
+Assets 0..*
+Organisations 0..*

IdentifiedObject
InfAssets::
FailureEvent

+ActivityRecords [0..*]
+Assets [0..*]
+Organisations [0..*]

IdentifiedObject
InfCustomers::
ComplianceEvent

+ActivityRecords [0..*]
+Assets [0..*]
+Organisations [0..*]

IdentifiedObject
Metering::
EndDeviceEvent

+ActivityRecords [0..*]
+Assets [0..*]
+Organisations [0..*]
WG16 CIM Market Extensions
CIM UML Release Cycles

• 61970 CIM UML tries for annual release cycle
  • Basis for IEC 61970-301 CIM Base Fifth Edition
    • Word document auto-generated from the UML electronic model
    • Information system and Profile documents are synchronized with UML model release

• 61968 CIM UML different update cycles
  • Basis for IEC 61968-11 CIM Distribution Information Exchange Model

• 62325 CIM UML on another update cycle
  • Basis for IEC 62325-301 CIM for Deregulated Markets
  • Complete CIM UML available as a combined model on CIMug Sharepoint site:
    - **Title:** draft CIM16 + DCIM12 + MCIM02
    - **Name:** [iec61970cim16v13_iec61968cim12v05_iec62325cim02v05](#)
CIM UML in Enterprise Architect

- The CIM UML model is maintained in Sparx Enterprise Architect (EA)

- Current Official CIM Releases of UML Model
  - `iec61970cim16v29a_iec61968cim12v08_iec62325cim03v01a` (official release 16 WG13)
  - `iec61970cim17v04_iec61968cim12v09_iec62325cim03v01a` (updated by WG14)
  - `iec61970cim17v07_iec61968cim12v10_iec62325cim03v02` (current model release)

- Go to UML model in EA
Break