2017 Technical Program

OpenFMB/Data Modeling

OpenFMB Data Modeling
Antitrust Guidelines for SEPA Meetings & Conferences

The antitrust laws and other business laws apply to SEPA, its members, funders, and advisers; violations can lead to civil and criminal liability. SEPA is committed to full compliance, as well as to maintaining the highest ethical standards in all of our operations and activities. These guidelines apply to all occasions: before, during, and after SEPA meetings and conferences, including in the hallways, over lunch, cocktails and at dinner.

• **SEPA’S MISSION** – is to facilitate the utility industry’s smart transition to a clean energy future through education, research, standards, and collaboration.

• **YOUR ROLE AT SEPA MEETINGS AND CONFERENCES** – varies based on what you are attending, but could include sharing information with and learning from peers, potential partners, and industry experts and/or to provide guidance to SEPA on its activities. Consult with your company counsel if at any time you believe discussions are touching on sensitive antitrust subjects such as pricing, bids, allocation of customers or territories, boycotts, tying arrangements and the like.

• **DO NOT DISCUSS** – pricing, price terms, such as, for example, discount and credit policies, promotions, or product category pricing levels and industry pricing levels, production capacity, or cost information which is not publicly available; confidential market strategies or business plans; or other competitively sensitive information. Do not disparage suppliers and/or competitors of SEPA and/or SEPA Members and participants.

• **BE ACCURATE, OBJECTIVE, AND FACTUAL** – in any discussions of goods and services offered in the market by others, including your competitors, suppliers, and customers.

• **SEPA DOES NOT RECOMMEND** – the use of particular vendors, contractors or consultants. SEPA will not promote or endorse commercial products or services of third parties. You must draw your own conclusions and make your own choices independently.

• **DO NOT AGREE WITH OTHERS** – to discriminate against or refuse to deal with (i.e., "boycott") a supplier; or to do business only on certain terms and conditions; or to set price, divide markets, or allocate customers.

• **PLEASE BE AWARE** – that an agreement regarding price need not relate to a specific price, but may relate to levels, discounts policy, allowance policy, and other terms affecting price levels or movements and may be inferred from a discussion and ensuing conduct.

• **DO NOT TRY TO INFLUENCE** – or advise others on their business decisions, and do not discuss yours (except to the extent that they are already public).

• **ASK** – for advice from your own legal department, if you have questions about any aspect of these guidelines or about a particular situation or activity at SEPA; or ask the responsible SEPA manager to contact SEPA’s Legal Counsel.
OpenFMB Modeling Approach

- Top-down business driven
- Layered architecture
  - Start with use cases and requirements
  - Structured in a single UML model
    - Using Sparx EA as modeling tool
  - Traceability among the layers
- Model driven artifacts generation
Use Case Layer

Standard UML Activity Diagram used to model use cases (e.g. Transition to Island)

10. Island recloser publishes its unsolicited status (open)

20. Battery Inverter receives status from Island Recloser and switches battery to voltage source mode (Sv).

30. Optimizer receives status from Island Recloser

40. Back-office SCADA receives status from Island Recloser

End
Data Requirements Layer

Data requirements are identified in the use case modeling process and modeled using UML Requirement element.
Integration Design Layer

- **UML Sequence Diagram** used to model interactions
- **Patterns based on Quality of Service**
Data Model Layer

The data exchanged between the devices and systems are modeled in UML Class Diagrams based on standards.
OpenFMB Data Modeling

Reference Models

**Reference Model**
- Standards such as IEC CIM & IEC 61850
- Provide objects and relationships for OpenFMB requirements
- Application independent, but defines all concepts needed for any application

Context (Profile)

**Contextual layer restricts information model and extends as needed**
- Select reference model for given profile
- Restrictions and extensions
- Mandatory and optional
- Propose extension to the standards / reference models

Message Syntax

**Message syntax describes format for instance data**
- Model driven artifacts generation
- Serialization of instance data
- May modify container or associations for message payloads
- Mappings to various technologies can be defined
Traceability
Platform Independent Model
Logical model (Profile) built based on the data requirements
Platform Specific Model

Physical implementation artifacts such as XSDs & IDLs are generated directly from the logical model.
Module Structure

Overall model
package structure
XSD Generation Tool

Native Sparx EA tool used for XSD generation
IDL Generation Tool

RTI IDL4 for IDL generation
Common Module

• Common Module contains reusable classes shared (imported) across other modules

• Each module may contain multiple profiles
Namespace

• Namespace for all individual module
  • http://openfmb.org/<version #>/openfmb/<Module Name>
    • e.g. http://openfmb.org/2017/05/openfmb/reclosermodule
Version Control

Two types of update in terms of version control:

• Backward NOT Compatible:
  - Namespace updated with new version #
  - Version # updated in header

• Backward Compatible:
  - Namespace NOT updated
  - Version # updated in header
Version Control
Defined in the Model

UML Tagged Value used for version attributes
Version Control Example
– Backward Incompatible

Version 1 updated but not backward compatible
• Both targetNamespace and version attribute updated

```xml
<x:schema ... targetNamespace="http://openfmb.org/xsd/2015/11/openfmb/reclosermodule" version="1.0">
    <xs:annotation>
        <xs:documentation>
            Version 1.0 created 2015/11
        </xs:documentation>
    </xs:annotation>
</xs:schema>
```

```xml
<x:schema ... targetNamespace="http://openfmb.org/xsd/2015/12/openfmb/reclosermodule" version="2.0">
    <xs:annotation>
        <xs:documentation>
            Version 2.0 created 2015/12
        </xs:documentation>
    </xs:annotation>
</xs:schema>
```
Version Control Example
–Backward Compatible

Version 1 updated and backward compatible

<xs:schema ... targetNamespace="http://openfmb.org/xsd/2015/11/openfmb/reclosermodule"
version="1.0">
  <xs:annotation>
    <xs:documentation>
      Version 1.0 created 2015/11
    </xs:documentation>
  </xs:annotation>
</xs:schema>

To →

<xs:schema ... targetNamespace="http://openfmb.org/xsd/2015/11/openfmb/reclosermodule"
version="1.1">
  <xs:annotation>
    <xs:documentation>
      Version 1.1 created 2015/12
    </xs:documentation>
  </xs:annotation>
</xs:schema>
Naming and Design Rules for XSD

Naming and Design Rules (NDRs)

- Garden of Eden type with elements and types defined at the global level for profiles and only type (complex & simple) for the CommonModule
- Element sequence using xs:sequence
  - mRID listed at the top
  - simpleType listed alphabetically
  - complexType listed alphabetically
  - Inherited attributes listed above native attributes
XSD Style

• Global level element & Type
  • Garden of Eden

```xml
<xsd:element name="Employee" type="EmployeeType"/>
<xsd:element name="ErpPerson" type="ErpPersonType"/>
<xsd:element name="ErpAddress" type="ErpAddressType"/>
<xsd:complexType name="EmployeeType">
    <xsd:sequence>
        <xsd:element name="ErpPerson" type="ErpPersonType"/>
        <xsd:element name="ErpAddress" type="ErpAddressType"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ErpPersonType">
    <xsd:sequence>
        <xsd:element name="lastName" type="xs:string"/>
        <xsd:element name="firstName" type="xs:string"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ErpAddressType">
    <xsd:sequence>
        <xsd:element name="streetNumber" type="xs:string"/>
        <xsd:element name="streetName" type="xs:string"/>
    </xsd:sequence>
</xsd:complexType>
```
Message Types

• Reading (both analog & discrete)
• Control (& Control Schedule)
• Event
  – Alarm
  – Informational
  – Protection
  – Workflow
# Quality of Service

<table>
<thead>
<tr>
<th>Interaction Pattern</th>
<th>QoS policy name</th>
<th>Policy value applied</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reading</strong></td>
<td>RELIABILITY</td>
<td>BEST_EFFORT</td>
</tr>
<tr>
<td></td>
<td>DURABILITY</td>
<td>VOLATILE</td>
</tr>
<tr>
<td><strong>Control</strong></td>
<td>RELIABILITY</td>
<td>RELIABLE</td>
</tr>
<tr>
<td></td>
<td>DURABILITY</td>
<td>VOLATILE</td>
</tr>
<tr>
<td></td>
<td>LIFESPAN</td>
<td>5 sec</td>
</tr>
<tr>
<td><strong>Event</strong></td>
<td>RELIABILITY</td>
<td>RELIABLE</td>
</tr>
<tr>
<td></td>
<td>DURABILITY</td>
<td>TRANSIENT</td>
</tr>
<tr>
<td></td>
<td>LIFESPAN</td>
<td>5 sec</td>
</tr>
</tbody>
</table>
Modeling Topics for Discussion

Data Model

• Use of 61850/CIM harmonization model
  - Include 61850 Logical Node (LN)

• Model structure
  - Common module & individual modules

• Restricted data type
  - Enumerations
  - Specialization

• XSD style (top level element)
  - Venetian Blind style

• Model management (centralized server proposed)
Model Structure Change from Phase II

• Use the 61850/CIM harmonization MeasurementValueSource & MeasurementValue as profile foundation

• Specialization:
  • Specialized IED at root (e.g. ResourceReadingIED)
  • Specialized MeasurementValue (e.g. ResourceReadingMeaValue)
  • Specialized IEC61850 LN (e.g. ResourceReadingMMXU)
  • Specialized data types (e.g. MV, BCR and etc.)

• Common module package
Use of IEC61850 (PoC)
XSD Structure

Restricted IEC61850 LN
Discussion – Q&A