Implementation Challenges of S-Series Specification, S3000L/S2000M

Parker Owen
Vice President, Technical Operations
Integrated Support Systems, Inc.
E-mail: parker.owen@isscorp.com
Integrated Support Systems: Brief Background

• Established: 1984
• Locations: USA / France
• Integrated Logistics Support (ILS) / Configuration Management (CM) software development
• Related technical services / custom software development / training
• ILS engineering consulting
• Active involvement with international ILS standards committees
Integrated Support Systems Deployed Solutions

• **Logistics Support Analysis (LSA) and Provisioning**
  - Active Participation on:
    • SAE Life Cycle Logistics Support Committee (LCLS)
    • ASD Data Model Exchange Working Group (DMEWG)
    • ASD S3000L Steering Committee
    • ASD PLCS (S2000M) Task Team
    • ASD S1000XTT (S1000D Interface) Task Team

• **Product Definition and Configuration**
  - Standards Supported: MIL-STD-973/2549, SAE GEIA-HB 649, EIA-836, AP 203/CC1, AP 232, pdmschema
    • Prior participation on the PDES, Inc. Technical Advisory Committee
Integrated Logistics Support ASD S Series Specifications
Challenging Areas

• **Support for S Series primitive data types**
  – Identifier
  – Property
  – Descriptor
  – Classification

• **Data Exchange**

• **Common Part definition (HardwarePartAsDesigned) between S3000L and S2000M**

• **Maintenance Task and Initial Provisioning data supporting S1000D DM generation**
Primitive Data Types: Building Blocks for S3000L and S2000M Schema

Note: The S2000M property and descriptor primitive optional characterizations are different than S3000L primitive optional characterizations.
• **ASD S3000L / S2000M:**

  – Business object identifiers in the S3000L and S2000M schema may or may not have multiple identifier values, dependent on the class. If multiple identifiers, optional characterizations must differentiate the value of the identifier.

    • Several S3000L classes allow multiple identifiers. S2000M classes only allow a single identifier

  – The optional characterizations for identifiers are identifierClassifier and identifierSetBy (Organization)

    • The identifierClassifier is a classification. Note: classifications may be tailored by project

As reference, the identifiers in GEIA-STD-0007 (and 1388-2B) are explicitly defined in clear text in all usages (all classes)
## Identifier Primitive

### Example of multiple S3000L identifiers with defining classification characterization

<table>
<thead>
<tr>
<th>Breakdown Element ID</th>
<th>Breakdown Element Type</th>
<th>Aggregated Element Type</th>
<th>Hardware Element Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>T050 LEAA 00 P</td>
<td>MM:Hardware</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T050 LEAA 01 P</td>
<td>MM:Hardware</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T050 LEAA 02 P</td>
<td>MM:Hardware</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T050 LEAA 03 P</td>
<td>MM:Hardware</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T050 LEAA01 00 P</td>
<td>MM:Hardware</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T050 LEAA01 01 P</td>
<td>MM:Hardware</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T050 LEAA01 02 P</td>
<td>MM:Hardware</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T050 LEAA01 03 P</td>
<td>MM:Hardware</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T050 LEAA01 04 P</td>
<td>MM:Hardware</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T050 LEAA01 05 P</td>
<td>MM:Hardware</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T050 LEAA01 07 P</td>
<td>MM:Hardware</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Primary Field Classification**
- 1850_LEAA_00_P
- 1850_LEAA_01_P
- 1850_LEAA_02_P
- 1850_LEAA_03_P
- 1850_LEAA01_00_P
- 1850_LEAA01_01_P
- 1850_LEAA01_02_P
- 1850_LEAA01_03_P
- 1850_LEAA01_04_P
- 1850_LEAA01_05_P
- 1850_LEAA01_07_P

**Classification**
- MM (Man:Machine)
- ECN (EngineeringChangeNumber)
- LG (Logistics:Generic)
- CN (CatalogNumber)
- D (Designation)
- BL (BreakdownElementID)

**Set by Organization**
- 1850_LEAA_00_P
- 1850_LEAA_01_P
- 1850_LEAA_02_P
- 1850_LEAA_03_P
- 1850_LEAA01_00_P
- 1850_LEAA01_01_P
- 1850_LEAA01_02_P
- 1850_LEAA01_03_P
- 1850_LEAA01_04_P
- 1850_LEAA01_05_P
- 1850_LEAA01_07_P

*Note: This example illustrates the use of identifiers with defining classification characterization in a database environment.*
• **ASD S3000L / S2000M:**

  - Property values in the S3000L and S2000M schema require a unit of measure. If multiple values, optional characterizations must differentiate the value.
  - All S3000L properties allow multiple values. Almost all S2000M properties only allow a single value.
  - S3000L has additional characterization attributes: `valueDetermination` and `valueRecordingDate`
  - S3000L properties may also be text properties

As reference, the properties in GEIA-STD-0007 (and 1388-2B) are typically numeric values (integer or real) with a related unit
Example of an S3000L property with defining unit of measure, date and value determination
Primitive Data Types: Descriptor

• **ASD S3000L / S2000M:**
  - Descriptor values in the S3000L and S2000M schema are string values. If multiple values, optional characterizations must differentiate the value.
  - All S3000L descriptors allow multiple values. S2000M descriptors only allow a single value.
  - S3000L has additional characterization attributes: descriptorLanguage, descriptorProvidedBy (Organization) and descriptorProvidedDate.

As reference, the descriptors in GEIA-STD-0007 (and 1388-2B) are single value strings
Example of an S3000L descriptor with defining language, providedByOrganization and providedByDate
• **ASD S3000L / S2000M:**
  
  – Classification values in the S3000L and S2000M schema are defined in schemas.
  – The classifications are able to be updated for project specific requirements
  
  • Classification values are used in the unique identification of business object identifiers and the multiple value elements within a business object
Classification Primitive

Classifications provide a standard set of element valid values that may be modified for program specific requirements.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:simpleType name="hardwareElementStructuralIndicatorCodeValues">
  <xs:simpleType name="" xsd:restriction base="xs:string">  
    <xs:enumeration value="SST"/>
    <xs:enumeration value="SS"/>
    <xs:enumeration value="SD"/>
    <xs:enumeration value="SI"/>
    <xs:enumeration value="NA"/>
  </xs:simpleType>
</xs:simpleType>
```
• The SX005G: S Series ILS specifications XML schema implementation guidance defines how the exchange files for the S Series specifications (S2000M, S3000L, S4000P, S5000F, S6000T) are implemented
  – Full file exchange, as well as net change is defined
Data Exchange: The support/management of the primitive data type, optional characterizations enable the accurate exchange of the data.

Example of a task business object with multiple identifiers and a property element of the business object that has multiple values.
• Both the S3000L and the S2000M specifications use a common definition of **HardwarePartAsDesigned**
  
  – The S3000L allows multiple identifier values
  – The S2000M allows one identifier value
  – Both the identifiers have the same optional characterizations of `partIdentifierClass` and `organizationReference`

• The `partIdentifierClass` value (PNO: `partNumber`) may be used to determine which `partIdentifier` is used in the S2000M exchange file.
Common Part Definitions between S3000L and S2000M

- `<xs:simpleType name="partIdentificationClassValues">
  - `<xs:restriction base="xs:string">
     - `<xs:enumeration value="CPNO">
     - `<xs:annotation/>
     - `<xs:appinfo/>
     </xs:enumeration>
     - `<xs:enumeration value="SNNO">
     - `<xs:annotation/>
     - `<xs:appinfo/>
     </xs:enumeration>
     - `<xs:enumeration value="P6NO">
     - `<xs:annotation/>
     - `<xs:appinfo/>
     </xs:enumeration>
     - `<xs:enumeration value="PNO">
     - `<xs:annotation/>
     - `<xs:appinfo/>
     </xs:enumeration>
     - `<xs:enumeration value="SPNO">
     - `<xs:annotation/>
     - `<xs:appinfo/>
     </xs:enumeration>
     - `<xs:enumeration value="REF">
     - `<xs:annotation/>
     - `<xs:appinfo/>
     </xs:enumeration>
     - `<xs:enumeration value="SN">
     - `<xs:annotation/>
     - `<xs:appinfo/>
     </xs:enumeration>
     - `<xs:enumeration value="STD">
     - `<xs:annotation/>
     - `<xs:appinfo/>
     </xs:enumeration>
     - `<xs:restriction base="xs:int">
     - `<xs:annotation/>
     - `<xs:appinfo/>
     </xs:restriction>
   </xs:restriction>
</xs:simpleType>`
• **IPD DM generation from S2000M source data:**
  – All S1000D data modules require a data module code
  – S2000M does not have a data module code defined in its schema; S3000L does have a document of type “Data Module” defined
  – An S3000L document may be classified as a data module, therefore it makes sense to use the S3000L document of data module type as the data module code for the S1000D IPD data module generated from the S2000M source data.

• **Procedural DM generation from S3000L source data:**
  – Information Control Numbers (ICNs) may be referenced in to S1000D Procedural data modules
  – S3000L does not have an ICN defined in its schema; ICNs are defined in the S2000M schema
  – An S2000M ICN may be referenced into an S3000L subtask description to support the generation the S1000D data module.
IPD DM generation from S2000M and S3000L source data

Example of an S3000L document used as a DMC for an S1000D IPD DM sourced from S2000M data
Procedural DM generation from S3000L and S2000M source data

Example of an ICN being referenced into a subtask.
Thank you for your attention! Questions?