EWIS Interoperability Forum
Test Suite v1.0

2019-10-25

Contacts		
Lothar Klein Steinweg 1 36093 Künzell / Germany	Sophie HERAIL CIMPA S.A.S. Centreda 1 4, Avenue Didier Daurat 31700 Blagnac, France Subcontractor for AIRBUS Operations SAS – IZMA	Daniel Ganser Gulfstream Aerospace Corporation BTC 171 Crossroads Parkway Savannah, GA 31407, U.S.A.
lothar.klein@lksoft.com	sophie.herail@airbus.com	dan.ganser@gulfstream.com

Table of Contents

List of Figures

Document History

Release	Date	Change
1.0	2019-09-11	Initial Release

1 Introduction

This document describes the suite of test cases to be used for the EWIS Interoperability Forum. The EWIS-IF is a joint testing forum, organized and facilitated by AFNeT and PDES.

2 Formal Test Syntax

This clause defines a formal syntax for the definition of synthetic test cases in the terms of the Domain Model. Purpose of this syntax is to formulate test cases in a clear, easy readable and unambiguous way. A macro capability allows to define standard patterns once and then apply them again and again. It is intended to convert test cases using this syntax into XML Schematron for the

The formal test syntax allows the definition of patterns of instances of Application Objects (AO); here they are AOs of the Domain Model. The test syntax is used to define the test specifications for both import and export of STEP XML files.

Instances of AOs and other values (string, real, ...) are identified (ID) by a leading “@” followed by a positive number. This number is unique within a particular test case and macro. If the same ID is used several times within a test case or macro, then this means the same AO instance or other value. If the same ID is used in different test cases or macros this does not have any meaning.

Every instance or other value has to have a definition statement. Such a statement starts with the ID, followed by a “:” and then followed by it’s type that is defined in the domain model. After this constraints on the attribute values of instances or the value itself can be stated within “(“...”)”.

The order of IDs within a macro is significantly. They should start with @1, @2, @3 … @N. When invoking a macro from another test case or higher level macro, these IDs are replaced with the IDs and values defined within the macro invocation.

3 Macros

3.1 Part_with_PartView

This macro constraints single instances of a Part, a PartVersion, a PartView and a ViewContext to be linked together. Only the name for a Part is constrained as the ID of a Part is often rather system dependent. The ViewContext used as the initialContext for the PartView is constrained for the predefined LifeCycleStage “design”.

```
Macro Part_with_PartView (
  @1:Part( Name=@2, Versions[i]=@3 );
  @2:CharacterString
  @3:PartVersion( Views[i]=@4 );
  @4:PartView( InitialContext=@5 );
  @5:ViewContext( LifeCycleStage=PredefinedApplicationDomainEnum(design) );
);
```

3.2 Part_with_WiringHarnessAssemblyDesign

This macro is similar to the macro Part_with_PartView and constraints single instances of a Part, a PartVersion, a WiringHarnessAssemblyDesign (that is a sub-subytype of PartView) and a ViewContext to be linked together. The Part is constrained for the PartType “wiring_harness”.

```
Macro Part_with_WiringHarnessAssemblyDesign (
  @1:Part( Name=@2,
```

```
            Versions[i]=@3,
            PartTypes[i]=PartCategoryEnum(wiring_harness) );
   @2:CharacterString
   @3:PartVersion( Views[i]=@4 );
   @4:WiringHarnessAssemblyDesign( InitialContext=@5 );
   @5:ViewContext( LifeCycleStage=PredefinedApplicationDomainEnum(design),
              ApplicationDomain=PredefinedApplicationDomainEnum(electrical)
                              );
);
```

3.3 Harness_design_with_segment_topology

This macro is an extension of the macro Part_with_WiringHarnessAssemblyDesign. In addition to this it adds a constraint for an additional ViewContext with the predefined LifeCycleStage “wiring_harness_segment_topology”.

```
Macro Harness_design_with_segment_topology (
   Part_with_WiringHarnessAssemblyDesign(@1,@2,@3,@4,@5);
   @1:Part;
   @2:CharacterString;
   @3:PartVersion;
   @4:WiringHarnessAssemblyDesign( AdditionalContexts[i]=@6 );
   @5:ViewContext;
   @6:ViewContext( ApplicationDomain=PredefinedApplicationDomainEnum(
           wiring_harness_segment_topology) );
);
```

3.4 Undirected_edge

This macro constrains an edge with two vertices so that either one of the Vertices is the EdgeStart and the other Vertices is the EdgeEnd. In STEP all Edges are by default directed, however for the design of the topology of an EWH the direction of an Edge is not relevant (however it might be relevant for the purpose of manufacturing).

```
Macro Undirected_edge (
   @1=Edge( ( EdgeStart=@2, EdgeEnd=@3)
            OR
            ( EdgeStart=@3, EdgeEnd=@2) );
   @2=Vertex;
   @3=Vertex;
);
```

4 Synthetic Test Case Specifications

4.1 EWH-Unit-Assembly1

This test case focusses on a very basic flat assembly structures as it might show up in EWH. This test does not address connectivity or topological information. This test is an extension of the typical assembly structure as provided in the document “Recommended Practices for AP242 Business Object Model XML Assembly Structure”.

The following elements are tested:

- Part with PartCategories: discrete_part, raw_material_by_length, wire, cable, connector, lug
- WiringHarnessAssemblyDesign that is a subtype of AssemblyDefinition

- specific kinds of Part Occurrences: SingleOccurrence, QuantifiedOccurrence, Wire-Occurrence, CableOccurrence

```
Test EWH-Unit-Assembly1 (
  @4:ViewContext;
  @5:ViewContext;
  @8:Unit( Name=ClassString("metre"), Quantity=ClassString("length") );

  @100:Part( PartTypes[i]=PartCategoryEnum(connector), PartTypes[i]=Part-
CategoryEnum(discrete) );
  @101:PartVersion;
  @102:PartView;
  Part_with_PartView( @100, "Connector-A", @101, @102, @4);
  @111:SingleOccurrence( Id=IdentifierString("J1"), Definition=@102 );
  @121:SingleOccurrence( Id=IdentifierString("J2"), Definition=@102 );

  @200:Part( PartTypes[i]=PartCategoryEnum(terminal_lug),
PartTypes[i]=PartCategoryEnum(discrete) );
  @201:PartVersion;
  @202:PartView;
  Part_with_PartView(@200, "Lug-B", @201, @202, @4);
  @211:SingleOccurrence( Id=IdentifierString("J3"), Definition=@202 );
  @221:SingleOccurrence( Id=IdentifierString("J4"), Definition=@202 );

  @300:Part( PartTypes[i]=PartCategoryEnum(wire), PartTypes[i]=PartCatego-
ryEnum(raw_material_by_length) );
  @301:PartVersion;
  @302:PartView;
  Part_with_PartView(@300, "Wire-C", @301, @302, @4);
  @311:WireOccurrence( Id=IdentifierString("W1"), Definition=@302, Quan-
tity=@312 );
  @312:NumericalValue( Unit=@8, ValueComponent=3.5 );
  @321:WireOccurrence( Id=IdentifierString("W2"), Definition=@302, Quan-
tity=@322 );
  @312:NumericalValue( Unit=@8, ValueComponent=2.7 );

  @400:Part( PartTypes[i]=PartCategoryEnum(cable), PartTypes[i]=PartCatego-
ryEnum(raw_material_by_length) );
  @401:PartVersion;
  @402:PartView;
  Part_with_PartView(@400, "Cable-D", @401, @402, @4);
  @411:CableOccurrence( Id=IdentifierString("W3"), Definition=@402, Quan-
tity=@412 );
  @412:NumericalValue( Unit=@8, ValueComponent=1,8 );
  @421:CableOccurrence( Id=IdentifierString("W4"), Definition=@402, Quan-
tity=@422 );
  @412:NumericalValue( Unit=@8, ValueComponent=7.2 );

  @900:Part;
  @901:PartVersion;
  @902:WiringHarnessAssemblyDesign;
  part_with_WiringHarnessAssemblyDesign( @900,"Unit-
Assembly1",@901,@902,@4,@5 );
  @911:NextAssemblyOccurrenceUsage( Relating=@902, Related=@111 );
```

```
   @912:NextAssemblyOccurrenceUsage( Relating=@902, Related=@121 );
   @913:NextAssemblyOccurrenceUsage( Relating=@902, Related=@211 );
   @914:NextAssemblyOccurrenceUsage( Relating=@902, Related=@221 );
   @915:NextAssemblyOccurrenceUsage( Relating=@902, Related=@311 );
   @916:NextAssemblyOccurrenceUsage( Relating=@902, Related=@321 );
   @917:NextAssemblyOccurrenceUsage( Relating=@902, Related=@411 );
   @918:NextAssemblyOccurrenceUsage( Relating=@902, Related=@421 );
);
```

4.2 EWH-UseCase-Topology1

This test case focuses on a very basic topological structure needed for EWH without any other information. The test consists of a flexible topological representation of the harness, consisting of 6 vertices and 5 edges with length.

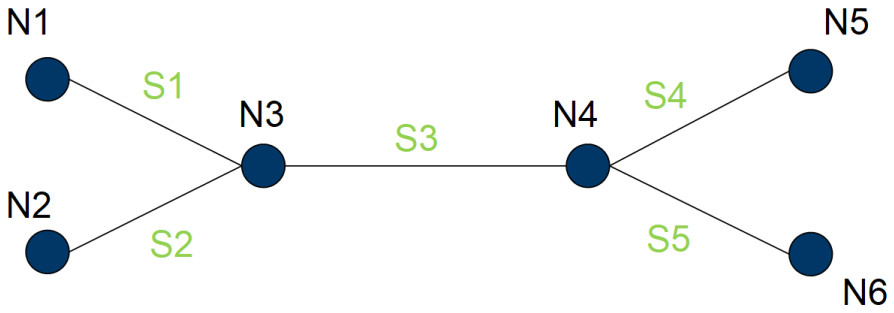

Figure 1: Topology1

```
Test EWH-UseCase-Topology1 (
   @1:Part;
   @2:PartVersion;
   @3:WiringHarnessAssemblyDesign( Topology=@6 );
   @4:ViewContext;
   @5:ViewContext;
   harness_design_with_segment_topology(@1,"UseCase-Topology1",@2,@3,@4,@5);

   @6:ConnectedEdgeWithLengthSetRepresentation( Items[i]=@10, ContextOf-
Items=@7 );
   @7:RepresentationContext( Units=@8 )
   @8:Unit( Name=ClassString("metre"), Quantity=ClassString("length") );

   @10:ConnectedEdgeSet( ConnectedEdges[i]=@11,
                         ConnectedEdges[i]=@12,
                         ConnectedEdges[i]=@13,
                         ConnectedEdges[i]=@14,
                         ConnectedEdges[i]=@15,
                         ConnectedEdges.sizeof=5
                        );
   @11:EdgeWithLength( name='S1', EdgeLength=PositiveLengthMeasure(2.0) );
   undirected_edge(@11, @21, @23)
   @12:EdgeWithLength( name='S2', EdgeLength=PositiveLengthMeasure(4.0) );
   undirected_edge(@12, @22, @23)
   @13:EdgeWithLength( name='S3', EdgeLength=PositiveLengthMeasure(6.0) );
```

```
   undirected_edge(@13, @23, @24);
   @14:EdgeWithLength( name='S4', EdgeLength=PositiveLengthMeasure(8.0) );
   undirected_edge(@14, @24, @25);
   @15:EdgeWithLength( name='S5', EdgeLength=PositiveLengthMeasure(10.0) );
   undirected_edge(@15,@24,@26)

  @21:Vertex( name='N1')
  @22:Vertex( name='N2')
  @23:Vertex( name='N3')
  @24:Vertex( name='N4')
  @25:Vertex( name='N5')
  @26:Vertex( name='N6')

   sizeof(@5:ConnectedEdgeWithLengthSetRepresentation.Items) = 1;
   sizeof(Part) = 1;
   sizeof(PartVersion) = 1;
   sizeof(WiringHarnessAssemblyDesign) = 1;
   sizeof(PartView) = 1;
   sizeof(ConnectedEdgeWithLengthSetRepresentation) = 1;
   sizeof(RepresentationContext) = 1;
   sizeof(ConnectedEdgeSet) = 1;
   sizeof(EdgeWithLength) = 5;
   sizeof(Vertex) = 6;
);
```

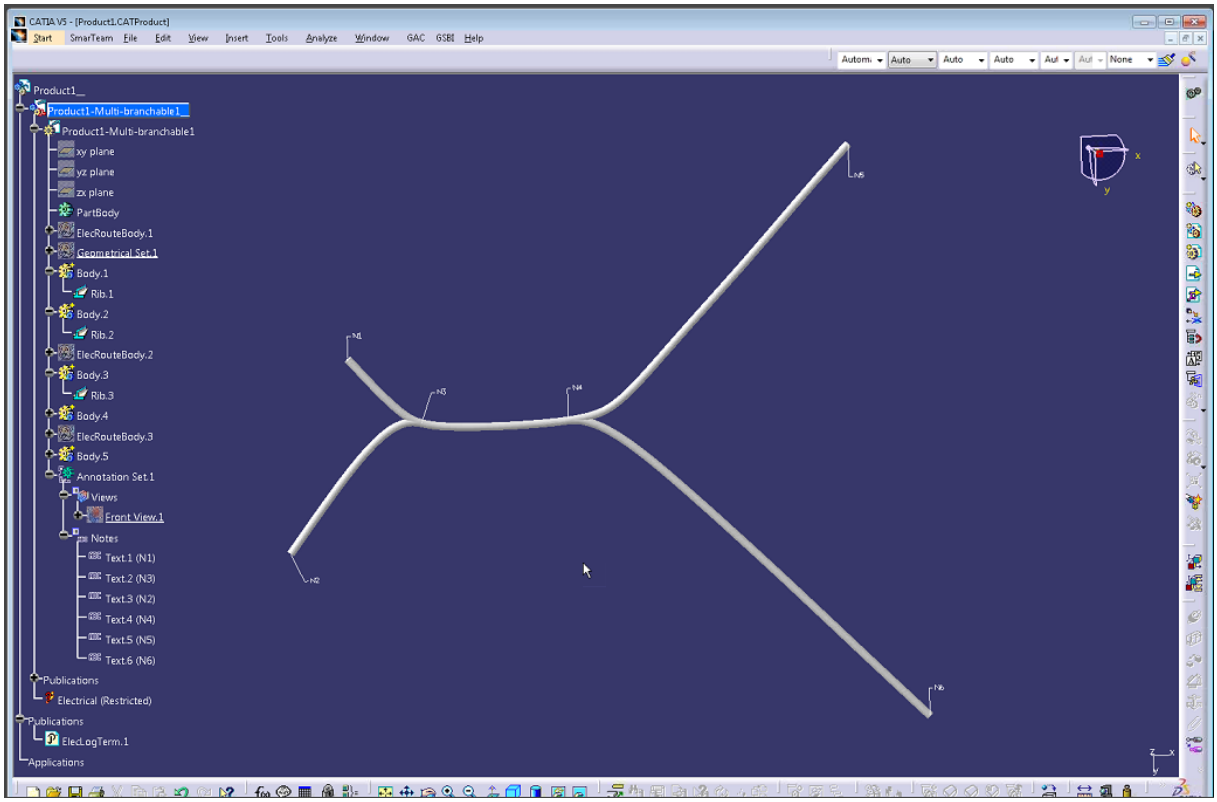

Figure 2: Example in CATIA

Provided test files:

AP242ed2 XML: EWH-UseCase-Topology1.xml

Native CATIA: Topology Test1 - Sample CatiaV5 STP.zip

Mentor/Siemens Capital Harness: Topology Test1 - Sample Capital HX2ML.xml

KBL, generated from Capital Harness:

Topology Test1 - Sample Capital KBL_2.3.kbl

Topology Test1 - Sample Capital KBL2.4.kbl