EWIS Interoperability Forum
Test Suite v2.0

2020-09-09/11

Contacts		
Lothar Klein	Sophie HERAIL	Daniel Ganser
Steinweg 1	CIMPA S.A.S.	Gulfstream Aerospace Corporation
36093 Künzell / Germany	Centreda 1	BTC
	4, Avenue Didier Daurat	171 Crossroads Parkway
	31700 Blagnac, France	Savannah, GA 31407, U.S.A.
	Subcontractor for AIRBUS	
	Operations SAS – IZMA	
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
2.0	2020-09-09/11	Update & extended for 2nd test round

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_ID_and_PartView

This macro is similar to the macro Part_with_PartView but instead of constraining the name of a Part the ID of a part is constraint.

```
Macro Part_with_ID_and_PartView (
  @1:Part( Id=@2, Versions=(@3) );
  @2:Identifier;
  @3:PartVersion( Views=(@4) );
  @4:PartView( InitialContext=@5 );
  @5:ViewContext( LifeCycleStage=PredefinedApplicationDomainEnum(design) );
);
```

3.3 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)
                                );
);
```

3.4 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) );
);
```

3.5 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;
);
```

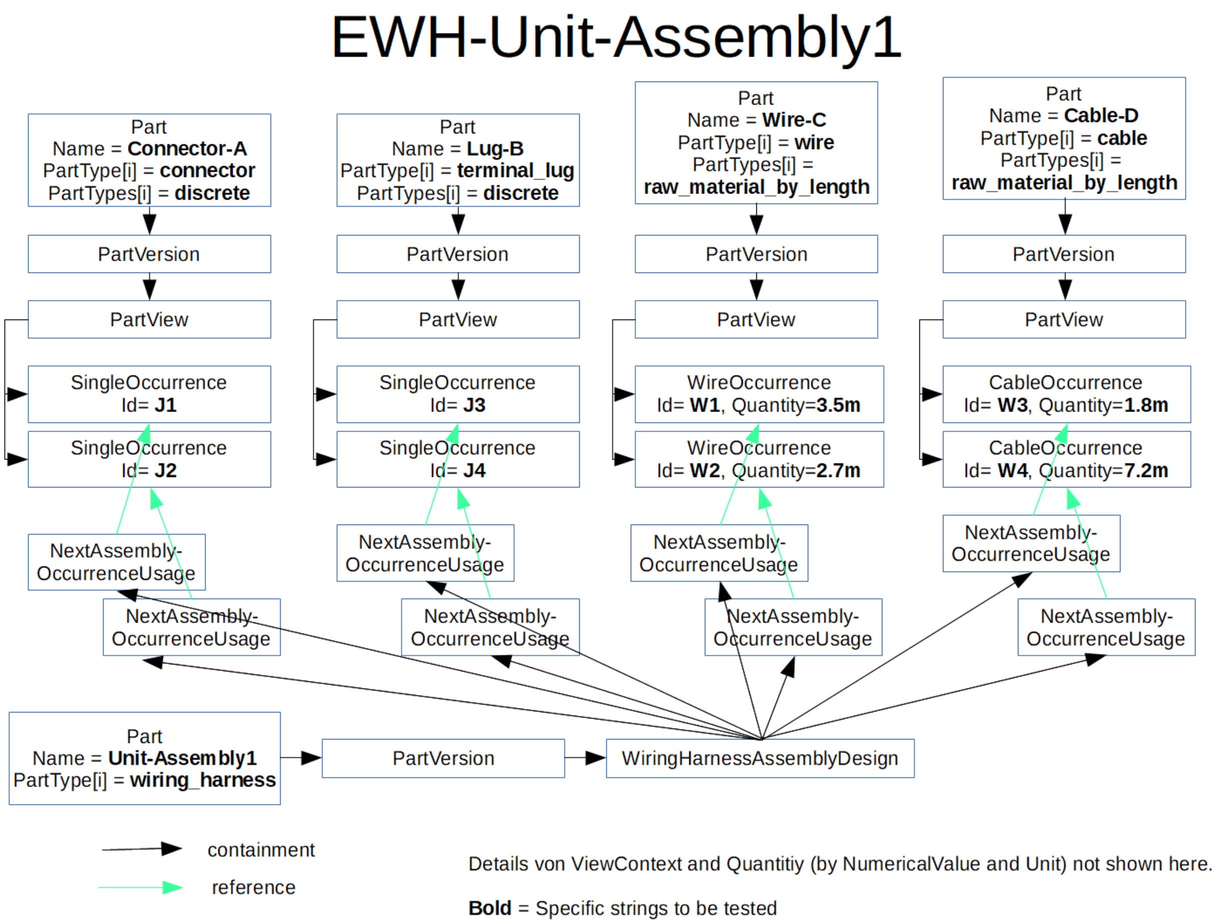
4 Test Case Specifications

4.1 EWH-Assembly1

This test case focuses 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, WireOccurrence, CableOccurrence



Details von ViewContext and Quantitiy (by NumericalValue and Unit) not shown here.

Bold = Specific strings to be tested

Formal test-case specification:

```
Test EWH-Assembly1 (

@4:ViewContext;
@5:ViewContext;
@8:Unit( Name=ClassString("metre"), Quantity=ClassString("length") );

@100:Part( PartTypes[i]=PartCategoryEnum(connector), PartTypes[i]=PartCate-
goryEnum(discrete) );
@101:PartVersion;
@102:PartView;
```

```
Part_with_Name_and_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_Name_and_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]=PartCategoryEnum(raw_material_by_length) );
@301:PartVersion;
@302:PartView;
Part_with_Name_and_PartView(@300, "Wire-C", @301, @302, @4);
@311:WireOccurrence( Id=IdentifierString("W1"), Definition=@302,
Quantity=@312 );
@312:NumericalValue( Unit=@8, ValueComponent=3.5 );
@321:WireOccurrence( Id=IdentifierString("W2"), Definition=@302,
Quantity=@322 );
@312:NumericalValue( Unit=@8, ValueComponent=2.7 );

@400:Part( PartTypes[i]=PartCategoryEnum(cable), PartTypes[i]=PartCategoryEnum(raw_material_by_length) );
@401:PartVersion;
@402:PartView;
Part_with_Name_and_PartView(@400, "Cable-D", @401, @402, @4);
@411:CableOccurrence( Id=IdentifierString("W3"), Definition=@402,
Quantity=@412 );
@412:NumericalValue( Unit=@8, ValueComponent=1.8 );
@421:CableOccurrence( Id=IdentifierString("W4"), Definition=@402,
Quantity=@422 );
@412:NumericalValue( Unit=@8, ValueComponent=7.2 );

@9000:Part;
@9001:PartVersion;
@9002:WiringHarnessAssemblyDesign;
@9003:ViewContext;
@9004:ViewContext;
part_with_WiringHarnessAssemblyDesign( @9000,"EWH Test-Case
Assembly1",@9001,@9002,@9003,@9004 );

@9101:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@111 );
@9102:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@121 );
@9103:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@211 );
@9104:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@221 );
@9105:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@311 );
@9106:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@321 );
@9107:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@411 );
@9108:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@421 );

sizeof(Part) = 5;
sizeof(PartVersion) = 5;
sizeof(PartView) = 4;
sizeof(WiringHarnessAssemblyDesign) = 1;
sizeof(NextAssemblyOccurrenceUsage) = 8;
sizeof(SingleOccurrence) = 4;
sizeof(WireOccurrence) = 2;
sizeof(CableOccurrence) = 2;

);
```

4.2 *EWH-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/geometric representation of the harness, consisting of 6 vertices and 5 edges with length.

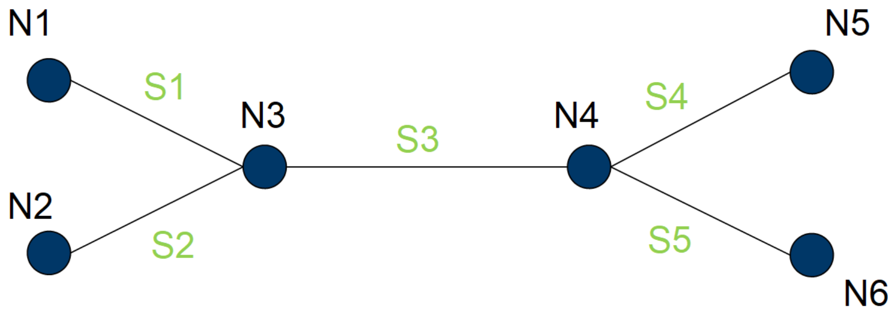


Figure 1: Topology1

Formal test-case specification:

```
Test EWH-Topology1 (

@8:Unit( Name=ClassString("metre"), Quantity=ClassString("length") );

@9000:Part;
@9001:PartVersion;
@9002:WiringHarnessAssemblyDesign( Topology=@9901 );
@9003:ViewContext;
@9004:ViewContext;
harness_design_with_segment_topology(@9000,
  "EWH Test-Case Topology1",@9001,@9002,@9003,@9004);

@9900:GeometricCoordinateSpace( Units=@8, DimensionCount=1 );
@9901:EdgeBasedTopologicalRepresentationWithLengthConstraint(
 Items=(@9902), ContextOfItems=@9900 );
@9902:ConnectedEdgeSet( ConnectedEdges=(@9931,@9932,@9933,@9934,@9935) );

@9911:Point();
@9912:Point();
@9913:Point();
@9914:Point();
@9915:Point();
@9916:Point();
@9921:VertexPoint( name='N1' VertexGeometry=@9911 );
@9922:VertexPoint( name='N2' VertexGeometry=@9912 );
@9923:VertexPoint( name='N3' VertexGeometry=@9913 );
@9924:VertexPoint( name='N4' VertexGeometry=@9914 );
@9925:VertexPoint( name='N5' VertexGeometry=@9915 );
@9926:VertexPoint( name='N6' VertexGeometry=@9916 );

@9931:EdgeBoundedCurveWithLength( name='S1', EdgeGeometry=@9941 );
undirected_edge(@9931, @9921, @9923)
@9932:EdgeBoundedCurveWithLength( name='S2', EdgeGeometry=@9942 );
undirected_edge(@9932, @9922, @9923)
@9933:EdgeBoundedCurveWithLength( name='S3', EdgeGeometry=@9943 );
undirected_edge(@9933, @9923, @9924);
@9934:EdgeBoundedCurveWithLength( name='S4', EdgeGeometry=@9944 );
undirected_edge(@9934, @9924, @9925);
@9935:EdgeBoundedCurveWithLength( name='S5', EdgeGeometry=@9945 );
undirected_edge(@9935,@9924,@9926);

@9941:BoundedCurveWithLength( EdgeLength=PositiveLengthMeasure(2.0) );
@9942:BoundedCurveWithLength( EdgeLength=PositiveLengthMeasure(4.0) );
@9943:BoundedCurveWithLength( EdgeLength=PositiveLengthMeasure(6.0) );
@9944:BoundedCurveWithLength( EdgeLength=PositiveLengthMeasure(8.0) );
@9945:BoundedCurveWithLength( EdgeLength=PositiveLengthMeasure(10.0) );

sizeof(Part) = 1;
sizeof(PartVersion) = 1;
sizeof(WiringHarnessAssemblyDesign) = 1;
sizeof(NextAssemblyOccurrenceUsage) = 0;

sizeof(GeometricCoordinateSpace) = 1;
sizeof(EdgeBasedTopologicalRepresentationWithLengthConstraint) = 1;
sizeof(ConnectedEdgeSet) = 1;
sizeof(BoundedCurveWithLength) = 5;
sizeof(EdgeBoundedCurveWithLength) = 5;
sizeof(VertexPoint) = 6;
sizeof(Point) = 6;
sizeof(CartesianPoint) = 0;
sizeof(PointOnCurve) = 0;

);
```

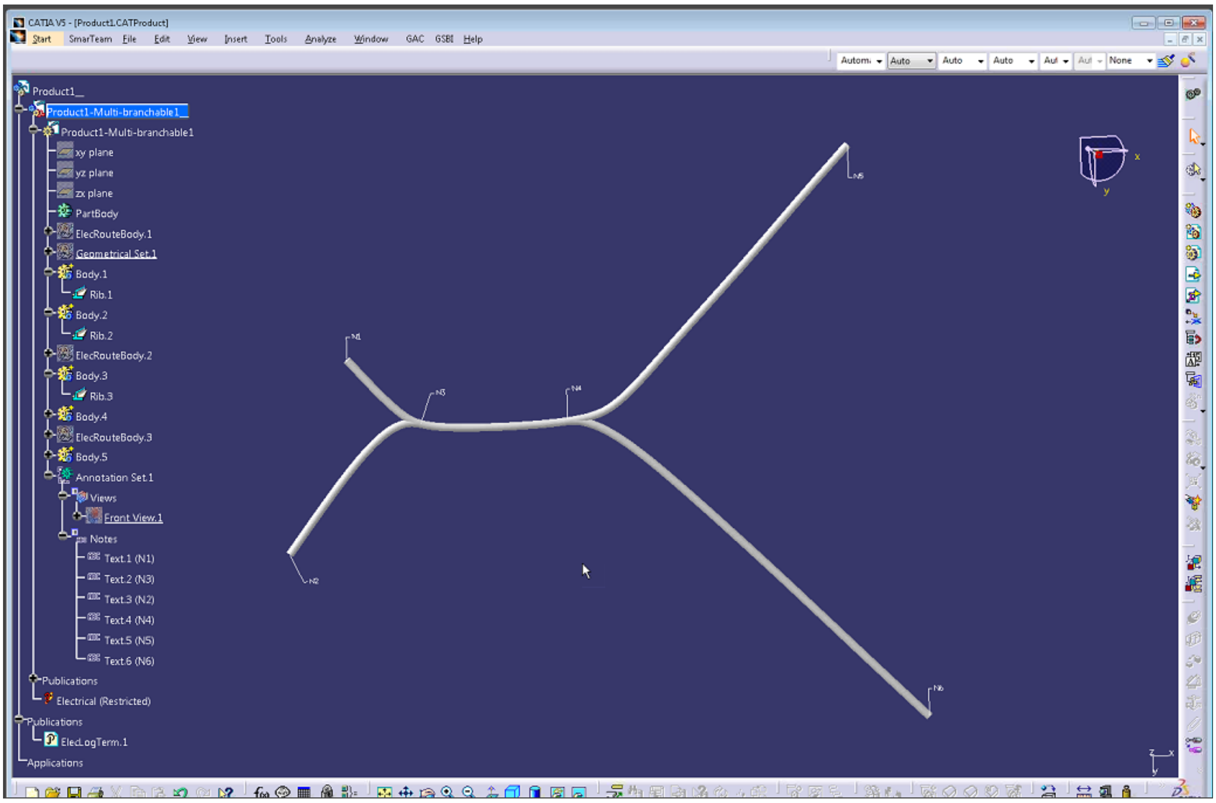


Figure 2: Example in CATIA

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

4.3 EWH-Topology2

This test case is an extension of test case EWH-Topology1 that is merged with a simplified EWH-Assembly1 test case.

- the topology model is extended for Paths, SubEdges and PointOnCurves
  - ○ Path P1 traverses the EdgeBoundedCurveWithLength S1, S3, S4
  - ○ Path P2 traverses the EdgeBoundedCurveWithLength S2, S3, S5
  - ○ Path P3 traverses the SubEdges S2.2, S3.1
  - ○ for the definition of the VertexPoints for the SubEdges, two PointOnCurves are defined in the middle of the underlying BoundedCurveWithLength
  - ○ it is up to the implementations to ensure that the orientations of the Edges in the EdgeList of a Path fits with the orientation of the underlying BoundedCurveWith-

Length. See the attributes *Path.OrientationList* and *EdgeCurve.SameSense* for this purpose.

- the simplified assembly structure consists of
  - a single wire
  - a single cable
  - a protective covering for only a certain region
- two simple 2-pin connectors and two terminal lugs; one at each extremity of the harness topology
- geometry-to-topology association of
  - wire/cable/protection *Occurrences* to *Paths*
  - connectors and terminal lug *Occurrences* to *VertexPoints*

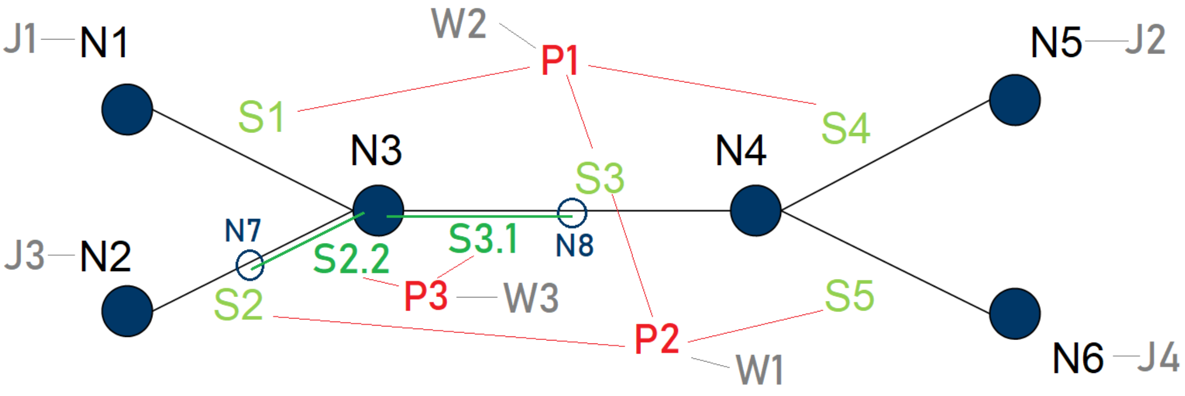


Figure 3: Topology2

Formal test-case specification:

(Draft; not completed yet; references depends on the available p21 files)

```
Test EWH-Topology2 (

@8:Unit( Name=ClassString("metre"), Quantity=ClassString("length") );

@100:Part( PartTypes[i]=PartCategoryEnum(connector),
           PartTypes[i]=PartCategoryEnum(discrete) );
@101:PartVersion;
@102:PartView( DefiningGeometry=@191 );
Part_with_Name_and_PartView( @100, "Connector-A", @101, @102, @4);
@111:SingleOccurrence( Id=IdentifierString("J1"), Definition=@102 );
@121:SingleOccurrence( Id=IdentifierString("J2"), Definition=@102 );
@190:GeometricCoordinateSpace( DimensionCount=3 );
@191:GeometricModel( items[i]=@192, ContextOfItems=@190 );
@192:AxisPlacement;

@200:Part( PartTypes[i]=PartCategoryEnum(terminal_lug),
           PartTypes[i]=PartCategoryEnum(discrete) );
@201:PartVersion;
@202:PartView( DefiningGeometry=@291 );
Part_with_Name_and_PartView(@200, "Lug-B", @201, @202, @4);
```

```
@211:SingleOccurrence( Id=IdentifierString("J3"), Definition=@202 );
@221:SingleOccurrence( Id=IdentifierString("J4"), Definition=@202 );
@290:GeometricCoordinateSpace( DimensionCount=3 );
@291:GeometricModel( items[i]=@292, ContextOfItems=@290 );
@292:AxisPlacement;

@300:Part( PartTypes[i]=PartCategoryEnum(wire),
           PartTypes[i]=PartCategoryEnum(raw_material_by_length) );
@301:PartVersion;
@302:PartView( DefiningGeometry=@391 );
Part_with_Name_and_PartView(@300, "Wire-C", @301, @302, @4);
@311:WireOccurrence( Id=IdentifierString("W1"),
  Definition=@302, Quantity=@312 );
@312:NumericalValue( Unit=@8, ValueComponent=3.5 );
@390:GeometricCoordinateSpace( DimensionCount=2 );
@391:GeometricModel( name='2D cross section',
  items[i]=@392, ContextOfItems=@390 );
@392:AxisPlacement; # placeholder for 2D centre

@400:Part( PartTypes[i]=PartCategoryEnum(cable),
           PartTypes[i]=PartCategoryEnum(raw_material_by_length) );
@401:PartVersion;
@402:PartView( DefiningGeometry=@491 );
Part_with_Name_and_PartView(@400, "Cable-D", @401, @402, @4);
@411:CableOccurrence( Id=IdentifierString("W2"), Definition=@402,
Quantity=@412 );
@412:NumericalValue( Unit=@8, ValueComponent=1.8 );
@490:GeometricCoordinateSpace( DimensionCount=2 );
@491:GeometricModel( name='cross section', items[i]=@492,
ContextOfItems=@490 );
@492:AxisPlacement;

@500:Part( PartTypes[i]=PartCategoryEnum(protective_covering),
           PartTypes[i]=PartCategoryEnum(raw_material_by_length) );
@501:PartVersion;
@502:PartView( DefiningGeometry=@591 );
Part_with_Name_and_PartView(@500, "Protection-E", @501, @502, @4);
@511:QuantifiedOccurrence( Id=IdentifierString("W3"),
                           Definition=@402, Quantity=@412 );
@512:NumericalValue( Unit=@8, ValueComponent=1.2 );
@590:GeometricCoordinateSpace( DimensionCount=2 );
@591:GeometricModel( name='cross section',
                     items[i]=@592, ContextOfItems=@590 );
@592:AxisPlacement;

@9000:Part;
@9001:PartVersion;
@9002:WiringHarnessAssemblyDesign( Topology=@9901 );
@9003:ViewContext;
@9004:ViewContext;
harness_design_with_segment_topology(@9000,
  "EWH Test-Case Topology2",@9001,@9002,@9003,@9004);
```

```
@9101:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@111,
                                   Placement=(@9111) ); # connector J1
@9102:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@121,
                                   Placement=(@9112) ); # connector J2
@9103:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@211,
                                   Placement=(@9113) ); # terminal lug J3
@9104:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@221,
                                   Placement=(@9114) ); # terminal lug J4
@9105:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@311,
                                   Placement=(@9115 ) ); # wire W1
@9106:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@411,
                                   Placement=(@9116 ) ); # cable W2
@9107:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@511,
                                   Placement=(@9117 ) ); # protection W3

@9111:GeometryToTopologyModelAssociation (
  Relating=@9901, Related=@191, Origin=@192, Target=@9921); # connector J1
@9112:GeometryToTopologyRepresentationAssociation(
  Relating=@9901, Related=@191, Origin=@192, Target=@9925); # connector J2
@9113:GeometryToTopologyRepresentationAssociation(
  Relating=@9901, Related=@291, Origin=@292, Target=@9922); # terminal lug
J3
@9114:GeometryToTopologyRepresentationAssociation(
  Relating=@9901, Related=@291, Origin=@292, Target=@9926); # terminal lug
J4
@9115:GeometryToTopologyRepresentationAssociation(
  Relating=@9901, Related=@391, Origin=@392, Target=@9952); # wire W1
@9116:GeometryToTopologyRepresentationAssociation(
  Relating=@9901, Related=@491, Origin=@492, Target=@9951); # cable W2
@9117:GeometryToTopologyRepresentationAssociation(
  Relating=@9901, Related=@591, Origin=@493, Target=@9953); # protection W3

@9900:GeometricCoordinateSpace( Units=@8, DimensionCount=1 );
@9901:EdgeBasedTopologicalRepresentationWithLengthConstraint(
  Items=(@9902,@9951,@9952,@9953), # the ConnectedEdgeSet + paths
  ContextOfItems=@9900 );
@9902:ConnectedEdgeSet( ConnectedEdges=(@9931,@9932,@9933,@9934,@9935) );
  # only main edges, not sub-edges

@9911:Point();
@9912:Point();
@9913:Point();
@9914:Point();
@9915:Point();
@9916:Point();
@9917:PointOnCurve( BasicCurve=@9942, Parameter=2.0 );
  # in the middle of the basic curve
@9918:PointOnCurve( BasicCurve=@9943, Parameter=3.0 );
  # in the middle of the basic curve

@9921:VertexPoint( name='N1', VertexGeometry=@9911 );
@9922:VertexPoint( name='N2', VertexGeometry=@9912 );
@9923:VertexPoint( name='N3', VertexGeometry=@9913 );
```

```
@9924:VertexPoint( name='N4', VertexGeometry=@9914 );
@9925:VertexPoint( name='N5', VertexGeometry=@9915 );
@9926:VertexPoint( name='N6', VertexGeometry=@9916 );
@9927:VertexPoint( name='N7', VertexGeometry=@9917 );
@9928:VertexPoint( name='N8', VertexGeometry=@9918 );

@9931:EdgeBoundedCurveWithLength( name='S1', EdgeGeometry=@9941 );
undirected_edge(@9931, @9921, @9923)
@9932:EdgeBoundedCurveWithLength( name='S2', EdgeGeometry=@9942 );
undirected_edge(@9932, @9922, @9923)
@9933:EdgeBoundedCurveWithLength( name='S3', EdgeGeometry=@9943 );
undirected_edge(@9933, @9923, @9924);
@9934:EdgeBoundedCurveWithLength( name='S4', EdgeGeometry=@9944 );
undirected_edge(@9934, @9924, @9925);
@9935:EdgeBoundedCurveWithLength( name='S5', EdgeGeometry=@9945 );
undirected_edge(@9935,@9924,@9926);
@9936:SubEdge( name='S2.2', ParentEdge=@9932 );
undirected_edge(@9936,@9927,@9923);
@9937:SubEdge( name='S3.1', ParentEdge=@9933);
undirected_edge(@9937,@9923,@9928);

@9941:BoundedCurveWithLength( EdgeLength=PositiveLengthMeasure(2.0) );
  # for S1
@9942:BoundedCurveWithLength( EdgeLength=PositiveLengthMeasure(4.0) );
  # for S2
@9943:BoundedCurveWithLength( EdgeLength=PositiveLengthMeasure(6.0) );
  # for S3
@9944:BoundedCurveWithLength( EdgeLength=PositiveLengthMeasure(8.0) );
  # for S4
@9945:BoundedCurveWithLength( EdgeLength=PositiveLengthMeasure(10.0) );
  # for S5

# vendors to ensure that the edge are oriented in correct way
@9951:Path( name="P1", EdgeList=(@9931,@9933,@9934) ); # S1+S3+S4
@9952:Path( name="P2", EdgeList=(@9932,@9933,@9935) ); # S2+S3+S5
@9953:Path( name="P3", EdgeList=(@9936,@9937) );       # S2.2+S3.1

sizeof(Part) = 6;
sizeof(PartVersion) = 6;
sizeof(PartView) = 5;
sizeof(WiringHarnessAssemblyDesign) = 1;
sizeof(NextAssemblyOccurrenceUsage) = 7;
sizeof(SingleOccurrence) = 4;
sizeof(WireOccurrence) = 1;
sizeof(CableOccurrence) = 1;

sizeof(GeometricCoordinateSpace) = 1;
sizeof(EdgeBasedTopologicalRepresentationWithLengthConstraint) = 1;
sizeof(ConnectedEdgeSet) = 1;
sizeof(BoundedCurveWithLength) = 5;
sizeof(EdgeBoundedCurveWithLength) = 5;
sizeof(VertexPoint) = 8;
sizeof(Point) = 6;
sizeof(CartesianPoint) = 0;
sizeof(PointOnCurve) = 2;
sizeof(SubEdge) = 2;
sizeof(Path) = 3;

);
```

4.4 EWH-Topology3

This test case is an extension of the content in EWH-Topology2 for:

- external references into p21 files:
  - complete p21 files for discrete parts "Connector-A" and "Lug-B"
  - element reference into p21 file for centre-curves and axis-placements
- topology-to-geometry association

This test case is likely to be refined later on as the topic of XML “external element references” is new to the community of STEP implementers, and there are no final recommended practices yet for this area (need common work with CAX-IF and PDM-IF). So even if only a subset of the below gets implemented would already be a success.

Formal test-case specification:

```
Test EWH-Topology3 (

@9:FormatProperty( DataFormat="ISO 10303-242", CharacterCode="ISO 8859-
1" );

@8:Unit( Name=ClassString("metre"), Quantity=ClassString("length") );

@100:Part( PartTypes[i]=PartCategoryEnum(connector),
           PartTypes[i]=PartCategoryEnum(discrete) );
@101:PartVersion;
@102:PartView( DefiningGeometry=@191 );
Part_with_Name_and_PartView( @100, "Connector-A", @101, @102, @4);
@111:SingleOccurrence( Id=IdentifierString("J1"), Definition=@102 );
@121:SingleOccurrence( Id=IdentifierString("J2"), Definition=@102 );
@190:GeometricCoordinateSpace( DimensionCount=3, Items=(@192) );
@191:ExternalGeometricModel( items=(@192), ContextOfItems=@190,
       ExternalFile=@193  ); # was GeometricModel in EWH-Topology2
@192:AxisPlacement( Position=(0.0, 0.0, 0.0) ); # Axis and RefDirection de-
faults
  # alternatively use ExternalRepresentationItem to select placement in p21
file
@193:DigitalFile( FileLocations=@194, FileFormat=@9, exists(Id) ); #
id=file name

@200:Part( PartTypes[i]=PartCategoryEnum(terminal_lug),
           PartTypes[i]=PartCategoryEnum(discrete) );
@201:PartVersion;
@202:PartView( DefiningGeometry=@291 );
Part_with_Name_and_PartView(@200, "Lug-B", @201, @202, @4);
@211:SingleOccurrence( Id=IdentifierString("J3"), Definition=@202 );
@221:SingleOccurrence( Id=IdentifierString("J4"), Definition=@202 );
```

```
@290:GeometricCoordinateSpace( DimensionCount=3, Items=(@192) );
@291:ExternalGeometricModel( Items=(@292), ContextOfItems=@290,
  ExternalFile=@293  ); # was GeometricModel in EWH-Topology2
@292:AxisPlacement( Position=(0.0, 0.0, 0.0) ); # Axis and RefDirection de-
faults
@293:DigitalFile( FileLocations=@294, FileFormat=@9, exists(Id) ); #
id=file name

@300:Part( PartTypes[i]=PartCategoryEnum(wire),
           PartTypes[i]=PartCategoryEnum(raw_material_by_length) );
@301:PartVersion;
@302:PartView( DefiningGeometry=@391 );
Part_with_Name_and_PartView(@300, "Wire-C", @301, @302, @4);
@311:WireOccurrence( Id=IdentifierString("W1"),
  Definition=@302, Quantity=@312 );
@312:NumericalValue( Unit=@8, ValueComponent=3.5 );
@390:GeometricCoordinateSpace( DimensionCount=2 );
@391:GeometricModel( name='2D cross section',
  items[i]=@392, ContextOfItems=@390 );
@392:AxisPlacement; # placeholder for 2D centre

@400:Part( PartTypes[i]=PartCategoryEnum(cable),
           PartTypes[i]=PartCategoryEnum(raw_material_by_length) );
@401:PartVersion;
@402:PartView( DefiningGeometry=@491 );
Part_with_Name_and_PartView(@400, "Cable-D", @401, @402, @4);
@411:CableOccurrence( Id=IdentifierString("W2"), Definition=@402,
Quantity=@412 );
@412:NumericalValue( Unit=@8, ValueComponent=1.8 );
@490:GeometricCoordinateSpace( DimensionCount=2 );
@491:GeometricModel( name='cross section', items[i]=@492,
ContextOfItems=@490 );
@492:AxisPlacement;

@500:Part( PartTypes[i]=PartCategoryEnum(protective_covering),
           PartTypes[i]=PartCategoryEnum(raw_material_by_length) );
@501:PartVersion;
@502:PartView( DefiningGeometry=@591 );
Part_with_Name_and_PartView(@500, "Protection-E", @501, @502, @4);
@511:QuantifiedOccurrence( Id=IdentifierString("W3"),
                           Definition=@402, Quantity=@412 );
@512:NumericalValue( Unit=@8, ValueComponent=1.2 );
@590:GeometricCoordinateSpace( DimensionCount=2 );
@591:GeometricModel( name='cross section',
                     items[i]=@592, ContextOfItems=@590 );
@592:AxisPlacement;

@9000:Part;
@9001:PartVersion;
@9002:WiringHarnessAssemblyDesign(
  Topology=@9901, DefiningGeometry=@9201  );
@9003:ViewContext;
@9004:ViewContext;
harness_design_with_segment_topology(@9000,
  "EWH Test-Case Topology2",@9001,@9002,@9003,@9004);

@9101:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@111,
  Placement=(@9111,@9801) ); # connector J1
@9102:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@121,
  Placement=(@9112,@9802) ); # connector J2
@9103:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@211,
  Placement=(@9113,@9803) ); # terminal lug J3
@9104:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@221,
  Placement=(@9114,@9804) ); # terminal lug J4
@9105:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@311,
  Placement=(@9115) ); # wire W1
@9106:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@411,
  Placement=(@9116) ); # cable W2
@9107:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@511,
  Placement=(@9117) ); # protection W3

@9111:GeometryToTopologyModelAssociation (
  Relating=@9901, Related=@191, Origin=@192, Target=@9921); # connector J1
@9112:GeometryToTopologyRepresentationAssociation(
  Relating=@9901, Related=@191, Origin=@192, Target=@9925); # connector J2
@9113:GeometryToTopologyRepresentationAssociation(
  Relating=@9901, Related=@291, Origin=@292, Target=@9922); # terminal lug
J3
@9114:GeometryToTopologyRepresentationAssociation(
  Relating=@9901, Related=@291, Origin=@292, Target=@9926); # terminal lug
J4
@9115:GeometryToTopologyRepresentationAssociation(
  Relating=@9901, Related=@391, Origin=@392, Target=@9952); # wire W1
@9116:GeometryToTopologyRepresentationAssociation(
  Relating=@9901, Related=@491, Origin=@492, Target=@9951); # cable W2
@9117:GeometryToTopologyRepresentationAssociation(
  Relating=@9901, Related=@591, Origin=@493, Target=@9953); # protection W3

@9200:GeometricCoordinateSpace( Units=@8, DimensionCount=3 );
@9201:ComposedGeometricModel( ContextOfItems=@9200;
Items=(@9211,@9212,@9213,@9214) );
  # contains the connectors and multi-branchable
@9211=AxisPlacement; # alternativel ExternalRepresentationItem from p21
files
@9212=AxisPlacement;
@9213=AxisPlacement;
@9214=AxisPlacement;
@9801:GeometricRepresentationRelationshipWithPlacementTransformation(
  origin=@192, target=@9211,
  relating=@9201, related=@191, Definitional=TRUE ); # connector J1
@9802:GeometricRepresentationRelationshipWithPlacementTransformation(
  origin=@192, target=@9212,
  relating=@9201, related=@191, Definitional=TRUE ); # connector J2
@9803:GeometricRepresentationRelationshipWithPlacementTransformation(
  origin=@292, target=@9213,
  relating=@9201, related=@292, Definitional=TRUE ); # terminal lug J3
@9804:GeometricRepresentationRelationshipWithPlacementTransformation(
  origin=@292, target=@9214,
  relating=@9201, related=@292, Definitional=TRUE ); # terminal lug J4
@9808:GeometricRepresentationRelationshipWithSameCoordinateSpace(
  relating=@9201, related=@9211, Definitional=TRUE ); # for stuff in the
multi-branchable

@9210:DigitalFile( FileFormat=@9, FileFormat=@9, exisit(Id) ); # Id=name of
p21 file
@9211:ExternalGeometricModel( items=(@9212,@9213,@9214,@9215),
  ContextOfItems=@9200, ExternalFile=@9210 ); # multi-branchable
@9220:AxisPlacement;
@9221:ExternalRepresentationItem( External=@9231 );
@9222:ExternalRepresentationItem( External=@9232 );
@9223:ExternalRepresentationItem( External=@9233 );
@9224:ExternalRepresentationItem( External=@9234 );
@9225:ExternalRepresentationItem( External=@9235 );
@9226:ExternalRepresentationItem( External=@9236 );
@9227:ExternalRepresentationItem( External=@9237 );

# for the following instance the ID attribute must be set
# corresponding to an anchor or instance-id in the target p21 file
@9231:ExternalEntityInstance( exist(Id), Source=@9210 );
@9232:ExternalEntityInstance( exist(Id), Source=@9210 );
@9223:ExternalEntityInstance( exist(Id), Source=@9210 );
@9224:ExternalEntityInstance( exist(Id), Source=@9210 );
@9225:ExternalEntityInstance( exist(Id), Source=@9210 );
@9226:ExternalEntityInstance( exist(Id), Source=@9210 ); # curve for S2.2
@9227:ExternalEntityInstance( exist(Id), Source=@9210 ); # curve for S3.1

# Alternative for @9226 and @9227
# use PointOnCurve with PARAMETER given in p21 file and
# construct a new curve in XML to associat to

@9299:TopologyToGeometryModelAssociation( Relating=@9201, Related=@9901,
  # order of pairs: connector J1, ... J2, terminal lug J3, ... J4, edges
S1..S5, S2.2, S3.1
  # maybe instead of paths we have to map single EdgeBoundedCurveWithLength
   Origin=(@9921,@9925,@9922,@9926, @9931,@9932,@9933,@9934,@9935,
@9936,@9937),
   Target=(@9211,@9212,@9213,@9214, @9221,@9222,@9223,@9224,@9225,
@9226,@9227) );

@9900:GeometricCoordinateSpace( Units=@8, DimensionCount=1 );
@9901:EdgeBasedTopologicalRepresentationWithLengthConstraint(
  Items=(@9902,@9951,@9952,@9953), # the ConnectedEdgeSet + paths
  ContextOfItems=@9900 );
@9902:ConnectedEdgeSet( ConnectedEdges=(@9931,@9932,@9933,@9934,@9935) );
  # only main edges, not sub-edges

@9911:Point();
@9912:Point();
@9913:Point();
@9914:Point();
@9915:Point();
@9916:Point();
```

```
@9917:PointOnCurve( BasicCurve=@9942, Parameter=2.0 );
  # in the middle of the basic curve
@9918:PointOnCurve( BasicCurve=@9943, Parameter=3.0 );
  # in the middle of the basic curve

@9921:VertexPoint( name='N1', VertexGeometry=@9911 );
@9922:VertexPoint( name='N2', VertexGeometry=@9912 );
@9923:VertexPoint( name='N3', VertexGeometry=@9913 );
@9924:VertexPoint( name='N4', VertexGeometry=@9914 );
@9925:VertexPoint( name='N5', VertexGeometry=@9915 );
@9926:VertexPoint( name='N6', VertexGeometry=@9916 );
@9927:VertexPoint( name='N7', VertexGeometry=@9917 );
@9928:VertexPoint( name='N8', VertexGeometry=@9918 );

@9931:EdgeBoundedCurveWithLength( name='S1', EdgeGeometry=@9941 );
undirected_edge(@9931, @9921, @9923)
@9932:EdgeBoundedCurveWithLength( name='S2', EdgeGeometry=@9942 );
undirected_edge(@9932, @9922, @9923)
@9933:EdgeBoundedCurveWithLength( name='S3', EdgeGeometry=@9943 );
undirected_edge(@9933, @9923, @9924);
@9934:EdgeBoundedCurveWithLength( name='S4', EdgeGeometry=@9944 );
undirected_edge(@9934, @9924, @9925);
@9935:EdgeBoundedCurveWithLength( name='S5', EdgeGeometry=@9945 );
undirected_edge(@9935,@9924,@9926);
@9936:SubEdge( name='S2.2', ParentEdge=@9932 );
undirected_edge(@9936,@9927,@9923);
@9937:SubEdge( name='S3.1', ParentEdge=@9933);
undirected_edge(@9937,@9923,@9928);

@9941:BoundedCurveWithLength( EdgeLength=PositiveLengthMeasure(2.0) );
  # for S1
@9942:BoundedCurveWithLength( EdgeLength=PositiveLengthMeasure(4.0) );
  # for S2
@9943:BoundedCurveWithLength( EdgeLength=PositiveLengthMeasure(6.0) );
  # for S3
@9944:BoundedCurveWithLength( EdgeLength=PositiveLengthMeasure(8.0) );
  # for S4
@9945:BoundedCurveWithLength( EdgeLength=PositiveLengthMeasure(10.0) );
  # for S5

# vendors to ensure that the edge are oriented in correct way
@9951:Path( name="P1", EdgeList=(@9931,@9933,@9934) ); # S1+S3+S4
@9952:Path( name="P2", EdgeList=(@9932,@9933,@9935) ); # S2+S3+S5
@9953:Path( name="P3", EdgeList=(@9936,@9937) );       # S2.2+S3.1

);
```

4.5 EWH-Connectivity1

This test case consists of a WiringHarnessAssemblyDesign that is composed of

- a terminal lug “LUG01” that is defined by Part “640903-1” with a single terminal “1”
- a connector “PLUG01” that is defined by Part “RCA123” with terminals “0” and “1”

- a connector "P-CONN01" that is defined by Part "IMC16-2002X" with terminals "1" and "2"
- a cable "CABLE01" that is defined by Part "9962 009100" with two wires, one black and the other white
- a wire "WIRE01" that is defined by Part "83027 001100"
- the two connectors are joint to the two ends of the cable.
- the single wire connects LUG01 with terminal "1" of "PLUG01"

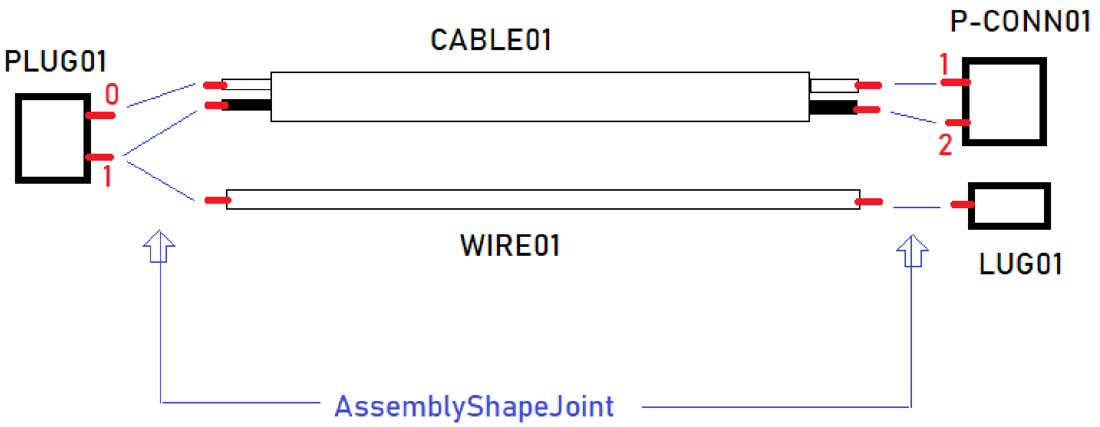


Figure 4: Test case: EWH-Connectivity1

Formal test-case specification:

```
Test EWH-Connectivity1 (

@4:ViewContext;
@5:ViewContext;
@8:Unit( Name=ClassString("metre"), Quantity=ClassString("length") );

@50:Organization( name="MIL ...??")
@51:Organization( name="Deutch Company Ltd ...")
@52:Organization( name="BELDEN company ...")
@60:Identifier( Id=IdentifierString("Standard RCA connector") )

@70:WireColourBasedIdentificationCode( Id="white" );
@71:WireColourBasedIdentificationCode( Id="black" );


# Terminal Lug
@100:Part( PartTypes[i]=PartCategoryEnum(terminal_lug),
  PartTypes[i]=PartCategoryEnum(discrete) );
@101:PartVersion;
@102:PartView;
@103:Identifier( Id=IdentifierString("640903-1"), IdentificationContext=@50
)
Part_with_ID_and_PartView(@100, @103, @101, @102, @4);
@104:PartTerminal( ElementOf=@102, Id="1", DomainType="electrical",
  InterfaceOrJoinTerminal="join_terminal" );
@111:SingleOccurrence( Id=IdentifierString("LUG01"), Definition=@102 );
@112:OccurrenceTerminal( ElementOf=@111, Definition=@104 );

# Connector with integrated contacts
@200:Part( PartTypes[i]=PartCategoryEnum(connector), PartTypes[i]=PartCate-
goryEnum(discrete) );
@201:PartVersion;
@202:PartView;
@203:Identifier( Id=IdentifierString("RCA123"), IdentificationContext=@60 )
Part_with_ID_and_PartView(@200, @203, @201, @202, @4);
@204:PartTerminal( ElementOf=@202, Id="0", DomainType="electrical",
  InterfaceOrJoinTerminal="join_terminal" );
@205:PartTerminal( ElementOf=@202, Id="1", DomainType="electrical",
  InterfaceOrJoinTerminal="join_terminal" );
@211:SingleOccurrence( Id=IdentifierString("PLUG01"), Definition=@202 );
  @214:OccurrenceTerminal( ElementOf=@211, Definition=@204 );
  @215:OccurrenceTerminal( ElementOf=@211, Definition=@205 );

# Simplified model for Deutch connector with direct PartTerminals
@300:Part( PartTypes[i]=PartCategoryEnum(connector), PartTypes[i]=PartCate-
goryEnum(discrete) );
@301:PartVersion;
@302:PartView;
@303:Identifier( Id=IdentifierString("IMC16-2002X"),
IdentificationContext=@51 )
Part_with_ID_and_PartView(@300, @303, @301, @302, @4);
@306:PartTerminal( ElementOf=@302, Id="1", DomainType="electrical",
  InterfaceOrJoinTerminal="join_terminal" );
@307:PartTerminal( ElementOf=@302, Id="2", DomainType="electrical",
  InterfaceOrJoinTerminal="join_terminal" );
@311:SingleOccurrence( Id=IdentifierString("P-CONN01"), Definition=@302 );
#316:OccurrenceTerminal( ElementOf=@311, Definition=@306 );
#317:OccurrenceTerminal( ElementOf=@311, Definition=@307 );

# Cable
@500:Part( PartTypes[i]=PartCategoryEnum(cable), PartTypes[i]=PartCatego-
ryEnum(raw_material_by_length) );
@501:PartVersion;
@502:PartView;
@503:Identifier( Id=IdentifierString("9962 009100"),
IdentificationContext=@52 )
Part_with_ID_and_PartView(@500, @503, @501, @502, @4);
@511:CableOccurrence( Id=IdentifierString("CABLE01"), Definition=@502,
Quantity=@512 );
@512:NumericalValue( Unit=@8, ValueComponent=1.8 );
@513:WireIdentification( ElementOf=@511, Id="CABLE01-WHT" code=@70 );
@514:WireIdentification( ElementOf=@511, Id="CABLE01-BLK" code=@71 );
@515:CableOccurrenceTerminalLocationGroup( ElementOf=@511, Name="end a" );
@516:CableOccurrenceTerminalLocationGroup( ElementOf=@511, Name="end b" );
@521:CableOccurrenceTerminal( ElementOf=@511,
AssociatedTransportFeature=@513,
  LocationGroup=@515 );
```

```
@522:CableOccurrenceTerminal( ElementOf=@511,
AssociatedTransportFeature=@513,
  LocationGroup=@516 );
@523:CableOccurrenceTerminal( ElementOf=@511,
AssociatedTransportFeature=@514,
  LocationGroup=@515 );
@524:CableOccurrenceTerminal( ElementOf=@511,
AssociatedTransportFeature=@514,
  LocationGroup=@516 );

# Wire
@600:Part( PartTypes[i]=PartCategoryEnum(wire),
  PartTypes[i]=PartCategoryEnum(raw_material_by_length) );
@601:PartVersion;
@602:PartView;
@603:Identifier( Id=IdentifierString("83027 001100"), IdentificationContext=@52 )
Part_with_ID_and_PartView(@600, @603, @601, @602, @4);
@611:WireOccurrence( Id=IdentifierString("WIRE01"), Definition=@602, Quantity=@612 );
@612:NumericalValue( Unit=@8, ValueComponent=3.5 );
@613=WireIdentification( ElementOf=@611, DomainType="electrical" ... )
@614=WireOccurrenceTerminal( ElementOf=@611,
AssociatedTransportFeature=@613,
  Name="end a" );
@615=WireOccurrenceTerminal( ElementOf=@611,
AssociatedTransportFeature=@613,
  Name="end b" );

# EWH-Assembly
@9000:Part;
@9001:PartVersion;
@9002:WiringHarnessAssemblyDesign;
@9003:ViewContext;
@9004:ViewContext;
part_with_WiringHarnessAssemblyDesign( @9000,
  "EWH Test-Case Connectivity1",@9001,@9002,@9003,@9004 );

@9101:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@111 );
@9102:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@211 );
@9103:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@311 );
@9106:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@511 );
@9107:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@611 );

# connections
@9210:AssemblyShapeJoint( ElementOf=@9002 );
  @9211:AssemblyShapeJointItemRelationship( Relating=@9210, Related=@214 );
# PLUG01 / 0
  @9212:AssemblyShapeJointItemRelationship( Relating=@9210, Related=@521 );
# CABLE01-WHT / end a
@9220:AssemblyShapeJoint( ElementOf=@9002 );
  @9221:AssemblyShapeJointItemRelationship( Relating=@9220, Related=@316 );
# P-CONN01 / 01
  @9222:AssemblyShapeJointItemRelationship( Relating=@9220, Related=@522 );
# CABLE01-WHT / end b
```

```
@9230:AssemblyShapeJoint( ElementOf=@9002 );
  @9231:AssemblyShapeJointItemRelationship( Relating=@9230, Related=@215 );
# PLUG01 / 1
  @9232:AssemblyShapeJointItemRelationship( Relating=@9230, Related=@523 );
# CABLE01-BLK / end a
  @9232:AssemblyShapeJointItemRelationship( Relating=@9230, Related=@614 );
# WIRE01 / end a
@9240:AssemblyShapeJoint( ElementOf=@9002 );
  @9241:AssemblyShapeJointItemRelationship( Relating=@9240, Related=@317 );
# P-CONN01 / 02
  @9242:AssemblyShapeJointItemRelationship( Relating=@9240, Related=@524 );
# CABLE01-BLK / end a
@9250:AssemblyShapeJoint( ElementOf=@9002 );
  @9251:AssemblyShapeJointItemRelationship( Relating=@9250, Related=@112 );
# LUG01 / 1
  @9252:AssemblyShapeJointItemRelationship( Relating=@9250, Related=@615 );
# WIRE01 / end b

sizeof(Part) = 6;
sizeof(PartVersion) = 6;
sizeof(PartView) = 5;
sizeof(WiringHarnessAssemblyDesign) = 1;
sizeof(NextAssemblyOccurrenceUsage) = 5;
sizeof(SingleOccurrence) = 3;
sizeof(WireOccurrence) = 1;
sizeof(CableOccurrence) = 1;

sizeof(PartTerminal) = 5; # only the join terminals
sizeof(OccurrenceTerminal) = 5; # only the join terminals
sizeof(WireColourBasedIdentificationCode) = 2;
sizeof(WireOccurrenceTerminal) = 2;
sizeof(CableOccurrenceTerminalLocationGroup) = 2;
sizeof(CableOccurrenceTerminal) = 4;
sizeof(AssemblyShapeJoint) = 5;
sizeof(AssemblyShapeJointItemRelationship) = 11;

);
```

4.6 EWH-Connectivity2

This test case is very similar to the test case EWH-Connectivity1. The difference is that the connector “P-CONN01” is now modeled more realistically. There is no direct terminal but instead there are the two cavities “1” and “2” for two separate connector contacts:

- a terminal lug “LUG01” that is defined by Part “640903-1” with a single terminal “1”
- a connector “PLUG01” that is defined by Part “RCA123” with terminals “0” and “1”
- a connector “P-CONN01” that is defined by Part “IMC16-2002X” with cavities “1” and “2”
- two connector contacts "P-CONN01-01" and "P-CONN01-02" that are defined by Part "6860-201-20278" that fits into the cavities of a connector of type “IMC16-2002X”. Each of the connector contacts has a single join terminal.

- a cable “CABLE01” that is defined by Part “9962 009100” with two wires, one black and the other white
- a wire “WIRE01” that is defined by Part “83027 001100”
- connector “PLUG01” is joint to one ends of the cable, and the two connector contacts are joint to the other end. The connector contacts are then inserted into connector “PLUG01”
- the single wire connects LUG01 with terminal “1” of “PLUG01”

Initial input data from users to this test:

Part Number	Occurrence (REFDES)	Terminals	Description	Images
640903-1	LUG01	1	MIL standard Receptacle (similar to Lug)	
RCA123	PLUG01	0	Standard RCA plug (Or Cinch) https://en.wikipedia.org/wiki/RCA_connector	
		1		
IMC16-2002X	P-CONN01		Deutch waterproof connector with two cavities	
6860-201-20278	P-CONN01-01		Deutch Plug Contact	
	P-CONN01-02			
9962 009100			BELDEN Cable	
83027 001100			BELDEN Wire	

Table 1: Original part list for connectivity test

From	From Pin	Wire Name	Material	To	To Pin
PLUG01	0	CABLE01-WHT	9962 009100	P-CONN01	P-CONN01-01
PLUG01	1	CABLE01-BLK	9962 009100	P-CONN01	P-CONN01-02
PLUG01	1	WIRE01	83027 001100	LUG01	

Table 2: Original wire list for connectivity test

Adaptions on the original input data to be used by AP242-EWH:

- the cavities of the connector “IMC16-2002X” are not numbered, but they are in the Deutsch documentation indicated with “1” and “2”. It is essential to not mix them up;
- no terminals are defined for the connector contact “6860-201-20278”, but of course there is an implicit join-terminal (for crimping) and an interface-terminal for the external connection (the later one is not covered here);
- there is no explicit information which connector contact P-CONN01-01/-02 goes into the cavities 1/2 of the connector. This can only be derived from the naming. For AP242-EWH it is essential to state which connector-contact is inserted into which cavity of the connector (by AssemblyShapeJoint);
- the wire list indicates two connections onto the PLUG01/1 pin. For AP242-EWH this is handled by a triple AssemblyShapeJoint of PLUG01/1 with the cable and single wire terminals.

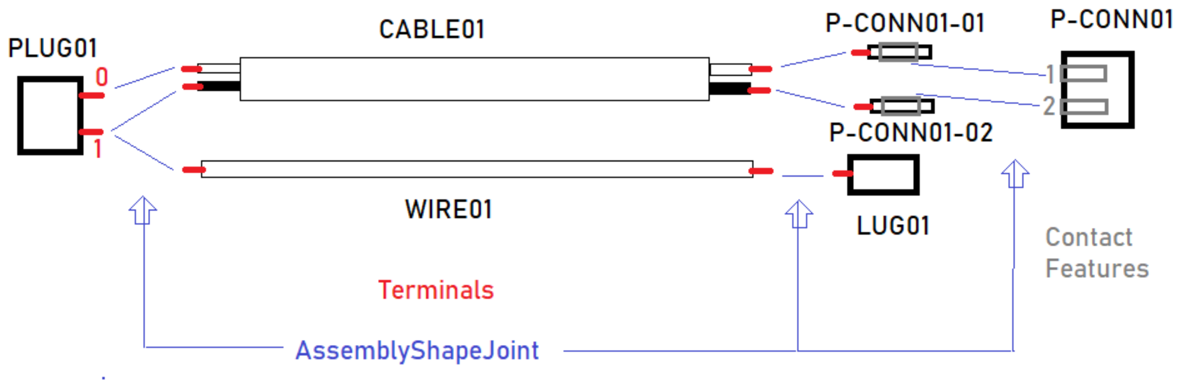


Figure 5: Test case: EWH-Connectivity2

Formal test-case specification:

```
Test EWH-Connectivity2 (

@4:ViewContext;
@5:ViewContext;
@8:Unit( Name=ClassString("metre"), Quantity=ClassString("length") );

@50:Organization( name="MIL ...??")
@51:Organization( name="Deutch Company Ltd ...")
@52:Organization( name="BELDEN company ...")
@60:Identifier( Id=IdentifierString("Standard RCA connector") )

@70:WireColourBasedIdentificationCode( Id="white" );
@71:WireColourBasedIdentificationCode( Id="black" );


# Terminal Lug
@100:Part( PartTypes[i]=PartCategoryEnum(terminal_lug),
  PartTypes[i]=PartCategoryEnum(discrete) );
@101:PartVersion;
@102:PartView;
@103:Identifier( Id=IdentifierString("640903-1"),
  IdentificationContext=@50 )
```

```
Part_with_ID_and_PartView(@100, @103, @101, @102, @4);
@104:PartTerminal( ElementOf=@102, Id="1", DomainType="electrical",
    InterfaceOrJoinTerminal="join_terminal" );
@111:SingleOccurrence( Id=IdentifierString("LUG01"), Definition=@102 );
  @112:OccurrenceTerminal( ElementOf=@111, Definition=@104 );

# Connector with integrated contacts
@200:Part( PartTypes[i]=PartCategoryEnum(connector),
  PartTypes[i]=PartCategoryEnum(discrete) );
@201:PartVersion;
@202:PartView;
@203:Identifier( Id=IdentifierString("RCA123"),
  IdentificationContext=@60 )
Part_with_ID_and_PartView( @200, "", @201, @202, @4);
@204:PartTerminal( ElementOf=@202, Id="0", DomainType="electrical",
  InterfaceOrJoinTerminal="join_terminal" ); # or left, right m GND ?
@205:PartTerminal( ElementOf=@202, Id="1", DomainType="electrical",
  InterfaceOrJoinTerminal="join_terminal" );
@211:SingleOccurrence( Id=IdentifierString("PLUG01"), Definition=@202 );
   @214:OccurrenceTerminal( ElementOf=@211, Definition=@204 );
   @215:OccurrenceTerminal( ElementOf=@211, Definition=@205 );

# Realistic model for Deutsch connector with cavities
@300:Part( PartTypes[i]=PartCategoryEnum(connector),
  PartTypes[i]=PartCategoryEnum(discrete) );
@301:PartVersion;
@302:PartView;
@303:Identifier( Id=IdentifierString("IMC16-2002X"),
  IdentificationContext=@51 )
Part_with_ID_and_PartView( @300, @303, @301, @302, @4);
@306:PartContactFeature( ElementOf=@302, Id="1", Definition=@1001 );
@307:PartContactFeature( ElementOf=@302, Id="2", Definition=@1001 );
@311:SingleOccurrence( Id=IdentifierString("P-CONN01"), Definition=@302 );
   #316:OccurrenceContactFeature( ElementOf=@311, Definition=@306 );
   #317:OccurrenceContactFeature( ElementOf=@311, Definition=@307 );

# Contact for Deutsch connector
@400:Part( PartTypes[i]=PartCategoryEnum(connector_contact),
  PartTypes[i]=PartCategoryEnum(discrete) );
@401:PartVersion;
@402:PartView;
@403:Identifier( Id=IdentifierString("6860-201-20278"),
  IdentificationContext=@51 )
Part_with_ID_and_PartView( @400, @403, @401, @402, @4);
@406:PartTerminal( ElementOf=@402, Id="j", DomainType="electrical",
  InterfaceOrJoinTerminal="join_terminal" );
@407:PartContactFeature( ElementOf=@402, Id="o", Definition=@1002 );
@411:SingleOccurrence( Id=IdentifierString("P-CONN01-01"),
Definition=@402 );
   @412:OccurrenceTerminal( ElementOf=@411, Definition=@406 );
   @413:OccurrenceContactFeature( ElementOf=@411, Definition=@407 );
@421:SingleOccurrence( Id=IdentifierString("P-CONN01-02"),
Definition=@402 );
```

```
 @422:OccurrenceTerminal( ElementOf=@421, Definition=@406 );
 @423:OccurrenceContactFeature( ElementOf=@421, Definition=@407 );

# Cable
@500:Part( PartTypes[i]=PartCategoryEnum(cable),
 PartTypes[i]=PartCategoryEnum(raw_material_by_length) );
@501:PartVersion;
@502:PartView;
@503:Identifier( Id=IdentifierString("9962 009100"),
IdentificationContext=@52 )
Part_with_Name_and_PartView(@500, "Cable-D", @501, @502, @4);
@511:CableOccurrence( Id=IdentifierString("CABLE01"), Definition=@502,
Quantity=@512 );
 @512:NumericalValue( Unit=@8, ValueComponent=1.8 );
 @513:WireIdentification( ElementOf=@511, Id="CABLE01-WHT" code=@70 );
 @514:WireIdentification( ElementOf=@511, Id="CABLE01-BLK" code=@71 );
 @515:CableOccurrenceTerminalLocationGroup( ElementOf=@511, Name="end
a" );
 @516:CableOccurrenceTerminalLocationGroup( ElementOf=@511, Name="end
b" );
 @521:CableOccurrenceTerminal( ElementOf=@511,
  AssociatedTransportFeature=@513, LocationGroup=@515 );
 @522:CableOccurrenceTerminal( ElementOf=@511,
  AssociatedTransportFeature=@513, LocationGroup=@516 );
 @523:CableOccurrenceTerminal( ElementOf=@511,
  AssociatedTransportFeature=@514, LocationGroup=@515 );
 @524:CableOccurrenceTerminal( ElementOf=@511,
  AssociatedTransportFeature=@514, LocationGroup=@516 );

# Wire
@600:Part( PartTypes[i]=PartCategoryEnum(wire),
 PartTypes[i]=PartCategoryEnum(raw_material_by_length) );
@601:PartVersion;
@602:PartView;
@603:Identifier( Id=IdentifierString("83027 001100"),
 IdentificationContext=@52 )
Part_with_Name_and_PartView(@600, "Wire-C", @601, @602, @4);
@611:WireOccurrence( Id=IdentifierString("WIRE01"), Definition=@602,
 Quantity=@612 );
@612:NumericalValue( Unit=@8, ValueComponent=3.5 );
@613=WireIdentification( ElementOf=@611, DomainType="electrical" ... )
@614=WireOccurrenceTerminal( ElementOf=@611,
 AssociatedTransportFeature=@613, Name="end a" );
@615=WireOccurrenceTerminal( ElementOf=@611,
 AssociatedTransportFeature=@613, Name="end b" );

# Deutsch IMC Series cavity & contact shapes
@1000:ContactFeatureDefinitionFitRelationship( Name="Deutsch IMC Series
Size 20 fit",
 Relating=@1001, Related=@1002 );
@1001:ContactFeatureDefinition( Name="Deutsch IMC Series Size 20 cavity",
 ShapeFeatureType=cavity_profile );
@1002:ContactFeatureDefinition( Name="Deutsch IMC Series Size 20 pin",
 ShapeFeatureType=contact_profile );
```

```
# EWH-Assembly
@9000:Part;
@9001:PartVersion;
@9002:WiringHarnessAssemblyDesign;
@9003:ViewContext;
@9004:ViewContext;
part_with_WiringHarnessAssemblyDesign( @9000,"EWH Test-Case Connectivity2",
  @9001,@9002,@9003,@9004 );

@9101:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@111 );
@9102:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@211 );
@9103:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@311 );
@9104:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@411 ); # "P-
CONN01-01"
@9105:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@421 ); # "P-
CONN01-02"
@9106:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@511 );
@9107:NextAssemblyOccurrenceUsage( Relating=@9002, Related=@611 );

# electrical connections
@9210:AssemblyShapeJoint( ElementOf=@9002 );
  @9211:AssemblyShapeJointItemRelationship( Relating=@9210, Related=@214 );
# PLUG01 / 0
  @9212:AssemblyShapeJointItemRelationship( Relating=@9210, Related=@521 );
# CABLE01-WHT / end a
@9220:AssemblyShapeJoint( ElementOf=@9002 );
  @9221:AssemblyShapeJointItemRelationship( Relating=@9220, Related=@412 );
# P-CONN01-01 / j
  @9222:AssemblyShapeJointItemRelationship( Relating=@9220, Related=@522 );
# CABLE01-WHT / end b
@9230:AssemblyShapeJoint( ElementOf=@9002 );
  @9231:AssemblyShapeJointItemRelationship( Relating=@9230, Related=@215 );
# PLUG01 / 1
  @9232:AssemblyShapeJointItemRelationship( Relating=@9230, Related=@523 );
# CABLE01-BLK / end a
  @9232:AssemblyShapeJointItemRelationship( Relating=@9230, Related=@614 );
# WIRE01 / end a
@9240:AssemblyShapeJoint( ElementOf=@9002 );
  @9241:AssemblyShapeJointItemRelationship( Relating=@9240, Related=@422 );
#  P-CONN01-02 / j
  @9242:AssemblyShapeJointItemRelationship( Relating=@9240, Related=@524 );
# CABLE01-BLK / end b
@9250:AssemblyShapeJoint( ElementOf=@9002 );
  @9251:AssemblyShapeJointItemRelationship( Relating=@9250, Related=@112 );
# LUG01 / 1
  @9252:AssemblyShapeJointItemRelationship( Relating=@9250, Related=@615 );
# WIRE01 / end b
# mechanical connections
@9260:AssemblyShapeJoint( ElementOf=@9002 );
  @9261:AssemblyShapeJointItemRelationship( Relating=@9260, Related=@316 );
# P-CONN01 / 1
  @9262:AssemblyShapeJointItemRelationship( Relating=@9260, Related=@317 );
# P-CONN01-01 / o
@9270:AssemblyShapeJoint( ElementOf=@9002 );
```

```
  @9271:AssemblyShapeJointItemRelationship( Relating=@9270, Related=@112 );
# P-CONN01 / 2
  @9272:AssemblyShapeJointItemRelationship( Relating=@9270, Related=@423 );
# P-CONN01-02 / o

sizeof(Part) = 7;
sizeof(PartVersion) = 7;
sizeof(PartView) = 6;
sizeof(WiringHarnessAssemblyDesign) = 1;
sizeof(NextAssemblyOccurrenceUsage) = 7;
sizeof(SingleOccurrence) = 5;
sizeof(WireOccurrence) = 1;
sizeof(CableOccurrence) = 1;

sizeof(PartTerminal) >= 4; # there might me interface terminals
sizeof(OccurrenceTerminal) >= 4; # there might me interface terminals
sizeof(PartContactFeature) = 3;
sizeof(OccurrenceContactFeature) = 4;
sizeof(WireColourBasedIdentificationCode) = 2;
sizeof(WireOccurrenceTerminal) = 2;
sizeof(CableOccurrenceTerminalLocationGroup) = 2;
sizeof(CableOccurrenceTerminal) = 4;
sizeof(AssemblyShapeJoint) = 7;
sizeof(AssemblyShapeJointItemRelationship) = 15;

);
```