This document is based on material provided in the document AP242_Electrical_Harness_Tutorial_XML.pdf

Version 2.4; 2021-12-02
(added ComposedGeometricModel and geometric assignment example)
Topics covered in Part 1

- High level XML structure, including “Unit of serialization” (Uos), cmn:BaseRootObject, cmn:DataContainer.
- Part with specific part categories for EWH such as “wire”, “connector”, “raw_material_by_length”
- Specific application domains for PartView and subtype WiringHarnessAssemblyDesign such as “complete_design” and “partial_design”, “wiring_harness_segment_topology”, ...
- SingleOccurrence and QuantifiedOccurrence (subtype WireOccurrence and CableOccurrence)
- ShapeElement with subtype OccurrenceShapeElement, PartShapeElement and ShapeFeatureDefinition
- The PartShapeElements HarnessSegment and HarnessNode for a WiringHarnessAssemblyDesign
- Topological model of a WiringHarnessAssemblyDesign, consisting of EdgeBoundedCurveWithLength or SubEdge with VertexPoints and underlying geometry BoundedCurveWithLength, Point or PointOnCurve (but not CartesianPoint)
Topics for this Part 2

- Different kinds of connectors
- Kinds of Terminals
- Highly modular & configurable connectors
- Single and multi contacts
- ShapeElement
- AssemblyShapeJoint - Basic
- Features, Contacts & Joints
- Transport Features and Terminals
- Wires, Cables & Conductors
- Contacting conductors at arbitrary locations
- Groups of ShapeElements – Multi Terminals
- ShapeFeatureDefinition and Elements
Different kind of connectors to support

- Terminal lugs with a single crimp or screw contact for typically one wire
- Audio connector with 2 (mono) or 3 (stereo) contacts
- Monolytical connectors such as a DSUB9 for welding
- Highly configurable connectors with very many variations such as the ARINC 600
Kinds of Terminals

- In many cases a single contact for a connector or a terminal lug has:
  - one “join terminal” to be connected permanently by e.g. a wire or cable,
  - and one “interface terminal” to be connected and disconnected at a next higher level
  - Note: terminal busbars and splices have typically only “join contacts”
- terminals are joined with other stuff by crimping, welding, screwing, ...
- Contacts are typically available as mating pairs (pin & socket)
- Connectors are typically available as mating pairs (plug & receptable)
Typical connector for aircrafts: ARINC 600

- A very modular and highly configurable family of connectors
- available from several vendors
Configurable Inserts for Size 1, slot A or B

Contact type single, different sizes or:

- C COAX
- T TWINAX OR TRIAX
- F FIBER
- Q QUADRAX
Contacts, Filler and Sealing Plugs that go into the Cavities of the Inserts

Crimped contacts are for joining with wires / cables.

There are also contacts with „PC Tail“ for direct welding with a PCB.

<table>
<thead>
<tr>
<th>CONTACT TYPE</th>
<th>SIZE</th>
<th>RECEPTACLE TYPE</th>
<th>PART NO.</th>
<th>Plug TYPE</th>
<th>PART NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal</td>
<td>22</td>
<td>SOCKET</td>
<td>AC-782222-301</td>
<td>Pin</td>
<td>AC-772222-301</td>
</tr>
<tr>
<td>Power</td>
<td>20</td>
<td>Pin</td>
<td>AC-772020-302</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>Pin</td>
<td>AC-771616-303</td>
<td>Socket</td>
<td>AC-781616-303</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>Pin</td>
<td>AC-771212-304</td>
<td>Socket</td>
<td>AC-781212-304</td>
</tr>
</tbody>
</table>

FILLER PLUGS

<table>
<thead>
<tr>
<th>Contact Cavity Size</th>
<th>Amphenol Part Numbers</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>AC-660022-701</td>
<td>Black</td>
</tr>
<tr>
<td>20</td>
<td>AC-660020-701</td>
<td>Red</td>
</tr>
<tr>
<td>16</td>
<td>AC-660016-701</td>
<td>Blue</td>
</tr>
<tr>
<td>16 Fiber</td>
<td>AC-660016F-701</td>
<td>Blue</td>
</tr>
<tr>
<td>12</td>
<td>AC-660012-701</td>
<td>Yellow</td>
</tr>
<tr>
<td>8 Coax</td>
<td>AC-660008-701</td>
<td>Red</td>
</tr>
<tr>
<td>5 Coax (Plug)</td>
<td>AC-660005-701</td>
<td>White</td>
</tr>
<tr>
<td>5 Coax (Recept.)</td>
<td>AC-660004-701</td>
<td>White</td>
</tr>
</tbody>
</table>

SEALING PLUGS

<table>
<thead>
<tr>
<th>Contact Cavity Size</th>
<th>Amphenol Part Numbers</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>AC-660022-801</td>
<td>Black</td>
</tr>
<tr>
<td>20</td>
<td>AC-660020-801</td>
<td>Red</td>
</tr>
<tr>
<td>16</td>
<td>AC-660016-801</td>
<td>Blue</td>
</tr>
<tr>
<td>12</td>
<td>AC-660012-801</td>
<td>Yellow</td>
</tr>
<tr>
<td>8 Coax</td>
<td>AC-660008-8701</td>
<td>Red</td>
</tr>
</tbody>
</table>
Coaxial, Triaxial, Quad and Fiber contacts

It is up to the use case, customer, and tool supplier to which level of detail AP242 is used. As a minimum all electrical connections have to be clearly identified.
A single ARINC 600 kit

- It is the job of the harness manufacturer to assemble all this
- Note that crimped contacts have be joined to the wires/cables before inserting them in the cavities of the Inserts.
Out of Scope:
Detailed Geometry, Dimensions and Tolerances

- The geometric shape (2D or 3D) with dimensions and tolerances is NOT the focus for EWH, but can be done by other capabilities of AP242 (see CAX-IF). For display purposes it is sufficient to reference into externally provided geometry.

- Focus for EWH is the identification of the features such as the slots A, B, C and which inserts go into them, and which contact into which cavity and how all is electrical connected by wires & cables.
Domain Model: **ShapeElement (from Part 1)**

- A *ShapeElement* is the identification of an element of the shape of a *ProductConfiguration*, *PartView*, *Occurrence*, *ShapeFeatureDefinition* or of another *ShapeElement* or ...
- some subtypes of *ShapeElement* might be defined by a *ShapeFeatureDefinition* or another *ShapeElement*
- A *XxxFeature* is a “definitional” *ShapeElement* that is visible/reachable from the outside
- There are many subtypes of *ShapeElement* including terminals (pins), joins, nets ...
AssemblyShapeJoint - Basic

- All assembly joints (mechanical, electrical, optical, piping ...) are based on general concepts on how ShapeElements of PartViews are replicated for Occurrences and then joined together.
- An OccurrenceShapeElement is typically defined by a PartShapeElement
- Two or more OccurrenceShapeElements are joined together in a AssemblyShapeJoint through AssemblyShapeJointItemRelationship
- A JointType indicates on how an AssemblyShapeJoint is realized
Features, Contacts and Joints

- **Part/OccurrenceShapeElements** represents any kind of ShapeElement for a Part/Occurrence
- **PartFeature/OccurrenceShapeFeatures** represents ShapeElements that are on the physical boundary of a Part/Occurrence
- **Part/OccurrenceContactFeature** represents ShapeElements that are intended to be connected with other contact features
- **AssemblyShapeJoint** joins two or more OccurrenceContactFeature (or OccurrenceShapeFeatue) by AssemblyShapeJointItemRelationships
Example: The two Inserts 1 and 2 that are joined into the Slots A and B of a Backshell

- SingleOccurrences of Parts are assembled together in a \textit{WiringHarnessAssemblyDesign} by \textit{NextAssemblyOccurrenceUsage}
- The PartContactFeatures of a PartView are replicated as OccurrenceContactFeature for the SingleOccurrences
- AssemblyShapeJoints for a \textit{WiringHarnessAssemblyDesign} join the OccurrenceContactFeatures by AssemblyShapeJointItemRelationships
Transport Features, Terminals & ConnectivityDef.

- Electrical energy or information, light, matter is transported by special **TransportFeatures** for **PartViews** and **Occurrences**. In the electrical world this is also called a **conductor**.

- A **TransportFeature** is accessed by **Terminals** (Part/Occurrence). **OccurrenceTerminals** are intended to be joined on an assembly level with other terminals by a special manufacturing method; e.g. by crimping or soldering.

- **PartConnectivityDefinition** is similar to AssemblyShapeJoint to state on a higher level which Occurrence- and PartTerminals are connected without providing details.
Example: simple connector

- a Part that is a connector
- ... with two PartTerminals identified as “signal” and “gnd”
- ... that are intended to be joint by crimping (“crimp_terminal”) on the next higher assembly level (“join terminal”)
- a SingleOccurrence of the connector with the ID “phone1”

... with two OccurrenceTerminals that are defined by the PartTerminals
Usage of a simple connector in an EWH assembly

• simple connectors are simple piece parts
• PartTerminals are reflected for the SingleOccurrences as OccurrenceTerminals
• OccurrenceTerminals can directly be connected in an EWH-assembly
Complex Connectors and their usage in an EWH-Assembly (1 of 5)

Problem:

- complex connectors may consists of many components such as a housing, a backshell, many contacts, inserts, strain relief, seals and more
- often these components are available as a **connector kit** and the harness manufacturer has to assemble them
- some CAx systems may represent an assembled connector as a part by its own. But physically these **connector assemblies** do not exist; they make no sense!
- the connector components are only assembled together **during** the assembly process of the whole EWH (not before!). E.g. a contact is first crimped together with a wire before inserting it into a cavity of the connector housing or insert.
- During the assembly process **AssemblyShapeJoints** are used in two different ways:
  - electrical joints between the contact terminals and the wire/cable terminals. Knowledge of these terminals is **essential** for the use of EWH
  - mechanical joints between mating features. e.g. between the outer feature of a connector contact and the inner feature of a cavity into which the connector contact is inserted
- mechanical contact features are essential for the full description of an EWH-assembly, but CAx-systems may not have this information
- However mechanical CAD systems used to provide assembly information with transformation. Together with the 3D model a CAM system may calculate the mechanical contact zones between the assembly components (not numerical stable, error prone)
AP242 provides the following alternative solutions so that the needed AssemblyShapeJoints can refer to the right occurrence features / terminals:

1) Hierarchical assembly with SpecifiedOccurrences
   - use of dummy sub-assembly parts
   - occurrences of lower level assemblies are reflected to higher assembly levels by SpecifiedOccurrence
   - terminals of lower level occurrences are reflected as terminals of SpecifiedOccurrences
   - AssemblyShapeJoints are joining the terminals of SpecifiedOccurrences with others
   - mechanical features can be skipped; only terminals are essential

2) Hierarchical assembly with reflecting lower level terminals to a higher level (New)
   - use of dummy sub-assembly parts
   - no need for SpecifiedOccurrence
   - terminals of lower level occurrences are reflected as terminals of the next higher assembly part (can be used recursively)
   - AssemblyShapeJoints are joining the terminals of sub-assembly occurrences with others
   - mechanical features can be skipped; only terminals are essential

3) Flat assembly
   - no need for dummy sub-assembly parts
   - no need for SpecifiedOccurrence
   - mechanical features are essential to know which contact is in which cavity or which insert is in which slot
   - AssemblyShapeJoints are joining the terminal and features of the occurrences

Flat assembly is most close to the reality and would be the basis for a later Process Plan extension
Complex Connectors and their usage in an EWH-Assembly (3 of 5)

to alternative 1): Hierarchical assembly with SpecifiedOccurrences
see also: HarnessExample_Hierarchical.xml
Complex Connectors and their usage in an EWH-Assembly (4 of 5)

to alternative 2): Hierarchical assembly with reflecting lower level terminals to a higher level (New)
Complex Connectors and their usage in an EWH-Assembly (5 of 5)

to alternative 3): Flat assembly
see also: HarnessExample_Flat.xml
Wires, Cables & Conductors (current, ed2)

- A wire consists of a single conductor/ (maybe of several strands) and typically an isolation.
- The isolation has to be somehow removed before the conductor/TransportFeature can be accessed by a Wire/CableOccurrenceTerminal (details not covered so far).
- A cable consists of several “wires”(conductors), and so it is essential to identify the wire (e.g. color).
- CableOccurrenceTerminals that are located close together (e.g. same end) are grouped together.
Wires, Cables & Conductors (new, upcoming)

- Problem in ed2: In very most cases the colour code for a wire is defined for the raw material / part, and not for a particular occurrence.
- Changes: new WirePartIdentification
  old WireIdentification renamed to WireOccurrenceIdentification
- Typical usage: A WireOccurrenceIdentification refers to a WirePartIdentification as Definition. A WireOccurrenceIdentification has no colour code, but inherit this from WirePartIdentification
Often conductors (e.g. Wires, Cables, Busbars) are connected only at the ends but sometimes at any location.
Example: WireOccurrenceTerminals

```
<Part uid="_101000">
  ...
  <PartTypes>
    <PartCategoryEnum>wire</PartCategoryEnum>
    <PartCategoryEnum>raw_material_by_length</PartCategoryEnum>
  </PartTypes>
  <Versions>
    <PartVersion uid="_101001">
      <Id id="Version 1"/>
      <Views>
        <PartView uid="_101002">
          <DefiningGeometry uidRef="_104890"/>
          <InitialContext uidRef="_100102"/>
          <Occurrence xsi:type="n0:WireOccurrence" uid="_201004">
            <Id id="wire1"/>
            <ShapeElement xsi:type="n0:WireOccurrenceTerminal" uid="_201006">
              <Name> <CharacterString>end a</CharacterString> </Name>
              <AssociatedTransportFeature uidRef="_201008"/>
            </ShapeElement>
            <ShapeElement xsi:type="n0:WireOccurrenceTerminal" uid="_201007">
              <Name> <CharacterString>end b</CharacterString> </Name>
              <AssociatedTransportFeature uidRef="_201008"/>
            </ShapeElement>
            <ShapeElement xsi:type="n0:WireIdentification" uid="_201008">
              <Code uidRef="_100201"/>
            </ShapeElement>
            <Quantity xsi:type="n0:NumericalValue" uid="_201010">
              <Unit uidRef="_100301"/>
              <ValueComponent>1.75</ValueComponent>
            </Quantity>
          </Occurrence>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

a Part with the categories “wire” and “raw_material_by_length”
a WireOccurrence of a particular length
with a single WireIdentification to identify the conductor of the wire
and two WireOccurrenceTerminals at the ends of the wire, “end_a” and “end_b”, both referencing the WireIdentification
additional WireOccurrenceTerminals between the end of the wire can be defined as needed
Groups of ShapeElements - Multi-Terminals

- In most cases a PartShapeElement is an ElementOf a PartView, and an OccurrenceShapeElement is an ElementOf an Occurrence … but not always

- We saw that a CableOccurrenceTerminal is an ElementOf a CableOccurrenceTerminalLocationGroup

- In general all ShapeElements can also be elements of another ShapeElement. This allows flat and hierarchical grouping

- This capability might also be useful to represent e.g. the connectivity of coax, triax, quad connectors, or e.g. Ethernet CAT-5 cable as there are standards on how to connect the detailed conductors. (So far no example had been worked out for this)
Example: CableOccurrenceTerminals (1)

- two properties of type “wire colour-based identification code” with the values “white” and “red”

- a part with the categories “cable” and “raw_material_by_length”

- the PartView has two conductors, indicated by WirePartIdentification that references the “wire colour-based identification code” for “white” and “red”
Example: CableOccurrenceTerminals (2)

- **CableOccurrence**
  - has a particular length
  - two **WireOccurrenceIdentifications**
    - that reference corresponding **PartOccurrenceIdentifications**
  - **CableOccurrenceTerminalLocationGroups**, one for each end (a and b)
  - each has two **CableOccurrenceTerminals**
    that corresponds to the two **WireOccurrenceIdentifications**
Example: AssemblyShapeJoint

- a Part that is an assembly, here WiringHarnessAssemblyDesign
- NextAssemblyOccurrenceUsage bring in Occurrences “phone1” and “cable3”
- a single AssemblyShapeJoint by crimping (“crimped_connection”)
- the joint is established by AssemblyShapeJointItemRelationship between cable3, wire A at end_b with phone1, terminal “signal”
a *ShapeFeatureDefinition* allows for the identification of an independent feature with/without a defining *GeometricModel*

a *ShapeFeatureDefinition* can be used as the definition for a *PartFeature*. This allows to define the common shape of a feature once, and then use it for many different *PartViews*

a *ContactFeatureDefinition* is a kind of *ShapeFeatureDefinition* that is intended to be contacted by other corresponding *ContactFeatureDefinitions*. Mating pairs can be identified by *ContactFeatureDefinitionFitRelationship*

a *ShapeFeatureDefinitionElement* identifies a part of a *ShapeFeatureDefinition*

with the subtype *ShapeFeatureDefinitionOccurrenceElement* it is possible to compose complex *ShapeFeatureDefinitions* from simpler ones.

... (more to follow)
Example: Deutsch IMC Series connector (1)

- an *Organization* to identify the company “Deutsch”
- two *ContactFeatureDefinitions* for size 20 *cavity_profile* and *contact_profile*
- the *ContactFeatureDefinitions* are related by a *ContactFeatureDefinitionFitRelationship*
- *this information is not essential for exchange of a EWH design, but needed for re-use of the library info*

```xml
<Organization uid="_100"> <!-- Deutsch company -->
  <Id id="http://www.deutsch.net"/> ...
</Organization>

<ShapeFeatureDefinition xsi:type="n0:ContactFeatureDefinition" uid="_101">
  <Id> <Identifier uid="_201" id="IMC Series Size 20 cavity" idContextRef="_100"/> </Id> ...
  <ShapeFeatureType>cavity_profile</ShapeFeatureType>
</ShapeFeatureDefinition>

<ShapeFeatureDefinition xsi:type="n0:ContactFeatureDefinition" uid="_102">
  <Id> <Identifier uid="_201" id="IMC Series Size 20 pin" idContextRef="_100"/> </Id> ...
  <ShapeFeatureType>contact_profile</ShapeFeatureType>
  <ShapeFeatureDefinitionRelationship xsi:type="n0:ContactFeatureDefinitionFitRelationship" uid="_103">
    <Related uidRef="_101"/>
  </ShapeFeatureDefinitionRelationship>
</ShapeFeatureDefinition>
```
Example: Deutsch IMC Series connector (2)

- A Part that is a connector with two PartContactFeature “1” and “2” that are defined by a ContactFeatureDefinition that is a cavity_profile.

- A Part that is a connector_contact with a PartContactFeature “o” that is defined by a ContactFeatureDefinition that is a contact_profile.
GeometryToTopologyModelAssociation & TopologyToGeometryModelAssociation

- the ...Associations allow to map to/from a EdgeBasedTopologicalRepresentationWithLengthConstraint with an (External)GeometricModel
- the (External)GeometricModels might be 2D or 3D
- the GeometryToTopologyModelAssociation is typically a single item association
- the TopologyToGeometryModelAssociation is typically a multi-item association; the list of Origin elements must match to the list of Target elements
Use of the Geometry/Topology Associations for assembly cases

- The attribute Placement of NextAssemblyOccurrenceUsage <= AssemblyOccurrenceRelationship can refer to several kinds of RepresentationRelationship or as a simplified solution directly to CartesianTransformation.

- For the purpose of the CAX-IF most often CartesianTransformation or GeometricRepresentationRelationshipWithPlacementTransformation is used/recommended.

- For the purpose of EWH the relationship GeometryToTopologyModelAssociations and TopologyToGeometryModelAssociation are needed additionally:
  - GeometryToTopologyModelAssociations to relate the 2D/3D models of assembly components to the EWH topology model.
  - TopologyToGeometryModelAssociation to relate a complete WiringHarnessAssemblyDesign as a SingleOccurrence into the GeometricModel of a higher assembly.
A simple Assembly of rigid Parts with Transformation. What to do with the flexible Cable?
Associations of geometric Models to the topological Harness Model

```
WHAD = WiringHarnessAssemblyDesign
NAOU = NextAssemblyOccurrenceUsage
G2TMAso = GeometryToTopologyModelAssociation
EBTRepWLC = EdgeBasedTopologicalRepresentationWithLengthConstraint
```

```
Geo.Coor.Space 2D

Geom.Model

AxisPlacement

Items 1..*

Part

PartVersion

CableOccurrence

Geo.Coor.Space 3D

Geom.Model

AxisPlacement

Items 1..*

Part

PartVersion

SingleOccurrence

Target

Relating

G2TMAso

NAOU

Related

Origin

Placement

the wiring harness

Related

Target

EBTRepWLC

Path

EdgeList 1..*

EdgeBCWL

VertexPoint

Topology 0..1

PartVersion

Part
```
ExternalGeometricModel & External(Element)References

- use ExternalGeometricModel.ExternalItem if there are several Models/Representations in the external files
- use ExternalGeometricModel.Items to refer ExternalRepresentationItems
- for DigitalFile use FileLocations attribute (ed2); don’t use Locations attribute (ed1)
- for external p21 files use ExternalEntityInstance to make clear what is the meaning of the Id
- if available use for the ExternalEntityInstance.Id the external anchor name instead of the instance ID (e.g. #1234)
- NextInstanceForward and NextInstanceInverse allow to follow a path of instance in a p21 file (for later tests)
A part with an external geometric model and 2 features/terminals.
Example: Terminal Lug with external Geometry (2/3)

An ExternalGeometricModel (here subtype for an ABRP) that references into items of a p21 file.

```xml
<!-- Geometry for terminal lug -->
<RepresentationContext xsi:type="n0:GeometricCoordinateSpace" uid="_103091">
  <Id id="/NULL"/>
  <Representations>
    <Representation xsi:type="n0:ExternalAdvancedBrepShapeRepresentation" uid="_103090">
      <Id id="c-51864-1-af-3d.stp"/>
      <Items>
        <RepresentationItem uidRef="_103092"/>
        <RepresentationItem uidRef="_103094"/>
        <RepresentationItem uidRef="_103096"/>
      </Items>
      <ExternalFile uidRef="_103080"/>
      <ExternalItem xsi:type="n0:ExternalEntityInstance" uid="_103097">
        <Id id="#1023"/>
      </ExternalItem>
    </Representation>
  </Representations>
</RepresentationContext>

<!-- terminal lug geometric representation -->
<RepresentationItem xsi:type="n0:ExternalRepresentationItem" uid="_103092">
  <External xsi:type="n0:ExternalEntityInstance" uid="_103093">
    <Id id="#521"/>
  </External>
</RepresentationItem>

<RepresentationItem xsi:type="n0:ExternalRepresentationItem" uid="_103094">
  <External xsi:type="n0:ExternalEntityInstance" uid="_103095">
    <Id id="#940"/>
  </External>
</RepresentationItem>

<RepresentationItem xsi:type="n0:AxisPlacement" uid="_103096">
  <Position>0.0 0.0 0.0</Position>
</RepresentationItem>

<DimensionCount>3</DimensionCount>
</RepresentationContext>
```
Example: Terminal Lug with external Geometry (3/3)

File: c-51864-1-af-3d.stp

ADVANCED_BREP SHAPE_REPRESENTATION #1023

Advanced_face #940          Advanced_face #521
Example of a p21 ed3 file with Anchors

- Use of anchors in a p21 file requires implementation level 4
  see FILE_DESCRIPTION below
- Recommendations:
  - use anchors only for entity instances
  - continue to use "syntactical conformance class" 1 for "internal mapping"
  - for the anchor names use the name given in the source system
    (in CATIA v5 called “publication, in NX called “port”)
    Ideally a source system would use UUIDs to achieve global unique and persistent anchor names
  - centre line curves for harness segments best contained in a
    GEOMETRICALLY_BOUNDED_WIREFRAME SHAPE REPRESENTATION

ISO-10303-21;
HEADER;
FILE_DESCRIPTION(...,'4;1');
FILE_NAME('star1.p21', ...);
FILE_SCHEMA(('CONFIG_CONTROL_DESIGN'));
ENDSEC;

ANCHOR;
<placement1> = #1011;
<placement2> = #1012;
<curve1> = #1021;
<2871d0c8-9f87-4349-aab0-7832e53fa25a> = #1022;         <== example of a UUID
ENDSEC;

DATA;
...
#1000=(GEOMETRIC_REPRESENTATION_CONTEXT(3)...)  
#1001=GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION("",(#1011,#1012,#1020),#1000);  
#1011=AXIS2_PLACEMENT_3D(...);  
#1012=AXIS2_PLACEMENT_3D(...);  
#1020=GEOMETRIC_CURVE_SET("",(#1021,#1022, ...));  
#1021=B_SPLINE_CURVE_WITH_KNOTS(...);  
#1022=B_SPLINE_CURVE_WITH_KNOTS(...);  
...
ENDSEC;
END-ISO-10303-21;
Use of a ComposedGeometricModel (1/6)

- The files HarnessExample_Hierarchical.xml and HarnessExample_HierarchicalReflect.xml reference into the STEP p21 file **H1_abrep.stp**
- The p21 file contains a main assembly part with the name **Harness** with an assembly component for the dummy part **Product5-Multi-branchable2**
- Here we show how it is possible to reference into centre lines for the harness segments using a ComposedGeometricModel for the **WiringHarnessAssemblyDesign**
Use of a ComposedGeometricModel (2/6)

- The DefiningGeometry of a WiringHarnessAssemblyDesign might be a ComposedGeometricModel that composes the simplified geometry of the harness segments and the geometry of the connectors.
- The geometric models of connectors are typically defined in their own GeometricCoordinateSpace and brought into the ComposedGeometricModel by a GeometricRepresentationRelationshipWithPlacementTransformation.
- The ExternalGeometricModel of the harness segments are typically defined in the same GeometricCoordinateSpace and thus brought into the ComposedGeometricModel by a GeometricRepresentationRelationshipWithSameCoordinateSpace.
- The Target of the placement transformations for the connectors has to fit with the geometry of the harness segments and is therefore defined in the ExternalGeometricModel of the harness segment or of another higher one.

```xml
<RepresentationContext xsi:type="n0:GeometricCoordinateSpace" uid="_314091"/>
<Representations>
  <Representation xsi:type="n0:ComposedGeometricModel" uid="_314090"/>
  <Id id="xxx"/>
  <Items>
    <RepresentationItem uidRef="_314096"/>
  </Items>
  <!--Transformation of lug1-->
  <RepresentationRelationship xsi:type="n0:GeometricRepresentationRelationshipWithPlacementTransformation" uid="_314210"/>
    <Definitional>true</Definitional>
    <Related uidRef="_103090"/>
    <Origin uidRef="_103096"/>
    <Target uidRef="_314096"/>
  </RepresentationRelationship>
  <!--include external representation-->
  <RepresentationRelationship xsi:type="n0:GeometricRepresentationRelationshipWithSameCoordinateSpace" uid="_314220"/>
    <Definitional>true</Definitional>
    <Related uidRef="_314100"/>
  </RepresentationRelationship>
</Representation>
<Representation xsi:type="n0:ExternalGeometricModel" uid="_314100"/>
...
</Representations>
<Items>...</Items>
<DimensionCount>3</DimensionCount>
</RepresentationContext>
Within the `ExternalGeometricModel` (that is part of the `ComposedGeometricModel`) we find a `TopologyToGeometryModelAssociation` that is associating items from `EdgeBasedTopologicalRepresentationWithLengthConstraint` to items of the `ExternalGeometricModel`.

Note that the number and order of the items under `Origin` and `Target` fit as the association is pairwise (1st with 1st, 2nd with 2nd, ...).

`ExternalFile` refers to the external p21 file that contains the geometry.

`ExternalItem` / `ExternalEntityInstance` refers to the SHAPE_REPRESENTATION instance in that file.
Use of a ComposedGeometricModel (4/6)

- The **WiringHarnessAssemblyDesign** is:
  - referring the **ComposedGeometricModel** as **DefiningGeometry**
  - referring the **EdgeBasedTopologicalRepresentationWithLengthConstraint** as **Topology**
Use of a ComposedGeometricModel (5/6)

Depending on how the p21 file is structured, it may be needed to traverse through a p21 file to find the right entity instance. This can be done with NextInstanceForward and NextInstanceInverse. Example:

1) we start from instance #15, a SHAPE_REPRESENTATION and follow inverse the attribute rep_2 of
2) instance #19, a REPRESENTATION_RELATIONSHIP_WITH_TRANSFORMATION where the transformation has to be taken care! From there we follow attribute rep_1
3) and reach instance #23, a GEOMETRIC_REPRESENTATION_CONTEXT and follow inverse the attribute context_of_items of
4) instance #96, a GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION and follow the attribute items
5) we reach instance #97, a GEOMETRIC_CURVE set and follow the attribute elements
6) and finally the ExternalEntityInstance #141, a COMPOSITE_CURVE is reached
Use of a ComposedGeometricModel (6/6)

- We can also reference an **AxisPlacement** that is defined in a p21 by, e.g. an AXIS2_PLACEMENT_3D.
- This is done by populating the attribute **External**.
- When the attribute **External** is used the **AxisPlacement** attributes **Axis**, **Position** and **RefDirection** must not be used as the placement information is taken from the external p21 file only.

```xml
...  
<RepresentationItem xsi:type="n0:AxisPlacement" uid="_314096">  
  <External uid="_314096_1">  
    | <Id id="#5143"/> <!-- #5143=AXIS2_PLACEMENT_3D(' ',#5146,#5150,#5149); -->  
  </External>  
</RepresentationItem>  
</Items>  
<DimensionCount>3</DimensionCount>  
</RepresentationContext>  
```