AP242 ed2
Electrical Wiring Harness (EWH)
Tutorial – Slides
part 2

Lothar Klein, LKSoftWare GmbH

Sophie HERAIL, CIMPA S.A.S.

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 catagories 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

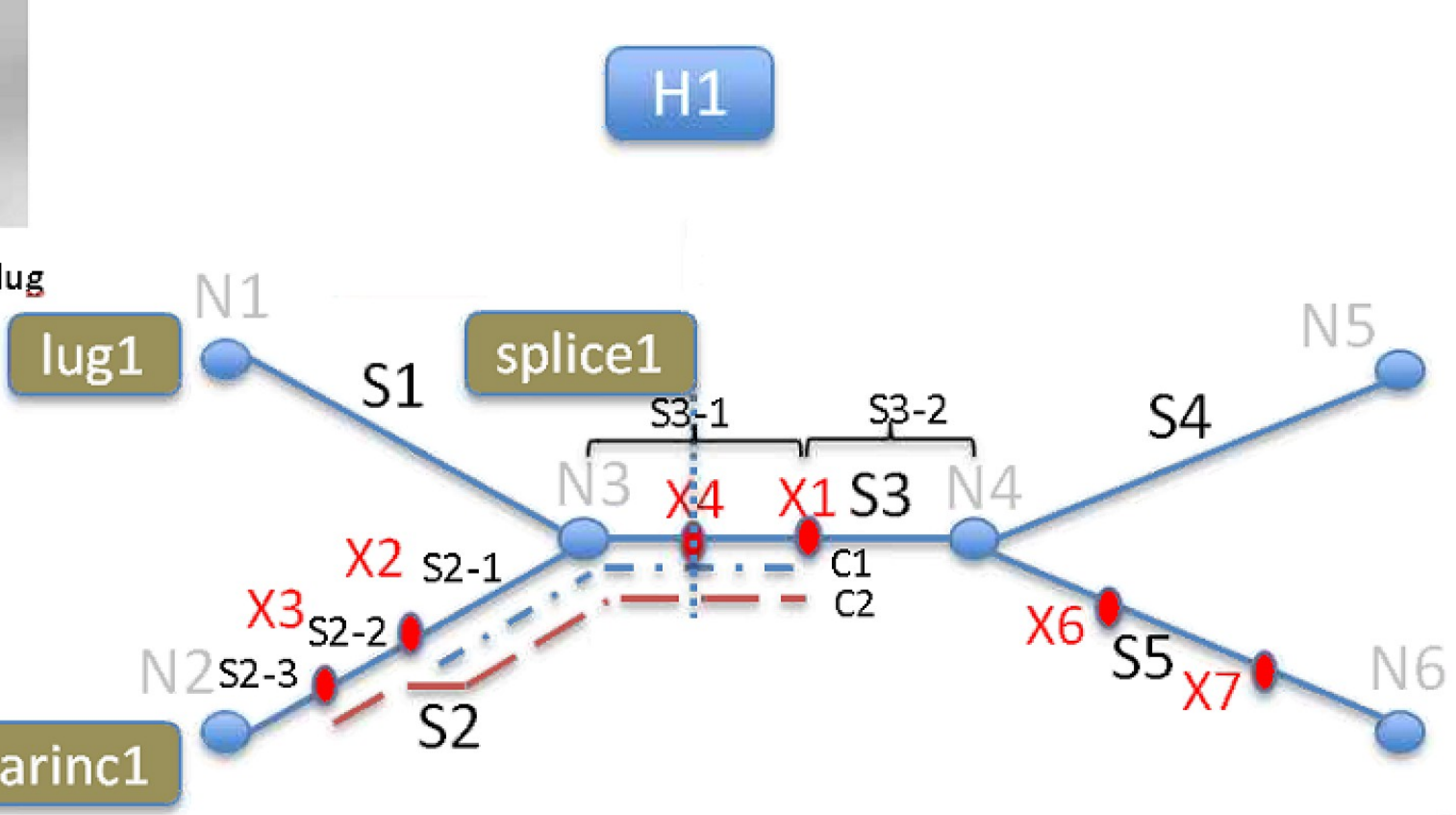
- Terminal lugs with a single crimp or screw contact for typically one wire
- Audio connector with 2 (mono) or 3 (stereo) contacts
- monolytical connectors such as a DSUB9 for welding
- highly configurable connectors with very many variations such as the ARINC 600

Terminal lug

Connector with inserts + backshell with EMI (ARINC600)

DSUB9

Phone connector (6,35mm)



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

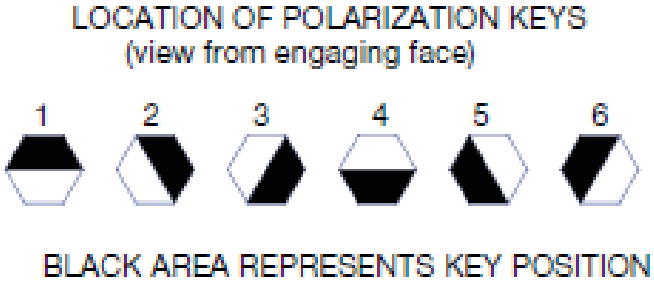
- A very modular and highly configurable family of connectors
- available from several vendors

Plug

Receptable

Inserts A, B, C

Polarisation keys



Shell Size 3

Shell Size 2

Amphenol CANADA
MILITARY & AEROSPACE

Shell Size 1

Configurable Inserts for Size 1, slot A or B

Size 1 Signal (A/B)

(0)	BLANK
QTY	Size

Q

(4)	4Q4
QTY	Size
4	8 Q

C T

(4)	4C4 4T4
QTY	Size
4	8 C, T

(8)	8
QTY	Size
8	12

(20)	20
QTY	Size
20	16

C T

(30)	30C2 30T2
QTY	Size
28	22
2	8 C, T

(32)	32
QTY	Size
8	16
24	20

(42)	42
QTY	Size
42	20

(60)	60
QTY	Size
60	22

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

CONTACT TYPE	SIZE	RECEPTACLE		Plug	
		TYPE	PART NO.	TYPE	PART NO.
Signal	22	SOCKET	AC-782222-301	Pin	AC-772222-301
Power	20	Pin	AC-772020-302	Socket	AC-782020-302
	16		AC-771616-303		AC-781616-303
	12		AC-771212-304		AC-781212-304

Crimped contacts are for joining with wires / cables.

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

FILLER PLUGS

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

SEALING PLUGS

Contact Cavity Size	Amphenol Part Numbers	Color
22	AC-660022-801	Black
20	AC-660020-801	Red
16	AC-660016-801	Blue
12	AC-660012-801	Yellow
8 Coax	AC-660008-8701	Red

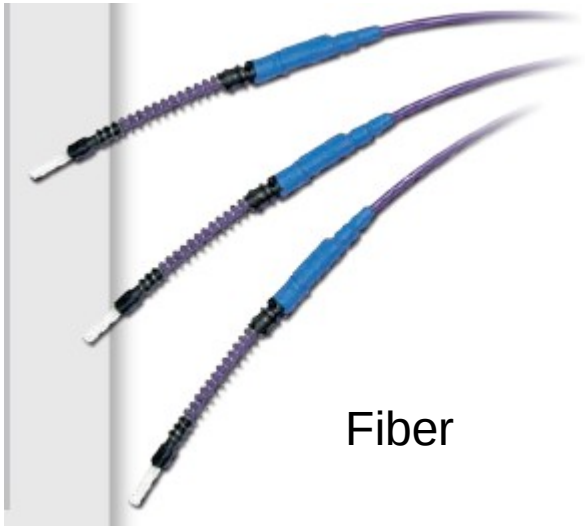
Coaxial, Triaxial, Quad and Fiber contacts

Coax

CRIMPED
PIN

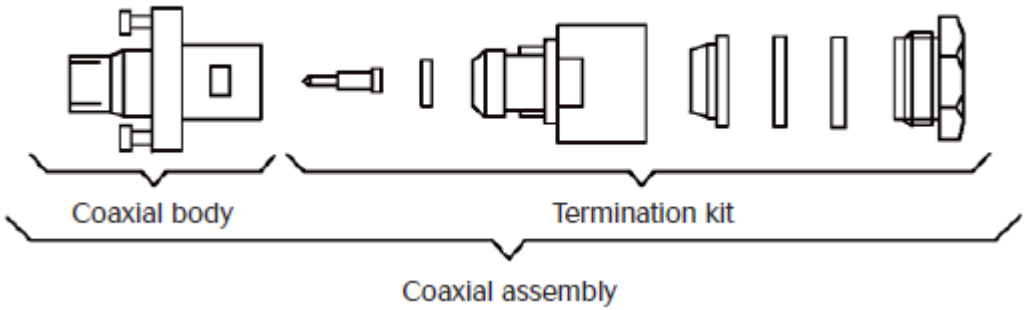
Quad

CRIMPED
SOCKET



Fiber

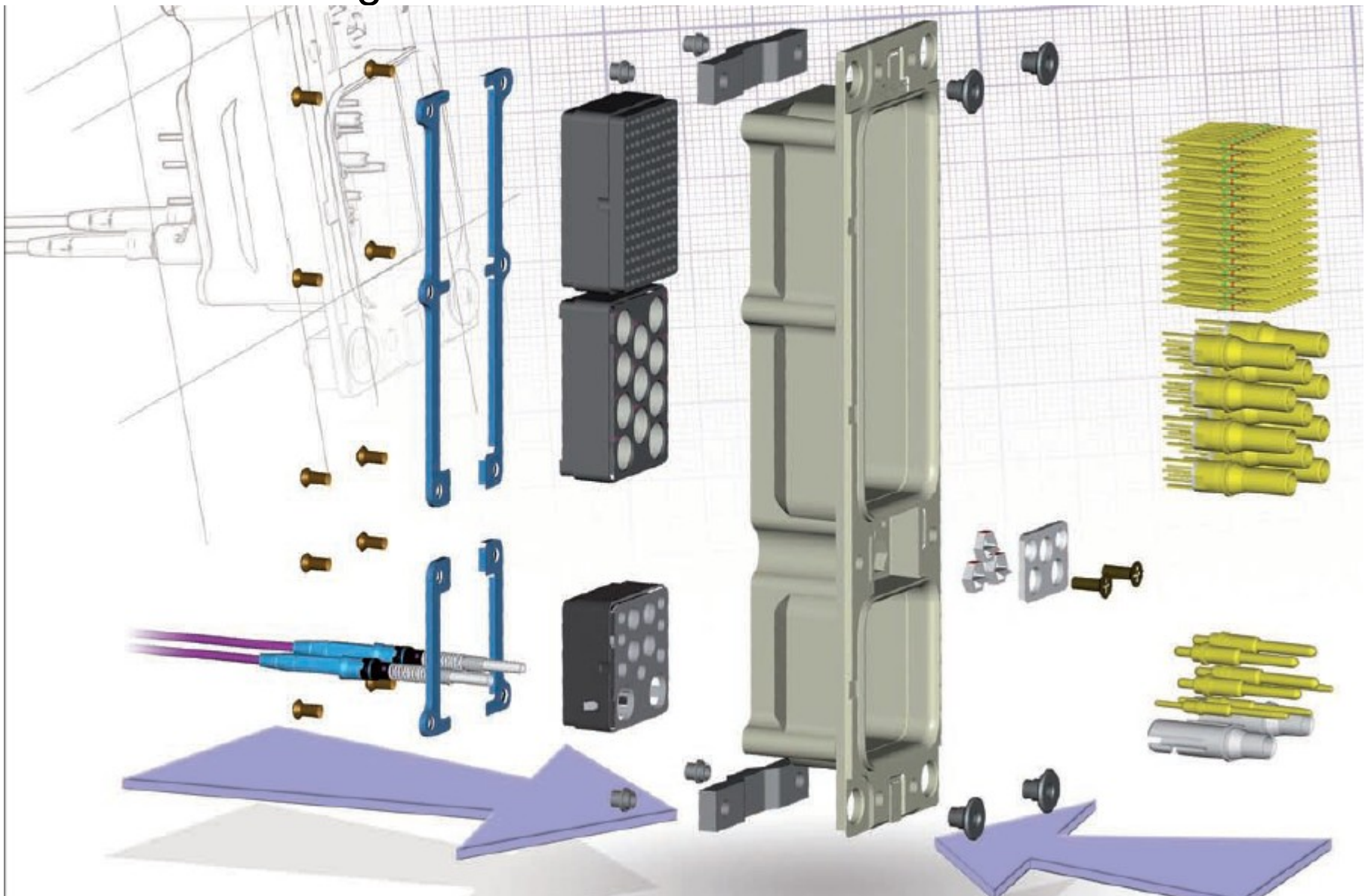
Coaxial assembly example



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.

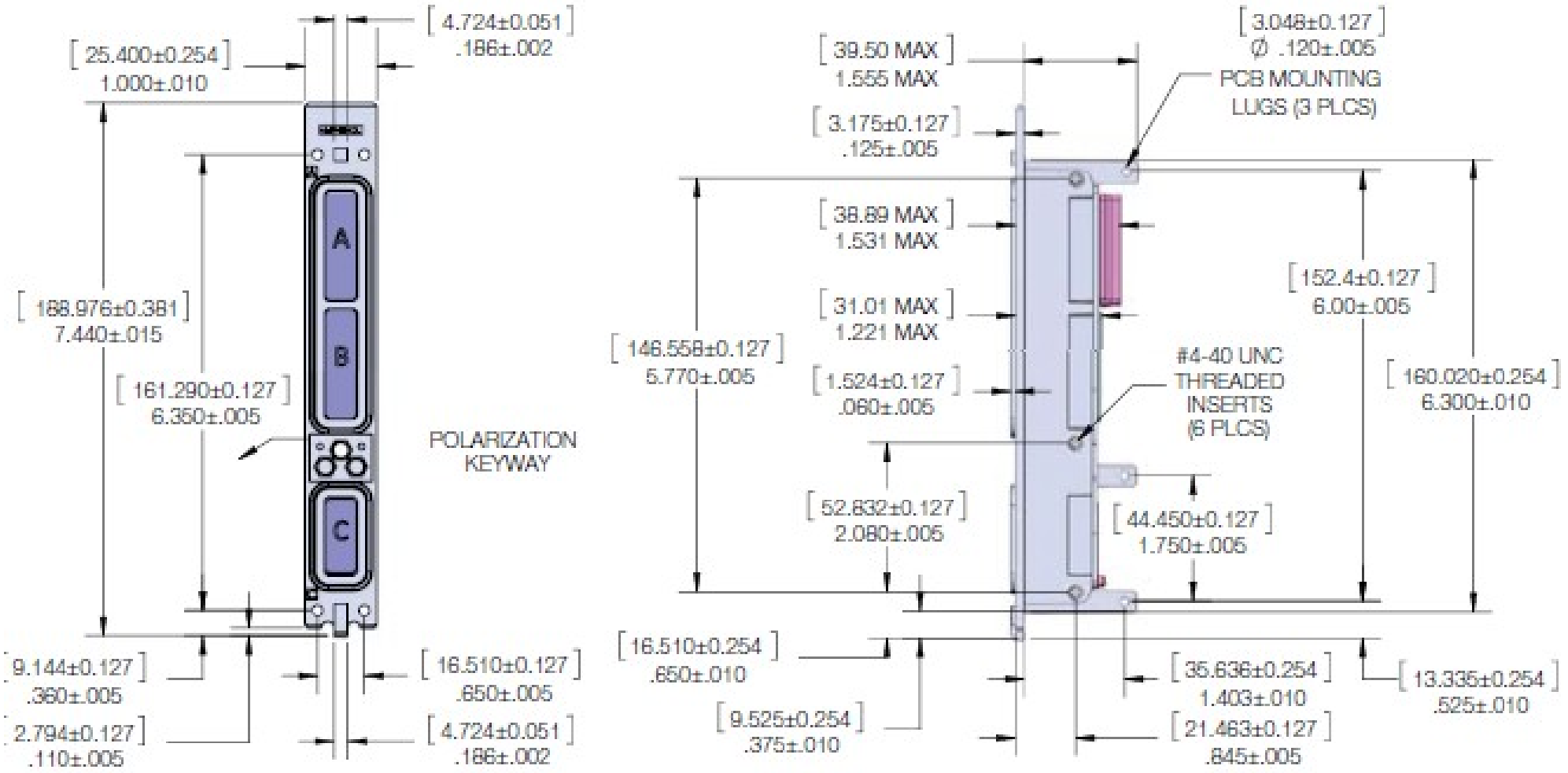
A single ARINC 600 kit

- 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.



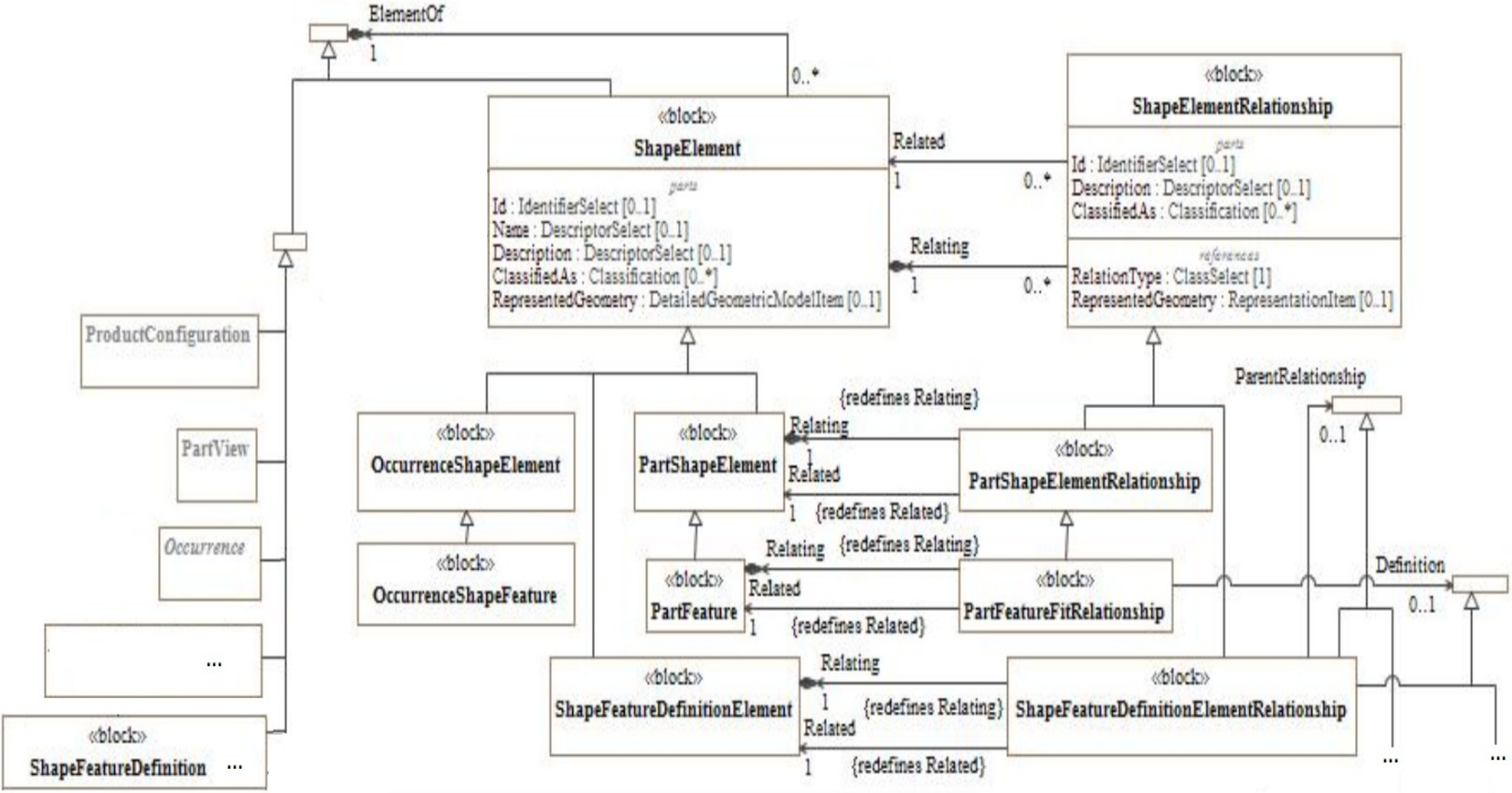
Out of Scope:
Detailed Geometry, Dimensions and Tolerances

- 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



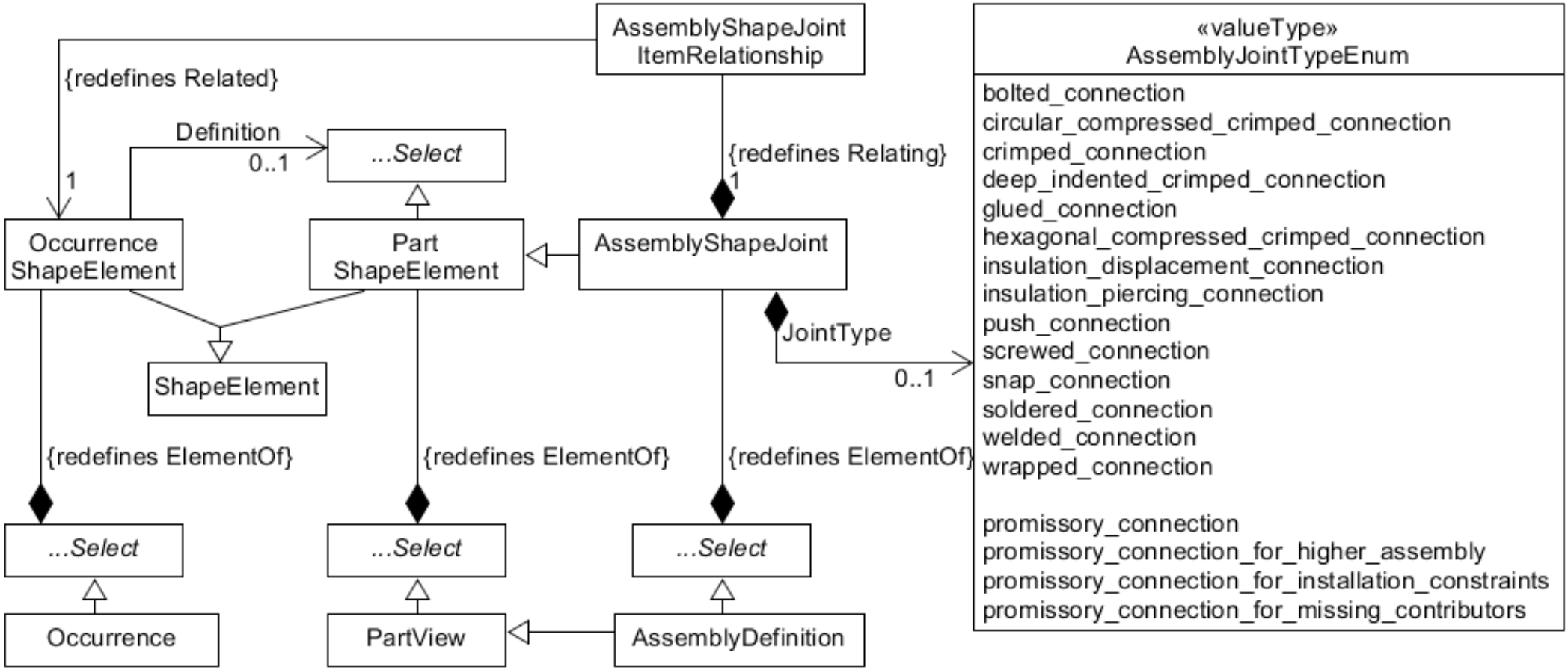
Domain Model: *ShapeElement (from Part 1)*

- 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 ...



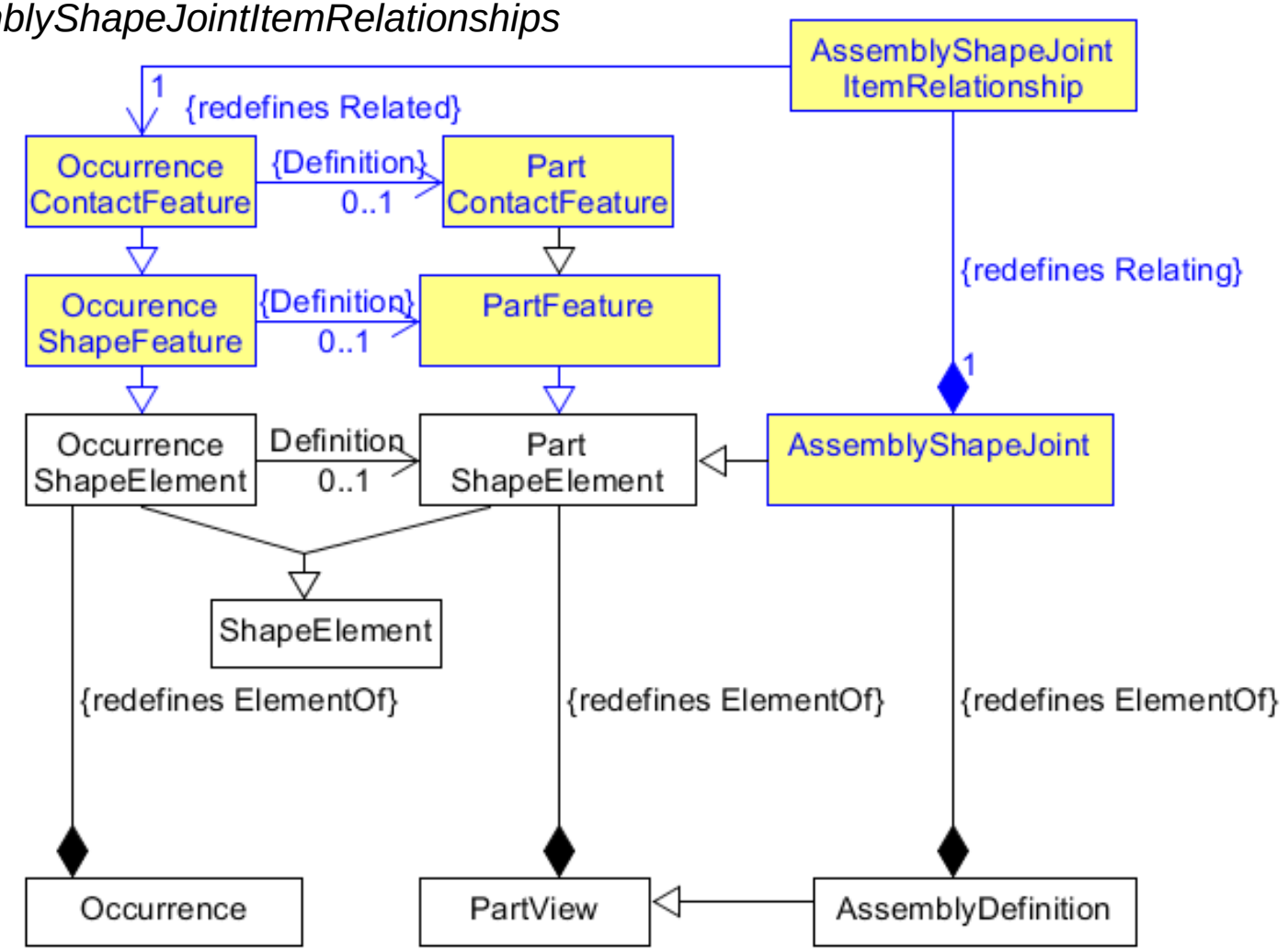
AssemblyShapeJoint - Basic

- 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



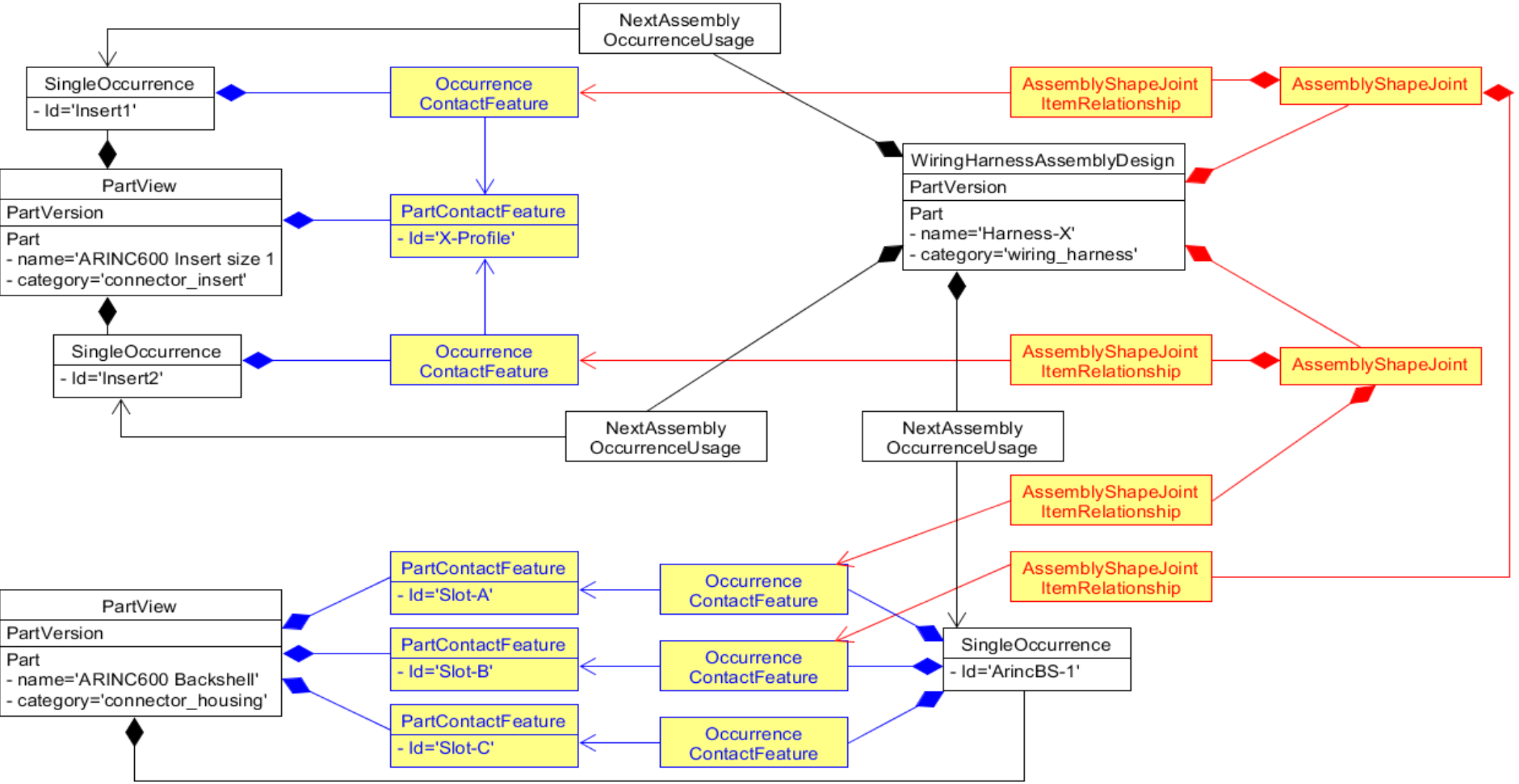
Features, Contacts and Joints

- *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*



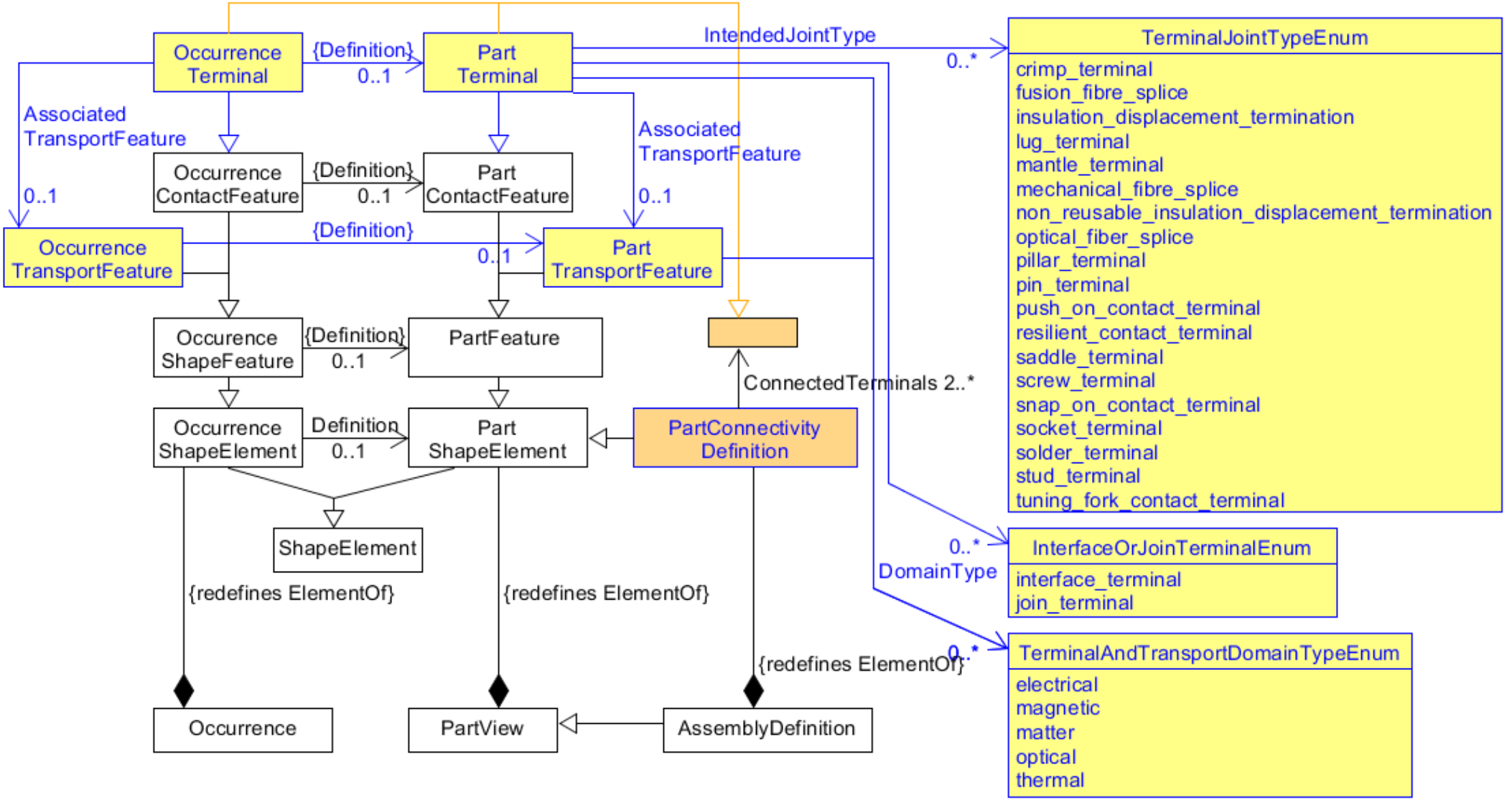
Example: The two Inserts 1 and 2 that are joined into the Slots A and B of a Backshell

- *SingleOccurrences* of Parts are assembled together in a *WiringHarnessAssemblyDesign* by *NextAssemblyOccurrenceUsage*
- The *PartContactFeatures* of a *PartView* are replicated as *OccurrenceContactFeature* for the *SingleOccurrences*
- *AssemblyShapeJoints* for a *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
- **PartConnecttivityDefinition** is similar to AssemblyShapeJoint to state on a higher level which Occurrence- and PartTerminals are connected without providing details



Example: simple connector

```
<Part uid="_117000">
  ...
  <PartTypes>
    <PartCategoryEnum>connector</PartCategoryEnum>
  </PartTypes>
  <Versions>
    <PartVersion uid="_117001">
      <Id id="Version 1"/>
      <Views>
        <PartView uid="_117002">
          <InitialContext uidRef="_100102"/>
          <Occurrence xsi:type="n0:SingleOccurrence" uid="_217100">
            <Id id="phone1"/>
            <ShapeElement xsi:type="n0:OccurrenceTerminal" uid="_217102">
              <Definition uidRef="_117004"/>
            </ShapeElement>
            <ShapeElement xsi:type="n0:OccurrenceTerminal" uid="_217104">
              <Definition uidRef="_117009"/>
            </ShapeElement>
          </Occurrence>
          <ShapeElement xsi:type="n0:PartTerminal" uid="_117004">
            <Id id="signal"/>
            <IntendedJointType>
              <TerminalJointTypeEnum>crimp_terminal</TerminalJointTypeEnum>
            </IntendedJointType>
            <InterfaceOrJoinTerminal>join_terminal</InterfaceOrJoinTerminal>
          </ShapeElement>
          <ShapeElement xsi:type="n0:PartTerminal" uid="_117009">
            <Id id="gnd"/>
            <IntendedJointType>
              <TerminalJointTypeEnum>crimp_terminal</TerminalJointTypeEnum>
            </IntendedJointType>
            <InterfaceOrJoinTerminal>join_terminal</InterfaceOrJoinTerminal>
          </ShapeElement>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

- 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)

Complex Connectors and their usage in an EWH-Assembly (2 of 5)

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

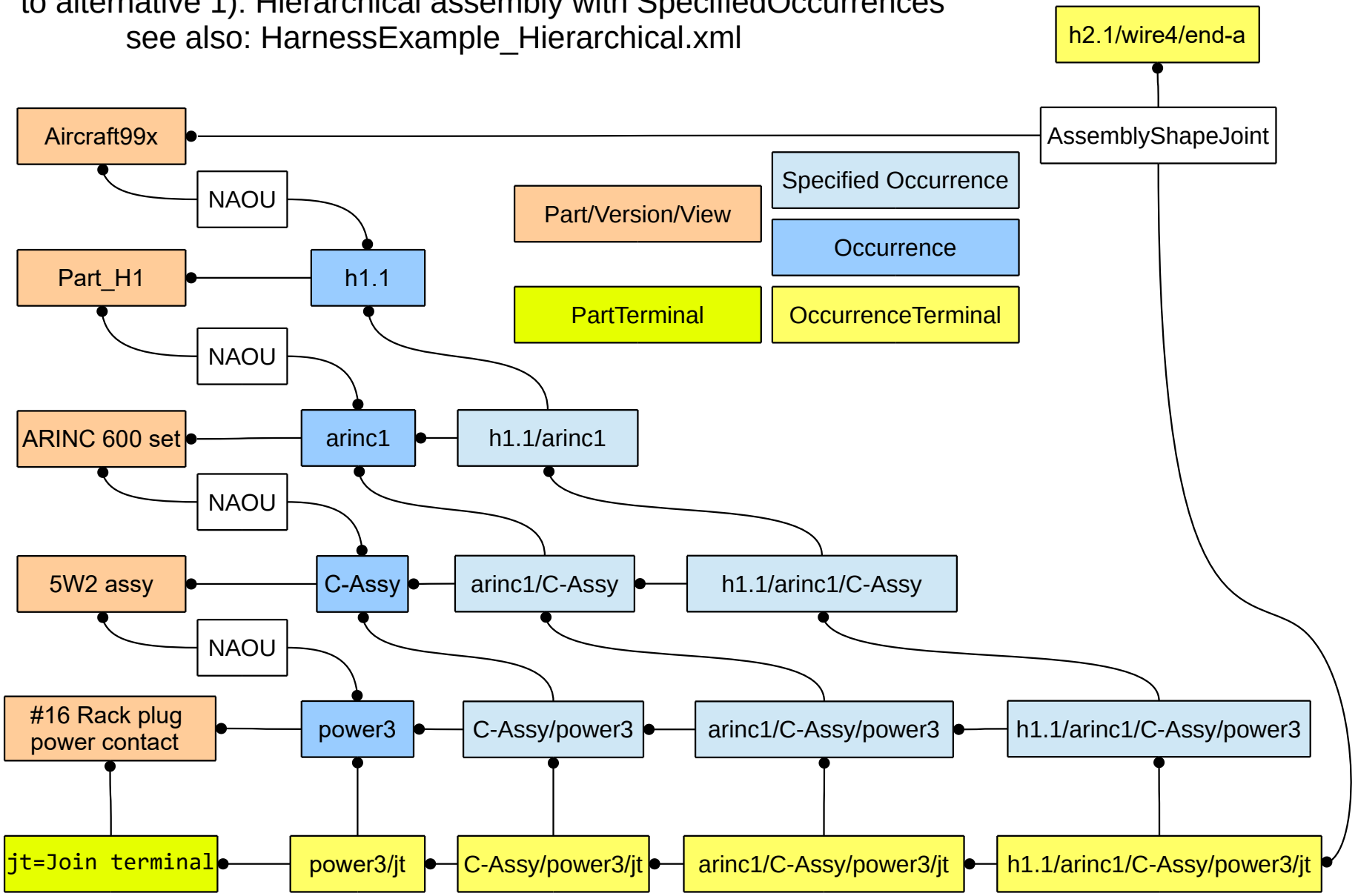
3) Flat assembly

- 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)

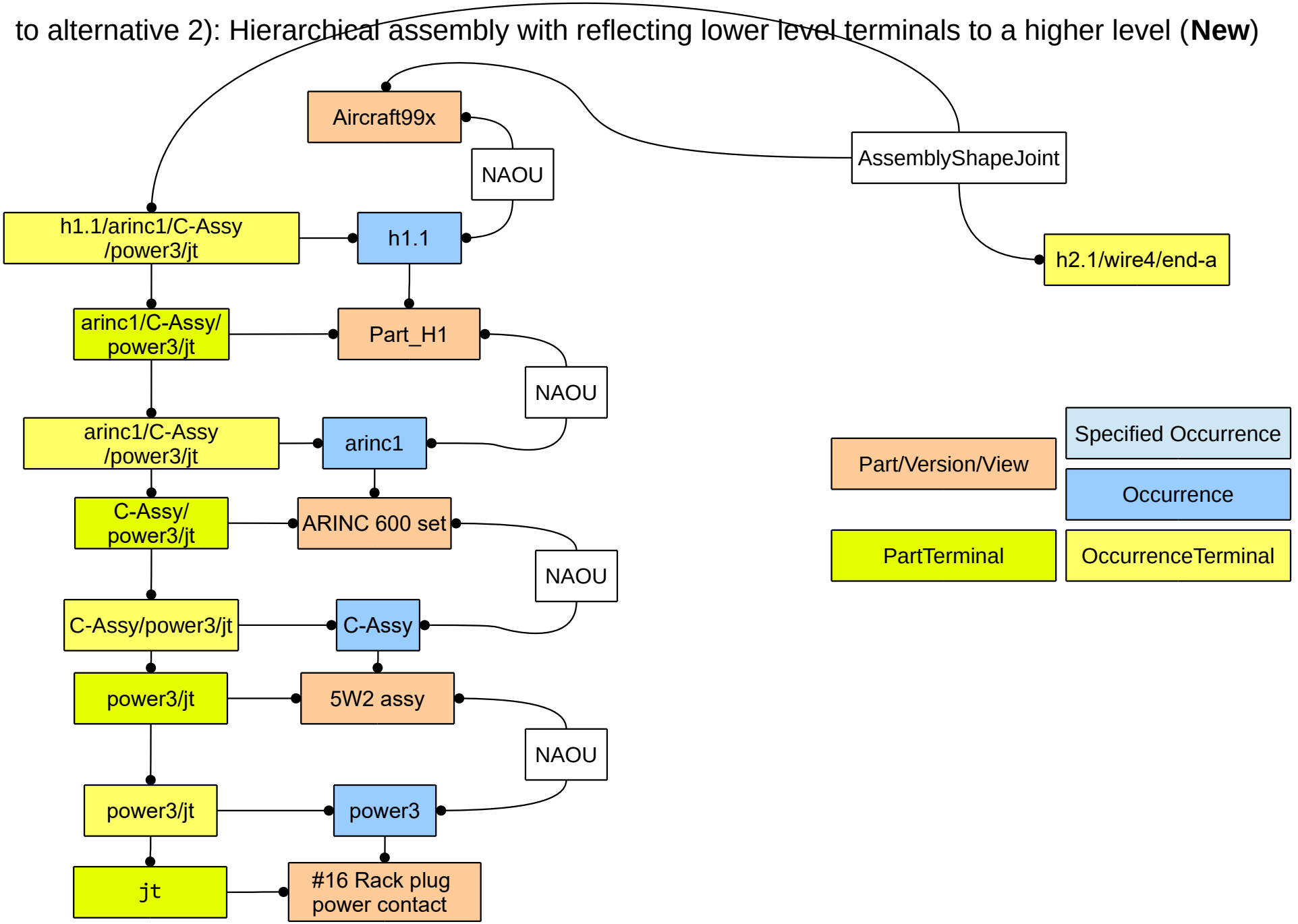
to alternative 1): Hierarchical assembly with SpecifiedOccurrences
see also: HarnessExample_Hierarchical.xml



Containment ==>>

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**)



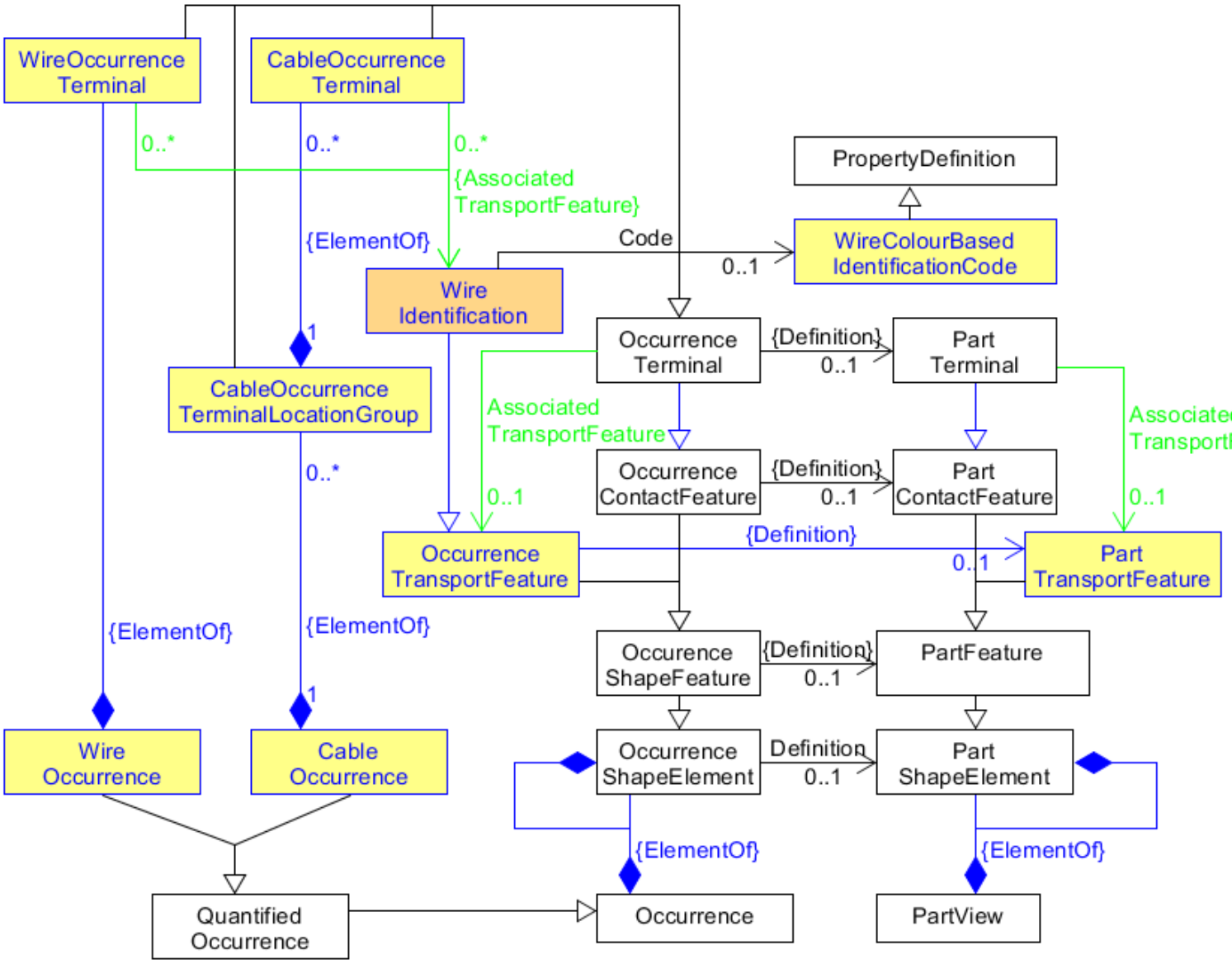
Complex Connectors and their usage in an EWH-Assembly (5 of 5)

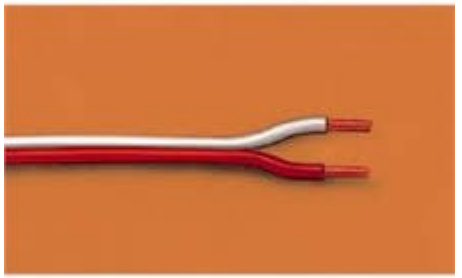
to alternative 3): Flat assembly

see also: HarnessExample_Flat.xml

Wires, Cables & Conductors (**current, ed2**)

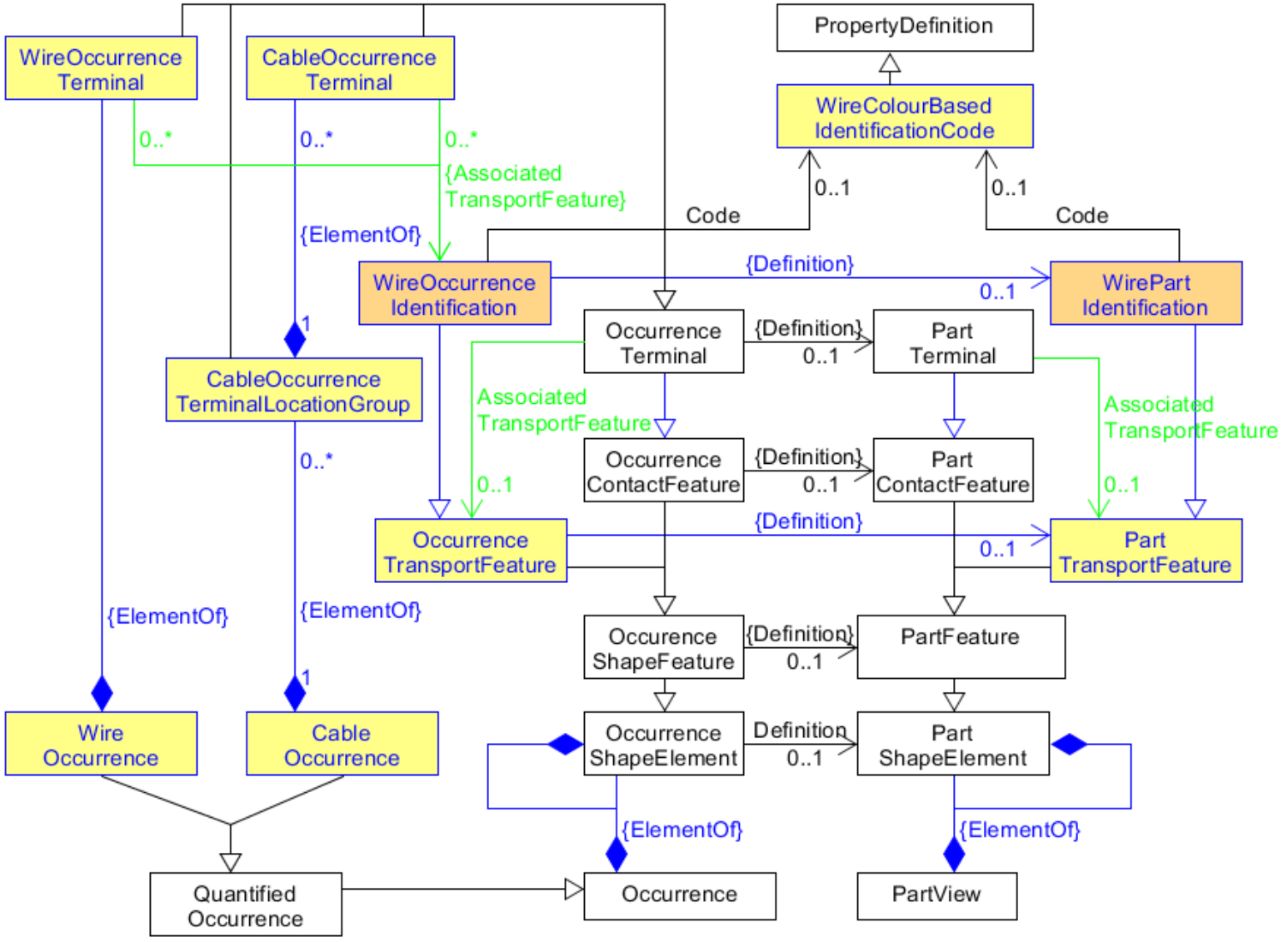
- A wire consists of a single conductor/ (maybe of several strands) and typically an isolation.
- The isolation has to be somehow removed before the conductor/*TransportFeature* can be accessed by a *Wire/CableOccurrenceTerminal* (details not covered so far).
- A cable consists of several “wires”(conductors), 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.

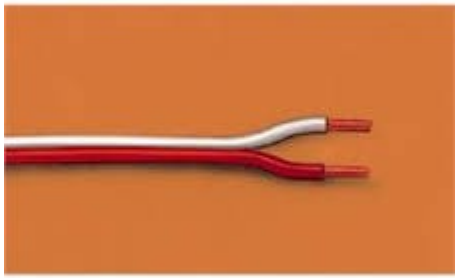




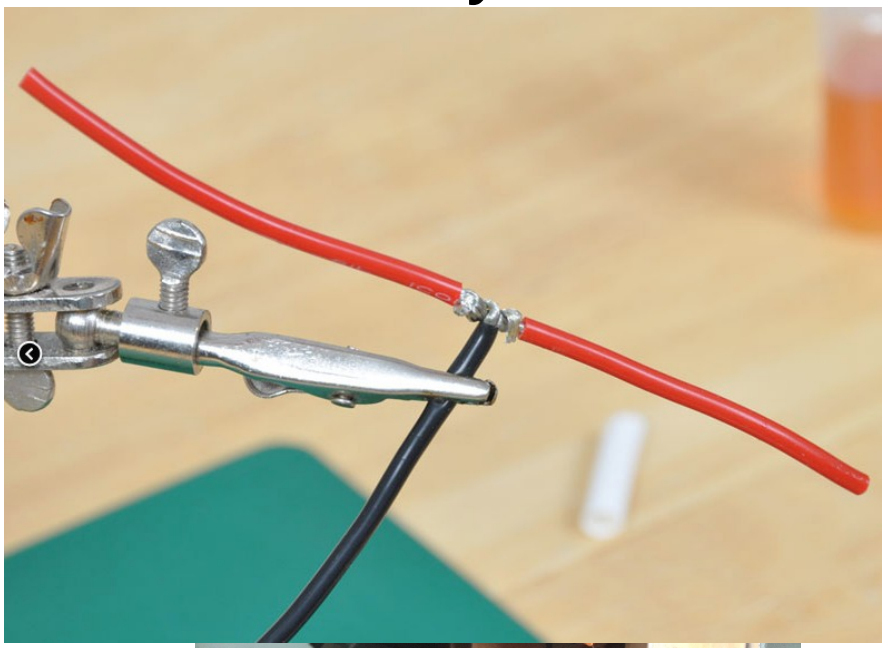
Wires, Cables & Conductors (**new, upcoming**)

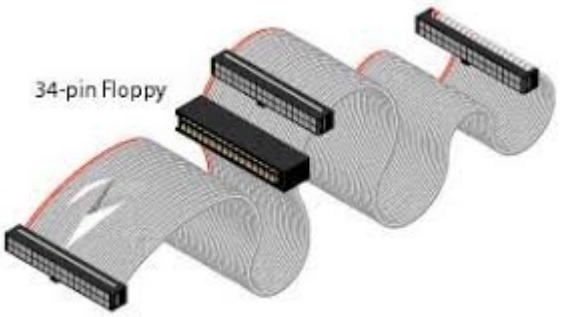
- Problem in ed2: In very most cases the colour code for a wire is defined for the raw material / part, and not for a particular occurrence.
- Changes: new *WirePartIdentification*
  old *WireIdentification* renamed to *WireOccurrenceIdentification*
- Typical usage: A *WireOccurrenceIdentification* refers to a *WirePartIdentification* as *Definition*.
  A *WireOccurrenceIdentification* has no colour code, but inherit this from *WirePartIdentification*

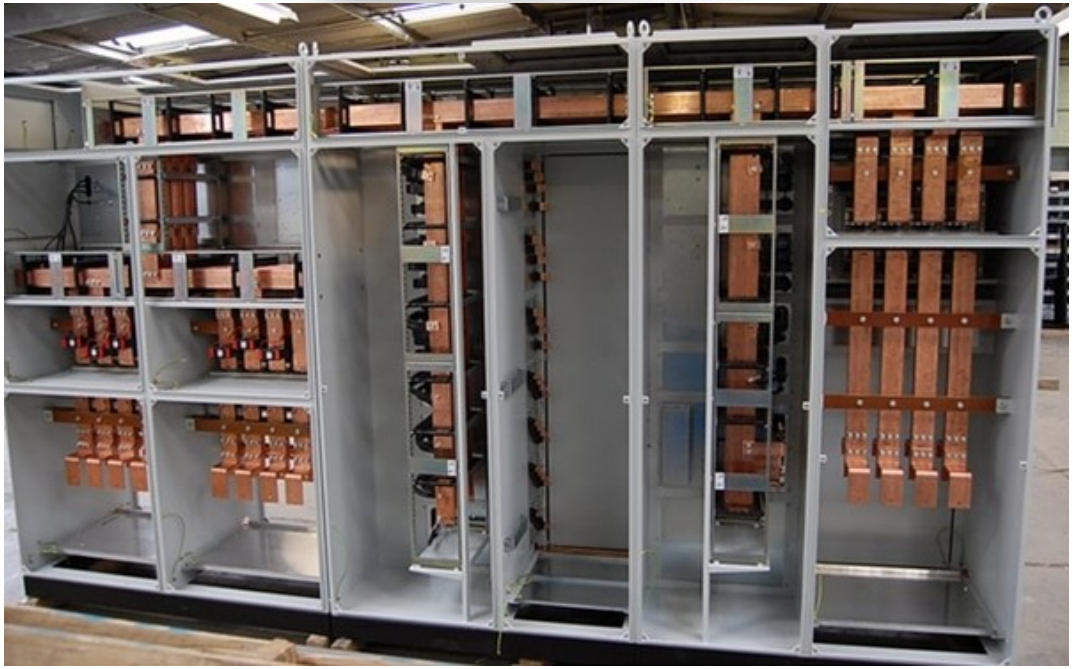




Often conductors (e.g. Wires, Cables, Busbars) are connected only at the ends but sometimes at any location







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

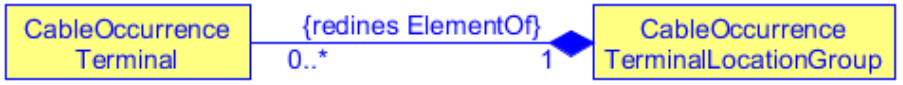
with a single *WireIdentification* to identify the conductor of the wire

and two *WireOccurrenceTerminals* at the ends of the wire, “end_a” and “end_b”, both referencing the *WireIdentification*

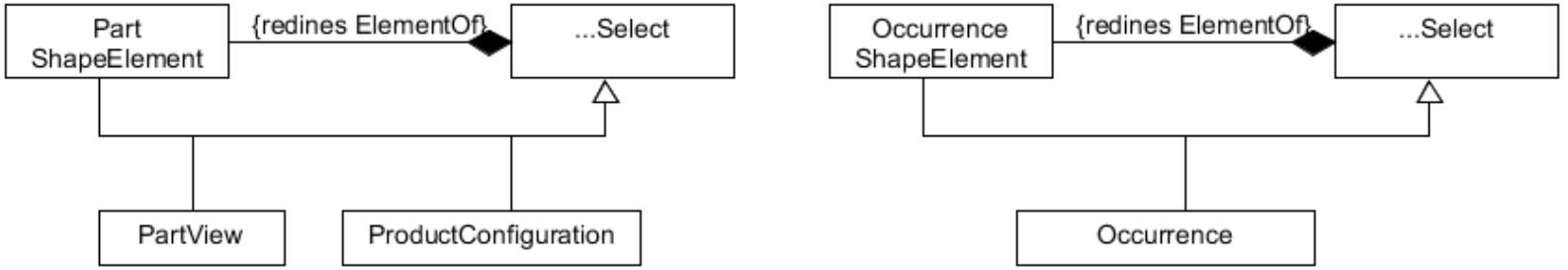
additional *WireOccurrenceTerminals* between the end of the wire can be defined as needed

Groups of ShapeElements - Multi-Terminals

- In most cases a PartShapeElement is an ElementOf a PartView, and an OccurrenceShapeElement is an ElementOf am 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)

```
<PropertyDefinition uid="_100205">
  <Id id="white"/>
  <PropertyType>
    <ClassString>wire colour-based identification code</ClassString>
  </PropertyType>
</PropertyDefinition>
<PropertyDefinition uid="_100206">
  <Id id="red"/>
  <PropertyType>
    <ClassString>wire colour-based identification code</ClassString>
  </PropertyType>
</PropertyDefinition >

<Part uid="_104000">
  ...
  <PartTypes>
    <PartCategoryEnum>cable</PartCategoryEnum>
    <PartCategoryEnum>raw_material_by_length</PartCategoryEnum>
  </PartTypes>
  <Versions>
    <PartVersion uid="_104001">
      <Id id="Version 1"/>
      <Views>
        <PartView uid="_104002">
          <DefiningGeometry uidRef="_104890"/>
          <InitialContext uidRef="_100102"/>
          ... => see occuurence on next page
          <ShapeElement xsi:type="n0:WirePartIdentification" uid="_104003">
            <Id id="A"/>
            <Code uidRef="_100205"/>
          </ShapeElement>
          <ShapeElement xsi:type="n0:WirePartIdentification" uid="_104004">
            <Id id="B"/>
            <Code uidRef="_100206"/>
          </ShapeElement>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

- 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)

```
<Occurrence xsi:type="n0:CableOccurrence" uid="_204006">
  <Id id="cable3"/>
  <ShapeElement xsi:type="n0:CableOccurrenceTerminalLocationGroup" uid="_204010">
    <Name>
      <CharacterString>end a</CharacterString>
    </Name>
    <ShapeElement xsi:type="n0:CableOccurrenceTerminal" uid="_204013">
      <AssociatedTransportFeature uidRef="_204003"/>
    </ShapeElement>
    <ShapeElement xsi:type="n0:CableOccurrenceTerminal" uid="_204014">
      <AssociatedTransportFeature uidRef="_204004"/>
    </ShapeElement>
  </ShapeElement>
  <ShapeElement xsi:type="n0:CableOccurrenceTerminalLocationGroup" uid="_204020">
    <Name>
      <CharacterString>end b</CharacterString>
    </Name>
    <ShapeElement xsi:type="n0:CableOccurrenceTerminal" uid="_204023">
      <AssociatedTransportFeature uidRef="_204003"/>
    </ShapeElement>
    <ShapeElement xsi:type="n0:CableOccurrenceTerminal" uid="_204024">
      <AssociatedTransportFeature uidRef="_204004"/>
    </ShapeElement>
  </ShapeElement>
  <ShapeElement xsi:type="n0:WireOccurrenceIdentification" uid="_204003">
    <Definition uidRef="_104003"/>
  </ShapeElement>
  <ShapeElement xsi:type="n0:WireOccurrenceIdentification" uid="_204004">
    <Definition uidRef="_104004"/>
  </ShapeElement>
  <Quantity xsi:type="n0:NumericalValue" uid="_204008">
    <Unit uidRef="_100301"/>
    <ValueComponent>4.25</ValueComponent>
  </Quantity>
</Occurrence>
```

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

Exampe: AssemblyShapeJoint

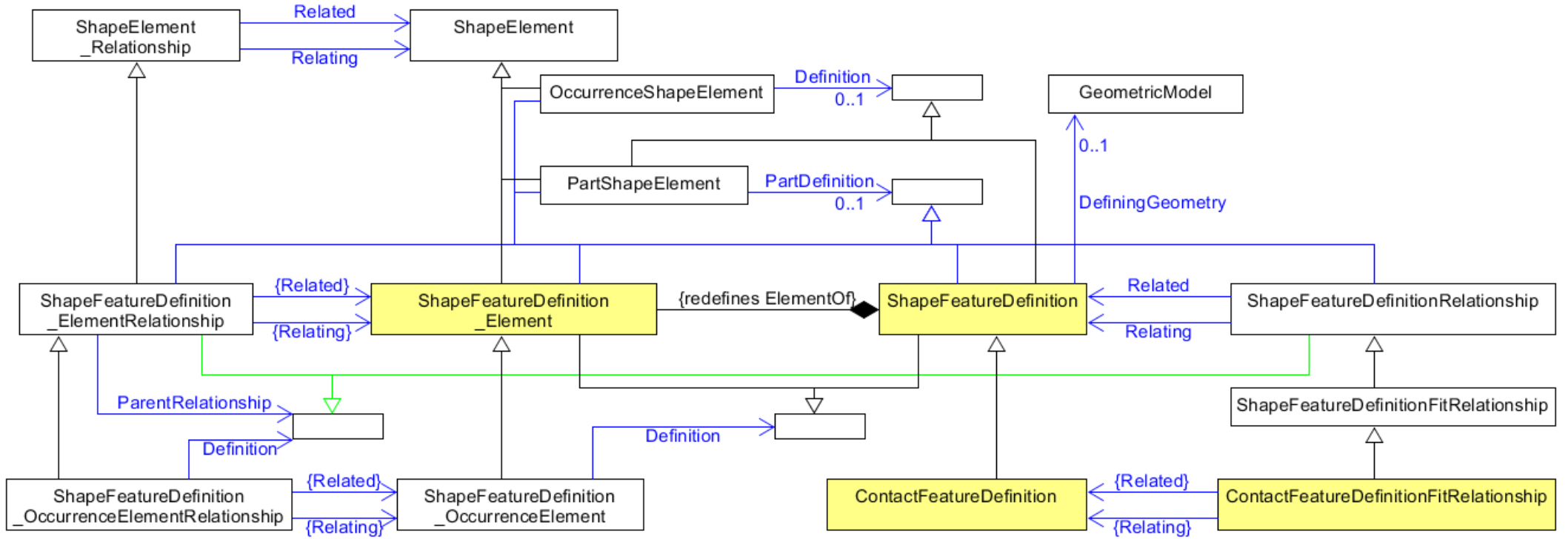
```
<Part uid="_311000"> <!-- Part_H1 -->
  ...
  <PartTypes> <PartCategoryEnum>wiring_harness</PartCategoryEnum> </PartTy
  <Versions>
    <PartVersion uid="_311001">
      ...
      <Views>
        <PartView xsi:type="n0:WiringHarnessAssemblyDesign" uid="_311002">
          ...
          <!--2 AssemblyJoints on Phone1 with cable3-->
          <ShapeElement xsi:type="n0:AssemblyShapeJoint" uid="_311020">
            <ShapeElementRelationship xsi:type="n0:AssemblyShapeJointItemR
              <Related uidRef="_204023"/> <!-- cable3#end b A -->
              <RelationType><ClassString></ClassString></RelationType>
            </ShapeElementRelationship>
            <ShapeElementRelationship xsi:type="n0:AssemblyShapeJointItemR
              <Related uidRef="_217102"/> <!-- phone1#Join signal -->
              <RelationType><ClassString></ClassString></RelationType>
            </ShapeElementRelationship>
            <JointType>crimped_connection</JointType>
          </ShapeElement>

          <ViewOccurrenceRelationship uid="_315021" xsi:type="n0:NextAssem
            <Related uidRef="_204006"/> <!--CableOccurrence cable3-->
            <RelationType>
              <ClassString>next assembly occurrence</ClassString>
            </RelationType>
          </ViewOccurrenceRelationship>
          <ViewOccurrenceRelationship uid="_315043" xsi:type="n0:NextAssemblyOccurrenceUsage">
            <Related uidRef="_217100"/> <!-- phone1 -->
            <RelationType>
              <ClassString>next assembly occurrence</ClassString>
            </RelationType>
          </ViewOccurrenceRelationship>
          ...
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

- a *Part* that is an assembly, here *WiringHarnessAssemblyDesign*
- *NextAssemblyOccurrenceUsages* 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”

ShapeFeatureDefinition and Elements

- 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 *ShapeFeatureDefinions* 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*

DEUTSCH INDUSTRIAL

Quick Connect & IMC Series Catalogue

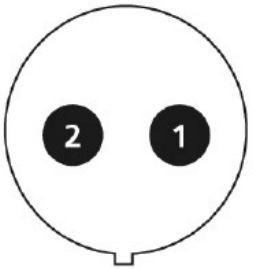
```
<Organization uid="_100"> <!-- Deutsch company -->
  <Id id="http://www.deutsch.net"/> ...
</Organization>

<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)

```
<Part uid="_200">  <!-- Deutsch connector -->
  <Id> <Identifier uid="_201" id="IMC16-2002X" idContextRef="_100"/> </Id>
  <PartTypes> <PartCategoryEnum>connector</PartCategoryEnum> </PartTypes>
  <Versions>
    <PartVersion uid="_202">
      <Id id="Version 1"/>
      <Views>
        <PartView uid="_202">
          <InitialContext uidRef="_100102"/>
          <ShapeElement xsi:type="n0:PartContactFeature" uid="_203">
            <Id id="1"/>
            <Definition uidRef="_101"/>
          </ShapeElement>
          <ShapeElement xsi:type="n0:PartContactFeature" uid="_204">
            <Id id="2"/>
            <Definition uidRef="_101"/>
          </ShapeElement>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

- a *Part* that is a *connector* with two *PartContactFeature* “1” and “2” that are defined by a *ContactFeatureDefinition* that is a cavity*_profile*

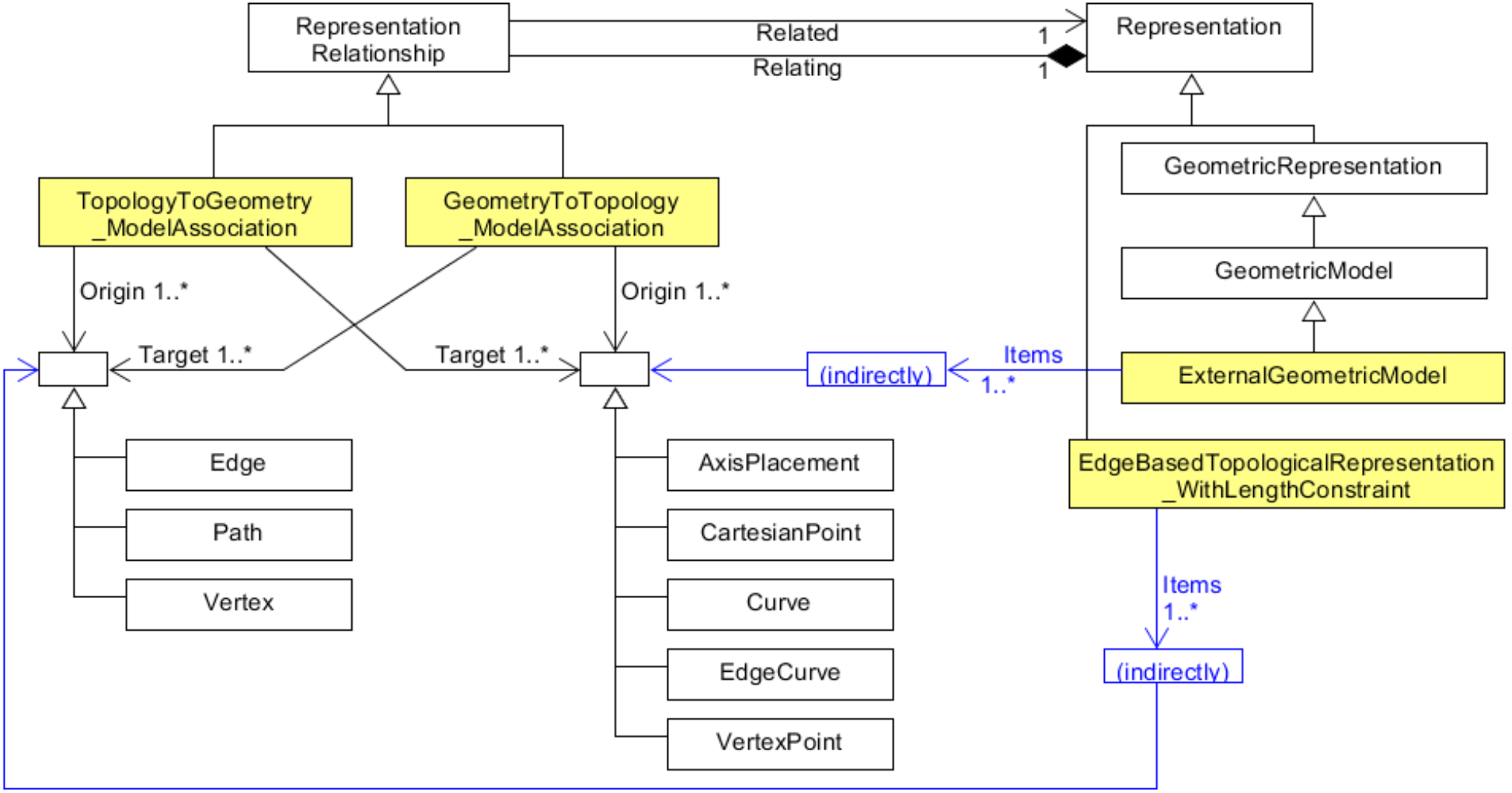



```
<Part uid="_300">  <!-- Deutsch connector-contact -->
  <Id> <Identifier uid="_201" id="6860-201-20278" idContextRef="_100"/> </Id>
  <PartTypes> <PartCategoryEnum>connector_contact</PartCategoryEnum> </PartTypes>
  <Versions>
    <PartVersion uid="_302">
      <Id id="Version 1"/>
      <Views>
        <PartView uid="_303">
          <InitialContext uidRef="_100102"/>
          <ShapeElement xsi:type="n0:PartContactFeature" uid="_304">
            <Id id="o"/>
            <Definition uidRef="_102"/>
          </ShapeElement>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

- a *Part* that is a *connector_contact* with a *PartContactFeature* “o” that is defined by a *ContactFeatureDefinition* that is a *contact_profile*

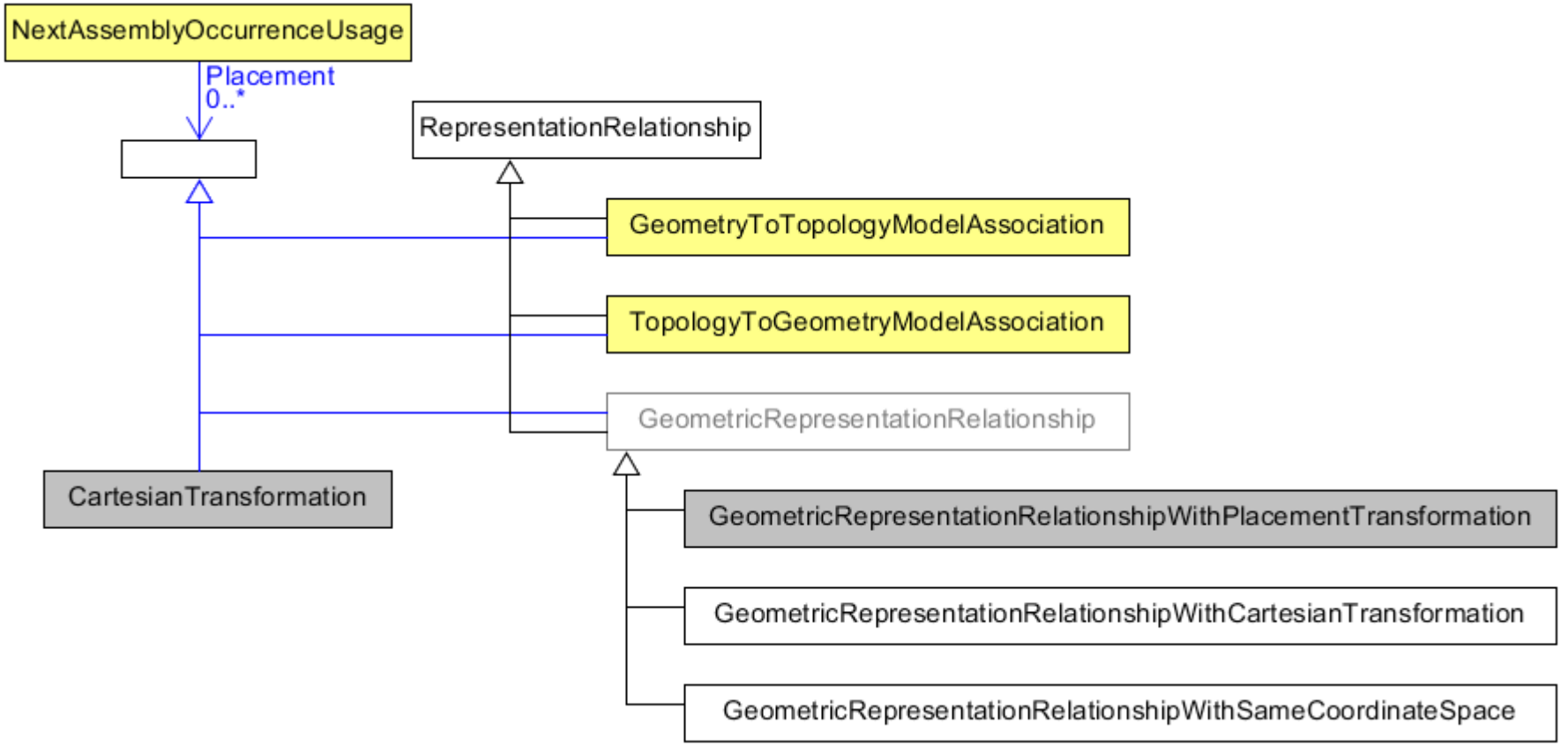
GeometryToTopologyModelAssociation & TopologyToGeometryModelAssociation

- 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

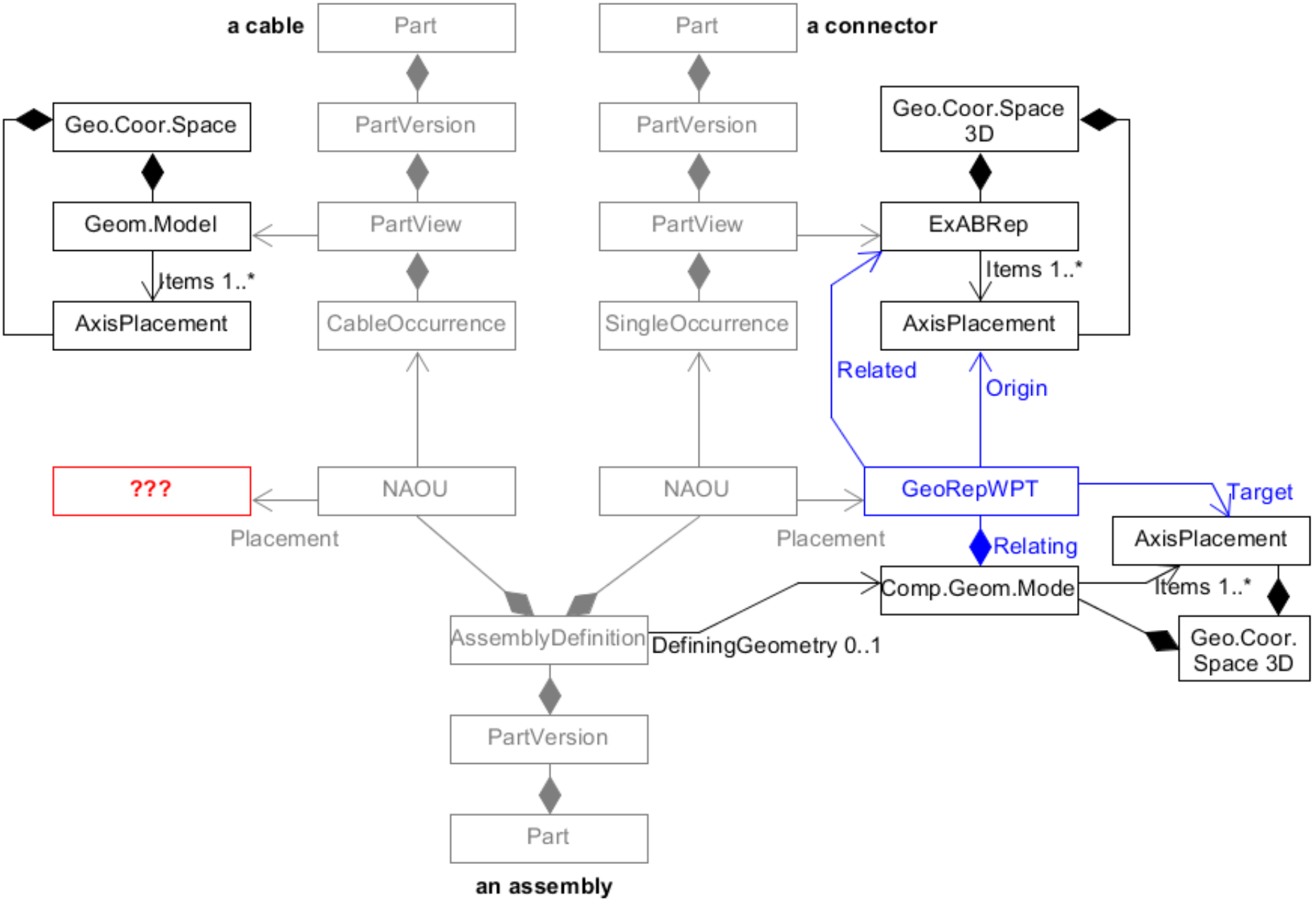


Use of the Geometry/Topology Associations for assembly cases

- 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 need 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

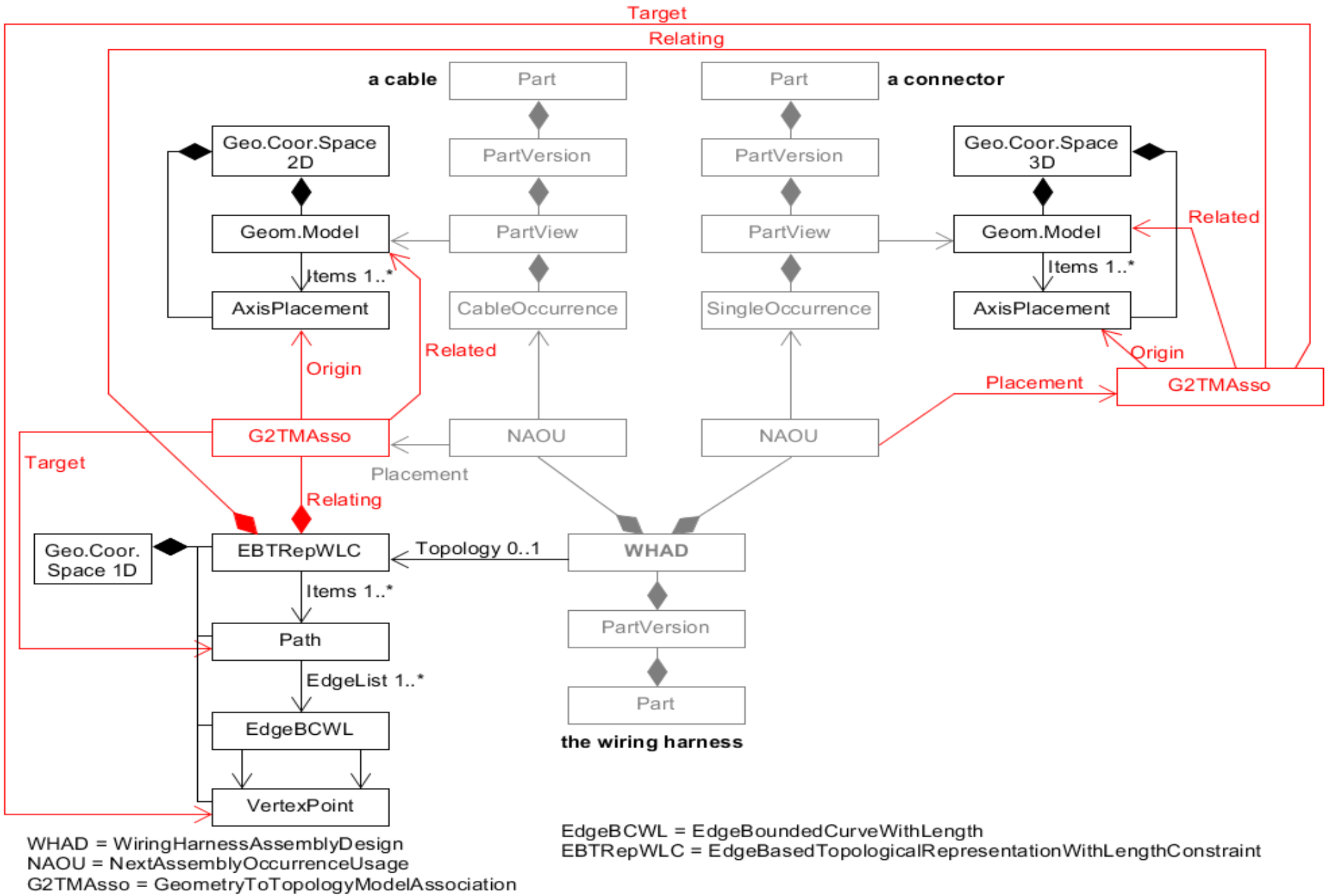


A simple Assembly of rigid Parts with Transformation. What to do with the flexible Cable?

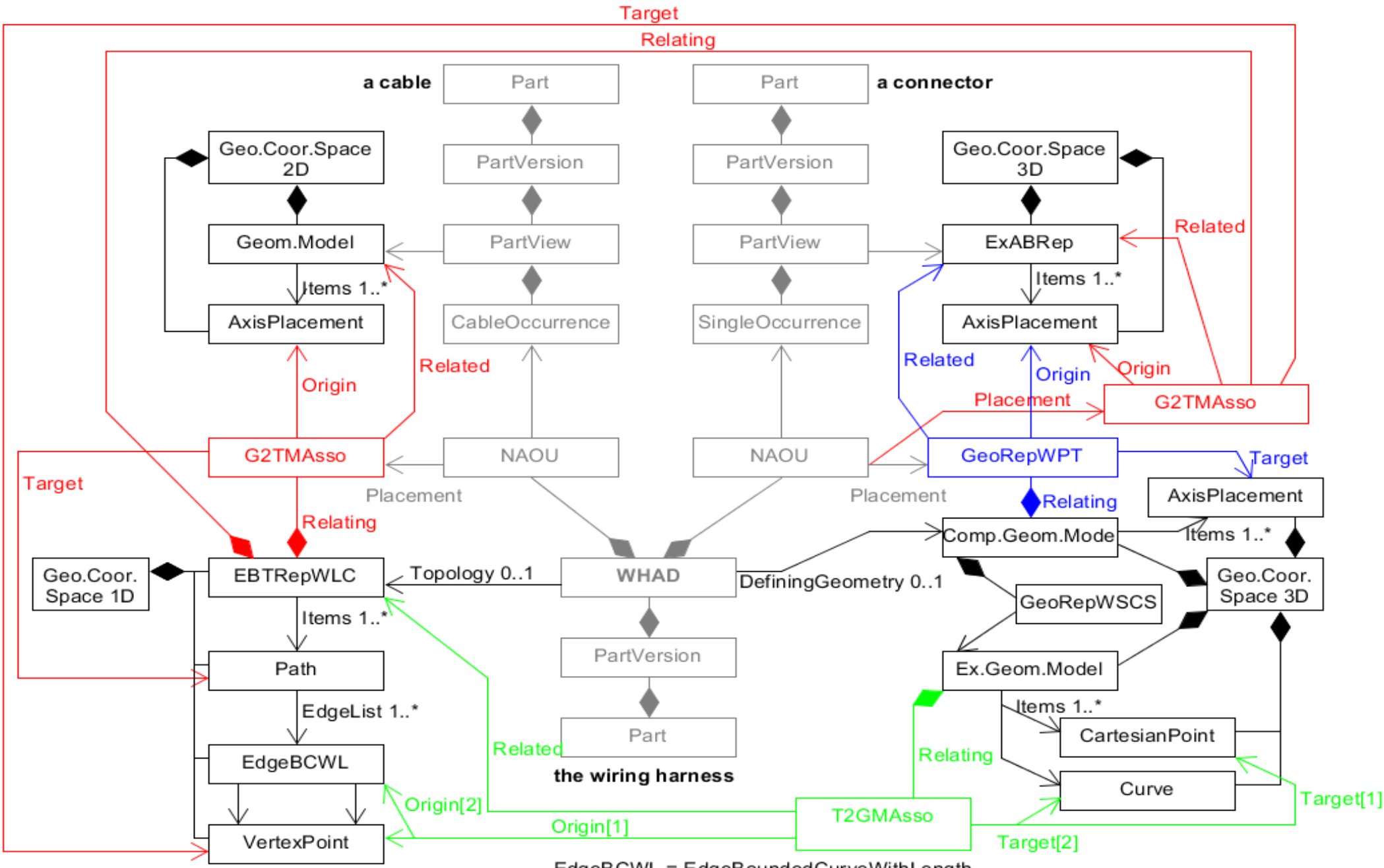


NAOU = NextAssemblyOccurrenceUsage

Associations of geometric Models to the topological Harness Model



A Wiring Harness with all Transformations and Associations

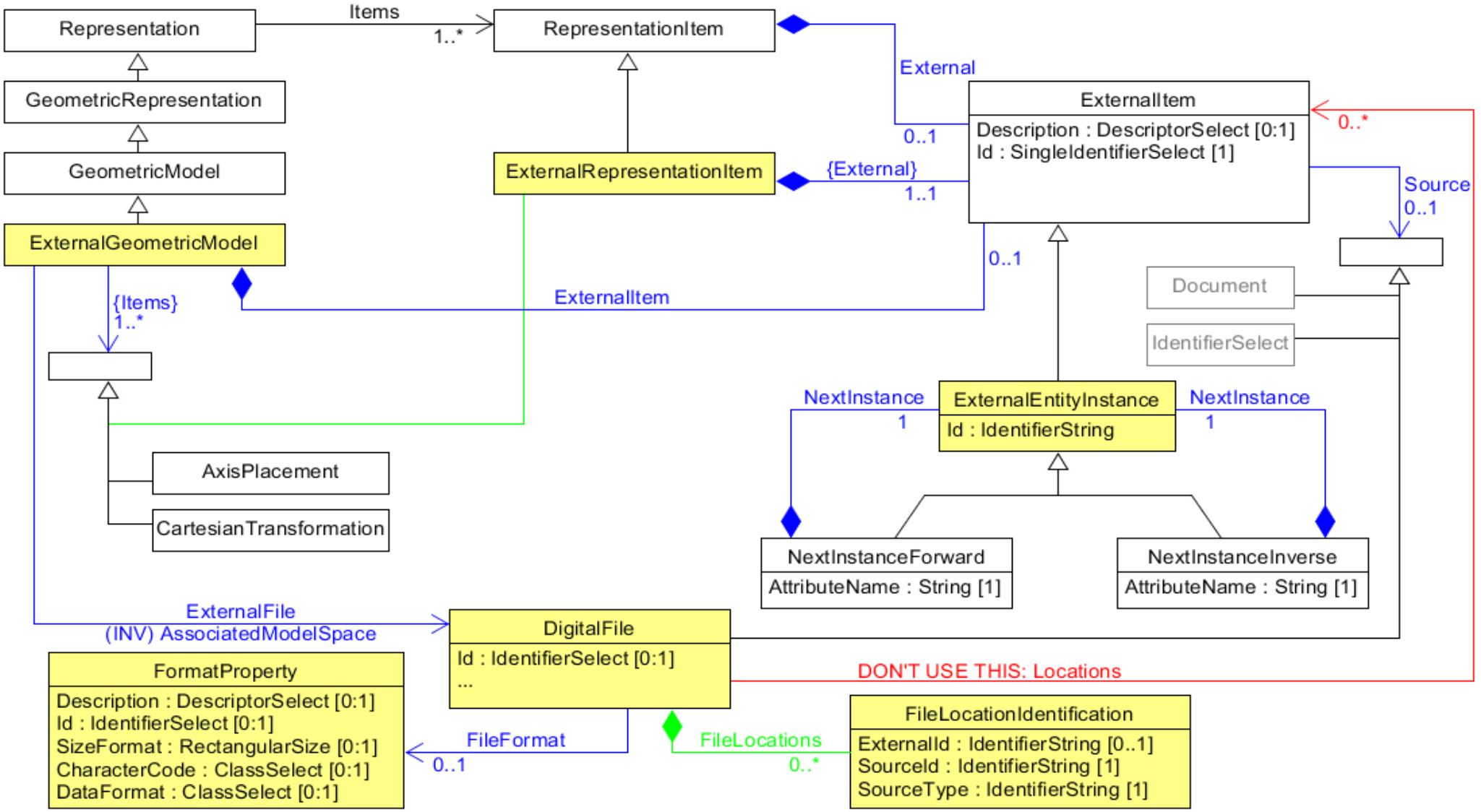


WHAD = WiringHarnessAssemblyDesign
NAOU = NextAssemblyOccurrenceUsage
G2TMAsso = GeometryToTopologyModelAssociation
T2GMAsso = TopologyToGeometryModelAssociation

EdgeBCWL = EdgeBoundedCurveWithLength
ExABRep = ExternalAdvancedBrepShapeRepresentation
EBTRepWLC = EdgeBasedTopologicalRepresentationWithLengthConstraint
GeoRepWPT = GeometricRepresentationRelationshipWithPlacementTransformation
GeoRepWSCS = GeometricRepresentationRelationshipWithSameCoordinateSpace

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)



Example: Terminal Lug with external Geometry (1/3)

```
<Part uid="_103000"> <!-- TERMINAL LUG CRIMP STYLE COPPER INSULATED RING TONGUE -->
  <Id id="MS5036-153"/>
  <Name>
    <LocalizedString lang="en-US">TERMINAL LUG CRIMP STYLE COPPER INSULATED RING TONGUE</LocalizedString>
    <LocalizedString lang="fr-FR">COSSE</LocalizedString>
  </Name>
  <PartTypes>
    <PartCategoryEnum>discrete</PartCategoryEnum>
    <PartCategoryEnum>terminal_lug</PartCategoryEnum>
  </PartTypes>
  <Versions>
    <PartVersion uid="_103001">
      <Id id="Version 1"/>
      <Views>
        <PartView uid="_103002">
          <DefiningGeometry uidRef="_103090"/>
          <InitialContext uidRef="_100102"/>
          ...
          <ShapeElement xsi:type="n0:PartTerminal" uid="_103003">
            <Id id="External"/>
            <RepresentedGeometry uidRef="_103092"/>
            <IntendedJointType>
              <TerminalJointTypeEnum>screw_terminal</TerminalJointTypeEnum>
            </IntendedJointType>
            <InterfaceOrJoinTerminal>interface_terminal</InterfaceOrJoinTerminal>
          </ShapeElement>
          <ShapeElement xsi:type="n0:PartTerminal" uid="_103004">
            <Id id="Internal"/>
            <RepresentedGeometry uidRef="_103094"/>
            <IntendedJointType>
              <TerminalJointTypeEnum>crimp_terminal</TerminalJointTypeEnum>
            </IntendedJointType>
            <InterfaceOrJoinTerminal>join_terminal</InterfaceOrJoinTerminal>
          </ShapeElement>
          ...
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

A part with an external geometric model and 2 features/terminals

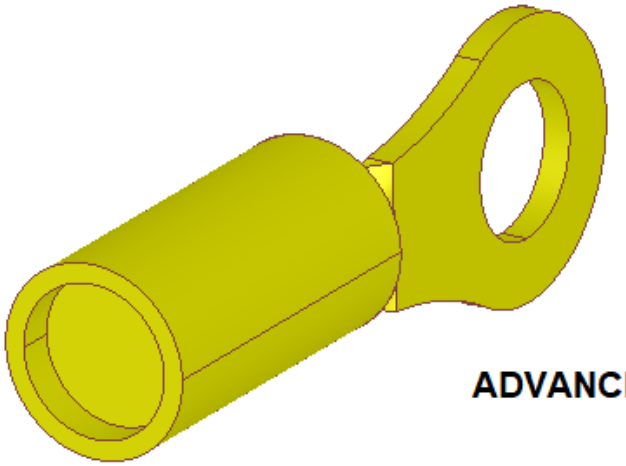
Example: Terminal Lug with external Geometry (2/3)

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

```
<!--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>
  <Items>
    <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>
  </Items>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>
```

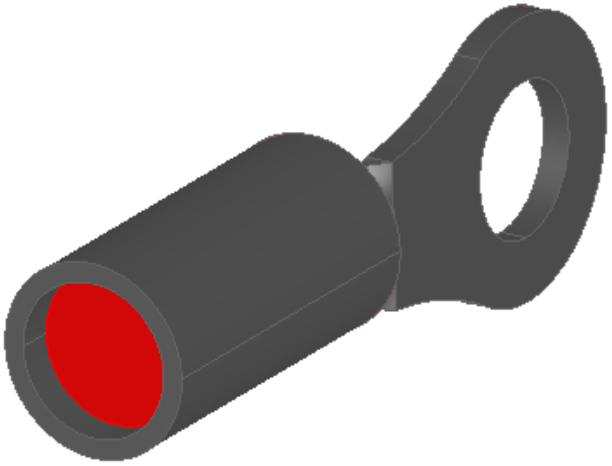
```
<!-- terminal lug geometric representation-->
<File xsi:type="n0:DigitalFile" uid="_103080">
  <FileFormat uidRef="_100300"/>
  <FileLocations>
    <FileLocationIdentification uid="_103081">
      <SourceId>c-51864-1-af-3d.stp</SourceId>
      <SourceType>file</SourceType>
    </FileLocationIdentification>
  </FileLocations>
</File>
```

Example: Terminal Lug with external Geometry (3/3)

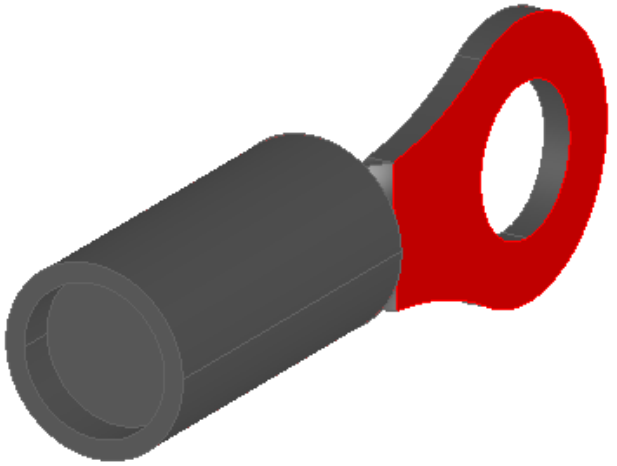


File: c-51864-1-af-3d.stp

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

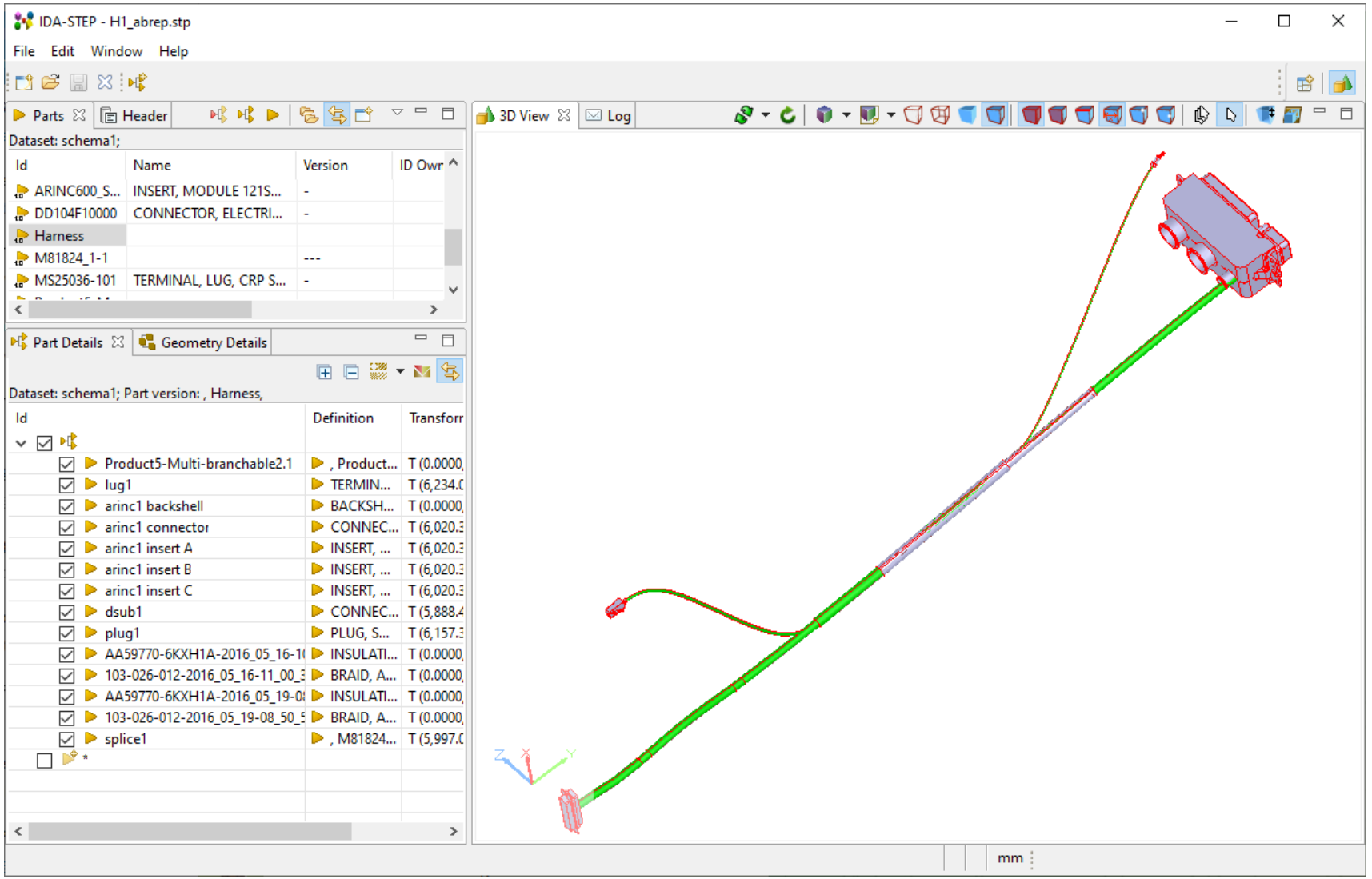
```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(...,'4;1');
FILE_NAME('star1.p21', ... );
FILE_SCHEMA(('CONFIG_CONTROL_DESIGN'));
ENDSEC;

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.

```
<RepresentationContext xsi:type="n0:GeometricCoordinateSpace" uid="_314091">
  <Id id="/NULL"/>
  <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">
      ...
    </Representation>
  </Representations>
  <Items> .., </Items>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>
```

Use of a ComposedGeometricModel (3/6)

- 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

```
<Representation xsi:type="n0:ExternalGeometricModel" uid="_314100">
  <Id id="xxx"/>
  <Items>
    <RepresentationItem uidRef="_314092"/>
    <RepresentationItem uidRef="_314096"/>
  </Items>
  <!--Transformation of topology model-->
  <RepresentationRelationship xsi:type="n0:TopologyToGeometryModelAssociation" uid="_314101">
    <Definitional>false</Definitional>
    <Related uidRef="_321010"/>  <!-- => EdgeBasedTopologicalRepresentationWithLengthConstraint -->
    <Origin>
      <Vertex uidRef="_321041"/>
      <Edge uidRef="_321021"/>
    </Origin>
    <Target>
      <AxisPlacement uidRef="_314096"/>
      <ExternalRepresentationItem uidRef="_314092"/>
    </Target>
  </RepresentationRelationship>
  <ExternalFile uidRef="_314080"/>  <!-- => H1_abrep.stp -->
  <ExternalItem xsi:type="n0:ExternalEntityInstance" uid="_314082">
    <Id id="#15"/> <!-- #15=SHAPE_REPRESENTATION(' ',(#1917,#5143,#31773,#44222,#45076,#45089,#45943,#47475,#50537,#50845,#51147,#51449,#
  </ExternalItem>
</Representation>
</Representations>
```

Use of a ComposedGeometricModel (4/6)

- The ***WiringHarnessAssemblyDesign*** is:
  referring the *ComposedGeometricModel*
  as *DefiningGeometry*
  refetring the *EdgeBasedTopologicalRepresentationWithLengthConstraint*
  as *Topology*

```
<Part uid="_311000"> <!-- Part_H1 -->
  <Id id="Part_H1"/>
  <Name>
    <CharacterString>Electrical Harness example 1</CharacterString>
  </Name>
  <PartTypes>
    <PartCategoryEnum>assembly</PartCategoryEnum>
    <PartCategoryEnum>wiring_harness</PartCategoryEnum>
  </PartTypes>
  <Versions>
    <PartVersion uid="_311001">
      <Id></Id>
      <Views>
        <PartView xsi:type="n0:WiringHarnessAssemblyDesign" uid="_311002">
          <AdditionalContexts>
            ...
          </AdditionalContexts>
          <DefiningGeometry uidRef="_314090"/>  <!-- => ComposedGeometricModel -->
          <InitialContext uidRef="_100102"/>
          ...
          <Topology uidRef="_321010" />  <!-- => EdgeBasedTopologicalRepresentationWithLengthConstraint -->
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

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 #1915, 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

```
<Items>
 <RepresentationItem xsi:type="n0:ExternalRepresentationItem" uid="_314092">
  <External  xsi:type="n0:NextInstanceInverse" uid="_314092_1">
   <Id id="#15"/> <!-- #15=SHAPE_REPRESENTATION(' ',(#1917,#5143,#31773,#44222,#45076,#45089,#45943,#47475,#50537,#50845,#51147,#51449,#51
   <AttributeName>rep_2</AttributeName>
   <NextInstance  xsi:type="n0:NextInstanceForward" uid="_314092_2">
    <Id id="#1915"/> <!-- #1915=(REPRESENTATION_RELATIONSHIP(' ',' ',#24,#15)REPRESENTATION_RELATIONSHIP_WITH_TRANSFORMATION(#1916)SHAPE_
    <AttributeName>rep_1</AttributeName>
    <NextInstance  xsi:type="n0:NextInstanceForward" uid="_314092_3">
     <Id id="#24"/> <!-- #24=SHAPE_REPRESENTATION(' ',(#1918),#23) ; -->
     <AttributeName>context_of_items</AttributeName>
     <NextInstance  xsi:type="n0:NextInstanceInverse" uid="_314092_4">
      <Id id="#23"/> <!-- #23=(GEOMETRIC_REPRESENTATION_CONTEXT(3)GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((#22))GLOBAL_UNIT_ASSIGNED_CONTEX
      <AttributeName>context_of_items</AttributeName>
      <NextInstance  xsi:type="n0:NextInstanceForward" uid="_314092_5">
       <Id id="#96"/> <!-- #96=GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION('NONE',(#97),#23) ; -->
       <AttributeName>items</AttributeName>
       <NextInstance  xsi:type="n0:NextInstanceForward" uid="_314092_6">
        <Id id="#97"/> <!-- #97=GEOMETRIC_SET('NONE',(#90,#99,#104,#109,#114,#119,#124,#141,#157)) ; -->
        <AttributeName>elements</AttributeName>
        <NextInstance  xsi:type="n0:ExternalEntityInstance" uid="_314092_7">
         <Id id="#141"/> <!-- #141=COMPOSITE_CURVE('Flexible Curve.2',(#140),.U.) ; -->
        </NextInstance>
       </NextInstance>
      </NextInstance>
     </NextInstance>
    </NextInstance>
   </NextInstance>
  </External>
  </RepresentationItem>
  ...
```

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

```
    ...
    <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>
```