LOTAR

LONG TERM ARCHIVING AND RETRIEVAL

STEP AP 242 Electrical Harness
XML Tutorial

Version: 1.1
Date: 2016-07-05
Status: Update to version 1.0, primarily on clause 9.

Lothar Klein
LKSoftWare GmbH
Steinweg 1
36093 Künzell, Germany

This document contains SysML data models and XML Examples and is based on an XML schema (XSD file) provided by:

Sophie HERAIL
CIMPA S.A.S.
Subcontractor for AIRBUS Operations SAS - EZMBB
Centreda 1
4, Avenue Didier Daurat
31700 Blagnac

Links:
http://www.lotar-international.org/
http://www.ap242.org/
https://en.wikipedia.org/wiki/ISO_10303
http://www.iso.org

Table of Contents

1. Overview

The purpose of this document is to walk through the Electrical Harness Extension of the upcoming AP242 edition 2 standard, and to explain how to use it.

Readers are expected to be familiar with the Recommended Practices for AP242 Business Object Model XML Assembly Structure

This document is based on the following documents:

- Recommended Practices for AP242 Business Object Model XML Assembly Structure Release 1.0.90; - Working Document –; October 26, 2015 (planned to be released as v1.2)
- STEP: ISO 10303 "Industrial automation systems and integration -- Product data representation and exchange"
  - AP242 ed1: ISO 10303-242:2014 Part 242: Application protocol: Managed model-based 3D engineering"
  - Change Request 11 for the SMRL, "STEP Module and Resource Library", that is intended to be published in future version of the SMRL after v6 and basis for the upcoming CD document for the second edition of AP242.
  - Current draft SysML data model and derived XSD for AP242 CD second edition

The work is performed for the joint ASD-Stan / AIA / ProSTEP iViP / PDES Inc. consortium "LOTAR International".

1.1. Reference Example

This tutorial comes together with a reference example that had been worked out by the electrical harness working group of the AP242 Edition 2 development team. The example is hand-crafted and is not a complete design. The intention of the example is to demonstrate how the different AP242 edition 2 extensions for electrical harness design play together.

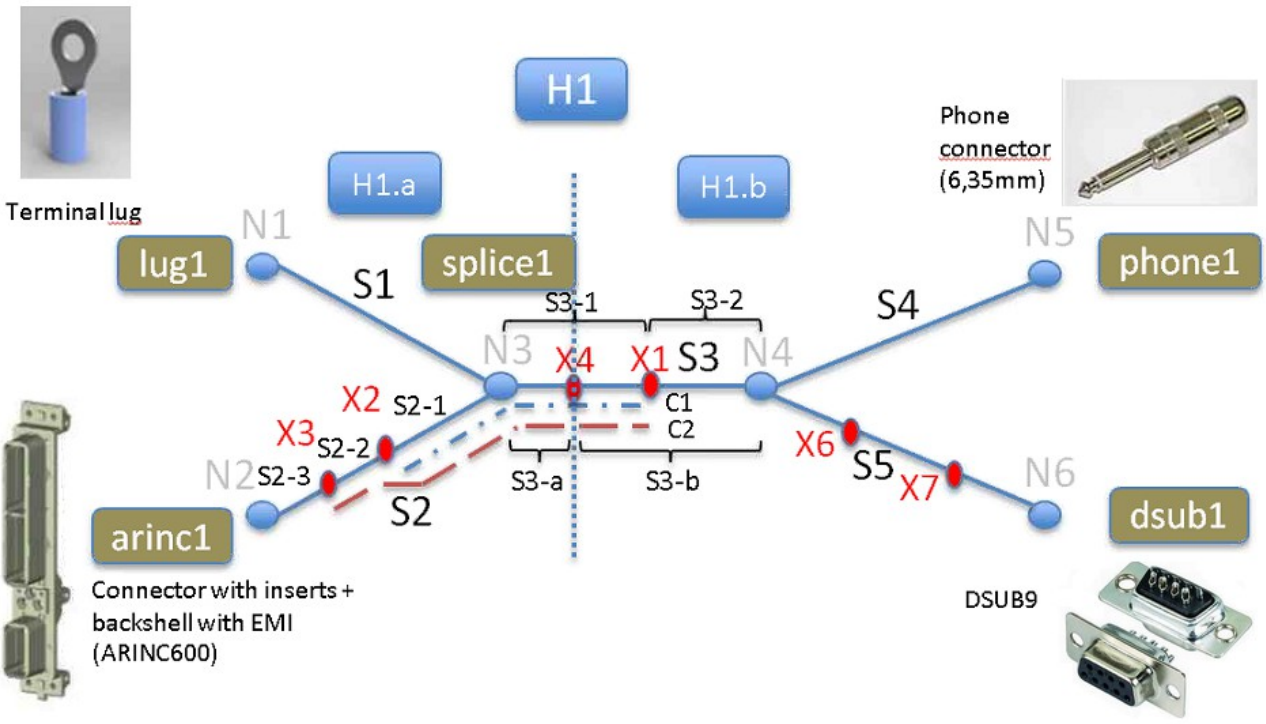

Figure 1: Topological structure and connectors of the provided example Part_H1

The main harness topology in this example consists of five segments S1 to S5 that connects six nodes N1 to N6. On the end-nodes N1, N2, N5 and N6 different kinds of connectors and a terminal lug are placed.

- Most wires and cables are connected to the ARINC600 connector at node N2 and go through segment S2. These are two coax RG 58 cables (cable1, cable2), a speaker cable (cable3), and a single wire (wire2).
- On the internal node N3 a splice is placed that is realizing a T connection between 3 wires (wire1, wire2, wire3).
- The terminal lug at node N1 is connected by a single wire arriving from segment S1. As S1 contains only a single component no grouping is needed for this segment.
- The phone connector at node N5 is connected by a single speaker wire (cable3) arriving through segment S4. As S4 contains only a single component no grouping is needed for this segment.
- The DSUB connector at node N6 is connected by two coax cables (cable1, cable2) and a wire (wire3). These three wires and cables are grouped together with lacing in Segment S5.
- In segment S2 wire2 and cable3 are grouped by twisting around each other. The result is then grouped together with cable1 and cable2 and covered by a shielding braid that is connected to the GND of the backshell of the ARINC600 connector.
- Segment S3 contains the same arrangement of cables and wires as in segment S2, but because of the splice at node N3, it contains cable3 instead of cable2.
- A path through the segments S2, S3 and S4 has been defined for the speaker wire (cable 3)
- A path through the segments S2, S3 and S5 has been defined for the two coax cables (cable1, cable2) and the alternative of wire1 respectively wire3

Occurrence	- name	Part-name	Segment / Path	Start-node	End-node
_201004	wire1	WIRE,ELEC,COMP, SNGL CONDUCTOR,150 DEG C	S1	N1	N3
_201104	wire2		S2	N2	N3
_201204	wire3		S3-S5	N3	N6
_202006	cable1	RG 58	S2-S3-S5	N2	N6
_202106	cable2	RG 58	S2-S3-S5	N2	N6
_204006	cable3	Speaker wire	S2-S3-S4	N2	N5
_220005	braid1	Braid 1/2inch	S2-S3	N2	N4
_221005	wrap1	1/2 inch Spiral Wrap - 100 foot spool - Black	C1 = S2-1 + S3-1	X2	X1
_222005	heatshrink1	Shrinkflex Polyolefin Heatshrink Tubing - 4/1 - 25mm 1	C2 = S2-2 + S2-2 + S3-1	X3	X1

Table 1: Wire / cable list for H1

A more detailed sub-topology on the main topology is available through the sub-nodes X1 to X7 that are placed on the main segments S2, S3 and S5. These sub-nodes divide the main segments. Segment S2 is divided by X2 and X3 into S2-1, S2-2 and S2-3. Segment S3 is divided in two alternative ways, S3-1 and S3-2 by X1 and S3-a and S3-B by X4. No sub-segments have been defined for the sub-nodes X6 and X7 on S5 because there had been no need for it. These detailed topology is used as follows:

- to define a path consisting of the segments S2-1 and S3-1 for a protective covering (wrap1)
- to define a path consisting of the segments S2-2, S2-1 and S3-1 for a heatshrink (heatshrink1)
- to split the overall harness H1 into two sub-harnesses H1.a and H1.b that different design groups are

responsible for

- to define fixing locations at nodes X6 and X7

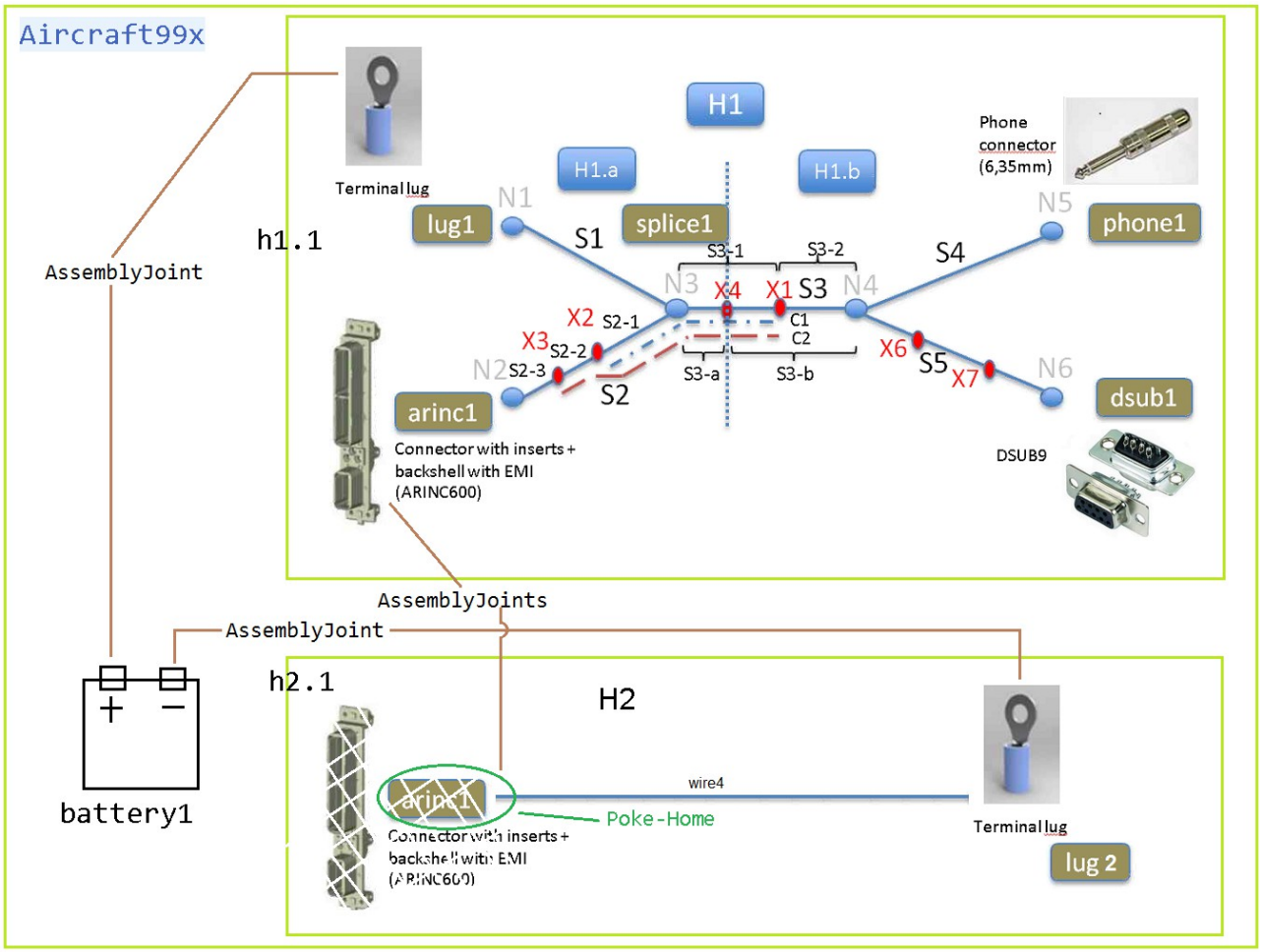

Figure 2: Harnesses H1 and H2 combined in top assembly Aircraft99x

The example also contains a second mini harness example Part_H2 and a top assembly Aircraft99x (figure 2). The mini harness example Part_H2 consists only of a single wire (wire4) and a terminal lug (lug2) to which one end of the wire is connected to. The other end of the wire is intended to be connected to the arinc1 connector from the Part_H1 example above, but this connector is not part of the H2 assembly. This situation is in some industries called “poke-home”.

The top assembly Aircraft99x that contains 3 Occurrences:

- h1.1 that is of type Part_H1
- h2.1 that is of type Part_H2
- battery1 that is connected with the interface terminals of the two terminal lugs

The PLUS and MINUS terminals of the battery is connected to the interface terminals of the two terminal lugs (lug1 and lug2).

Also the “poke-home” connection from the H2 assembly can finally be connected in that top assembly because arinc1 from Part_H1 and wire4 from Part_H2 are available in this assembly.

In the following clauses this example is further detailed. The example file “HarnessExample1.xml“ is publicly available via downloaded from:
http://stepmod.cvs.sourceforge.net/viewvc/stepmod/stepmod/etc/ap242/EH_XML_Examples/HarnessExample1.xml?view=log

Occurrence	Name	Occurrence-Terminal	Name	Assembly Join
	Part_H1			
_201004	wire1	_201006	end a	_311010-1
		_201007	end b	_311050-1
_201104	wire2	_201106	end a	_311040-1
		_201107	end b	_311050-2
_201204	wire3	_201206	end a	_311052-2
		_201207	end b	_311034-2
_204006	cable3	_204013	end a A	_311045-1
		_204014	end a B	_311046-1
		_204023	end b A	_311020-1
		_204024	end b B	_311021-1
_202006	cable1	_202013	end a signal	_311041-1
		_202014	end a shield	_311042-1
		_202023	end b signal	_311030-2
		_202024	end b shield	_311031-2
_202106	cable2	_202113	end a signal	_311043-1
		_202114	end a shield	_311044-1
		_202123	end b signal	_311032-2
		_202124	end b shield	_311033-2
_203005	lug1	_203006	External	no
		_203008	Internal	_311010-2
_217100	phone1	_217101	Interface signal	no
		_217102	Join signal	_311020-2
		_217103	Interface gnd	no
		_217104	Join gnd	_311021-2
_218100	splice1	_218103	end a	_311051-2
		_218104	end b	_311052-1
_219100	dsub1	_219113	Interface 1	no
		_219114	Join 1	_311030-1
		_219123	Interface 2	no
		_219124	Join 2	_311031-1
		_219133	Interface 3	no
		_219134	Join 3	_311032-1
		_219143	Interface 4	no
		_219144	Join 4	_311033-1
		_219153	Interface 5	no
		_219154	Join 5	_311034-1
		_219163	Interface 6	no
		_219164	Join 6	n.c.
		_219173	Interface 7	no
		_219174	Join 7	n.c.
		_219183	Interface 8	no
		_219184	Join 8	n.c.
		_219193	Interface 9	no
		_219194	Join 9	n.c.

Occurrence	Name	Occurrence-Terminal	Name	Assembly Join
Part_H1 (continued)				
_213100	arinc1/C-Assy/coax1	_213121	Interface signal	no
		_213122	Join signal	_311041-2
		_213123	Interface gnd	no
		_213124	Join gnd	_311042-2
_213210	arinc1/C-Assy/coax2	_213221	Interface signal	no
		_213222	Join signal	_311043-2
		_213223	Interface gnd	no
		_213224	Join gnd	_311044-2
_214110	arinc1/C-Assy/power3	_214201	Int term	no
		_214223	Join term	_311045-2
_214220	arinc1/C-Assy/power4	_214201	Int term	no
		_214224	Join term	_311046-2
_216120	arinc1/C-Assy/signal5	_216101	Int term	no
		_216122	Join term	_311040-2
_223200	arinc1/backshell1	_223201	GND1	_311047-1
_220005	braid1	_220007	shield-A	_311047-2
AssemblyJoint		_311050		_311051-1
Part_H2				
_203005	lug2	_203006	External	no
		_203008	Internal	_411010-1
_201204	wire4	_201206	end a	(_411011-1)
		_201207	end b	_411010-2
_213220	arinc1/C-Assy/power3	_214101	Int term	no
		_214122	Join term	(_411011-2)
Aircraft99x				
_311105	h1.1			
_201114	h1.1/wire2	_201116	end a	_511010-1
_203015	h1.1/lug1	_203016	External	_511012-1
_214130	h1.1/arinc1/C-Assy/power3	_214134	Join term	_511011-2
_411105	h2.1			
_201314	h2.1/wire4	_201316	end a	_511010-2
_203115	h2.1/lug2	_203116	External	_511013-1
_224100	battery1	_224101	PLUS	_511012-2
		_224102	MINUS	_511013-2
AssemblyJoint		_511010		_511011-1

Table 2: AssemblyJoints

2. Part, PartVersion, Part category

Objects of type **Part** are used for the all the parts that are used to build an electrical harness. In addition an electrical harness is modelled as *Part* as well, so that it can be used in a next higher assembly; e.g. for a complete aircraft. For each Part at least one PartVersion shall be defined. A PartVersion may have one or several views (see next chapter).

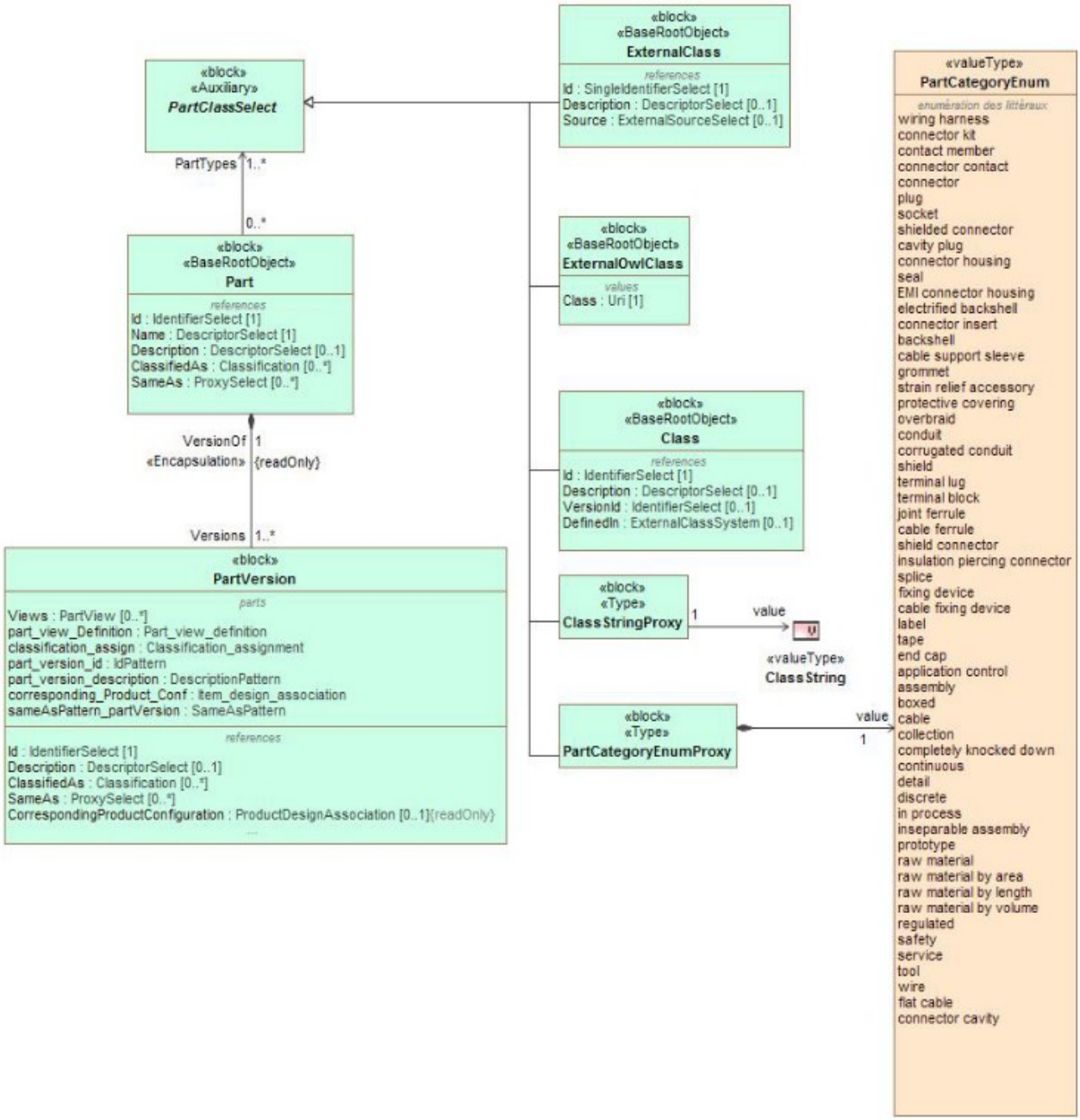

Figure 3: Part, PartVersion and categories for Parts

Example: A wire part that is provided as bulk material. For the usage of the *Part* in an *Occurrence* it needs to be cut to a specific length.

```
<Part uid="_101000">
   <Id id="04034-22-9"/>
   <Name>
```



```
      <CharacterString>WIRE,ELEC,COMP,SNGL CONDUCTOR,150 DEG C</CharacterString>
    </Name>
    <PartTypes>
      <PartCategoryEnum>wire</PartCategoryEnum>
      <PartCategoryEnum>raw material by length</PartCategoryEnum>
    </PartTypes>
    <Versions>
      <PartVersion uid="_101001">
        <Id id="Version 1"/>
        <Views>
          ...
        </Views>
      </PartVersion>
    </Versions>
</Part>
```

The example provided by this tutorial contains the following Parts:

```
<Part uid="_101000"> <!-- WIRE,ELEC,COMP,SNGL CONDUCTOR,150 DEG C -->
<Part uid="_102000"> <!-- RG 58 (COAX cable) -->
<Part uid="_103000"> <!-- TERMINAL LUG CRIMP STYLE COPPER INSULATED RING TONGUE -->
<Part uid="_104000"> <!-- Speaker wire -->
<Part uid="_117000"> <!-- Phone connector 6.35mm -->
<Part uid="_118000"> <!-- Splice-->
<Part uid="_119000"> <!-- Dsub9-->
<Part uid="_110000"> <!-- ARINC 600-->
<Part uid="_111000"> <!-- ARINC 600 shell 1 A and B dummy -->
<Part uid="_112000"> <!-- ARINC 600 shell 1 C 5W2 insert -->
<Part uid="_113000"> <!-- #5 Coax contact -->
<Part uid="_114000"> <!-- #16 Rack plug power contact -->
<Part uid="_115000"> <!-- Insert 5W2 assembly -->
<Part uid="_116000"> <!-- 12 Rack plug signal contact -->
<Part uid="_123000"> <!-- Backshell EMI connector -->
<Part uid="_120000"> <!-- Braid 1/2inch -->
<Part uid="_121000"> <!-- Wrap -->
<Part uid="_122000"> <!-- HeatShrink -->
<Part uid="_311000"> <!-- Part_H1 -->
```

The main attributes for a Part are its Id (used as master part number without version information) and Name. For the design exchange of electrical harness the specification of a part category is essential so that a receiving system can filter the data in a certain way. In this example the pre-defined category “wire” is used.

2.1. Generic Predefined Part Categories

AP242 defines the following generic categories (not specific for electrical harness):

- **application control**: this type of classification is used to indicate that a part shall be considered under certification aspects
  EXAMPLE Prior to the release of a new car to the market, both function and quality of certain parts have to be certified by an authority, e.g., the department of transportation. For such item objects, certification requirements have to be considered during the design phase.
- **assembly**: this type of classification shall be used for any part that has an *AssemblyDefinition* provided for at least one of its versions, i.e., it is decomposed further;
- **boxed**: specifies the role of the boxed part;
  EXAMPLE A can with one litre of a specific oil.
- **collection**: this type of classification shall be used for any part that has a *CollectionDefinition* provided for at least one of its versions;

- **completely knocked down**: this type of classification is used to indicate that a part is used in a production site that has assembling facilities only; EXAMPLE The 'completely knocked down' may indicate that the components are shipped to and assembled in a different country.
- **continuous**: this type of classification is used to indicate the part that can be measured by its volume or weight;
- **detail**: this type of classification shall be used for any part that has no *AssemblyDefinition* provided for any of its versions, i.e., it is not further decomposed;
- **discrete**: this type of classification is used to indicate that a part is one single piece.
- **in process**: this type of classification is used to indicate that the part identifies an intermediate object in a manufacturing process; NOTE Detailed information about the stage the part has within the manufacturing process may be obtained from the *ProcessOperationOccurrence*.
- **inseparable assembly**: the part plays the role of an inseparable assembly;
- **prototype**: this type of classification is used to indicate that the part identifies a prototype and is not intended for serial production;
- **raw material**: the part plays the role of raw material;
- **raw material by area**: the part plays the role of raw material, cut by area, e.g. chipboard;
- **raw material by length**: the part plays the role of raw material, cut by length, e.g. cable;
- **raw material by volume**: the part plays the role of raw material, cut by volume, e.g. a raw block of marble or oil;
- **regulated**: this type of classification is used to indicate that for a part certain regulations have to be considered;
- **safety**: this type of classification is used to indicate that a part is relevant for safety purposes;
- **service**: this type of classification is used to indicate that a part is relevant for service purposes;
- **tool**: the part plays the role of a tool.

2.2. Part Categories specific for Electrical Harness and other electrical applications

Here the pre-defined categories for electrical harness and other electrical applications. Most of these categories are standardized by IEC in the Electropedia: http://www.electropedia.org/

Preprocessors are required to provide these categories if they apply. This is needed so that a post-processor can rely on the incoming information and sort the parts suitable for the receiving system.

- **wiring harness**: assembly with a harness topology, consisting of cables or wires to enable electrical or optical connectivity, grouped together in one or more harness segments, each between two harness nodes in which the cables or wires either switch to other harness nodes or in which the cables and wires ends in connectors, contact members or terminal lugs
- **wire** [IEV ref 151-12-28]: flexible cylindrical conductor, with or without an insulating covering, the length of which is large with respect to its cross-sectional dimensions
  Note – The cross-section of a wire may have any shape, but the term "wire" is not generally used for ribbons or tapes.
- **cable** [IEV ref 151-12-38]: assembly of one or more conductors and/or optical fibres, with a protective covering and possibly filling, insulating and protective material
- **flat cable** [IEV ref 461-06-05]: multi-core cable having cores or groups of cores arranged in parallel flat formation

- **connector kit**: collection of connector parts that are intended to be assembled together and that contain at least one connector housing and that may contain alternative parts that may or may not be used in the final assembly
- **contact member** [IEV ref 151-12-16]: conductive element intended to make an electric contact
- **connector contact**: contact member that is intended to be contained in a connector
- **connector** [IEV ref 151-12-19]: device providing connection and disconnection to a suitable mating component
  Note – A connector has one or more contact members.
- **plug** [IEV ref 151-12-21]: connector attached to a cable
- **socket** [IEV ref 151-12-20]: connector attached to an apparatus, or to a constructional element or alike
  Note – Contact members of a socket may be socket contacts, pin contacts or both.
- **shielded connector** [IEV ref 581-26-19]: connector designed to prevent the radiation of electromagnetic interference to and from the internal conductor(s)
- **cavity plug**: plug for a connector cavity for the purpose of sealing
- **seal**: mechanical object that helps join other mechanical objects together by preventing leakage, containing pressure, or excluding contamination
- **connector housing** [IEV ref 581-27-10]: part of a connector into which the connector insert and contacts are assembled
- **emi connector housing**: connector housing that shields against electromagnetic interference
- **connector insert** [IEV ref 581-27-11]: insulating element designed to support and position contacts in a connector housing
- **backshell**: connector accessory that is closing a connector from the back side and guide the wires and cables
- **electrified backshell**: backshell that is intended to be conductive
- **cable support sleeve** [IEV ref 581-27-23]: flexible accessory or a part of a component placed around the cable to minimize flexing of the cable at the point of entry into the component
- **grommet** [IEV ref 581-27-19]: part of a component or an accessory, used to support and protect the wires or cable at the point of entry; it may also prevent the ingress of moisture or contaminants
- **strain relief accessory**: connector accessory to guide and provide strain relief to wires and cables
- **protective covering**, sheat (North America jacket) [IEV ref 461-05-03]: uniform and continuous tubular covering of metallic or non-metallic material, generally extruded
  Note – The term sheath is only used for metallic coverings in North America, whereas the term jacket is used for non-metallic coverings.
- **overbraid**: protective covering (sheath) that is also a braid
- **conduit** [IEV ref 442-02-03]: a part of a closed wiring system of generally circular cross section for insulated conductors and/or cables in electrical or communication installations, allowing them to be drawn in and/or replaced
- **corrugated conduit** [IEV ref 442-02-06]: a conduit in which the profile is corrugated in the longitudinal section
  Note – Both annular and helical corrugated conduits are permissible and a combination of both corrugated and plain conduit is possible.
- **shield** (of a cable) [IEV ref 461-03-04]: surrounding earthed metallic layer which serves to confine the electric field within the cable and/or to protect the cable from external electrical influence

Note 1 – Metallic sheaths, foils, braids, armours and earthed concentric conductors may also serve as shields.
Note 2 – In French, the term "blindage" may be used when the main purpose of the screen is the protection from external electrical influence.

- **terminal lug** [IEV ref 461-17-01]: metallic device to connect a cable conductor to other electrical equipment
- **terminal block** [IEV ref 581-26-26]: part of a component or an accessory, used to support and protect the wires or cable at the point of entry; it may also prevent the ingress of moisture or contaminants
- **joint ferrule**, through connector (of cables) [IEV ref 461-17-04]: metallic device for connecting two consecutive lengths of conductor
- **cable ferrule** [IEV ref 581-27-18]: accessory in the form of a short tube to provide cable support or termination for a cable screen
- **shield connector**, screen connector [IEV ref 461-17-12]: device used to make a connection to the screen or shield of a cable for the purpose of continuity or earthing
- **insulation piercing connector** [IEV ref 461-11-08]: connector in which electrical contact with the conductor is made by metallic protrusions which pierce the insulation of the cable core
- **splice** [IEV ref 581-24-19]: connecting device with barrel(s) accommodating conductor(s) with or without additional provision to accommodate and secure the insulation
- **fixing device** [IEV ref 442-02-40]: system component specifically designed to secure other components to the wall, ceiling, floor or other structure
- **cable fixing device**: fixing device for a cable on a structure
- **label**: part that is intended to attach written information to other parts
- **tape** [IEV ref 212-15-03]: sheeting or plastic film of limited width and in long continuous lengths
  Note – The width is typically less than some hundred millimetres.
- **end cap** [IEV ref 461-20-07]: device placed on the ends of a cable to prevent the ingress of moisture during storage, transportation and installation
- **connector cavity**: cavity in a connector, connector housing or insert intended to receive a connector contact or multi contact

3. PartView and ViewContext

The meaning and structure for Part, PartVersion, PartView with reference to ViewContext and also how to associate a geometric model is explained in great detail in the AP242 BO-Model XML Assembly Structure document and so we do not need to repeat this here.

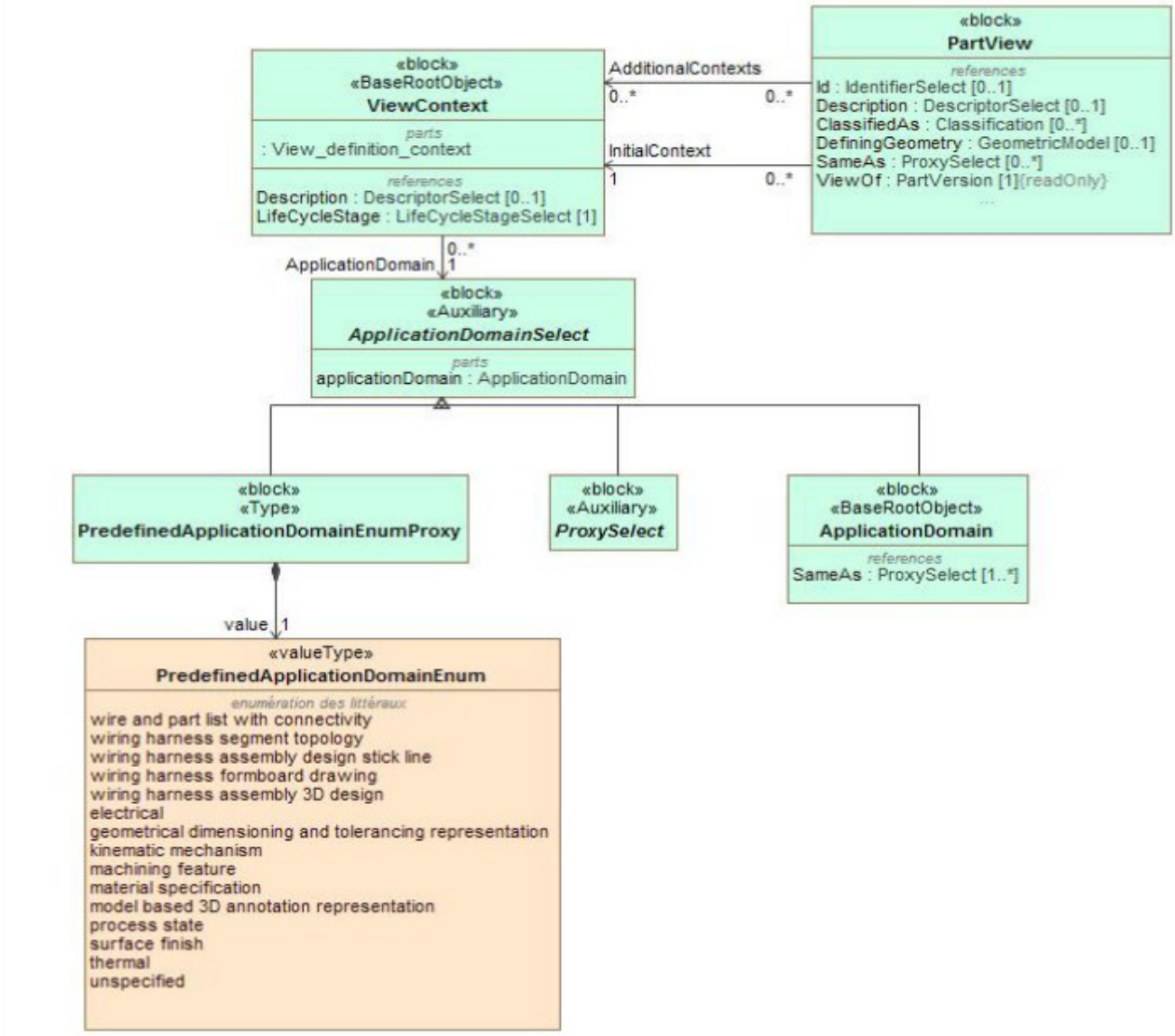

Figure 4: Data model for PartView and ViewContext

Example: The Part “terminal lug” used in the reference example; incomplete

```
<ViewContext uid="_100102">
   <ApplicationDomain>
      <ProxyString>electrical design</ProxyString>
   </ApplicationDomain>
   <LifeCycleStage>
      <ProxyString>design</ProxyString>
   </LifeCycleStage>
</ViewContext>

<Part uid="_103000">
   <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>terminal lug</PartCategoryEnum>
   </PartTypes>
   <Versions>
     <PartVersion uid="_103001">
       <Id id="Version 1"/>
       <Views>
         <PartView uid="_103002">
           <DefiningGeometry uidRef="_103090"/>
           <InitialContext uidRef="_100102"/>
           ...
         </PartView>
       </Views>
     </PartVersion>
   </Versions>
</Part>
```

The specifics for electrical harness, compared to the Assembly Structure Recommended Practises, is explained in the following sub-clauses:

3.1. ViewContext: ApplicationDomain

A *PartView* shall have at least one *ViewContext* for the *InitialContext* and may have further *ViewContexts* as *AdditionalContexts*. For the design of an electrical harness all these *ViewContexts* shall have the value "design" for the *LifeCycleStage* attribute.

In the recommended practises for assembly design an example of *ViewContext* with the value **"mechanical design"** for the *ApplicationDomain* is provided. The expectation for *PartViews* that call out such a *ViewContext* as either initial or additional context is that these *PartViews* also provide either a 2D or a 3D geometric model that represents the part.

For the purpose of the design of an electrical harness the *ApplicationDomain* **"electrical design"** shall be used for all *PartViews* that define elements relevant for electrical connectivity such as electrical terminals and nodes. When this *ApplicationDomain* is used for a *PartView* it tells the receiver of the data that all relevant electrical connectivity information is provided. The *ApplicationDomain* "electrical design" may be used as either *InitialContext* or *AdditionalContexts*.

As detailed in the assembly recommended practises, an *AssemblyDefinition* is a specialization of a *PartView* that is to be used to define an assembly of parts. In the case that the assembly of an electrical harness is defined, the subtype of ***WiringHarnessAssemblyDesign*** shall be used . The following values for the *ApplicationDomain* of additional contexts for a *WiringHarnessAssemblyDesign* are defined:

- *wire and part list with connectivity*: the application domain comprises all wires and wire caps, cables, connectors and associated material, contact members, terminal lugs, terminal blocks, splices and electrically non conductive through termination, electrical shielding definition in the assembly, tape marking of a wiring harness assembly.
- *wiring harness segment topology*: the application domain comprises the complete harness topology of a wiring harness assembly.
- *wiring harness assembly design stick line*: the application domain comprises a stick line drawing of a wiring harness assembly.
- *wiring harness formboard drawing*: the application domain comprises a formboard drawing of a wiring harness assembly.
- *wiring harness assembly 3d design*: the application domain comprises a 3D geometry model of the harness topology of a wiring harness assembly.

3.2. Parts used in the reference example

Table 1 lists all the Part used in the reference example, together with the used PartVersion, PartView and part category.

Part	Version	View	Id	Name	Category/type
_101000	_101001	_101002	04034-22-9	WIRE,ELEC,COMP,SNGL CONDUCTOR,150 DEG C	wire, raw material by length
_102000	_102001	_102002	RG 58	RG 58	cable, raw material by length
_103000	_103001	_103002	MS5036-153	TERMINAL LUG CRIMP STYLE COPPER INSULATED RING TONGUE	terminal lug
_104000	_104001	_104002	16 AWG	Speaker wire	cable, raw material by length
_110000	_110001	_110002	SB6 4 1 M G 05 W2 P E1 01 AA	ARINC 600 set	connector kit
_111000	_111001	_111002	8660-31A-100-01A/AA	ARINC 600 shell 1 A and B dummy	connector insert
_112000	_112001	_112002	8660-5W2	ARINC 600 shell 1 C 5W2 insert	connector insert
_113000	_113001	_113002	8660-2485	#5 Coax contact	connector contact
_114000	_114002	_103090	8660-249	#16 Rack plug power contact	connector contact
_115000	_115001	_115002	8660-5W2x1	Insert 5W2 assembly	connector insert
_116000	_116001	_116002	8660-250	#12 Rack plug signal contact	connector contact
_117000	_117001	_117002	TM2PB	Phone connector 6.35mm	connector
_118000	_118001	_118002	M81824	Butt Splice Terminal	splice
_119000	_119001	_119002	DEP09S065TLF	D-sub 9 Pin Db9 Female Solder Type Socket Connector	connector
_120000	_120001	_120002	TC122	Braid 1/2inch	overbraid
_121000	_121001	_121002	F6W1.50BK	Wrap	protective covering
_122000	_122001	_122002	H4N1.00BK	HeatShrink	protective covering
_123000	_123001	_123002	8660-140	Backshell EMI connector	electrified backshell
_311000	_311001	_311002	Part_H1	Electrical Harness team reference example	wiring harness
_411000	_411001	_411002	Part_H2	Electrical Harness example 2 (minmal)	wiring harness
_511000	_511001	_511002	Aircraft99x	Aircraft99x	

Table 3: Parts used in the example

4. Part Occurrence, and Multi-level Assembly Structure

AP242 allows to build up single and multi level assembly structures. Each assembly, either a top assembly or an intermediate sub-assembly, is represented by an *AssemblyDefinition* that is a subtype of *PartView*. The "components" of an *AssemblyDefinition* are subtypes of *Occurrence* that are related into the assembly through *AssemblyOccurrenceRelationships*. Its supertype *ViewOccurrenceRelationship* allows in a generic way to relate an *Occurrence* to a *PartView*.

In cases that a set of Occurrences belong together but are not physically combined with each other a *CollectionDefinition* is to be used, that is also a subtype of PartView.

An AssemblyDefinition is the relating target of an AssemblyOccurrenceRelationship that is a subtype of ViewOccurrenceRelationship that in a generic way establishes a relation between an Occurrence and a PartView. An AssemblyOccurrenceRelationship is either a NextAssemblyOccurrenceUsage or a PromissoryAssemblyOccurrenceUsage. A NextAssemblyOccurrenceUsage makes an Occurrence a direct constituent of an AssemblyDefinition. A PromissoryAssemblyOccurrenceUsage only indicates that an Occurrence may become an constituent in some way, maybe in some indirect way.

An Occurrence may be defined in three different ways:

1. In most cases an Occurrence is a particular usage of a PartView that inherits all its attributes including the geometric models from the PartView unless it is said otherwise.
2. But an Occurrence may also be a usage of a ProductConfiguration that is a member of a ProductClass. This case is used especially for standard parts that are bought off the shelf with only limited design information.
3. A third case is when an Occurrence is defined by a lower level Occurrence in the context of a higher level Occurrence. The specific subtype SpecifiedOccurrence is used in this case, see below and clause 5.6.

Occurrence itself is an abstract concept. One of the following subtype has to be used:

- A DefinitionBasedOccurrence is an Occurrence that is also abstract and that has as its definition either a PartView or a ProductConfiguration (but not another Occurrence).
    - A SingleOccurrence is a DefinitionBasedOccurrence that is taken from its definition as a single piece.
    - A QuantifiedOccurrence is a DefinitionBasedOccurrence that is taken from its definition in some quantity. The quantity can be defined in the number of pieces (e.g. 10 rivets), or by length (e.g. 2m wire), or by surface (e.g. 3 sqm. of glass) or by volume (2.5 litre of oil). It the quantity can not be specified exactly an alternative criterion can be specified (e.g. enough glue for the whole surface). At least either the quantity or the criterion must be specified.
- A SpecifiedOccurrence is an Occurrence that has as its definition another Occurrence. This definition is used by a NextAssemblyUsage in an AssemblyDefinition. In parallel the SpecifiedOccurrence calls out by the attribute UpperUsage one more other Occurrence, that has as its Definition (directly or indirectly) this AssemblyDefinition.

A single or multi level assembly structure shall form a directed acyclic graph. *ProductConfiguration*s, *PartView*s, *AssemblyDefinition*s and *Occurrence*s are the nodes, while *AssemblyOccurrenceRelationship*s, and the attribute *Occurrence.Definition* are the links between the nodes in this graph. Other than that this graph shall be "directed" and "acyclic" the following constraints apply:

- An Occurrence can be related at most once into an AssemblyDefinition, but an Occurrence can be related several times into different AssemblyDefinitions; e.g. to represent variants.
- In the case that an Occurrence is defined by a PartView, this PartView must not be the AssemblyDefinition into which the Occurrence is related to nor any of its higher AssemblyDefinitions.

- PartViews or AssemblyDefinitions at different levels of an assembly structure shall not share the same PartVersion nor the same Part.

See the “Recommended Practices for AP242 Business Object Model XML Assembly Structure” for detailed information on how to implement assembly structures in general.

Clause 5.6 provides a detailed example of a hierarchical assembly structure with SpecifiedOccurrence.

Attention: Do not use the types AssemblyViewRelationship, NextAssemblyViewUsage and PromissoryAssemblyViewUsage for electrical harness. These types had been introduced in AP242 for the harmonization with AP239 (PLCS), but they do in no way support the functionality needed for electrical harness.

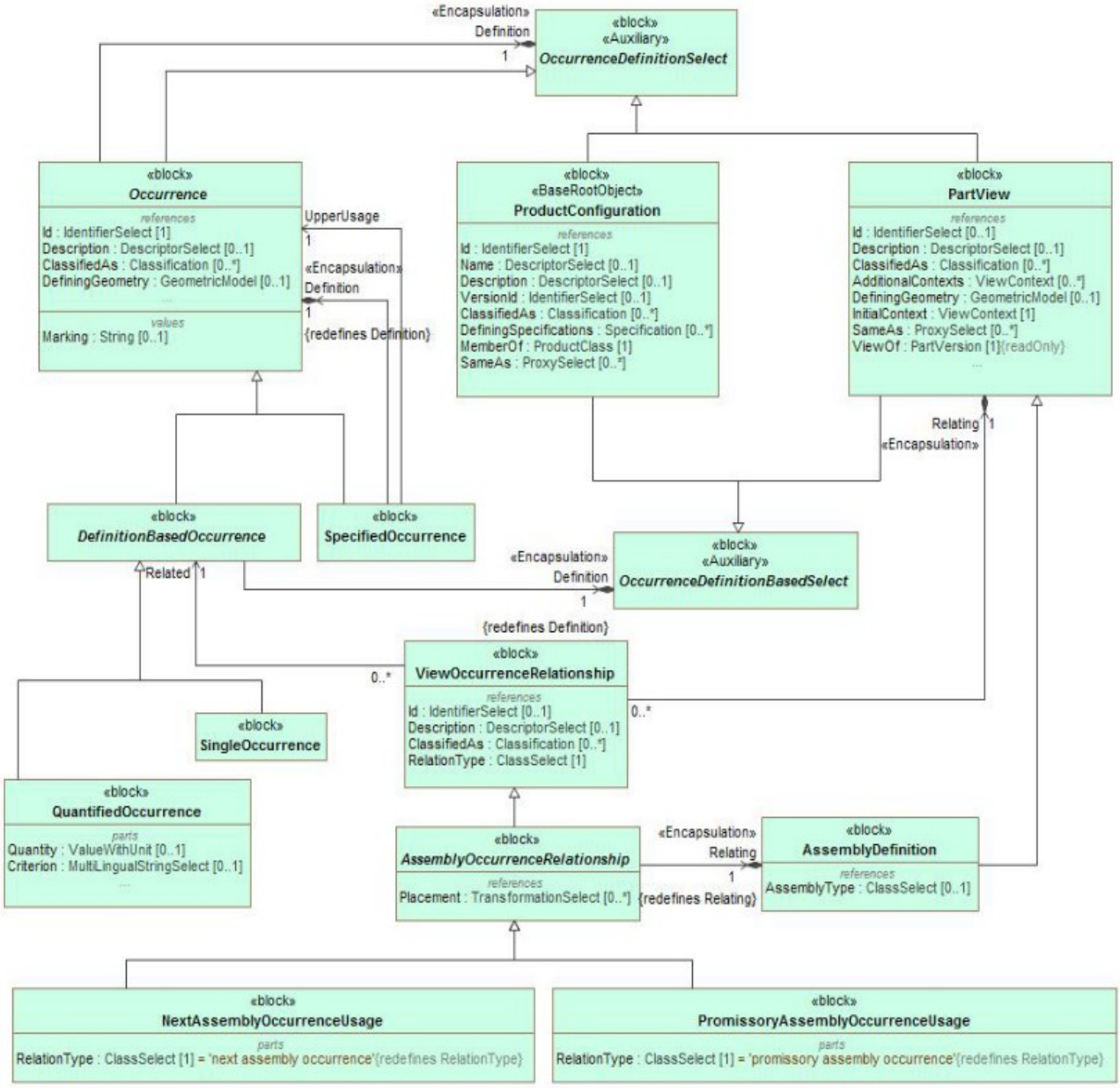

Figure 5: Detailed assembly structure model

The attribute Id of Occurrences shall be unique within all Occurrences used within an AssemblyDefinition. It is used as reference designator. In most cases a simple IdentifierString is sufficient. If there is a need to call out a specific reference designator system (e.g. IEC, IEEE or company specific) the Identifier Object with a Classification can be used.

See an overview of “IEC Reference Designations” at:
http://myelectrical.com/notes/entryid/24/iec-reference-designations

The attribute Id of ViewOccurrenceRelationship and all its subtypes is optional. Recommendation is to not use this attribute.

5. Electrical Connectivity & AssemblyJoint

The electrical connectivity in AP242 follows the so called “CFI five-box model of electrical connectivity”, defined originally by the CAD Framework Initiative (CFI), see
http://www10.edacafe.com/book/ASIC/CH09/CH09.5.php

The CFI model is used in AP242 for both, functional and physical connectivity. As this tutorial focuses on the physical design aspects of an electrical harness, only the physical connectivity aspects are presented here.

5.1. Adopted CFI Five-Box Model for Electrical Connectivity

Adopting the CFI five-box model to the modelling style and terminology used in AP242-XML and applying several simplifications the model would looks like this:

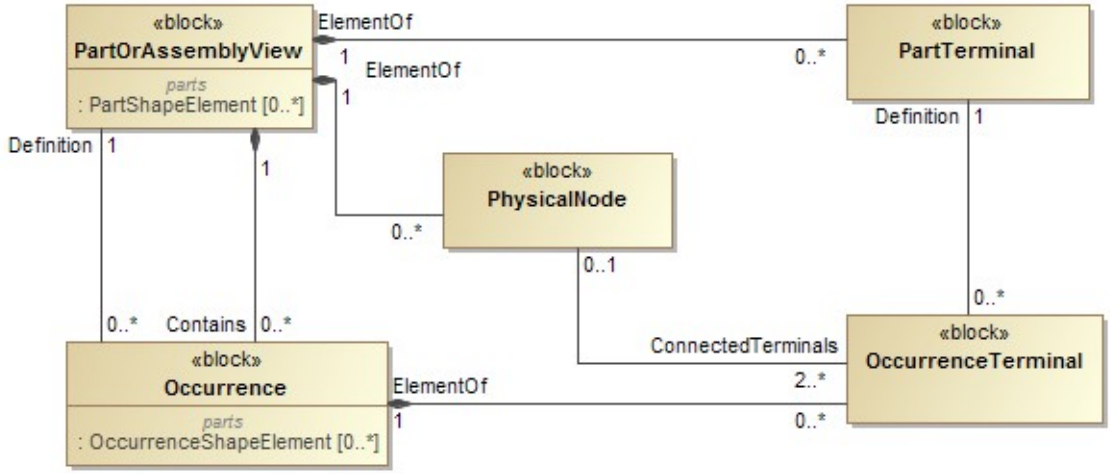

Figure 6: Simplified adopted CFI five-box model for physical connectivity

- A *PartView* represents any kind of a physical design and all the underlying parts it is made of. So the design of an electrical harness is a part and also all the connectors, cables and other things it is made of are represented as *PartView*s.
- A *PartTerminal* represents some area of a *PartView* where it can be electrical or optical connected; e.g. one of the inside or outside terminals of a connector.
  A *PartTerminal* belongs to exactly one *PartView.*
  A *PartView* may have zero, one or many *PartTerminal*s
- An *Occurrence* is a particular usage of a *PartView* and is therefore defined by one *PartView.*
  A *PartView* may serve as the definition for many *Occurrence*s.
  A higher level *PartView* may “contain” *Occurrence*s of lower level *PartView*s In STEP the subtype AssemblyView is to be used in this case. The structure is recursive and so also occurrences of higher level PartView can be constructed that may be contained in an even higher level PartView and so on. PartViews and Occurrences form a Direct Acyclic Graph, so a PartView can not contain Occurrences of itself.
  Example: An electrical harness PartView that contains two Occurrences of a connector.
  Note: This simplified model assumes that an Occurrence is ues in exactly one Part/AssemblyView. In STEP an Occurrence can be used in zero, one or many Part/AssemblyViews.
- An OccurrenceTerminal represents some area of an Occurrence where it can be connected.
  An OccurrenceTerminal belongs to exactly one Occurrence.
  An Occurrence can have zero, one or many OccurrenceTerminals
  An OccurrenceTerminal is defined by one PartTerminal from which it inherits all characteristics.

A PartTerminal may serve as the definition for many OccurrenceTerminals.

- A PhysicalNode represents a node of one or several electrical connected elements. It is equivalent to the concept of a Kirchhoff node.
  A PhysicalNode belongs to one PartView.
  A PartView may have zero, one or many PhysicalNodes,
  A PhysicalNode connects two or more OccurrenceTerminals. These terminals must belong to Occurrences that are contained on the PartView of the node. It is not allowed to connect OccurrenceTerminals of Occurrences that are not used in the PartView.
  An OccurrenceTerminal can be used by at most one PhysicalNode for the same PartView.

The following clauses explain in detail how the simplified CFI model is realised in AP242.

5.2. Adopted CFI Five-Box Model for any Joints in an Assembly

AP242 also contains a variation of the CFI five-box model to represent any kind of mechanical connectivity in an assembly (figure 7). This is based on the entities PartShapeElement and OccurrenceShapeElement that are supertypes of PartTerminal respectively OccurrenceTerminal. An AssemblyJoint is itself a PartShapeElement and this belongs to one PartView that is an assembly. An AssemblyJoint connects in a binary way two OccurrenceShapeElements. So an AssemblyJoint plays a similar role as a PhysicalNode with the more generic supertypes PartShapeElement and OccurrenceShapeElement.

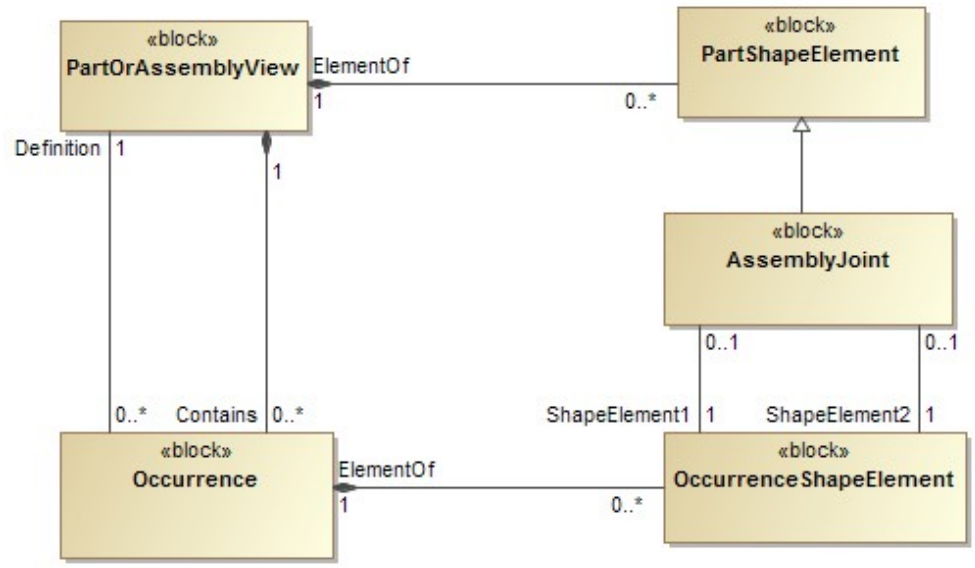

Figure 7: Simplified adopted CFI five-box model for joints in an assembly

Figure 8 shows an example of the CFI 5 box model for an assembly of two occurrences of a part with 2 terminals.

- top: PartView A with two PartTerminals A1 and A2
- middle: Aa and Ab of PartView A. Each has two OccurrenceTerminals that are defined by the PartTerminals of PartView A.
  Occurrence Aa has the OccurrenceTerminals Aa1 and Aa2.
  Occurrence Ab has the OccurrenceTerminals Ab1 and Ab2.
- bottom: AssemblyView B with two AssemblyJoints.
  AssemblyJoint Bx joints the OccurrenceTerminals Ab1 with Aa2.
  AssemblyJoint By joints the OccurrenceTerminals Ab2 with Aa1.

OccurrenceTerminals are essential to exactly specific which terminals are connected or joined together. The same is true for the more generic supertype OccurrenceShapeElement.

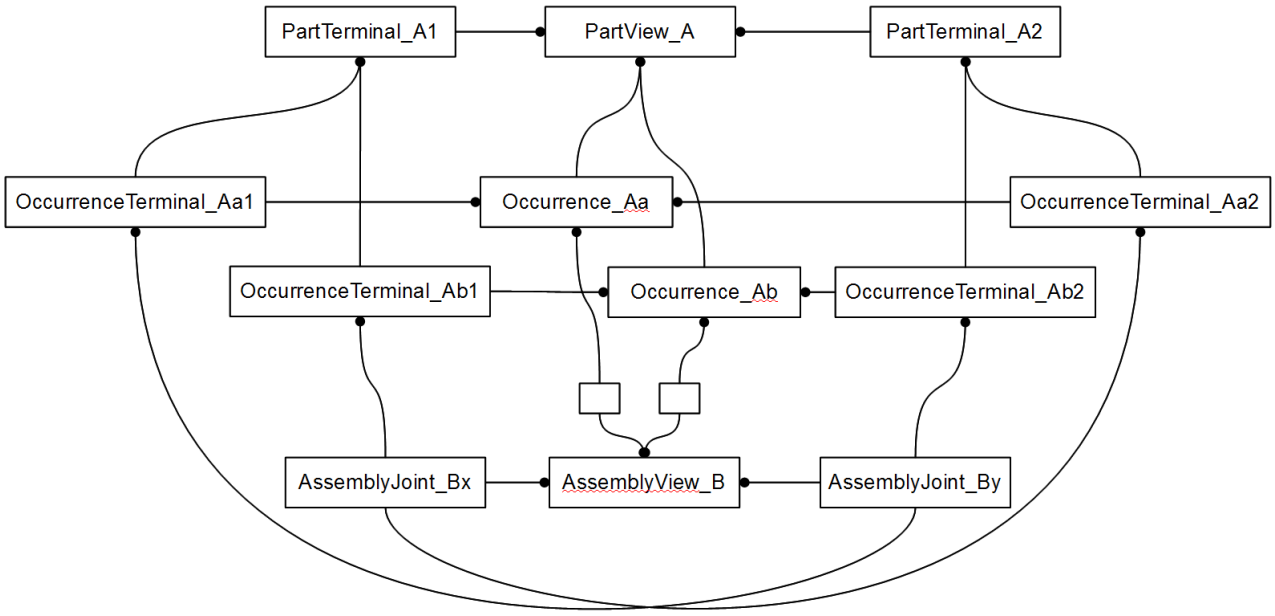

Figure 8: Example of adopted CFI 5 box pattern for AssemblyJoint of terminals

5.3. Detailed ShapeElement and AssemblyJoint Model

This clause explains in detail how two *OccurrenceTerminals* within an assembly are joined together, e.g. an end of a wire with an "internal" terminal of a connector, or two "external" terminals of two connectors.

While in the previous clauses we showed pieces of the AP242 connectivity model in a rather simplified way to highlight the principles, we will now look to it in more detail.

As stated earlier *OccurrenceShapeElement* and *PartShapeElement* are both subtypes of *ShapeElement*. A *PartShapeElement* represents an element of the shape of a *PartView*, while an *OccurrenceShapeElement* represents an element of the shape of an *Occurrence*. To enforce this the *ElementOf* attribute of a *PartShapeElement* must call out a *PartView*, while the *ElementOf* attribute of an *OccurrenceShapeElement* must call out an *Occurrence*.

Note: ElementOf of a ShapeElement might also refer to an AssemblyViewRelationship, BreakdownElementView or IndividualPartView, but these cases are not addressed in this tutorial.

An *OccurrenceShapeElement* calls out as its *Definition* a *PartOrOccurrenceShapeElementSelect* that is either a *PartShapeElement* or an *OccurrenceShapeElement*. There is a constraint on the combination of the values for *ElementOf* and *Definition*. If and only if *ElementOf* has a value of type *SpecifiedOccurrence* the *Definition* attribute must have a value of type *OccurrenceShapeElement*.

An *AssemblyJoint* is a *PartShapeElement* that belongs to an *AssemblyDefinition*. In most cases an *AssemblyJoint* connects two *OccurrenceShapeElements* of two different *Occurrences*. These two *Occurrences* must be related into the same *AssemblyDefinition* as the one the *AssemblyDefinition* belongs to.

In some cases there is a need to joint three or more *OccurrenceShapeElements*, e.g. when two or more wires join in a single screw terminal. In this case a tree of two or more *AssemblyJoints* are used. The bottom *AssemblyJoint*(s) in that tree connect two *OccurrenceShapeElement*, while the higher ones in that three connect either two *AssemblyJoints* or an *AssemblyJoint* with an *OccurrenceShapeElement*.

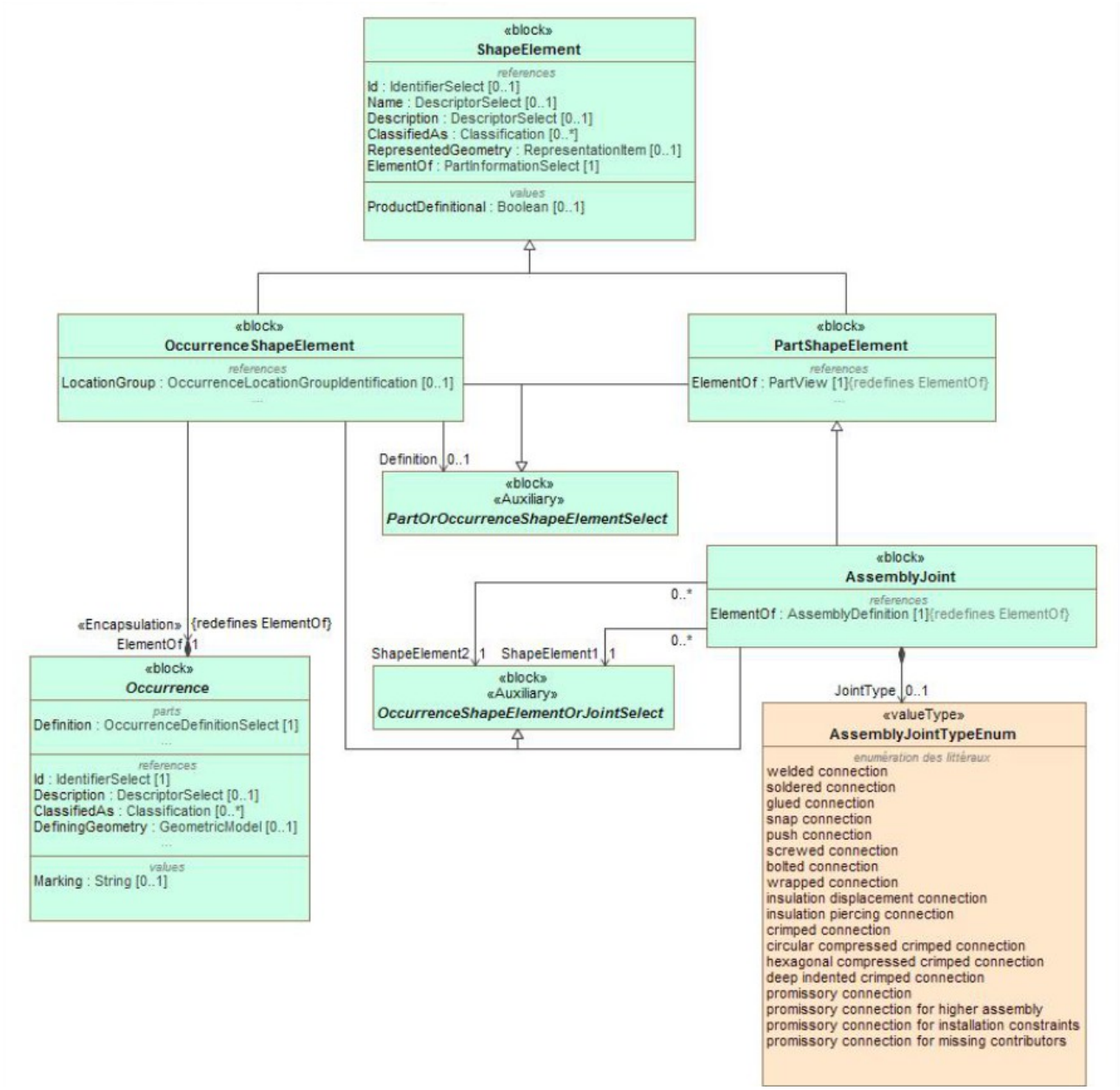

Figure 9: Detailed AssemblyJoint model

For each *AssemblyJoint* the kind of joint can be specified. The following *JointTypes* are pre-defined:

- **welded connection** [IEV ref 581-23-06]: connection made by welding
- **soldered connection** [IEV ref 581-23-04]: connection made by soldering
- **glued connection**: connection made by glueing
- **snap connection**: connection made by snapping
- **push connection**: connection made by pushing
- **screwed connection**: connection made by screwing
- **bolted connection** [IEV ref 461-19-05]: connection in which the pressure to the conductor is applied by bolting
- **wrapped connection** [IEV ref 581-23-07]: solderless connection achieved by wrapping a solid conductor around a wrap post

- **insulation displacement connection** [IEV ref 581-23-35]: solderless connection made by inserting a single wire into a slot in an insulation displacement termination
- **insulation piercing connection** [IEV ref 461-19-06]: connection made by metallic protrusions which pierce the insulation of the cable core
- **crimped connection** [IEV ref 461-19-01] permanent connection made by the application of pressure inducing the deformation or reshaping of the barrel around the conductor of a cable
  Note – In some cases, the deformation or reshaping of the barrel may affect the form of the conductor.
- **circular compressed crimped connection** [IEV ref 461-19-02]: crimp connection in which the barrel is compressed maintaining essentially its circular form
- **hexagonal compressed crimped connection** [IEV ref 461-19-03]: crimp connection in which the barrel is compressed and reshaped essentially to a hexagonal form
- **deep indented crimped connection** [IEV ref 461-19-04]: crimp connection in which the barrel and the cable conductor are deformed by deep indentations
- **promissory connection**: connection that is promised or expected to be established in some way, but that is by purpose not realised for this PartView
- **promissory connection for higher assembly**: promissory connection that will be realised in a higher assembly in which this assembly is an Occurrence
- **promissory connection for installation constraints**: the connection can not be realised on this assembly level because of installation constraints and this must be realised in a higher assembly
- **promissory connection for missing contributors**: the connection can not be realised on this assembly level because one or more needed OccurrenceShapeElements are not part of this assembly. So the final AssemblyJoint can only be realised in a higher assembly where the other OccurrenceShapeElements become accessible.

The attribute *JointTypes* is optional; it should only be provided if available in the sending system. Also the *JointTypes* does not need to be specified if one of the *JointTypes* of the affected *PartTerminals* that are called out as *Definition* of the *OccurrenceTerminal* is already specific enough. However if both *AssemblyJoint JointTypes* and one or two *PartTerminals.JointTypes* are provided, the *JointTypes* of the *AssemblyJoint* overrides the *JointTypes* of the *PartTerminal(s)*. E.g. if the terminal type is a *crimp terminal*, and this terminal is used for an *AssemblyJoint* with type *solder*, then this means the joint is realised by soldering, not by crimping.

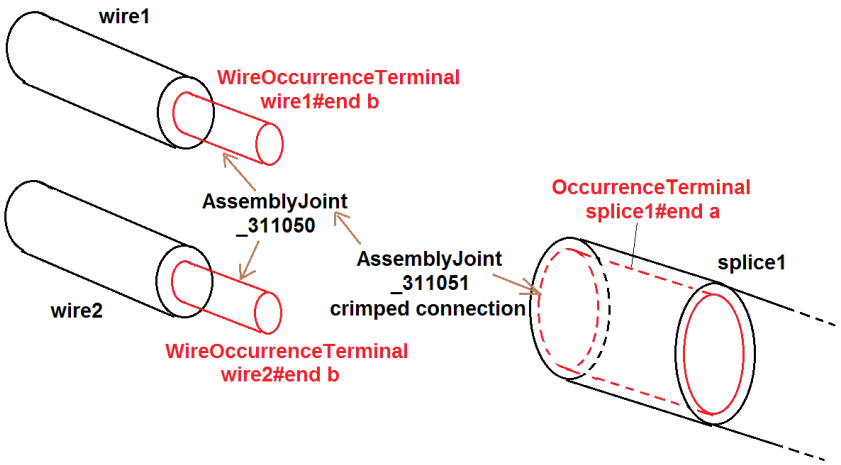

Figure 10: Tree of AssemblyJoints

Example of a two *AssemblyJoint*s for joining three *OccurrenceTerminal*s (figure 10). First the ends of two

wires (wire1 and wire2) are placed together to form a couple by *AssemblyJoint* _311050 that does not specify a particular JointType. The higher level *AssemblyJoint* _311051 is taking this couple and an *OccurrenceTerminal* from splice1 and only this is crimped.

```
<ShapeElement xsi:type="n0:AssemblyJoint" uid="_311050">
  <ShapeElement1 uidRef="_201007"/> <!--wire1#end b-->
  <ShapeElement2 uidRef="_201107"/> <!--wire2#end b-->
</ShapeElement>
<ShapeElement xsi:type="n0:AssemblyJoint" uid="_311051">
  <JointType>crimped connection</JointType>
  <ShapeElement1 uidRef="_311050"/> <!--AssemblyJoint(wire1 & wire2) -->
  <ShapeElement2 uidRef="_218103"/> <!--splice1#end a-->
</ShapeElement>
```

5.4. Detailed PartTerminal Model

This clause explains in detail the concept of *PartTerminal* that identifies a feature of a part that can be connected with others either electrical or optical.

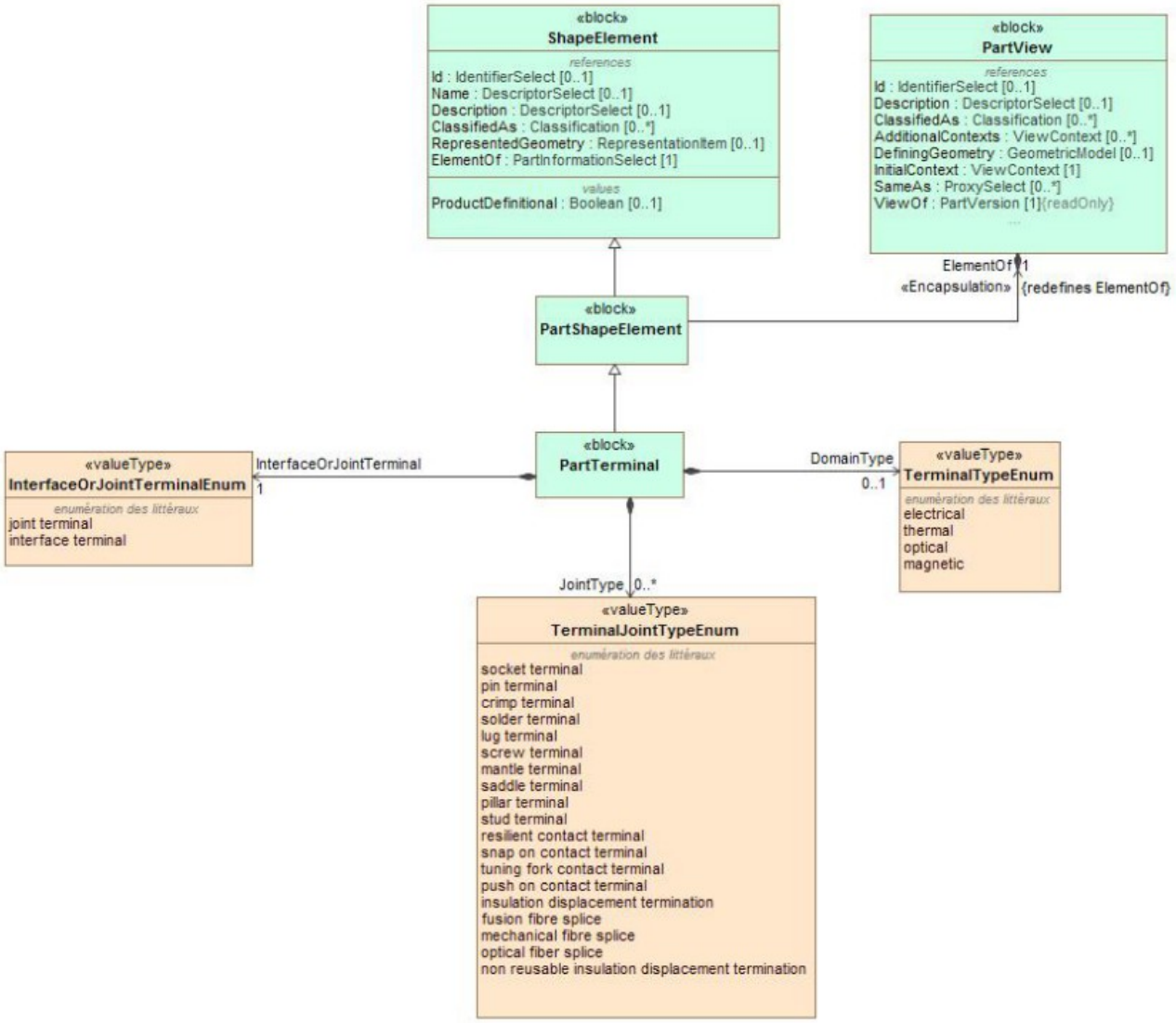

Figure 11: Detailed PartTerminal model

A PartTerminal carries three additional attributes in addition to those it inherits from PartShapeElement:

JointTypes characterises the intended way on how to contact the PartTerminal. Most of the allowed values listed below are taken from IEC definitions. To achieve a systematic naming convention the word “contact” has been replaced by the word “terminal” or the word “terminal” has been added at the end. The terms written in bold characters shall be used. Alternative IEC terms are separated by comma.

- **socket terminal**, socket contact, female contact [IEV ref 151-12-17]: contact member intended to make electric engagement on its inner surface for mating with the outer surface of another contact member
  Note – In English, the term "socket contact" does not imply that socket contacts are always mounted in a socket nor that sockets have only socket contacts.
- **pin terminal**, pin contact, male contact [IEV ref 151-12-18]: contact member intended to make electric engagement on its outer surface for mating with the inner surface of another contact member
- **crimp terminal**, crimp contact [IEV ref 581-22-05]: contact having a conductor barrel designed to be crimped
- **solder terminal** [IEV ref 442-06-20]: a conductive part of a connecting device provided to enable a termination to be made by means of solder
- **lug terminal** [IEV ref 442-06-16]: a screw-type terminal designed for clamping a cable lug or bar directly or indirectly by means of a screw or nut
- **screw terminal** [IEV ref 442-06-08]: a terminal, in which the conductor is clamped under the head of one or more screws, and where the clamping pressure can be applied directly by the head of the screw or through an intermediate part, such as a washer, clamping plate or an anti-spread device
- **mantle terminal** [IEV ref 442-06-14]: a terminal, in which the conductor is clamped against the base of a slot in a threaded stud by means of a nut, by a suitably shaped washer under the nut, by a central peg if the nut is a cap nut, or by an equally effective means for transmitting the pressure from the nut to the conductor within the slot
- **saddle terminal** [IEV ref 442-06-09]: a terminal, in which the conductor is clamped against the base of a slot in a threaded stud by means of a nut, by a suitably shaped washer under the nut, by a central peg if the nut is a cap nut, or by an equally effective means for transmitting the pressure from the nut to the conductor within the slot
- **pillar terminal** [IEV ref 442-06-22]: a screw type terminal, in which the conductor(s) is (are) inserted into a hole or cavity, where it is clamped under the shank of the screw
  Note – The clamping pressure can be applied directly by the shank of the screw or through an intermediate part, to which pressure is applied by the shank of the screw.
- **stud terminal** [IEV ref 442-06-23]: a screw-type terminal in which the conductor is clamped under a nut
  Note – The clamping pressure can be applied directly by a suitably shaped nut or through an intermediate part, such as a washer, a clamping plate or an anti-spread device.
- **resilient contact terminal** [IEV ref 581-22-09]: contact having elastic properties to provide a force to its mating part
- **snap on contact terminal** [IEV ref 581-22-10]: push-on contact in which retention is achieved by means of a deformation of the contact area which provides positive axial location
- **tuning fork contact terminal** [IEV ref 581-22-12]: resilient contact having a shape similar to that of a tuning fork, the two arms of which apply contact force in opposite directions
- **push on contact terminal** [IEV ref 581-27-04]: contact with which a connection is achieved by axial force, separation being restricted by friction
- **insulation displacement terminal**, insulation displacement termination [IEV ref 581-23-39]: termination having slots with precisely controlled sides, which are intended to accept a wire and to

displace its insulation, deform its conductor and to produce a gas-tight solderless connection

- **non reusable insulation displacement termination** [IEV ref 581-23-41]: insulation displacement termination that can be terminated only once
- **fusion fibre splice**, fusion splice [IEV ref 731-05-06]: a splice accomplished by the application of localised heat sufficient to fuse or melt the ends of two lengths of optical fibre, to produce a continuous single optical fibre
- **mechanical fibre splice**, mechanical splice [IEV ref 731-05-07]: a fibre splice accomplished by fixtures or materials, rather than by thermal fusion
- **optical fibre splice** [IEV ref 731-05-05]: a permanent joint whose purpose is to couple optical power between two optical fibres
  Note – Associated terms : to splice, splicing.

DomainType: Either electrical or optical. In principle also the domain types thermal or magnetic are available, but these are not used in practise so far.

InterfaceOrJointTerminal: This indicates on whether a PartTerminal is intended to be connected on the next higher assembly level (e.g. the internal terminals of a connector) or whether the PartTerminals are intended to be used to “interface” on an even higher assembly level (e.g. the external terminals of a connector)

5.5. Detailed OccurrenceTerminal and PhysicalNode Model

This clause explains in detail how “components” of an electrical assembly are connected with each other, either by AssemblyJoints or by *PhysicalNodes* that represents nodes in the sense of the Kirchhoff's laws on electrical circuits.

For the exchange of electrical or optical connectivity information, *PartTerminal*, subtype of *PartShapeElement*, and *OccurrenceTerminal*, subtype of *OccurrenceShapeElement*, are to be used. The *Definition* attribute of *OccurrenceTerminal* is of type *PartOrOccurrenceTerminalSelect* which is either a *PartTerminal* or another *OccurrenceTerminal*. Similar to *OccurrenceShapeElement*, the *Definition* of *OccurrenceTerminal* shall only, and only if, be of type *OccurrenceTerminal* when *ElementOf* has a value of type *SpecifiedOccurrence*.

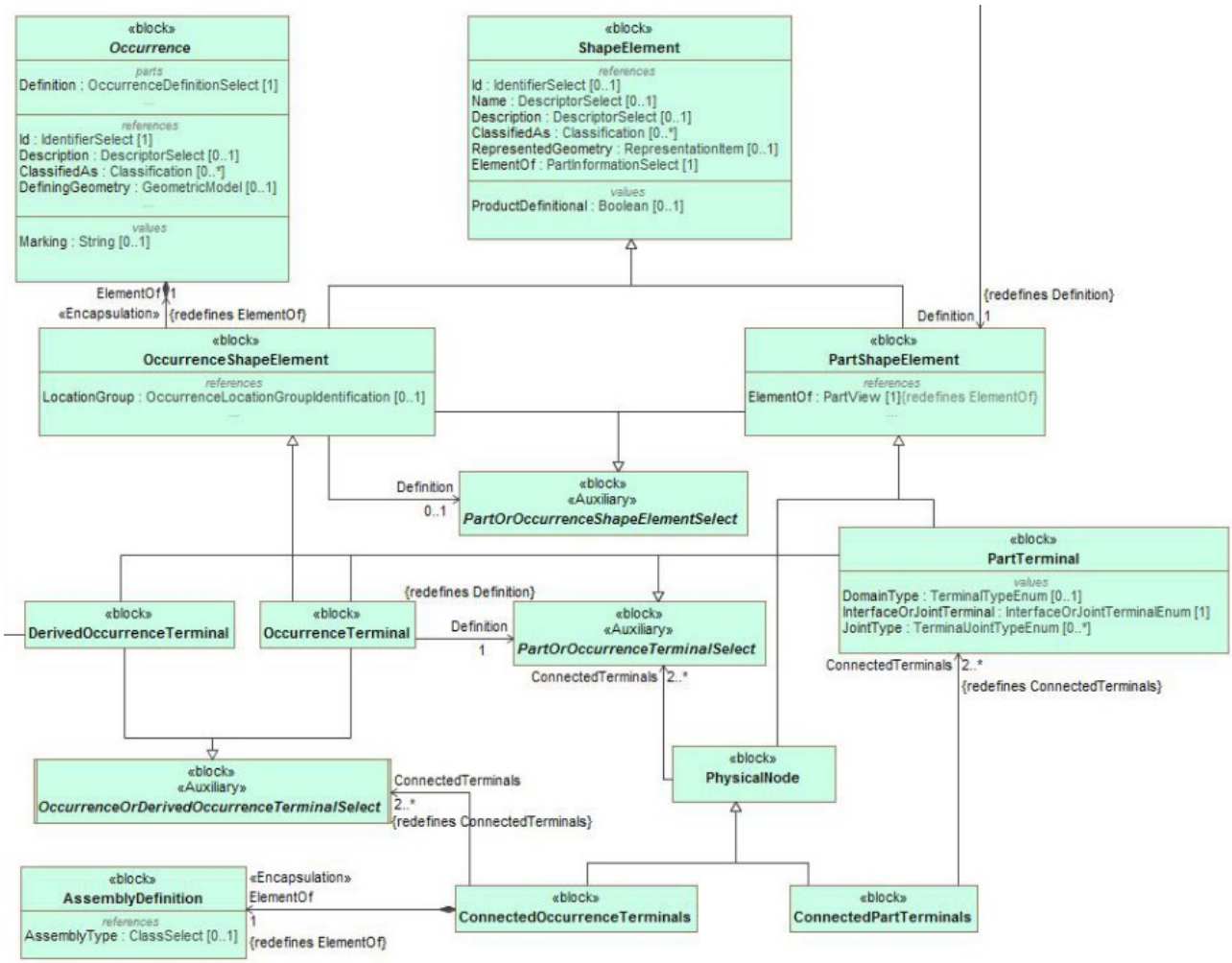

Figure 12: PhysicalNode & OccurrenceTerminal

Like *AssemblyJoint*, *PhysicalNode*, is also a subtype of *PartShapeElement*. It defines in a more generic way either electrical or optical connection between several terminals without providing all the details. This is done through the attribute *ConnectedTerminals* that is of type *PartOrOccurrenceShapeElement*. The following constraints apply:

- If a member of the *ConnectedTerminals* is of type *PartTerminal*, then this *PartTerminal* shall belong to the same *PartView* to which also *PhysicalNode* belongs to
- If a member of the *ConnectedTerminals* is of type *OccurrenceTerminal*, then the *Occurrence* this *OccurrenceTerminal* belongs to shall be related into an *AssemblyDefinition* via a *AssemblyOccurrenceRelationship* that is also the same *AssemblyDefinition* the *PhysicalNode* belongs

to.

Two subtypes of *PhysicalNode* are defined:

- *ConnectedPartTerminals* defines connectivity between several PartTerminals only. This subtype shall be used when *ElementOf* points to a *PartView* that is not an *AssemblyDefinition* and so there are no *OccurrenceTerminals* among the *ConnectedTerminals*.
- *ConnectedOccurrenceTerminals* defines connectivity between several *OccurrenceTerminals* only. So this subtype can only be used when *ElementOf* refers to an *AssemblyDefinition*. This subtype shall be used to indicate that the node is internal to the part and not visible to the outside.
- The supertype *PhysicalNode* shall be used when *ElementOf* refers to an *AssemblyDefinition* and the *ConnectedTerminals* attribute contains at least one *PartTerminal* and contains (or may contain) at least one *OccurrenceTerminals*. So an inner node is accessible from the outside.

Example:

A PartView contains two PartTerminals whose internal connection is stated by a ConnectedPartTerminals.

```
<PartView uid="_103002">
  <DefiningGeometry uidRef="_103090"/>
  <InitialContext uidRef="_100102"/>
  ...
  <ShapeElement xsi:type="n0:PartTerminal" uid="_103003">
    <Id id="External"/>
    <RepresentedGeometry uidRef="_103092"/>
    <InterfaceOrJointTerminal>interface terminal</InterfaceOrJointTerminal>
    <JointType>
      <TerminalJointTypeEnum>screw terminal</TerminalJointTypeEnum>
    </JointType>
  </ShapeElement>
  <ShapeElement xsi:type="n0:PartTerminal" uid="_103004">
    <Id id="Internal"/>
    <RepresentedGeometry uidRef="_103094"/>
    <InterfaceOrJointTerminal>join terminal</InterfaceOrJointTerminal>
    <JointType>
      <TerminalJointTypeEnum>crimp terminal</TerminalJointTypeEnum>
    </JointType>
  </ShapeElement>
  <ShapeElement xsi:type="n0:ConnectedPartTerminals" uid="_103007">
    <ConnectedTerminals>
      <PartTerminal uidRef="_103003"/>
      <PartTerminal uidRef="_103004"/>
    </ConnectedTerminals>
  </ShapeElement>
</PartView>
```

5.6. Referencing Terminals in Hierarchical Assembly Structures

So far we explained only on how to use *AssemblyJoints* and *PhysicalNodes* within a single assembly level as these concepts can refer only to *OccurrenceShapeElements* respectively *OccurrenceTerminals* within the same assembly level. But what to do when the *AssemblyJoints* and *PhysicalNodes* need to span over a multi level assembly structure?

The answer is to use *SpecifiedOccurrence* to make *OccurrenceShapeElements*/*OccurrenceTerminals* from a lower assembly level visible on a higher assembly level.

A *SpecifiedOccurrence* has as its *Definition* another *Occurrence* (or subtype *SpecifiedOccurrence*) and calls out with its attribute *UpperUsage* a higher level *Occurrence*, and thus making the lower level *Occurrence* visible in the higher level *Occurrence*. In a similar way the *OccurrenceShapeElements* and *OccurrenceTerminals* from the lower level *Occurrence* can be reflected for the higher level Occurrence by creating copies of *OccurrenceShapeElements* and *OccurrenceTerminals* that refer with their *Definition*

attributes to the lower level *OccurrenceShapeElements* and *OccurrenceTerminals*.

Here an example on how this mechanism works through an assembly hierarchy with four levels:

- The top assembly “Aircraft99x” contains an occurrence “h1.1” that is a “Part_H1”
- The assembly “Part_H1” contains an occurrence “arinc1” that is a “ARINC 600 set”
- The assembly “ARINC 600 set” contains an occurrence “C-Assy” for the insert at position C that is a “5W2 assy”
- The assembly “5W2 assy” contains an occurrence “power3” that is a “#16 Rack plug power contact”

A series of SpecifiedOccurrences flatten this hierarchical assembly structure:

- “h1.1/arinc1” represents the “arinc1” occurrence for the higher occurrence “h1.1”
- “arinc1/C-Assy” represents the “A-Assy” occurrence for the higher occurrence “arinc1”
- “h1.1/arinc1/C-Assy” represents the “arinc1/A-Assy” occurrence for the higher occurrence “arinc1/C-Assy”
- “C-Assy/power3” represents the “power3” occurrence for the higher occurrence “C-Assy”
- “arinc1/C-Assy/power3” represents the “C-Assy/power3” occurrence for the higher occurrence “arinc1”
- “h1.1/arinc1/C-Assy/power3” represents the “arinc1/C-Assy/power3” occurrence for the higher occurrence “h1.1”

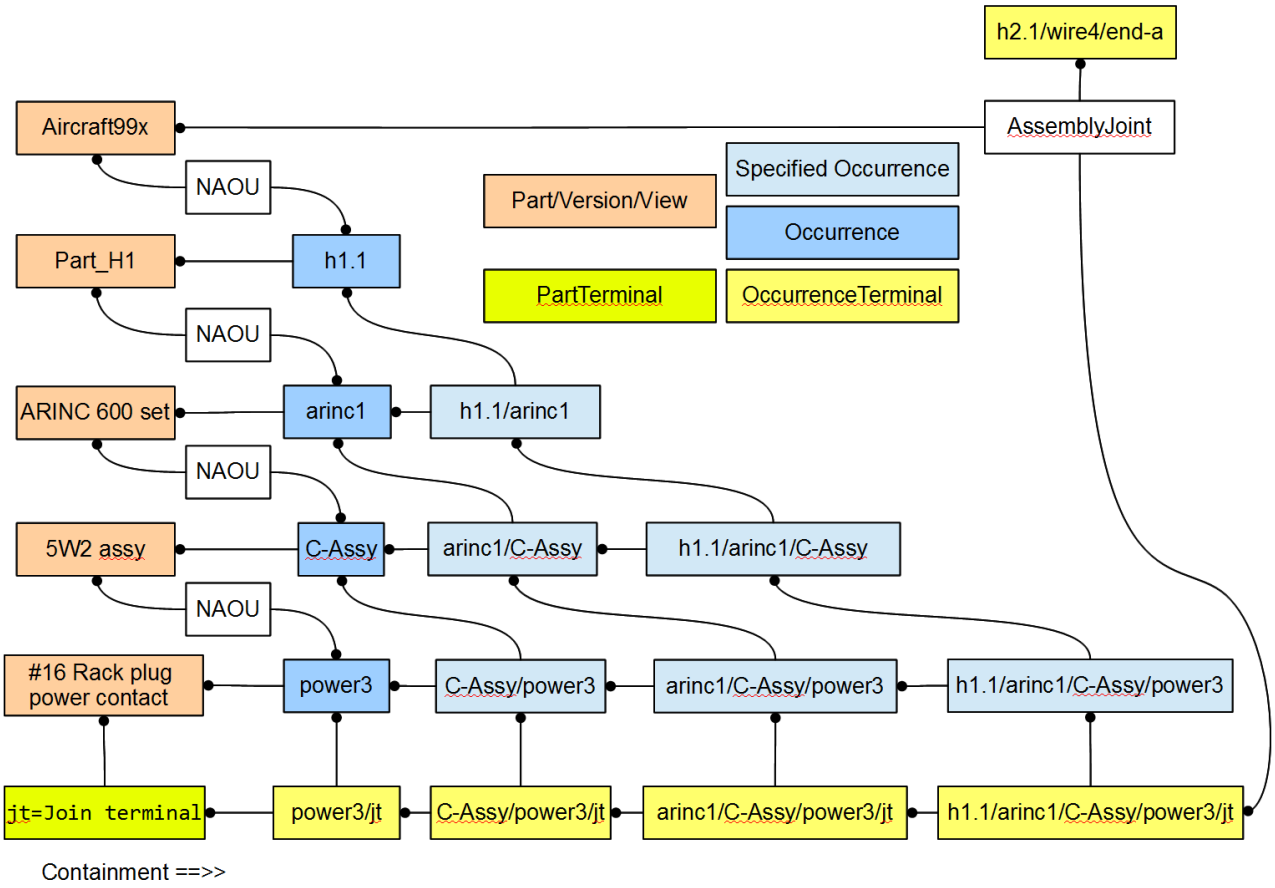

Figure 13: SpecifiedOccurrence Tree with OccurrenceTerminals

Now the terminal “jt” is made visible on the top assembly level and it can there be connected by an AssemblyJoint to another terminal “h2.1/wire1/end-a” for which a similar structure will be build up.

Please note that the flattening of the hierarchical structure into a flat structure is essential because:

- the “5W2 assy” might contain another occurrence “power4” of an “#16 Rack plug power contact” (which is the case in our example)
- the “ARINC 600 set” might contain another occurrence “D-Assy” of an “5W2 assy”
- the “Part_H1” might contain another occurrence “arinc2” of an “ARINC 600 set”
- the “ Aircraft99x” might contain another example “h1.2” of an “Part_H1”

If all this would be the case, an AssemblyJoint on the top assembly level would need to be specific as to which one of the following “jt” OccurrenceTerminals a connection is made:

- “h1.1/arinc1/C-Assy/power3”
- “h1.1/arinc1/C-Assy/power4”
- “h1.1/arinc1/D-Assy/power3”
- “h1.1/arinc1/D-Assy/power4”
- “h1.1/arinc2/C-Assy/power3”
- “h1.1/arinc2/C-Assy/power4”
- “h1.1/arinc2/D-Assy/power3”
- “h1.1/arinc2/D-Assy/power4”
- “h1.2/arinc1/C-Assy/power3”
- “h1.2/arinc1/C-Assy/power4”
- “h1.2/arinc1/D-Assy/power3”
- “h1.2/arinc1/D-Assy/power4”
- “h1.2/arinc2/C-Assy/power3”
- “h1.2/arinc2/C-Assy/power4”
- “h1.2/arinc2/D-Assy/power3”
- “h1.2/arinc2/D-Assy/power4”

6. Cable and Wire Occurrences

Let's start with the IEC definitions of what a wire and cable is:

- **wire** [IEV ref 151-12-28]: flexible cylindrical conductor, with or without an insulating covering, the length of which is large with respect to its cross-sectional dimensions
  Note – The cross-section of a wire may have any shape, but the term "wire" is not generally used for ribbons or tapes.
- **cable** [IEV ref 151-12-38]: assembly of one or more conductors and/or optical fibres, with a protective covering and possibly filling, insulating and protective material

For the purpose of AP242 the definitions may be used in a slightly different way. Important is that a wire has exactly one conductor, but it might be rigid (not flexible). And it is recommended to use cable only if there are two or more conductors.

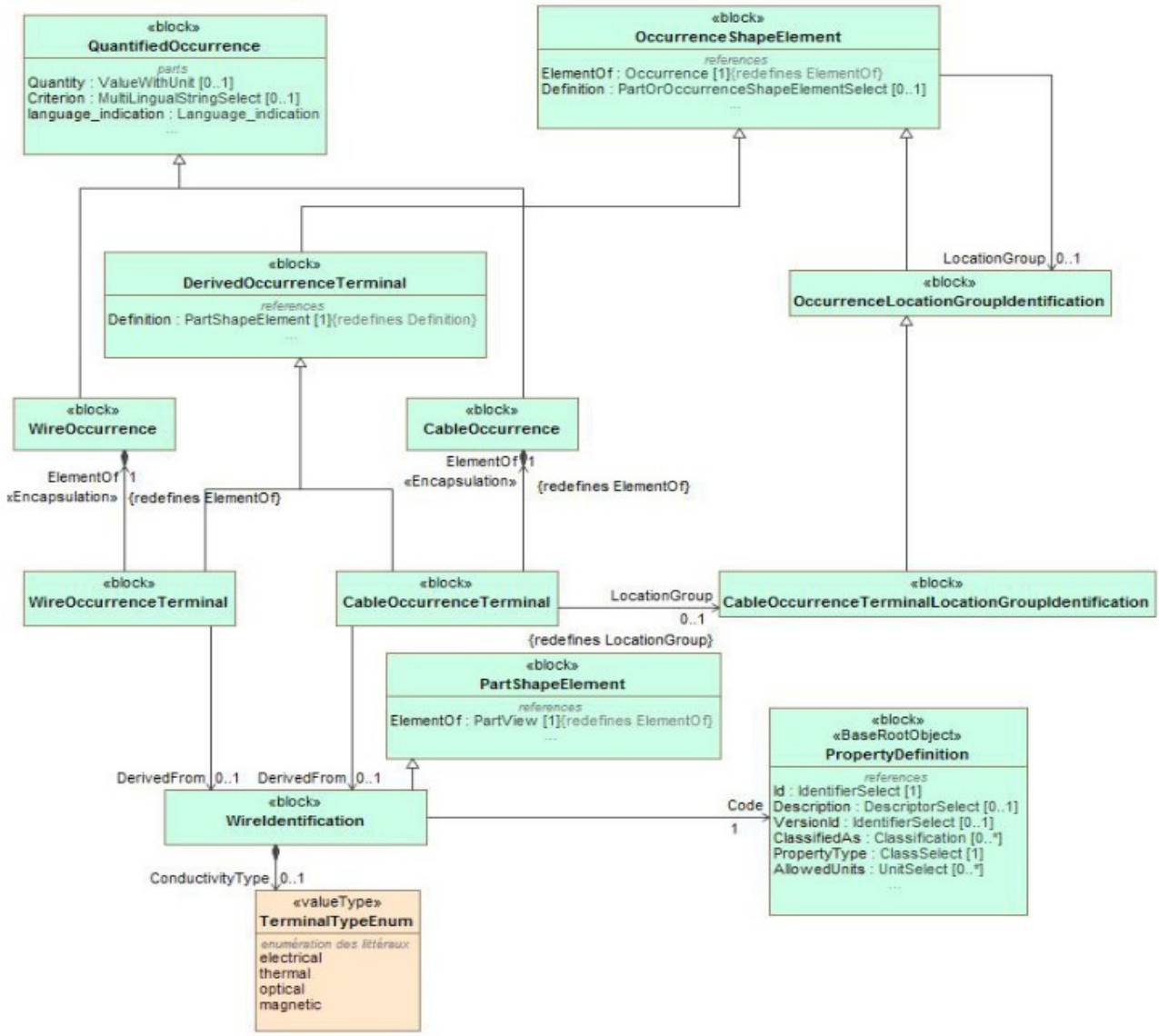

Figure 14: Cable- and WireOccurrence and their Terminals

Cables and Wires as raw materials are treated as Parts with the PartTypes attribute set to the PartCategoryEnum values “cable” respectively “wire”. Cables and wires are typically produced and sold in big lengths that are coiled. The length is often much bigger that the length needed for a particular usage and so a piece of wire or cable has to be cut from the coil in a certain length. Because of this the Part for the raw

material get for PartType the additional PartCategoryEnum values “raw material by length”.

If the exact cut length is know it can directly be specified as a quantity with unit, e.g. 1.5 m. Otherwise a criterion can be specified e.g. “not determined in engineering; to be defined by manufacturer”.

CableOccurrence and WireOccurrence are Occurrences of Parts with a PartTypes that has the *PartCategoryEnum* values “cable” respectively “wire”. Because each *CableOccurrence* and *WireOccurrence* has a particular length they are subtypes of *QuantifiedOccurrence*.

So far we looked only to *OccurrenceTerminals* that are defined by *PartTerminals*. There is another class of terminals on Occurrences that are derived from the more generic *PartShapeElement* which are called *DerivedOccurrenceTerminals*.

Because the raw cable and wire material are typically rolled up and protected by some isolation, it is not possible to identify any particular *PartTerminal* on that raw material. So instead of identifying *PartTerminals* we only identify conductors (one or several) using *WireIdentification*. Both *PartTerminal* and *WireIdentification* are subtypes of *PartShapeElement*. A *WireIdentification* has a pre-defined property named *Code* for the “wire colour-based identification code”.

Cable and wire parts are raw material; they do no have specific *PartTerminals*. *CableOccurrences* and *WireOccurrences* on the other hand have terminals. These terminals are derived from the *WireIdentifications* of the raw materials.

In the same way we could have e.g. a GND-bar and define any number of terminals on it.

7. Harness Topology

The topological structure of a wire harness, is primarily build up from the root topological elements *Vertex* and *Edge*, that are combined in a *ConnectedEdgeWithLengthSetRepresentation* that is referenced by a *WiringHarnessAssemblyDesign* with the *Topology* attribute. Unlike a 2D or 3D geometrical model, the topological model does not define any geometry other than the fact that Edges have a particular length. In most cases the wires, cables and tubular covers an electrical harness is made of are flexible and thus can be arranged in many possible 2D and 3D geometries. By associating these wires, cables and tubular covers to the Edges and Paths (see below) of the *ConnectedEdgeWithLengthSetRepresentation* a receiving system will know:

- at which Vertex a wire, cable or tubular cover will start and where it will end
- which wires, cables and tubular covers are associated to each single edge

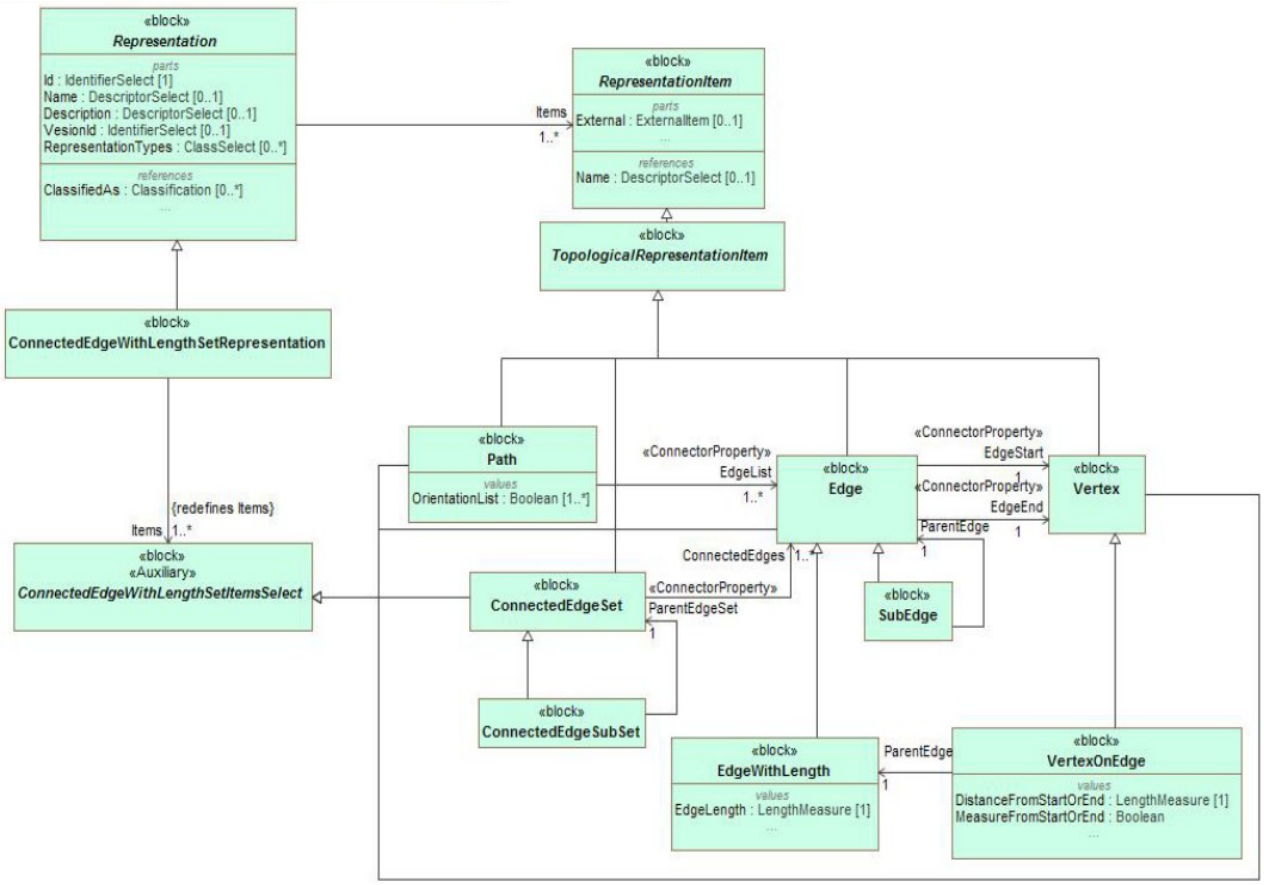

Figure 15: ConnectedEdgeWithLengthSetRepresentation

The main topological structure consists of the following subtypes of *RepresentationItems*:

- *Vertex*
- *Edge*
- *EdgeWithLength*
- *ConnectedEdgeSet*
- *Path*

These items are grouped together in a *ConnectedEdgeWithLengthSetRepresentation.*

An *Edge* is starting in one *Vertex* and is ending in another *Vertex*. Because of this an *Edge* has a direction. An *EdgeWithLength* is an *Edge* for which the length is defined. *Edges* that are connected with each other through common vertices can be grouped together in a *ConnectedEdgeSet*. A *ConnectedEdgeSet* defines a directed graph that may be cyclic (in other words, it may contain loops). Within this graph *Path*s can be

specified that consists of an ordered list of connected *Edges* and a corresponding ordered list of boolean values for the orientation of the *Edges*. The *Edges* in a *Path* are constrained, so that the start *Vertex* of a following *Edge* must start at the end *Vertex* of a previous *Edge*. As needed the direction of an Edge can be inverted in its usage by the corresponding boolean attribute in the *OrientationList*.

Sub-topological structures can be defined on the main topological structure or a higher sub-topological structure with the following *RepresentationItems*:

- *SubEdge*
- *VertexOnEdge*
- *ConnectedEdgeSubSet*

A *SubEdge* is an *Edge* that represents a portion of a *ParentEdge.* A *SubEdge* might have the same start or same end Vertex or none of the Vertices of the parent *Edge*. In all other cases the start and end Vertex of a SubEdge must be VertexOnEdge.
A VertexOnEdge is a Vertex on a parent Edge that is an EdgeWithLength in a defined distance from either the start or the end Vertex. A VertexOnEdge must only be referenced by SubEdges with the same parent Edge.
A ConnectedEdgeSubSet is a ConnectedEdgeSet that defines a sub-set of connected Edges of a ParentEdgeSet.

A *ConnectedEdgeWithLengthSetRepresentation* is a *Representation* that consists of exactly one *ConnectedEdgeSet* (with all the underlying edges and vertices) and any number of *Paths* on that *ConnectedEdgeSet* and any number of additional *SubEdge*s and *VertexOfEdge*s defined on the *Edges* of the *ConnectedEdgeSet*.

As with all kinds of *Representation*s, the *RepresentationItem*s are defined under the *Items* section of a *RepresentationContext* and referenced in the Items section of a *ConnectedEdgeWithLengthSetRepresentation*. Several *ConnectedEdgeWithLengthSetRepresentation* can be defined under the same RepresentationContext, and thus allowing to share the same RepresentationItems.

Example: The RepresentationContext _321000 contains items for which the length unit meter applies. Within this context three ConnectedEdgeWithLengthSetRepresentations are defined. The first one (_321010) defines the complete harness topology for the H1 example, while the other two defines sub-sets of the main one (_323010 for “H1-a” and _323010 for “H1-b”); see figure 1.

```
<RepresentationContext uid="_321000">
   <Id id="H1.x Harness topology context"/>
        <Items>
           ... Definitions of RepresentationItem
        </Items>
   <Representations>
     <Representation xsi:type="n0:ConnectedEdgeWithLengthSetRepresentation"
uid="_321010">
        <Id id="Topological representation of H1 harness"/>
        <Items>
           ... References to RepresentationItems
        </Items>
     </Representation>
     <Representation xsi:type="n0:ConnectedEdgeWithLengthSetRepresentation"
uid="_322010">
        <Id id="SubRep H1.a"/>
        <Items>
           ... References to RepresentationItems
        </Items>
     </Representation>
     <Representation xsi:type="n0:ConnectedEdgeWithLengthSetRepresentation"
uid="_323010">
        <Id id="SubRep H1.b"/>
        <Items>
           ... References to RepresentationItems
        </Items>
```



```
      </Representation>
   </Representations>
   <Units> <Unit uidRef="_100301"/> </Units>
</RepresentationContext>
...
<Unit uid="_100301">
   <Name><ClassString>metre</ClassString></Name>
   <Quantity><ClassString>length</ClassString></Quantity>
</Unit>
```

Example: The *ConnectedEdgeWithLengthSetRepresentation* _321010 is referred by the PartView "Part_H1 " as its topological model:

```
<Part uid="_311000">
   <Id id="Part_H1"/>
   ...
   <Versions>
      <PartVersion uid="_311001">
         <Id></Id>
         <Views>
            <PartView xsi:type="n0:WiringHarnessAssemblyDesign" uid="_311002">
               ...
               <Topology uidRef="_321010" />
            </PartView>
         </Views>
      </PartVersion>
   </Versions>
</Part>
```

Example for the main topological representation items in the *RepresentationContext*:
The *ConnectedEdgeSet* _321020 consists of the *Edges* _321022 and _321023 (and others). These *Edges* have the names S2 and S3, are defined as *EdgeWithLength* with the lengths 1.7m and 2.0m, and use the 3 *Vertices* _ 321042, _ 321043 and _ 321043 that have the names N2, N3 and N4. The *Vertex* _321043 (N3) is shared by both *Edges* as *EdgeEnd*. The *Path* _321065 with the name S2-S3 is defined along the edges S2 in original direction and S3 in inverse direction, so that the Path starts at Vertex N2, then follows to N3 and ends at N4.

```
<RepresentationItem xsi:type="n0:ConnectedEdgeSet" uid="_321020">
   <ConnectedEdges> ...
      <Edge uidRef="_321022"/> <!-- S2 -->
      <Edge uidRef="_321023"/> <!-- S3 -->
   </ConnectedEdges>
</RepresentationItem>
<RepresentationItem xsi:type="n0:EdgeWithLength" uid="_321022">
   <Name> <CharacterString>S2</CharacterString> </Name>
   <EdgeEnd uidRef="_321043"/>   <!-- N3 -->
   <EdgeStart uidRef="_321042"/> <!-- N2 -->
   <EdgeLength>1.7</EdgeLength>
</RepresentationItem>
<RepresentationItem xsi:type="n0:EdgeWithLength" uid="_321023">
   <Name> <CharacterString>S3</CharacterString> </Name>
   <EdgeEnd uidRef="_321043"/>    <!-- N3 -->
   <EdgeStart uidRef="_321044"/>  <!-- N4 -->
   <EdgeLength>2.0</EdgeLength>
</RepresentationItem>
<RepresentationItem xsi:type="n0:Vertex" uid="_321042">
   <Name> <CharacterString>N2</CharacterString> </Name>
</RepresentationItem>
<RepresentationItem xsi:type="n0:Vertex" uid="_321043">
   <Name> <CharacterString>N3</CharacterString> </Name>
</RepresentationItem>
<RepresentationItem xsi:type="n0:Vertex" uid="_321044">
```



```
   <Name> <CharacterString>N4</CharacterString> </Name>
</RepresentationItem>
<RepresentationItem xsi:type="n0:Path" uid="_321065">
   <Name> <CharacterString>S2-S3</CharacterString> </Name>
   <EdgeList>
     <Edge uidRef="_321022"/>  <!-- S2 -->
     <Edge uidRef="_321023"/>  <!-- S3 -->
   </EdgeList>
   <OrientationList>
     <Boolean>true</Boolean>
     <Boolean>false</Boolean>
   </OrientationList>
</RepresentationItem>
```

Example of sub-topological elements for the main topological elements defined above:
The VertexOnEdge _321051 with the name X1 is defined on the Edge _321023 (S3) at a distance of 0.8 m from Vertex _321044 (N4). The SubEdge _321035 (S3-2) is defined on _321023 (S3) between the start vertex _321051 (X1) and the end vertex _321044 (N4). Another SubEdge _321034 (S3-1) is defined on the same parent edge between the start vertex _321051 (X1) and the end vertex _321043 (N3). Because the main edge S3 is 2.0 m long, a receiving system can deduce that the length of the S3-1 SubEdge is 1.2 m. The ConnectedEdgeSubSet defines a sub-set of Edges from the parent _321020 and contains the main edge _321022 (S2), the sub-edge _321034 (S-1) and others.

```
<RepresentationItem xsi:type="n0:ConnectedEdgeSubSet" uid="_322020">
   <ConnectedEdges> ...
     <Edge uidRef="_321022"/> <!-- S2   -->
     <Edge uidRef="_321034"/> <!-- S3-1 -->
   </ConnectedEdges>
   <ParentEdgeSet uidRef="_321020"/>
</RepresentationItem>
<RepresentationItem xsi:type="n0:SubEdge" uid="_321034">
   <Name> <CharacterString>S3-1</CharacterString> </Name>
   <EdgeEnd uidRef="_321043"/>    <!-- N3 -->
   <EdgeStart uidRef="_321051"/>  <!-- X1 -->
   <ParentEdge uidRef="_321023"/> <!-- S3 -->
</RepresentationItem><RepresentationItem xsi:type="n0:SubEdge" uid="_321035">
   <Name> <CharacterString>S3-2</CharacterString> </Name>
   <EdgeEnd uidRef="_321044"/>    <!-- N4 -->
   <EdgeStart uidRef="_321051"/>  <!-- X1 -->
   <ParentEdge uidRef="_321023"/> <!-- S3 -->
</RepresentationItem>
<RepresentationItem xsi:type="n0:VertexOnEdge" uid="_321051">
   <Name> <CharacterString>X1</CharacterString> </Name>
   <DistanceFromStartOrEnd>0.8</DistanceFromStartOrEnd>
   <MeasureFromStartOrEnd>true</MeasureFromStartOrEnd>
   <ParentEdge uidRef="_321023"/>  <!-- S3 -->
</RepresentationItem>
```


8. Cross Section of Cables or Harness Segments

When cutting through a harness segment perpendicular to its centre line we will see cables, wires, and other auxiliary materials such as isolations, shields, protections and more. The following data model is provided to describe such a structure.

The data model defines a directed acyclic graph, with CrossSectionalOccurrenceShapeElements as the leave elements that are defined by CrossSectionalPartShapeElements. The higher level nodes and root nodes in this graph are:

- CrossSectionalAlternativePartShapeElement
- TwistedCrossSectionalGroupShapeElement
- CrossSectionalGroupShapeElement
- TwistedCrossSectionalGroupShapeElement
- CrossSectionalGroupShapeElementWithTubularCover
- CrossSectionalGroupShapeElementWithLacing

The cross-section of a HarnessSegment can be defined by any ones of these nodes. However the definition of any of these nodes and leaves is not restricted to a single HarnessSegment. These nodes may span several segments and are therefore defined and reusable for the whole harness.

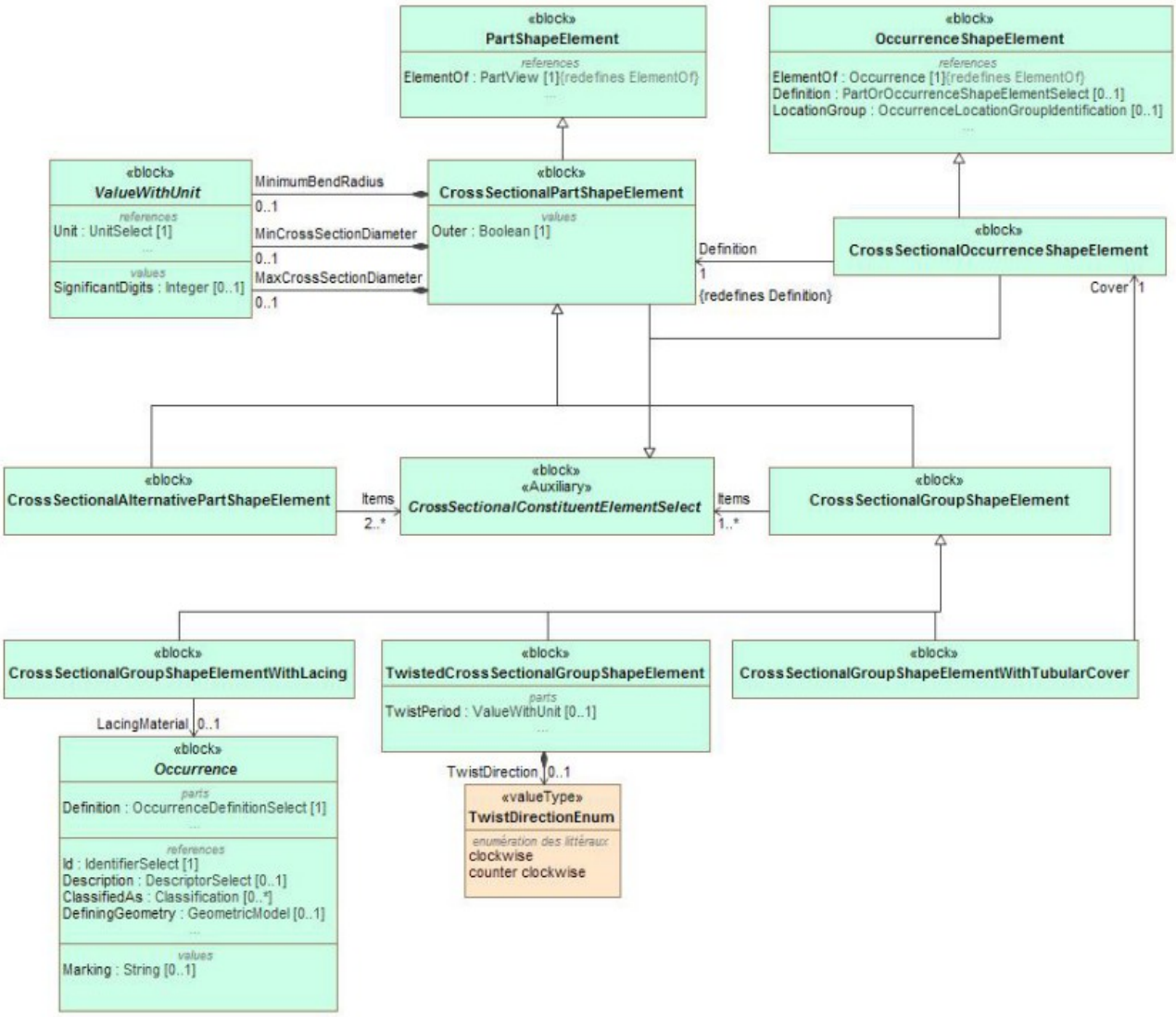

Figure 16: Cross-sections of extruded structures

A CrossSectionalPartShapeElement, subtype of PartShapeElement represents an extruded shape along some centre-line of a part. The minimum bend radius for this shape (to the centre-line) as well as the minimal and maximal cross sectional diameter of the shape is provided. The exact geometry of the cross sectional ShapeElement (e.g. circular or rounded rectangle) is not specified as this may vary along the centre line. The boolean attribute Outer specifies if TRUE the outer shape along the centre line. If FALSE an inner shape is provided. A PartView can have at most one CrossSectionalPartShapeElement with value TRUE for Outer, but it may have zero, one or more CrossSectionalPartShapeElements with the value FALSE.

A CrossSectionalOccurrenceShapeElement is an OccurrenceShapeElement defined by a CrossSectionalPartShapeElement for an Occurrence of a Part. CrossSectionalOccurrenceShapeElements are the leave elements from which the cross section of a harness segment is build up.

A CrossSectionalConstituentElementSelect defines a select of either a CrossSectionalPartShapeElement or a CrossSectionalOccurrenceShapeElement.

A CrossSectionalAlternativePartShapeElement is a CrossSectionalPartShapeElement that defines that one of two or more CrossSectionalConstituentElementSelects is to be used as alternatives in a cross section.

A CrossSectionalGroupShapeElement is a CrossSectionalPartShapeElement that groups one or more items defined by CrossSectionalConstituentElementSelect. How the grouping is realised is not defined.

A TwistedCrossSectionalGroupShapeElement is a CrossSectionalGroupShapeElement where the items are grouped by twisting. The twist direction and period can be specified.

A CrossSectionalGroupShapeElementWithTubularCover is a CrossSectionalGroupShapeElement where the items are grouped by covering them all around with a specified side of material from a specific side of this material.

CrossSectionalGroupShapeElementWithLacing is a CrossSectionalGroupShapeElement where the items are laced together by some material. The details of lacing such as the period length are not specified.

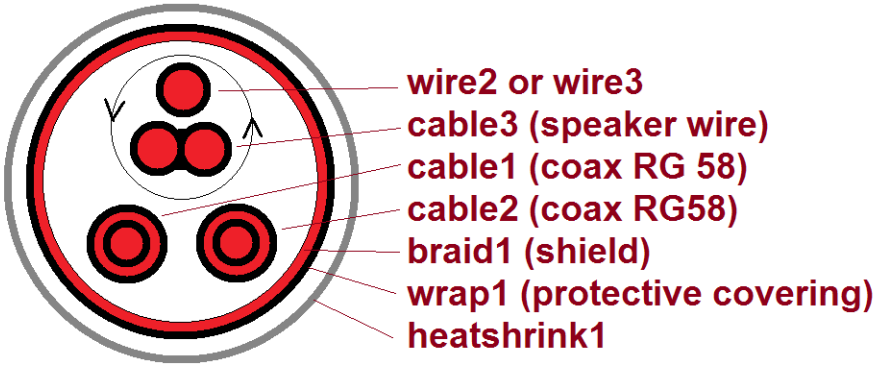

Figure 17: Example cross-section for harness segments S2-1 and S3-1

Explanation of the cross sectional structure in the provided example:

- for the wire, coax cable RG 48, and the Speaker Cable no detailed information is provided about the inner structure, insulation etc. So only the **outer** CrossSectionalPartShapeElement is provided for these parts, together with corresponding CrossSectionalOccurrenceShapeElements for the Occurrences. See the upper part of table 1
- for the Braid, Wrap and HeatShrink the **inner** CrossSectionalPartShapeElement and corresponding CrossSectionalOccurrenceShapeElement is needed as they need to go over something else. See the lower part of table 1.
- wire2 in segment S2 is equivalent to wire3 in segment S3; so these are taken as alternatives for a AlternativeLongitudinalExtendPart (_313001)

- the alternative of wire2 and wire3 are twisted with cable3, covered by TwistedCrossSectionalGroupShapeElement (_313002). The twist-direction is clockwise and the length of the period is 5 cm.
- the result of the upper twist is together with cable1 and cable2 grouped together and covered by braid1 for shielding purpose by CrossSectionalGroupShapeElementWithTubularCover (_313004).
- the group covered by braid1 is protected by wrap1 through CrossSectionalGroupShapeElementWithTubularCover (_313005).
- the group protected by wrap1 is further protected by heatshrink1 through CrossSectionalGroupShapeElementWithTubularCover _313006
- in some places (S2-2) the heatshrink1 is directly on braid1, CrossSectionalGroupShapeElementWithLacing (_313008).

In table4 and table5 the details of the cross sectional structure used in H1 are provided.

Name	Occurrence	in/out	uid
WIRE,ELEC,COMP,SNGL CONDUCTOR,150 DEG C	Part	outer	_101020
wire1	_201004	outer	_201009
wire2	_201104	outer	_201109
wire3	_201204	outer	_201209
RG 58	Part	outer	_102020
cable1	_202006	outer	_202007
cable2	_202106	outer	_202107
Speaker wire	Part	outer	_104020
cable3	_204006	outer	_204007
Braid 1/2inch	Part	inner	_120003
braid1	_220005	inner	_220006
Wrap	Part	inner	_121003
wrap1	_221005	inner	_221006
HeatShrink	Part	inner	_122003
heatshrink1	_222005	inner	_222006

Table 4: Cross-sectional ShapeElements of "bought" parts

Type	uid / uidRef	Segment	comment
AlternativeLongitudinalExtendPart	uid _313001	internal	wire2 or wire3
item	uidRef _201109		wire2/outer
item	uidRef _201209		wire3/outer
TwistedCSGSE	uid _313002	internal	twist (wire2 or wire3) with cable3
item	uidRef _313001		wire2 or wire3
item	uidRef _204007		cable3/outer
CSGSEWithTubularCover	uid _313004	S2-3,S3-2	braid1 on (twist...)
item	uidRef _313002		twist...
item	uidRef _202007		cable1/outer
item	uidRef _202107		cable2/outer
cover	uidRef _220006		braid1/inner
CSGSEWithTubularCover	uid _313005	internal	wrap1 on braid1 ...
item	uidRef _313004		braid1 on (twist...
cover	uidRef _221006		wrap1/inner
CSGSEWithTubularCover	uid _313006	S2-1,S3-1	heatshrink1 on wrap1
item	uidRef _313005		wrap1 on ...
cover	uidRef _222006		heatshrink1/inner
CSGSEWithTubularCover	uid _313007	S2-2	heatshrink1 on braid1 ...
item	uidRef _313004		braid1 on (twist...
cover	uidRef _222006		heatshrink1/inner
CSGSEWithLacing	uid _313008	S5	Lacing of cable1, cable2 & wire3
item	uidRef _201209		wire3/outer
item	uidRef _202007		cable1/outer
item	uidRef _202107		cable2/outer

Table 5: Cross-sectional structure definitions in H1

9. Bringing all together

A *WiringHarnessAssemblyDesign* is an *AssemblyDefinition* in which all the physical elements and the physical connectivity of an electrical harness is defined. Unless the harness has a very primitive structure, it refers to a *ConnectedEdgeWithLengthSetRepresentation* that defines the topological structure.

A *HarnessSegment* is a kind of *PartShapeElement* for a *WiringHarnessAssemblyDesign* that links an *Edge* from the topological representation of the harness to a *CrossSectionalGroupShapeElement*. For manufacturing purpose the length of a referenced *Edge* can be overridden by the ForcedLength attribute.

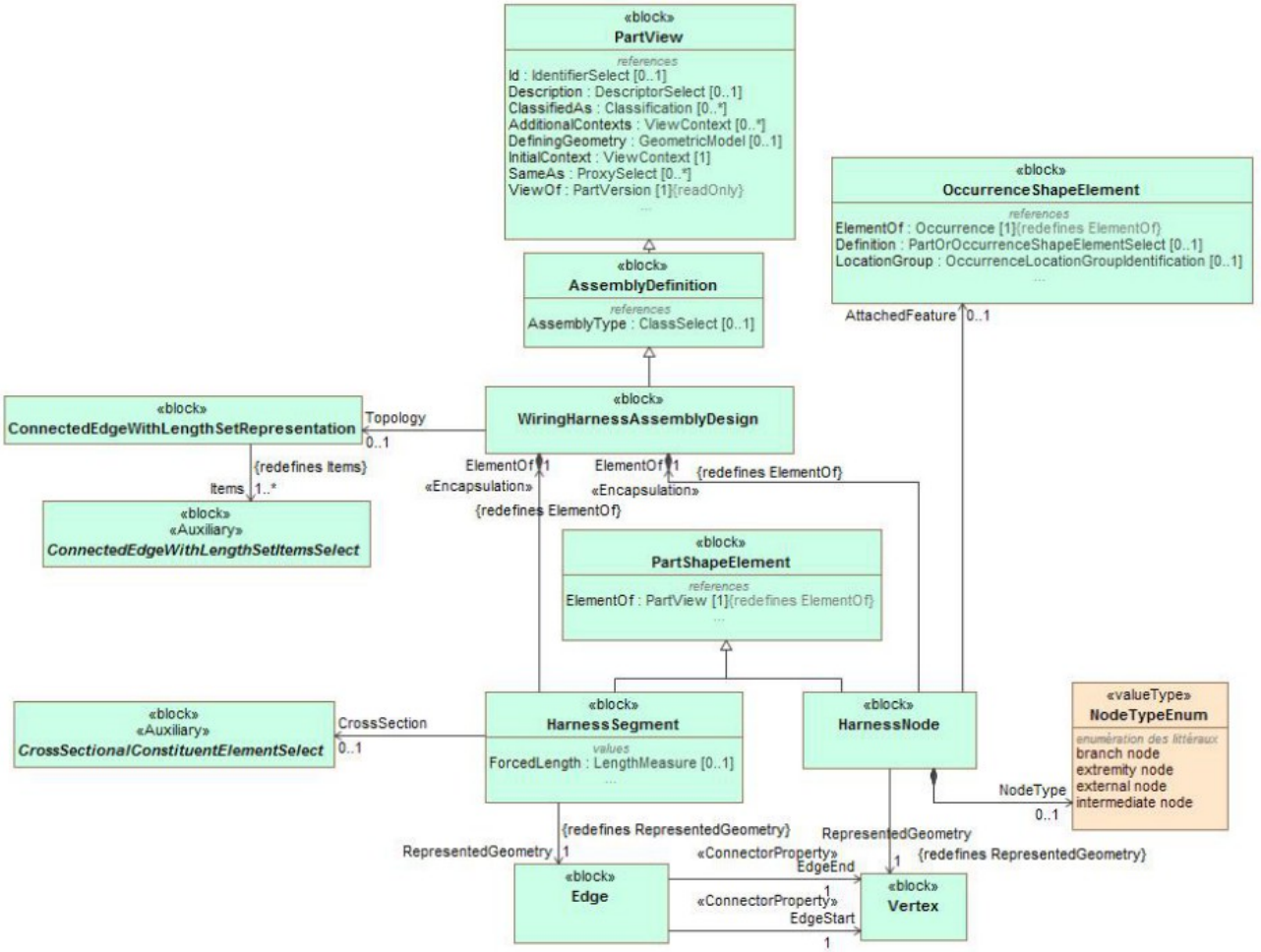

Figure 18: WiringHarnessAssemblyDesign with HarnessSegement and -Node

A *HarnessNode* is another kind of *PartShapeElement* for a *WiringHarnessAssemblyDesign* that links a *Vertex* from the topological representation to an *OccurrenceShapeElement* that represents an aspect of an *Occurrence*. With the attribute NodeType it is possible to characterise the kind of node:

- branch node: A node that is used by more than one Edge or HarnessSegment. The nodes N3 and N4 of the example are of this kind.
- extremity node: A node that is used by a single Edge or HarnessSegment. The nodes N1, N2, N5 and N6 of the example are of this kind.
- external node: A node in a partial harness design that is used by a single Edge or HarnessSegment and that is indicating that the harness continues beyond that node in another design. The node X4 in the example is of this kind for the sub-harnesses H1.a and H1.b.
- intermediate node: Additional nodes defined on Edges. The nodes X1, X2, X3, X4 (for H1), X5 and X6 of the example are of this type.

9.1. General Representation / Model and Transformation Capabilities

Figure 19 shows the data model for *Representation*, *GeometricRepresentation* and -*Model* and transformations between them. In this tutorial we only explain the concepts needed for electrical harness. For a more thorough understanding of the concepts see ISO 10303-42 “Representation structures”.

STEP defines many specializations of *RepresentationItem* (e.g. *AxisPlacement*, *Curve*, *CartesianPoint*, *Edge*, *Vertex*, *Path* ...). Each *RepresentationItem* is defined within a *RepresentationContext*. A *Representation* groups a set of *RepresentationItems* for a specific purpose. A *RepresentationItem* is included in a *Representation* either directly (via the *Items* attribute or indirectly because it it referenced by another *RepresentationItem*. E.g. when a *Representation* contains an *Edge*, it also contains the two *Vertices* called out by *EdgeStart* and *EdgeEnd*.

Representations can be related to each other by *RepresentationRelationship*. If the *Definitional* attribute is set to TRUE, the *Related Representation* becomes a definitional part of the *Relating Representation*. Note that the *Relating* attribute is covered as containment in XML.

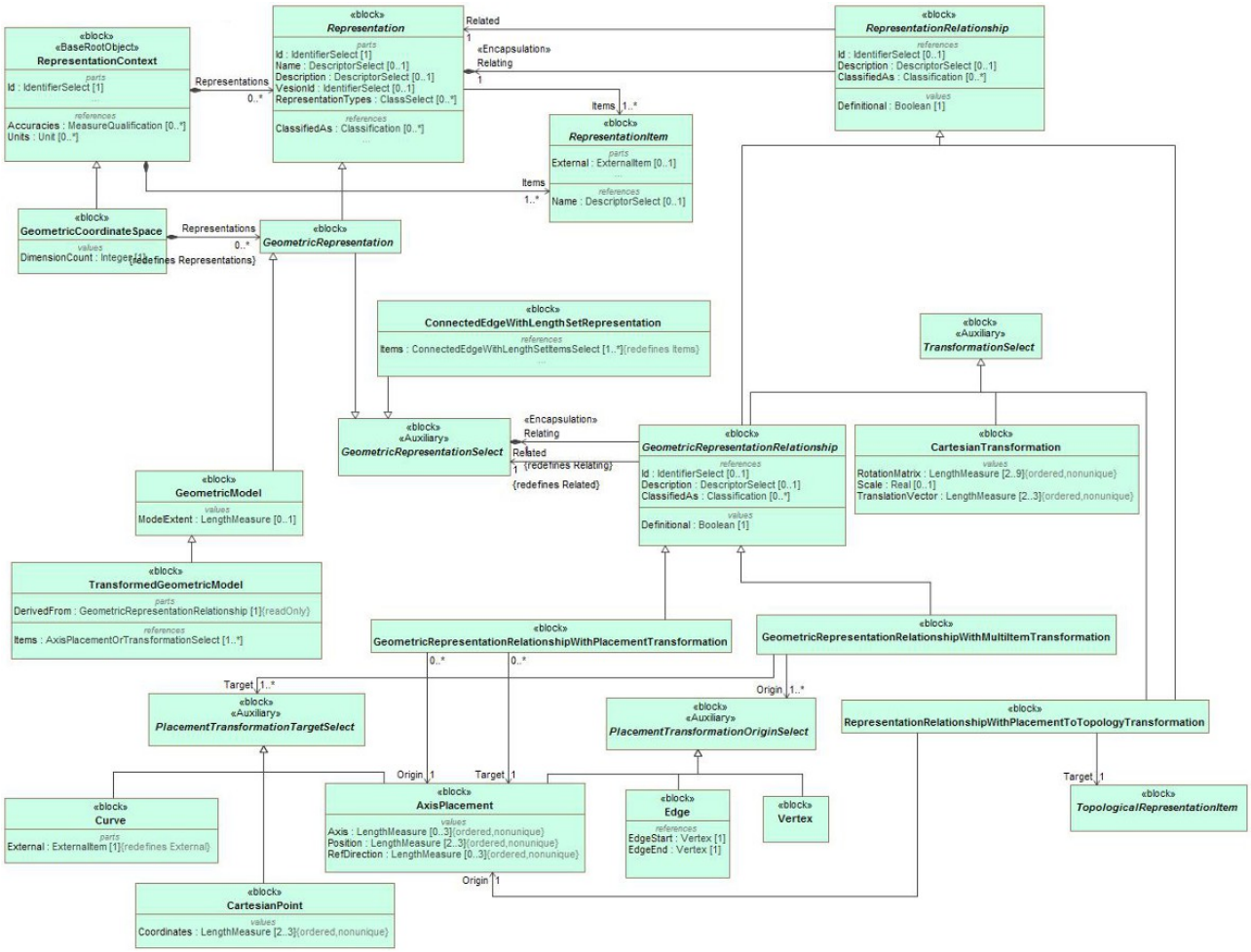

Figure 19: GeometricRepresentation, Relationship and Transformation

For the purpose of geometry *GeometricRepresentation* (a specialization of *Representation*) and *GeometricCoordinateSystem* (a specialization of *RepresentationContext*) is used. The attribute *DimensionCount* of *GeometricCoordinateSystem* defined on whether it is a 2D or a 3D coordinate system.

Transformations between *Representations* are defined by specializations of *RepresentationRelationship*. Most widely used in STEP is *GeometricRepresentationRelationshipWithPlacementTransformation* where the transformation is indirectly defined by placing an *AxisPlacement* from the related *Representation* (attribute *Origin*) onto an *AxisPlacement* (attribute *Target*) of the relating *Representation*.

Especially for the purpose of electrical harness the transformations

GeometricRepresentationRelationshipWithMultiItemTransformation and *RepresentationRelationshipWithPlacementToTopologicalTransformation* are defined that are further explained below.

9.2. Defining the top part with a WiringHarnessAssemblyDesign view

The top object of an electrical wiring harness is a *Part* with the *PartCategoryEnum* “wiring harness”.

The specific associations needed to represent the design of an electrical wiring harness are supported by *WiringHarnessAssemblyDesign* that is a specialization of *PartView* (and *AssemblyDefinition*).

As with any other *PartView* one or several 2D or 3D geometric models can be associated with the *DefiningGeometry* attribute.

For a full design documentation a *WiringHarnessAssemblyDesign* shall have an associated topological model in the form of a *ConnectedEdgeWithLengthSetRepresentation* topology. The attribute *Topology* might not be set if the *ConnectedEdgeWithLengthSetRepresentation* is not available or if the topology is so simple (e.g. a single segment with two nodes) that it is not needed.

Example:

```
<Part uid="_311000"> <!-- Part_H1 -->
  <Id id="Part_H1"/>
  <Name>
    <CharacterString>Electrical Harness example 1</CharacterString>
  </Name>
  <PartTypes>
    <PartCategoryEnum>wiring harness</PartCategoryEnum>
  </PartTypes>
  <Versions>
    <PartVersion uid="_311001">
      <Id></Id>
      <Views>
        <PartView xsi:type="n0:WiringHarnessAssemblyDesign" uid="_311002">
          <AdditionalContexts>
            <ViewContext uidRef="_100104"/>
            <ViewContext uidRef="_100105"/>
          </AdditionalContexts>
          <DefiningGeometry uidRef="_314090"/>
          <InitialContext uidRef="_100102"/>
          ...
          <Topology uidRef="_321010" />
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

9.3. Associating rigid and flexible components to the Topological Model

This clause describes how occurrences (components) of rigid and flexible parts are associated to items of a *ConnectedEdgeWithLengthSetRepresentation* (CEWLSR) of a *WiringHarnessAssemblyDesign* (WHAD), see figure 20.

As with all other assemblies, *Single-* or *QuantifiedOccurrences* are brought into a *WiringHarnessAssemblyDesign* through *NextAssemblyOccurrenceUsages* (NAOU).

To be able to link with items of the *ConnectedEdgeWithLengthSetRepresentation*, the *Occurrences* or their underlying *PartViews* must have an associated *GeometryModel*. This might be an externally defined model or a placeholder model with just a single *AxisPlacement*.

A *RepresentationRelationshipWithPlacementToTopologyTransformation* (RRWPTTT) is used to link the geometric model of an underlying *PartView* or *Occurrence* with a *ConnectedEdgeWithLengthSetRepresentation*. To be able to distinguish several occurrences of the same *PartView*, the attribute *Transformation* of *NextAssemblyOccurrenceUsages* must refer to the corresponding *RepresentationRelationshipWithPlacementToTopologyTransformation*. Note that the transformation is defined as a set so it can support additional transformations, e.g. for an additional 2D stick-line representation and a 3D representation.

In addition we need to specify the details of the geometrical to topological mapping. This is done by the *Origin* and *Target* attribute of the *RepresentationRelationshipWithPlacementToTopologyTransformation*. The *Origin* attribute calls out an *AxisPlacement* of the geometric model of the *Occurrence* or underlying *PartView* and maps it to either an *Edge, Vertex* or *Path* of the *ConnectedEdgeWithLengthSetRepresentation*.

This mapping is the key point to support the flexible nature of a wire harness. Typically for rigid parts the *AxisPlacements* of the geometric model of the *PartView* is mapped onto a *Vertex* or specialization *VertexOnEdge*, while flexible parts such as wires and cables are mapped onto an *Edge* with its specializations *EdgeWithLength* and *SubEdge* or to a *Path* that consists of several *Edges*.

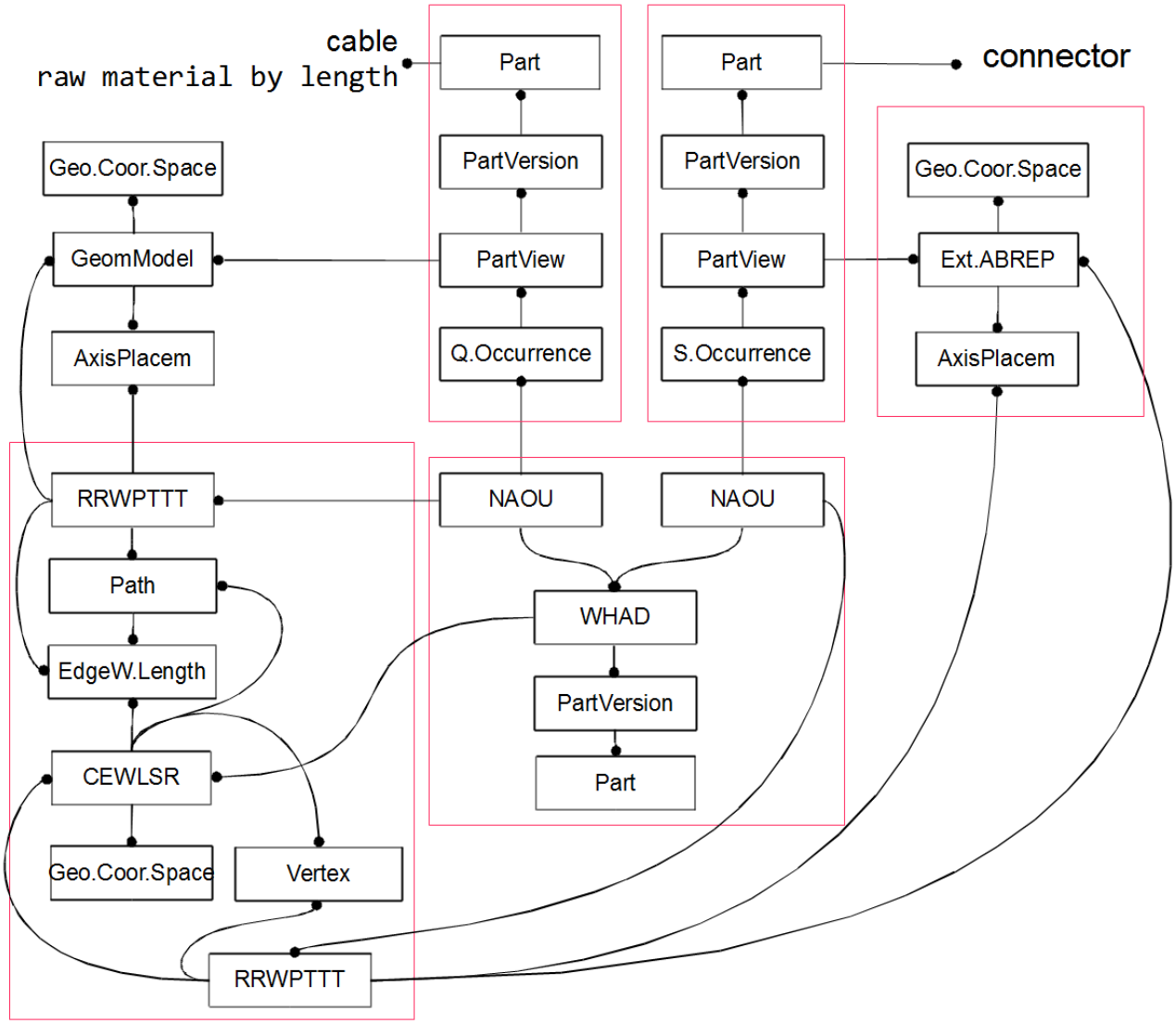

Figure 20: Transformation of Flexible and rigid parts onto the topological model

Example:

The mapping of flexible objects such as wires, cables and tubular covers to the topological model is done as

follows.

1) A generic geometric model (example _104890) is representing the centreline of a linear extruded - possibly flexible - objects by a single *RepresentationItem* of type *Vector*. This model might be used by several independent *PartViews*.

```
<!--Generic AxisPlacement representation for all wire, cable, covering  -->
<RepresentationContext xsi:type="n0:GeometricCoordinateSpace" uid="_104891">
   <Id id="/NULL"/>
   <Items>
      <RepresentationItem xsi:type="n0:AxisPlacement" uid="_104896">
        <Position>0.0 0.0 0.0</Position>
      </RepresentationItem>
   </Items>
   <Representations>
      <Representation xsi:type="n0:GeometricModel" uid="_104890">
        <Id id=""/>
        <Items>
          <RepresentationItem uidRef="_104896"/>
        </Items>
      </Representation>
   </Representations>
   <DimensionCount>3</DimensionCount>
</RepresentationContext>
```

2) A *PartView* refers to the generic centreline representation as its *DefiningGeometry*. In addition the *Occurrences* of the *PartView* are defined (in the example these are wire2/_201104 and wire3/_201204).

```
<Part uid="_101000"> <!-- WIRE,ELEC,COMP,SNGL CONDUCTOR,150 DEG C -->
   ...
   <Versions>
      <PartVersion uid="_101001">
        <Id id="Version 1"/>
        <Views>
          <PartView uid="_101002">
            <DefiningGeometry uidRef="_104890"/>
            ...
            <Occurrence xsi:type="n0:WireOccurrence" uid="_201104">
              <Id id="wire2"/> ...
            </Occurrence>
            <Occurrence xsi:type="n0:WireOccurrence" uid="_201204">
              <Id id="wire3"/> ...
            </Occurrence>
          </PartView>
        </Views>
      </PartVersion>
   </Versions>
</Part>
```

3) The *WiringHarnessAssemblyDesign* (Part_H1/_311002) contains *NextAssemblyOccurrenceUsage*s (wire2/_315012, wire3/_315041, ...) with *Placement* attribute to *RepresentationRelationshipWithPlacementToTopologyTransformation* (subtypes _321080, _321082) and a *NextAssemblyOccurrenceUsage*s (lug1/_315041) with *Placement* attribute to *GeometricRepresentationRelationship* specialization *GeometricRepresentationRelationshipWithPlacementTransformation*.

```
   <PartView xsi:type="n0:WiringHarnessAssemblyDesign" uid="_311002">
      ...
      <DefiningGeometry uidRef="_314090"/>
      ...
      <ViewOccurrenceRelationship uid="_315012"
xsi:type="n0:NextAssemblyOccurrenceUsage">
        <Related uidRef="_201104"/> <!--WireOccurrence wire2-->
```



```
        <RelationType>
          <ClassString>next assembly occurrence</ClassString>
        </RelationType>
        <Placement>
          <RepresentationRelationshipWithPlacementToTopologyTransformation
uidRef="_321080"/>
        </Placement>
      </ViewOccurrenceRelationship>
      <ViewOccurrenceRelationship uid="_315013"
xsi:type="n0:NextAssemblyOccurrenceUsage">
        <Related uidRef="_201204"/> <!--WireOccurrence wire3-->
        <RelationType>
          <ClassString>next assembly occurrence</ClassString>
        </RelationType>
        <Placement>
          <RepresentationRelationshipWithPlacementToTopologyTransformation
uidRef="_321082"/>
        </Placement>
      </ViewOccurrenceRelationship>
      ...
      <ViewOccurrenceRelationship uid="_315041"
xsi:type="n0:NextAssemblyOccurrenceUsage">
        <Related uidRef="_203005"/> <!--SingleOccurrence lug1 -->
        <RelationType>
          <ClassString>next assembly occurrence</ClassString>
        </RelationType>
        <Placement>
          <GeometricRepresentationRelationship uidRef="_314210"/>
        </Placement>
      </ViewOccurrenceRelationship>
      ...
      <Topology uidRef="_321010" />
    </PartView>
```

4) A *ConnectedEdgeWithLengthSetRepresentation* is representing the topology of a wiring harness. It contains *(Geometric)RepresentationRelationship*s that are referenced by the *NextAssemblyOccurrenceUsage*s of the wire harness. The specific subtype *RepresentationRelationshipWithPlacementToTopologyTransformation* is used to indicate how the flexible centreline of the extruded parts is mapped to either a single *Edge* or a *Path* that is a sequence of *Edge*s.

```
<RepresentationContext uid="_321000">
  <Id id="H1.x Harness topology context"/>
  <Representations>
    <Representation xsi:type="n0:ConnectedEdgeWithLengthSetRepresentation"
uid="_321010">
      <Id id="Topological representation of H1 harness"/> <!--This should be
the name but Id is mandatory-->
      <Items> ... </Items> ...
      <!--Transformation of wire2-->
      <RepresentationRelationship
xsi:type="n0:RepresentationRelationshipWithPlacementToTopologyTransformation"
uid="_321081">
        <Definitional>false</Definitional>
        <Related uidRef="_104890"/>
        <Origin uidRef="_104896"/>
        <Target uidRef="_321022"/> <!--Edge S2-->
      </RepresentationRelationship>
      <!--Transformation of wire3-->
      <RepresentationRelationship
xsi:type="n0:RepresentationRelationshipWithPlacementToTopologyTransformation"
uid="_321082">
        <Definitional>false</Definitional>
```



```
          <Related uidRef="_104890"/>
          <Origin uidRef="_104896"/>
          <Target uidRef="_321066"/> <!--Path S3-S5-->
        </RepresentationRelationship>
        ...
      </Representation>
    </Representations>
    <Units>
      <Unit uidRef="_100301"/>
    </Units>
</RepresentationContext>
```

9.4. Mapping harness topology to a Geometry Model, 2D or 3D

On the right side of figure 21 we see the transformation of rigid parts into the assembly. The structure is identical to what is detailed in the Recommended Practices for AP242 Business Object Model XML Assembly Structure. E.g. the *PartView* of a connector calls out an *ExternalAdvancedBrepShapeRepresentation* as its *DefiningGeometry*. This representation is defined within a *GeometricCoordinateSpace* (Geo.Coor.Space) and contains an *AxisPlacement* (usually at the position 0/0/0 with no rotation) for placement purposes.

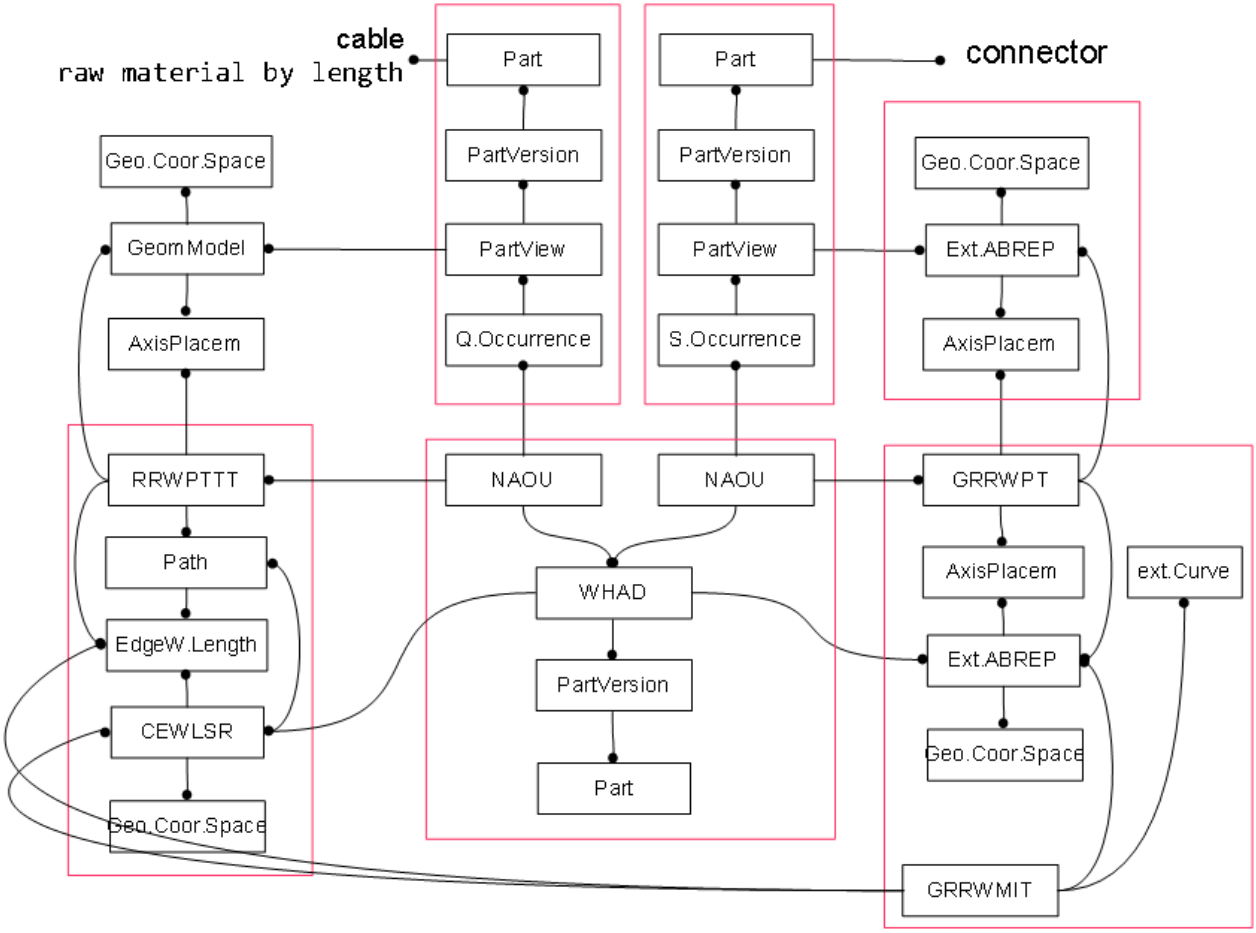

Figure 21: Flexible and rigid part transformation into a 3D (2D) model

A *SingleOccurrence* of the *PartView* of the connector is brought into the *WiringHarnessAssemblyDesign* by a *NextAssemblyOccurrenceUsage*. The *DefiningGeometry* of a *WiringHarnessAssemblyDesign* is represented by another *GeometricCoordinateSpace* that also contains several *AxisPlacements* that are on the positions and orientations on where to place the occurrences. A

GeometricRepresentationRelationshipWithPlacementTransformation (GRRWPT) brings the ABREP of the connector into the ABREP of the *WiringHarnessAssemblyDesign*. The transformation is defined by aligning two *AxisPlacements* onto each other. The *Placement* attribute of the *NextAssemblyOccurrenceUsage* refers to this GRRWPT. This is needed to be able to distinguish the geometry of several occurrences of the same type.

On the left side of figure 21 we see the transformation of flexible parts such as cables and wires into the wiring harness assembly that is different, but similar to the one of a rigid part. As a first step the occurrence of a cable is transformed onto a *Path* or *Edge* of the *ConnectedEdgeWithLengthSetRepresentation* that defines the *Topology* of the *WiringHarnessAssemblyDesign* by a *RepresentationRelationshipWithPlacementToTopologyTransformation* (RRWPTTT). Here the position of the *AxisPlacement* represents the start of the cable that is mapped onto the start *Vertex* of the *Path*. The z-direction (*Axis[3]*) of the *AxisPlacement* represents the direction of the centreline of the cable. As the target *Path* and underlying *Edges* have no particular geometry defined yet, we only know that this direction refers to the direction of the first *Edge* of the *Path* at the start *Vertex*.

At a next step the topological elements of the *ConnectedEdgeWithLengthSetRepresentation* are transformed onto items of the *ExternalAdvancedBrepShapeRepresentation* that defines one of possible many representations of the wire harness. For this a *GeometricRepresentationRelationshipWithMultiItemTransformation* is used that allows a pair-wise transformation of the items of the *ConnectedEdgeWithLengthSetRepresentation* onto the items of a *GeometricModel*. Usually *Edges* are transformed onto *Curves* and *Vertices* onto *AxisPlacements*. The figure shows a single *Edge* that is mapped onto an external *Curve* within the *ExternalAdvancedBrepShapeRepresentation*.

Note: In the provided example a *GeometricRepresentationRelationshipWithMultiItemTransformation* is only used to map *Edges* and *Paths* onto *Curves*. It could also be used to map *Vertices* to *AxisPlacement*. This possible alternative had not been chosen because traditional 3D STEP implementations are used to directly map the 3D model of the piece-parts to the 3D model of the assembly.

