AP242 ed2
Electrical Wire Harness (EWS)
Tutorial – Slides
part 1

Lothar Klein, LKSoftWare GmbH

Sophie HERAIL, CIMPA S.A.S.

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

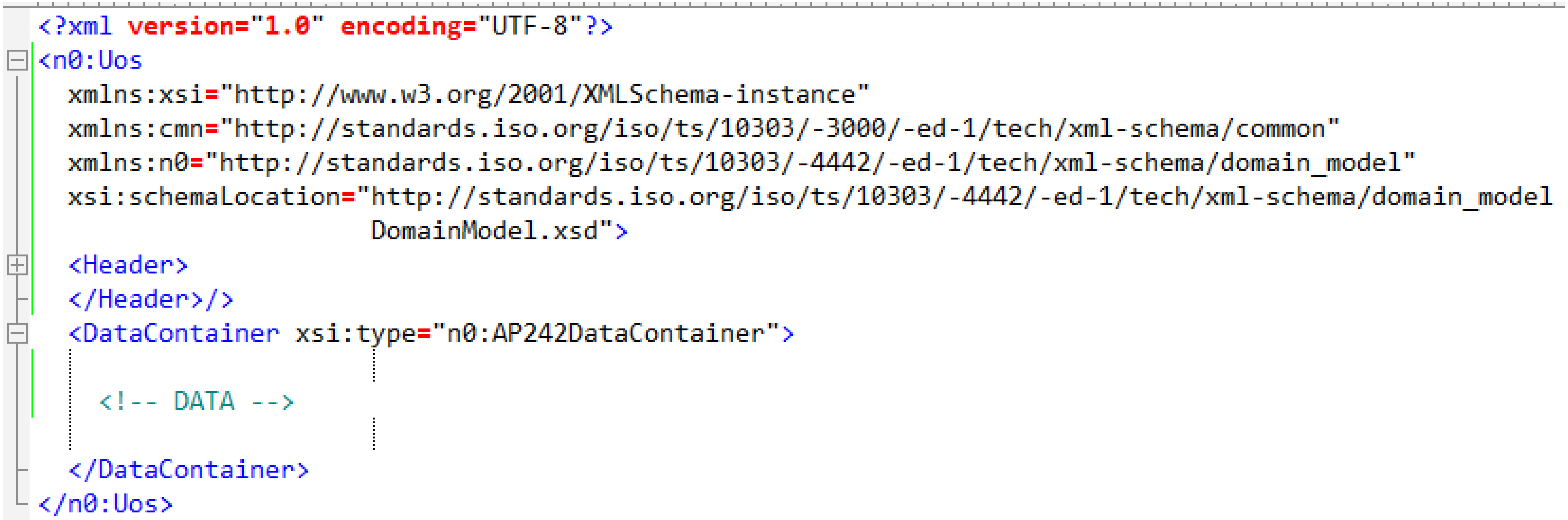
Version 2.1; 2020-11-16

General

- This document is based on ISO/IS 10303-242:2020
  (2nd edition) – and incorporates changes of the upcoming amendment
- This document focuses on XML implementations based on the
  AP242 ed2 Domain Model (DO-Model).
  Note: Electrical wire harness can also be implemented using the
  MIM (Modular Integrated Model) of AP242 ed2
- All provided files will be made available on https://www.cax-if.org/
- This tutorial is an extension of the:
  Recommended Practices for AP242 Business Object Model
  XML Assembly Structure (RECPRAC)
  Release 2.1; 2019-12-20; on: www-cax-if.org
  Please refer to that document for all general concepts.
  Note that release 2.1 is based on the first edition AP242 (technical corrigendum).
  Relevant changes for 2nd edition are indicated here.
- This document covers only the specific extensions of AP242 ed2
  for the purpose of electrical wire harness (EWH)
- The EWH Domain Model covers also general electrical systems and so can be used to describe electrical installations for e.g. machines or in buildings

XML File Format, XML Schema

- Every AP242 XML file has to call out the XML-Schema of AP242. For the 2nd edition of AP242 this is defined as a DomainModel in the 1st edition of part ISO/TS 10303-4442.
- The “Unit of serialization” (Uos) element is the top most element that encloses all needed schema definitions
- All the application objects (AO, entities) not being declared as contained in any other AO are defined as a subtype of cmn:BaseRootObject and show up within the cmn:DataContainer
- Example:



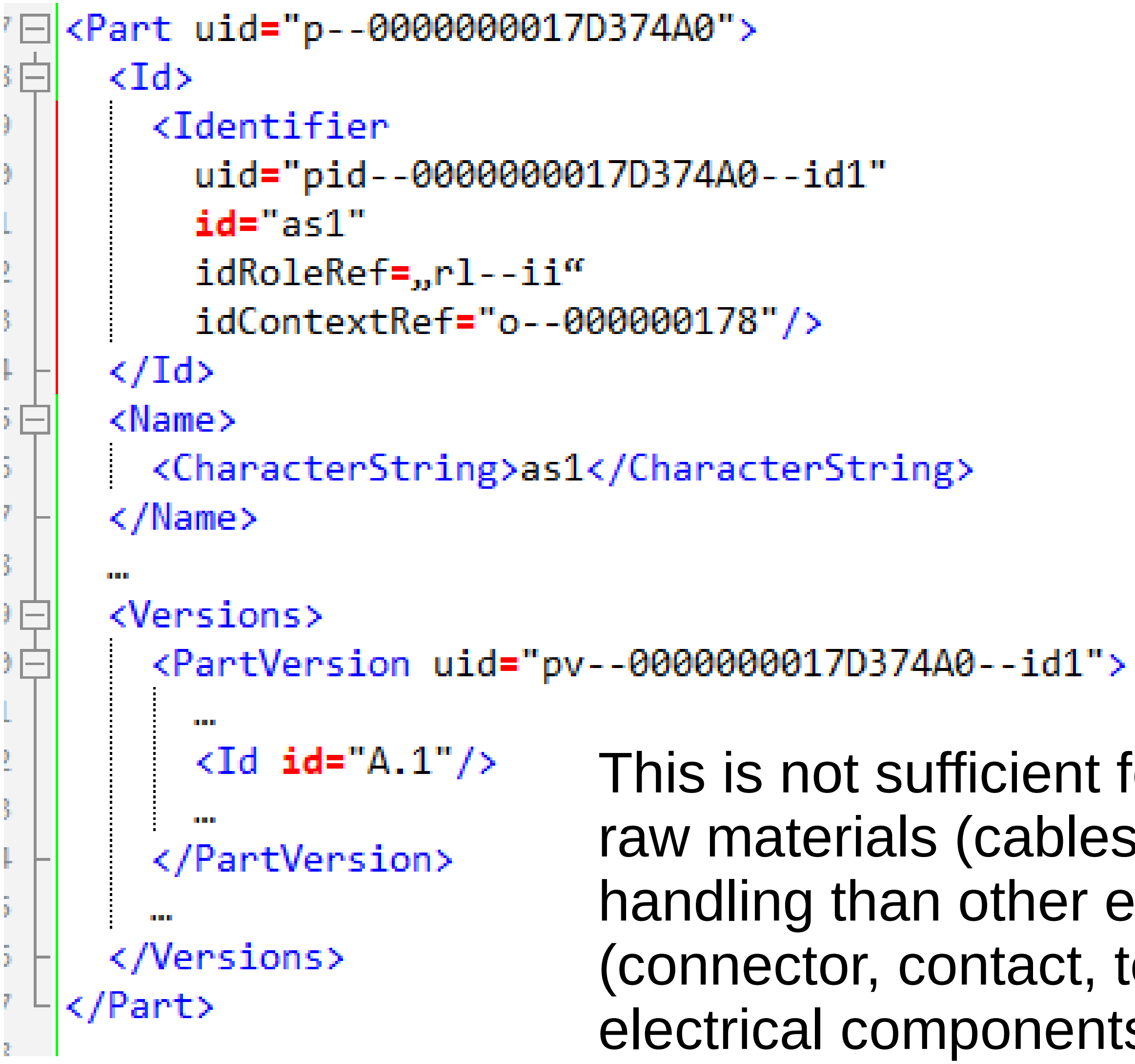
```
<?xml version="1.0" encoding="UTF-8"?>
<n0:Uos
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cmn="http://standards.iso.org/iso/ts/10303/-3000/-ed-1/tech/xml-schema/common"
  xmlns:n0="http://standards.iso.org/iso/ts/10303/-4442/-ed-1/tech/xml-schema/domain_model"
  xsi:schemaLocation="http://standards.iso.org/iso/ts/10303/-4442/-ed-1/tech/xml-schema/domain_model
                     DomainModel.xsd">
  <Header>
  </Header>/>
  <DataContainer xsi:type="n0:AP242DataContainer">

    <!-- DATA -->

  </DataContainer>
</n0:Uos>
```

Part, general

- For the purpose of RECPRAC the representation of a `Part` and a `PartVersion` may just look like this:

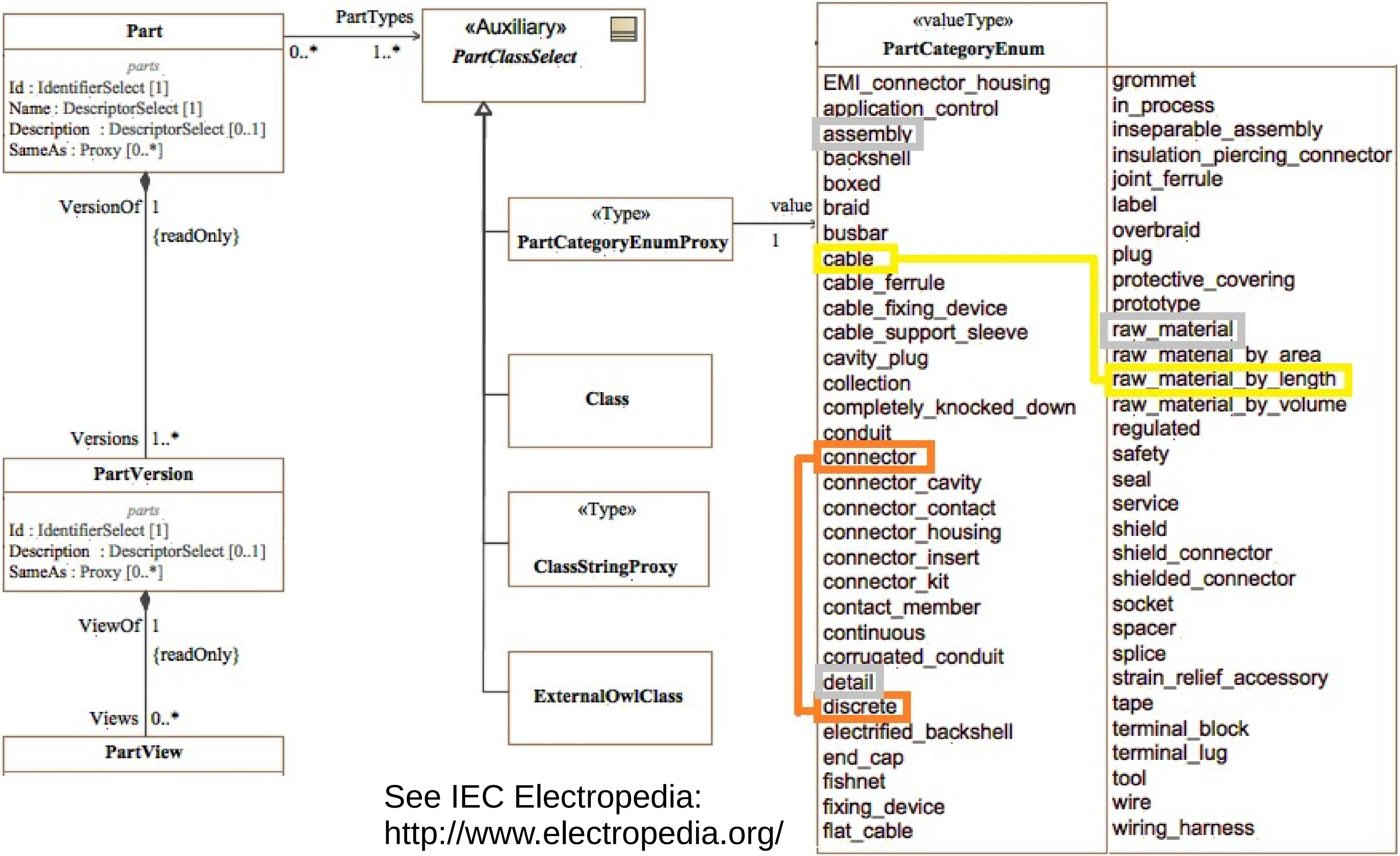


```
<Part uid="p--0000000017D374A0">
  <Id>
    <Identifier
      uid="pid--0000000017D374A0--id1"
      id="as1"
      idRoleRef=„rl--ii“
      idContextRef="o--000000178"/>
  </Id>
  <Name>
    <CharacterString>as1</CharacterString>
  </Name>
  …
  <Versions>
    <PartVersion uid="pv--0000000017D374A0--id1">
      …
      <Id id="A.1"/>
      …
    </PartVersion>
    …
  </Versions>
</Part>
```

This is not sufficient for an EWH converter as raw materials (cables, wires …) require different handling than other electrical relevant parts (connector, contact, terminal) and other non electrical components.

Domain Model: *Part*, *PartVersion*

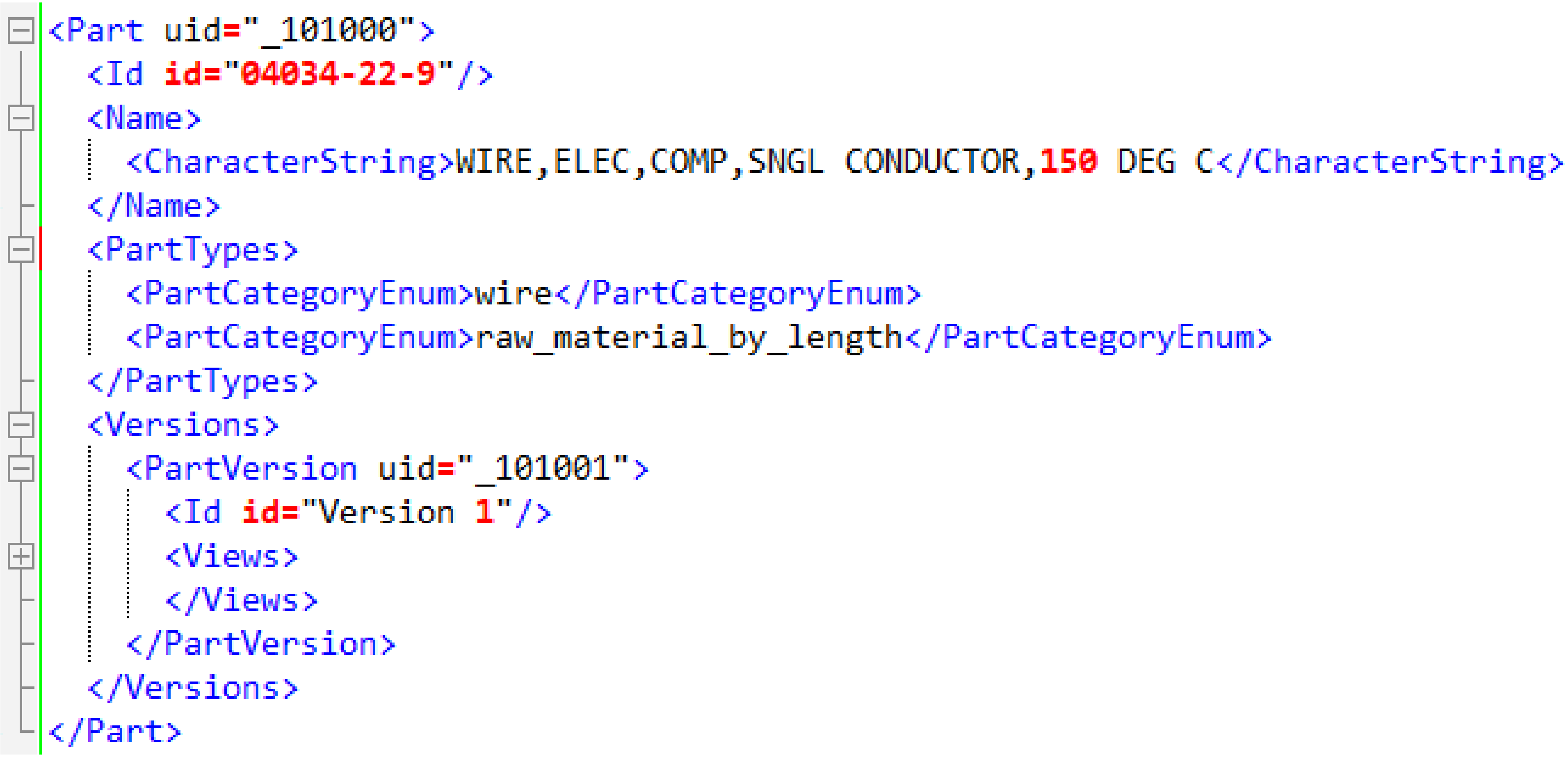
- The traditional part categories “detail”, “assembly”, “raw_material” are not sufficient for EWH
- New categories on the nature of a part (cable, connector) and
  how it can be used in an assembly (discrete, raw_material_by_length) are added



See IEC Electropedia:
http://www.electropedia.org/

Example: Electrical Wire Harness Part

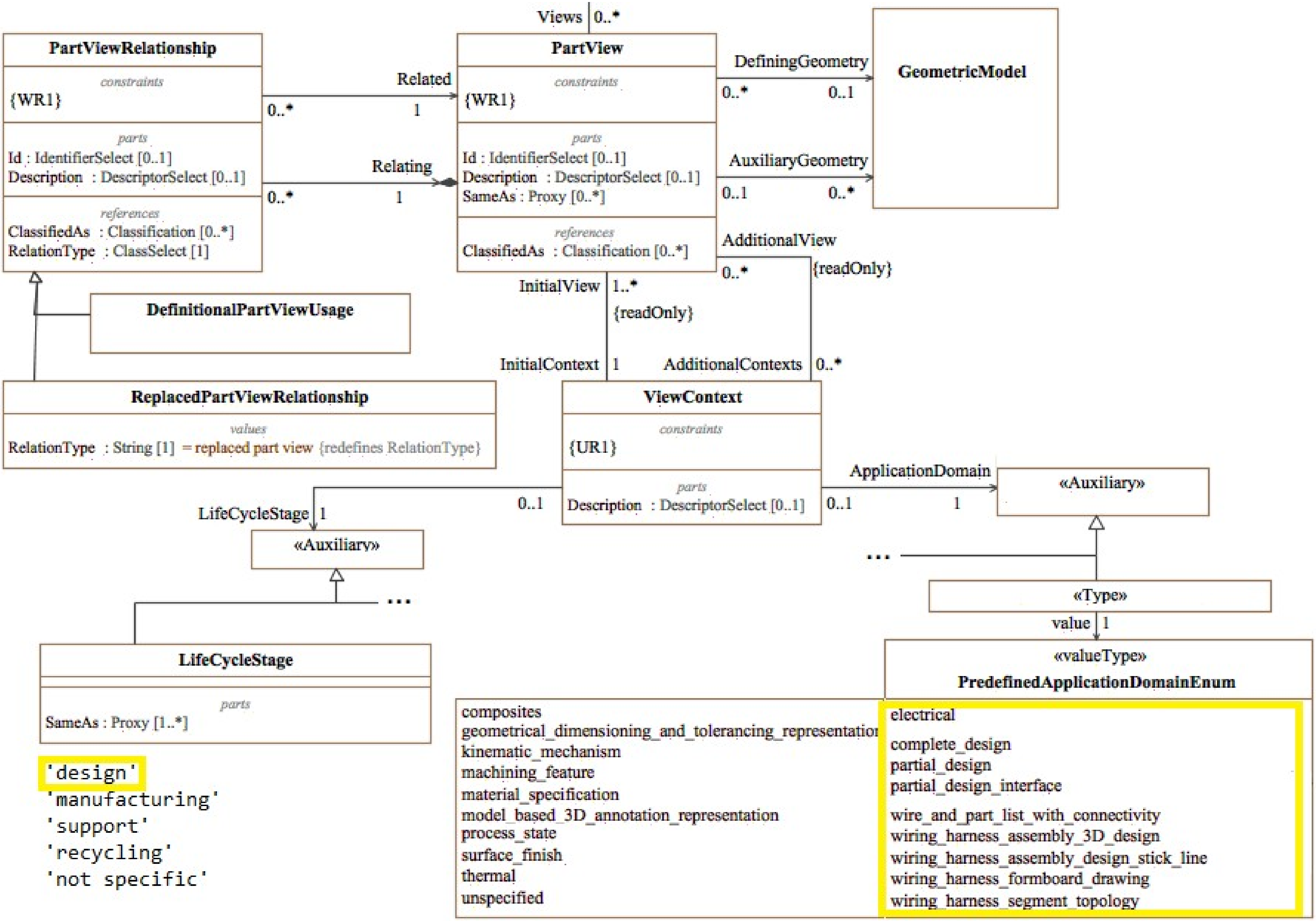
- The part has categories “wire” and “raw_material_by_length.
  So a converter importing this file knows
  - The part is a wire (in the meaning of Electropedia. IEC 60050-151)
  - The part can only be “used” in some specified lengths



```xml
<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>
```

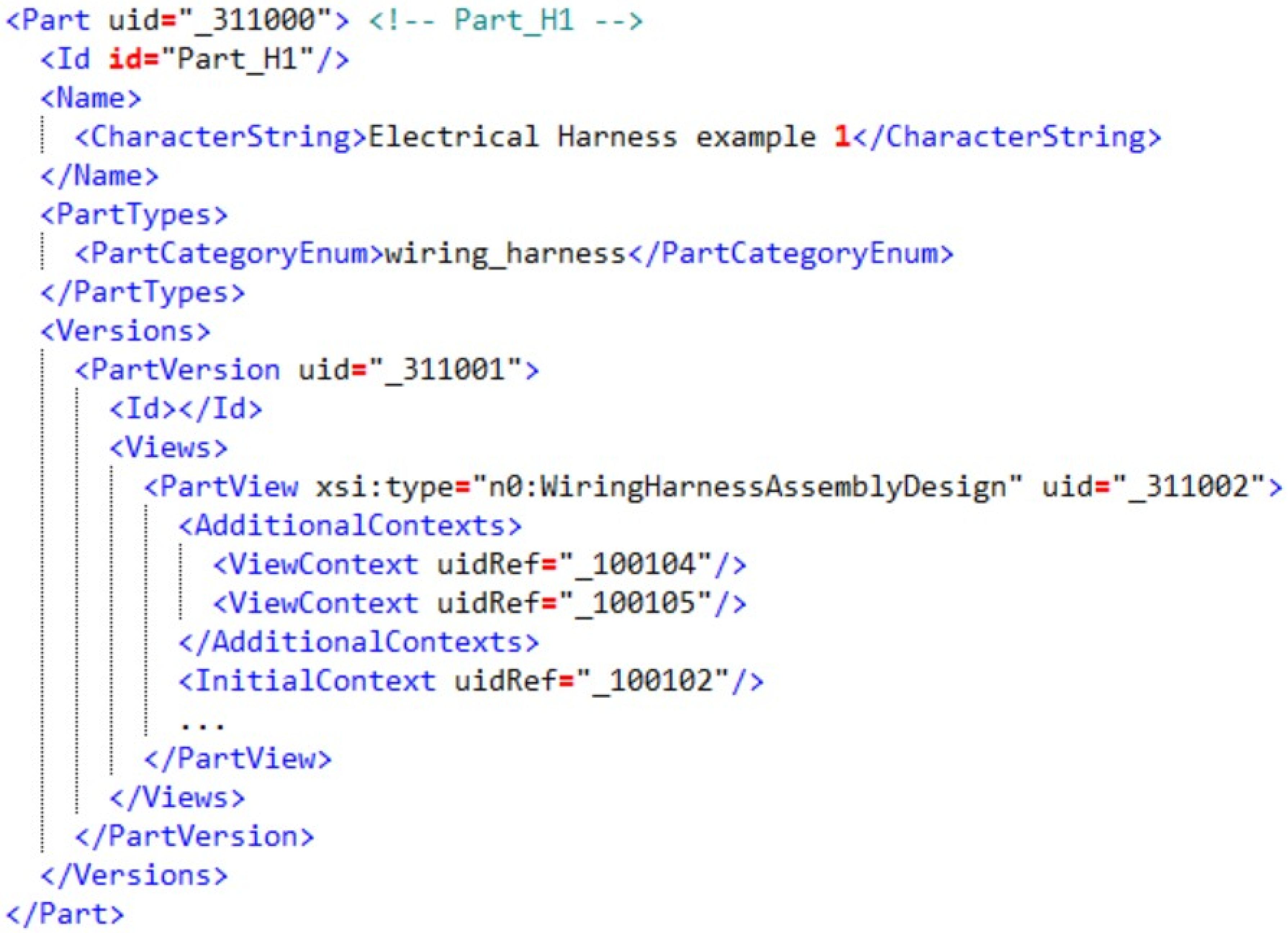
Domain Model: PartView, ViewContext

New ApplicationDomains for EWH


Views 0..*
PartViewRelationship
constraints
{WR1}
parts
Id : IdentifierSelect [0..1]
Description : DescriptorSelect [0..1]
references
ClassifiedAs : Classification [0..*]
RelationType : ClassSelect [1]
Related
0..*
1
Relating
0..*
1
PartView
constraints
{WR1}
parts
Id : IdentifierSelect [0..1]
Description : DescriptorSelect [0..1]
SameAs : Proxy [0..*]
references
ClassifiedAs : Classification [0..*]
DefiningGeometry
0..*
0..1
AuxiliaryGeometry
0..1
0..*
GeometricModel
AdditionalView
0..*
{readOnly}
InitialView
1..*
{readOnly}
DefinitionalPartViewUsage
InitialContext
1
AdditionalContexts
0..*
ReplacedPartViewRelationship
values
RelationType : String [1] = replaced part view {redefines RelationType}
ViewContext
constraints
{UR1}
parts
Description : DescriptorSelect [0..1]
ApplicationDomain
0..1
1
«Auxiliary»
0..1
LifeCycleStage 1
«Auxiliary»
...
...
«Type»
value 1
LifeCycleStage
parts
SameAs : Proxy [1..*]
«valueType»
PredefinedApplicationDomainEnum
composites
geometrical_dimensioning_and_tolerancing_representation
kinematic_mechanism
machining_feature
material_specification
model_based_3D_annotation_representation
process_state
surface_finish
thermal
unspecified
electrical
complete_design
partial_design
partial_design_interface
wire_and_part_list_with_connectivity
wiring_harness_assembly_3D_design
wiring_harness_assembly_design_stick_line
wiring_harness_formboard_drawing
wiring_harness_segment_topology

'design'
'manufacturing'
'support'
'recycling'
'not specific'

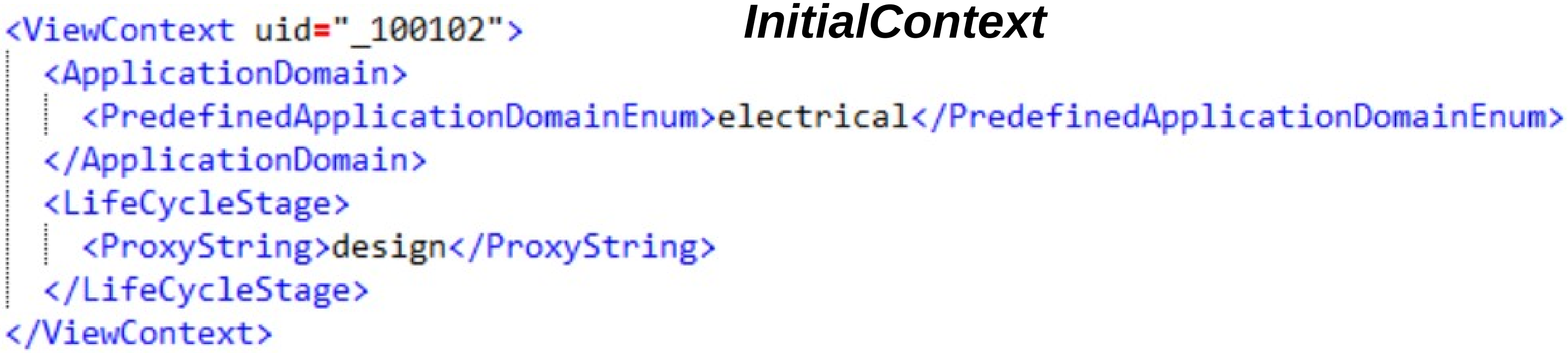
Example: Part/PartVersion with a WiringHarnessAssemblyDesign View



```xml
<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>
          <InitialContext uidRef="_100102"/>
          ...
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

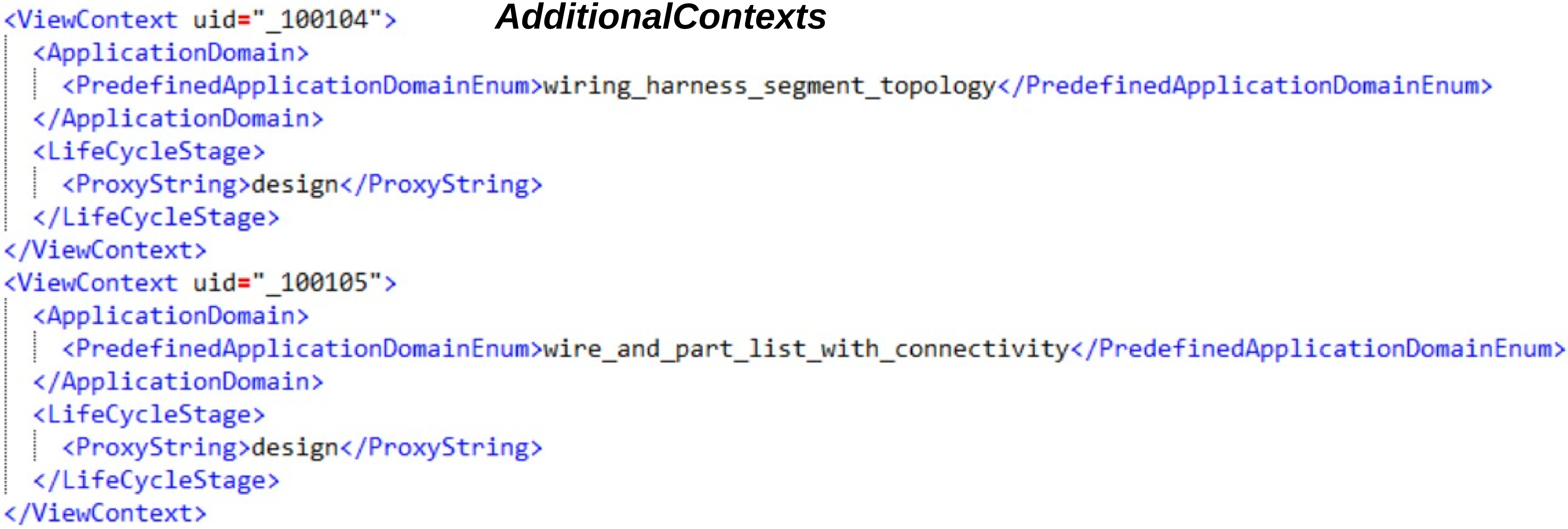
Example: *ViewContexts*

InitialContext



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

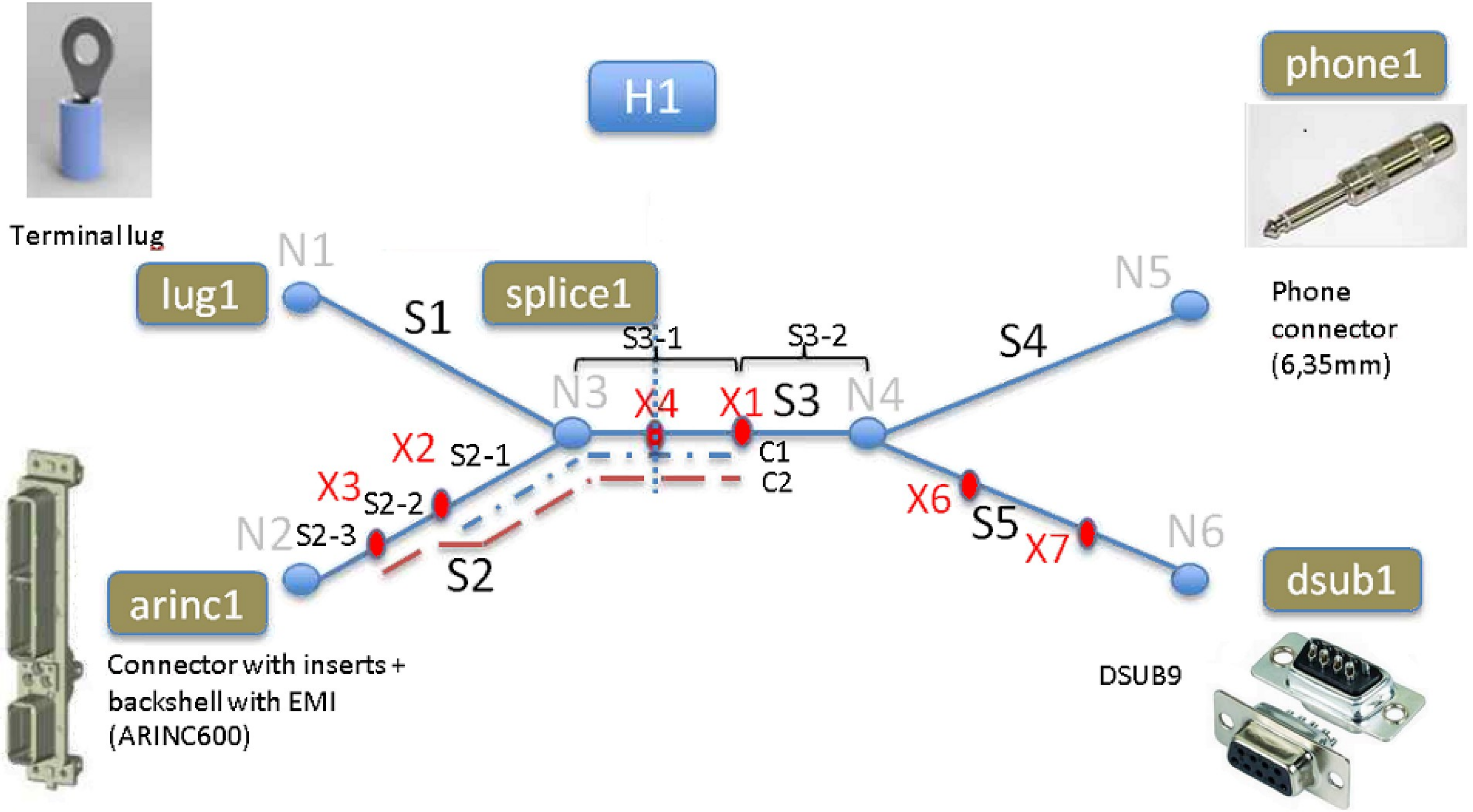
AdditionalContexts



```
<ViewContext uid="_100104">
  <ApplicationDomain>
    <PredefinedApplicationDomainEnum>wiring_harness_segment_topology</PredefinedApplicationDomainEnum>
  </ApplicationDomain>
  <LifeCycleStage>
    <ProxyString>design</ProxyString>
  </LifeCycleStage>
</ViewContext>
<ViewContext uid="_100105">
  <ApplicationDomain>
    <PredefinedApplicationDomainEnum>wire_and_part_list_with_connectivity</PredefinedApplicationDomainEnum>
  </ApplicationDomain>
  <LifeCycleStage>
    <ProxyString>design</ProxyString>
  </LifeCycleStage>
</ViewContext>
```

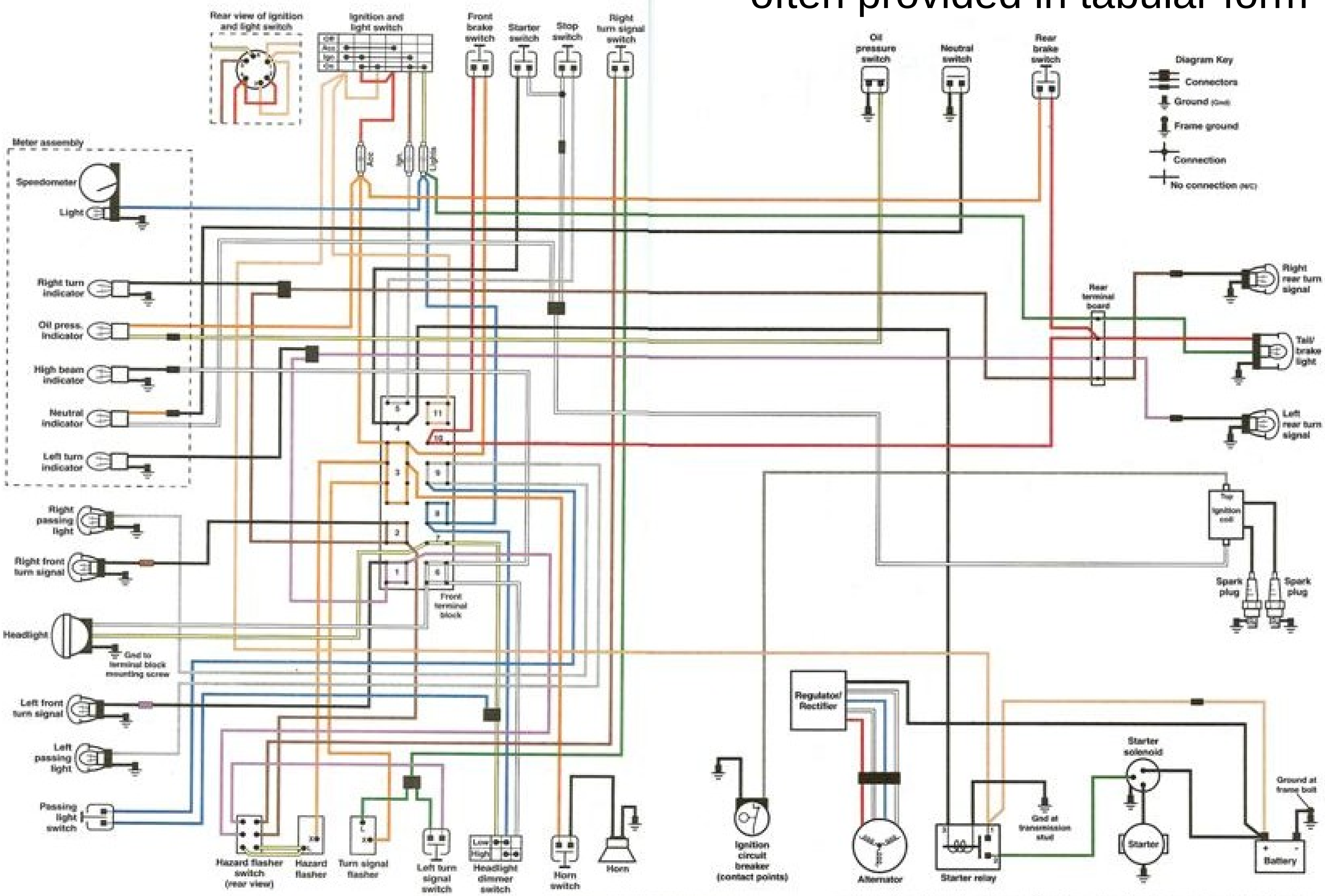
Example: ApplicationDomain *wiring_harness_segment_topology*

- consists of harness segments and nodes in a topological defined arrangement
- consists of all the connectors and splices
- consists of special partial coatings, shields, fixture places ...
- no 2D or 3D geometry, no wires & cables, no connectivity



Example: ApplicationDomain *wire_and_part_list_with_connectivity*

- captures the connectivity information in a schematics diagram, which wires and connector/device terminals are connected,
  but not the lines & symbols the diagram is made of
- no information which wires are bundled together into a harness segment
  - often provided in tabular form

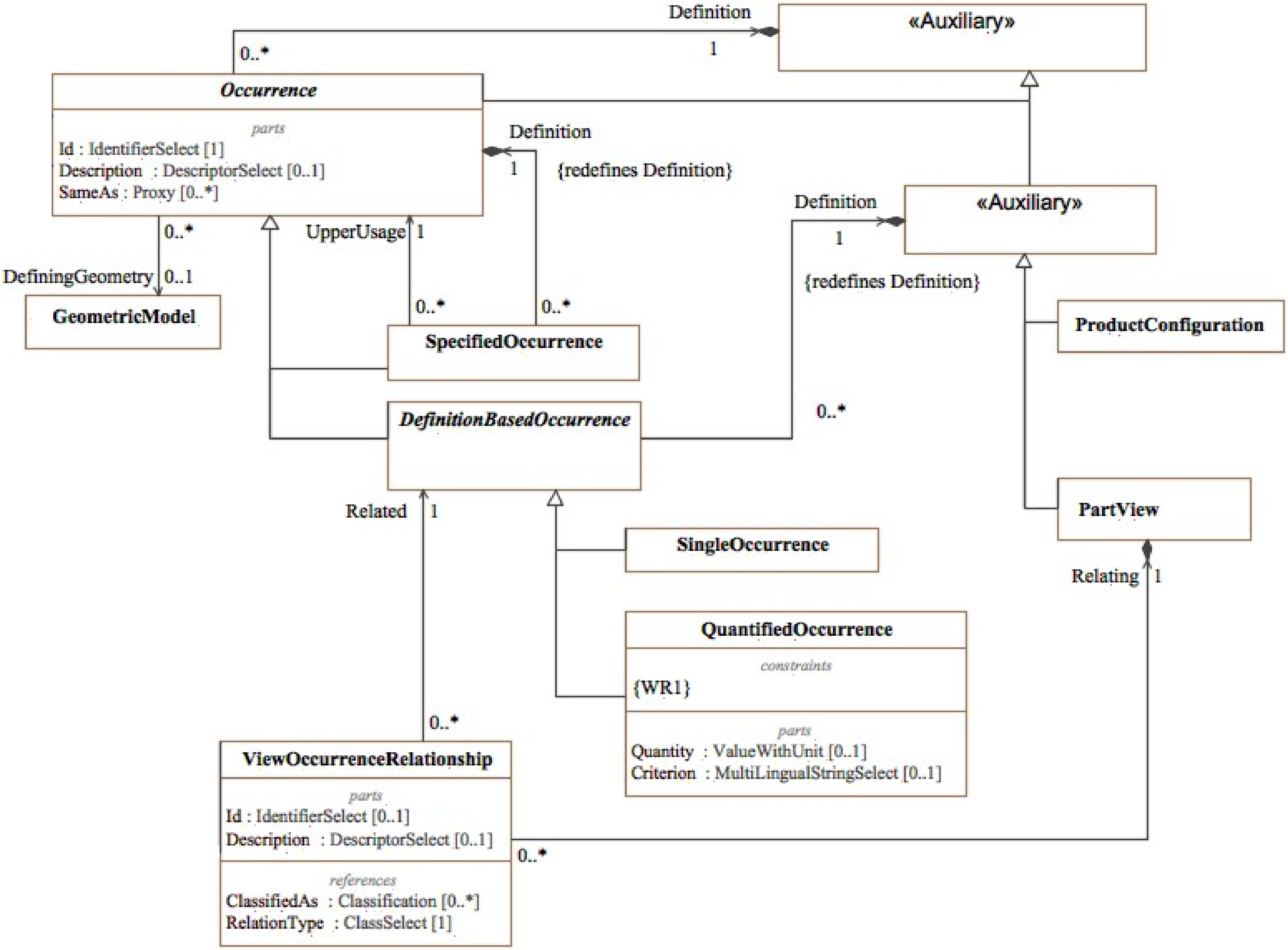


1973-1974 FL/FLH 1200

Quelle: Clymer / Harley-Davidson / Shovelheads 1966 - 1984

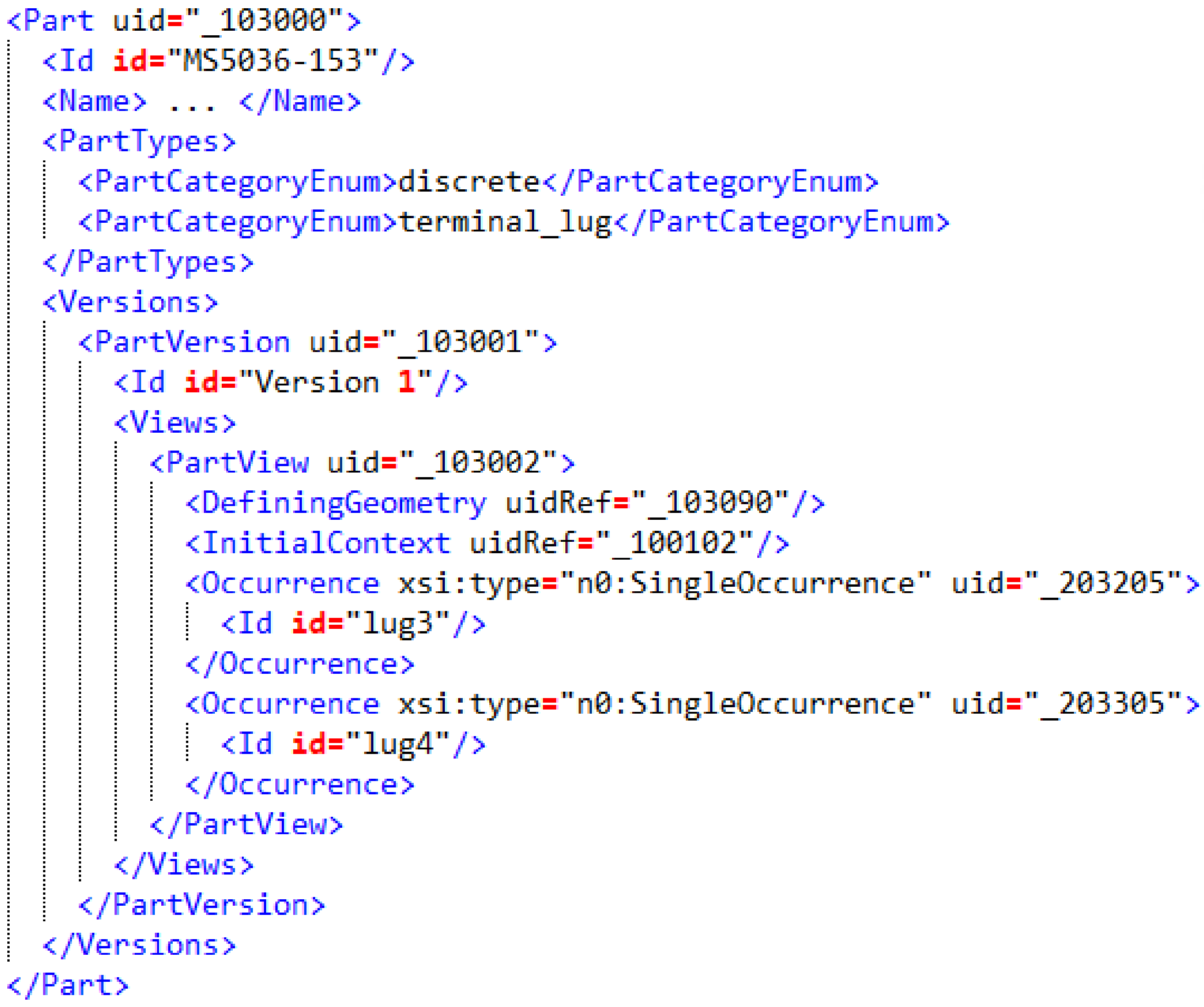
Domain Model: *Occurrence*

- An Occurrence is a member of a product structure that is defined either by a PartView, a ProductConfiguration or indirectly by another Occurrence
- An Occurrence is either a SingleO..., a QuantifiedO.., or a SpecifiedO…. (for hierachy)



Example: *SingleOccurrence*

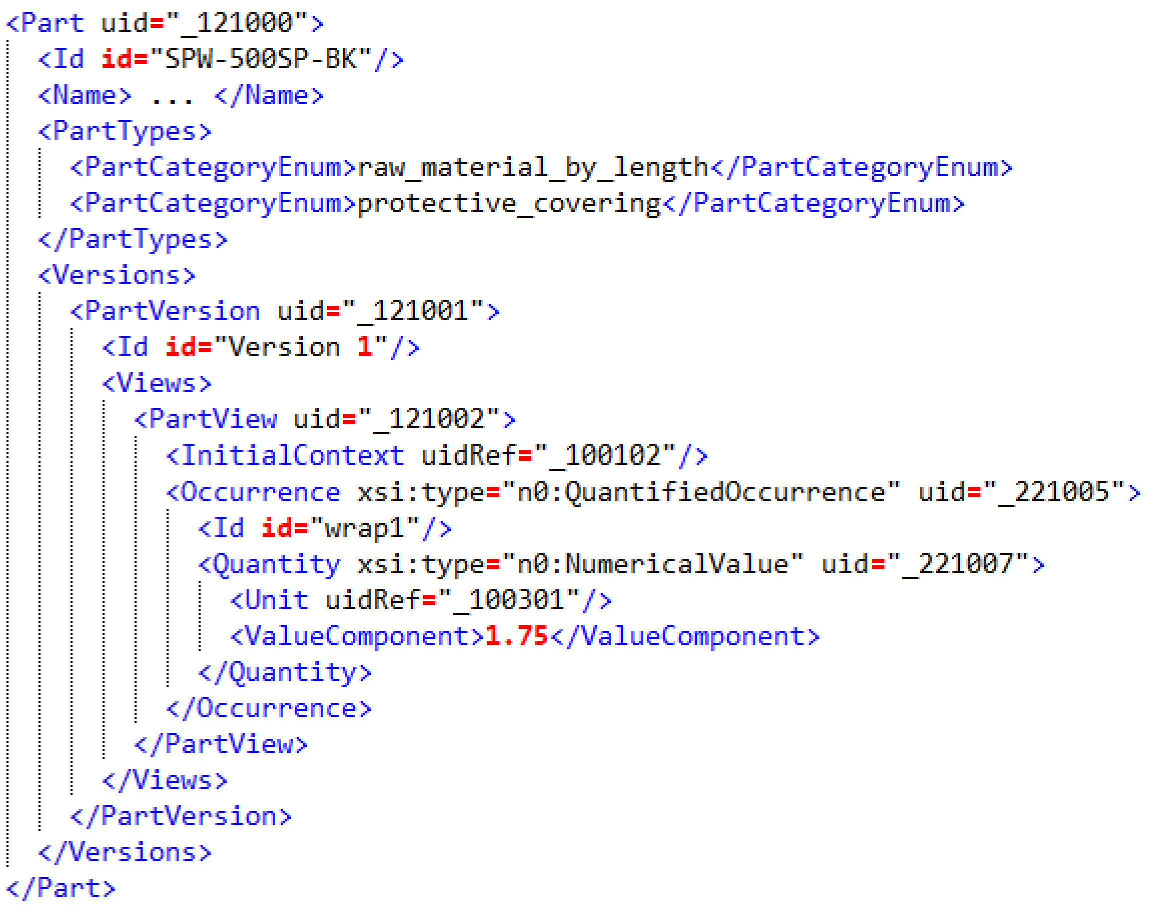
- Definition of a discrete part that is categorized as terminal lug with two occurrences, “lug3” and “lug4”



```
<Part uid="_103000">
  <Id id="MS5036-153"/>
  <Name> ... </Name>
  <PartTypes>
    <PartCategoryEnum>discrete</PartCategoryEnum>
    <PartCategoryEnum>terminal_lug</PartCategoryEnum>
  </PartTypes>
  <Versions>
    <PartVersion uid="_103001">
      <Id id="Version 1"/>
      <Views>
        <PartView uid="_103002">
          <DefiningGeometry uidRef="_103090"/>
          <InitialContext uidRef="_100102"/>
          <Occurrence xsi:type="n0:SingleOccurrence" uid="_203205">
            <Id id="lug3"/>
          </Occurrence>
          <Occurrence xsi:type="n0:SingleOccurrence" uid="_203305">
            <Id id="lug4"/>
          </Occurrence>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

Example: *QuantifiedOccurrence*

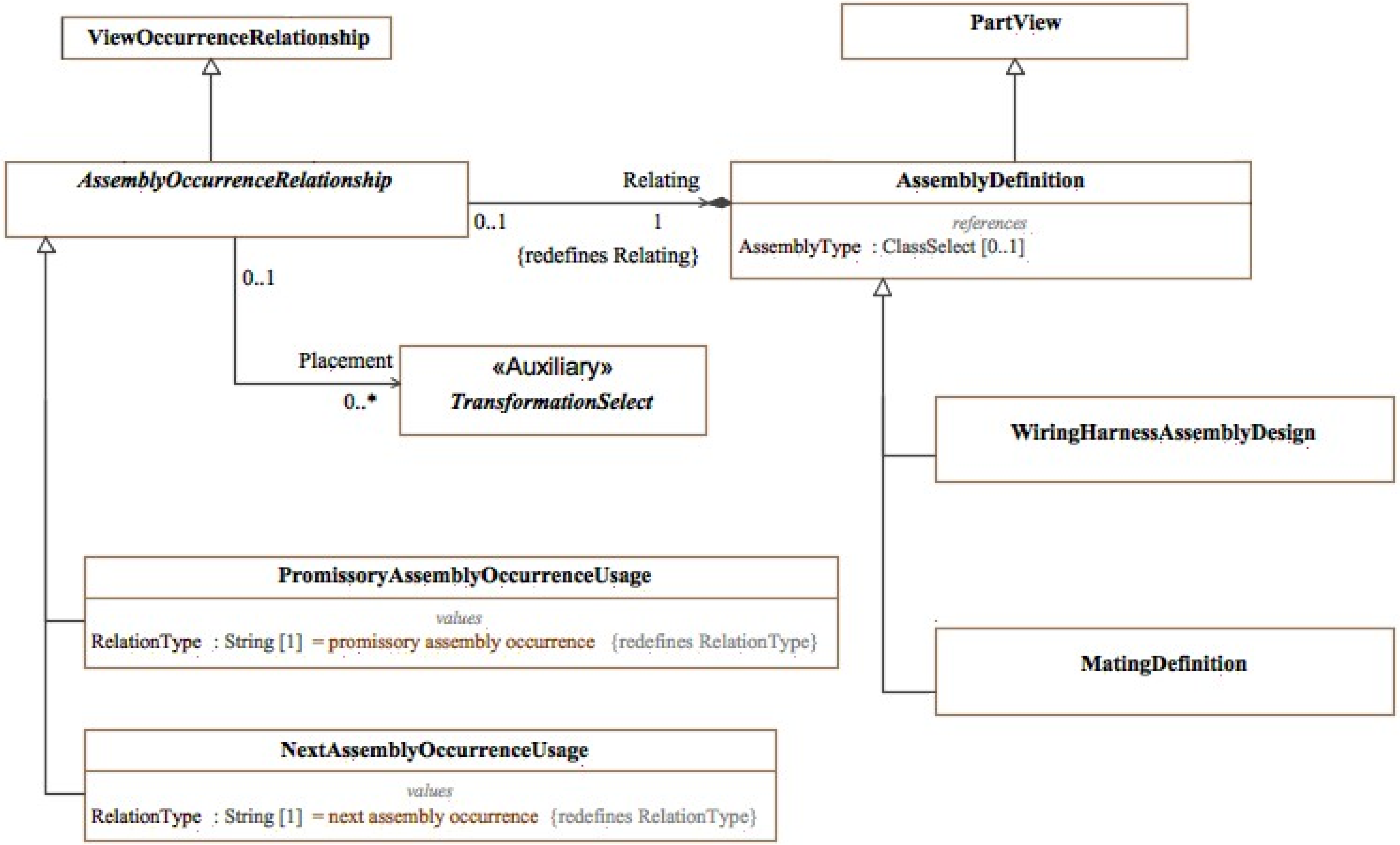
- A raw material part that is measured by length and categorized as “protective covering”
- The QuantifiedOccurrence “wrap1” of this part has a value of “1.75”, the reference length unit is metre



```
<Part uid="_121000">
  <Id id="SPW-500SP-BK"/>
  <Name> ... </Name>
  <PartTypes>
    <PartCategoryEnum>raw_material_by_length</PartCategoryEnum>
    <PartCategoryEnum>protective_covering</PartCategoryEnum>
  </PartTypes>
  <Versions>
    <PartVersion uid="_121001">
      <Id id="Version 1"/>
      <Views>
        <PartView uid="_121002">
          <InitialContext uidRef="_100102"/>
          <Occurrence xsi:type="n0:QuantifiedOccurrence" uid="_221005">
            <Id id="wrap1"/>
            <Quantity xsi:type="n0:NumericalValue" uid="_221007">
              <Unit uidRef="_100301"/>
              <ValueComponent>1.75</ValueComponent>
            </Quantity>
          </Occurrence>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

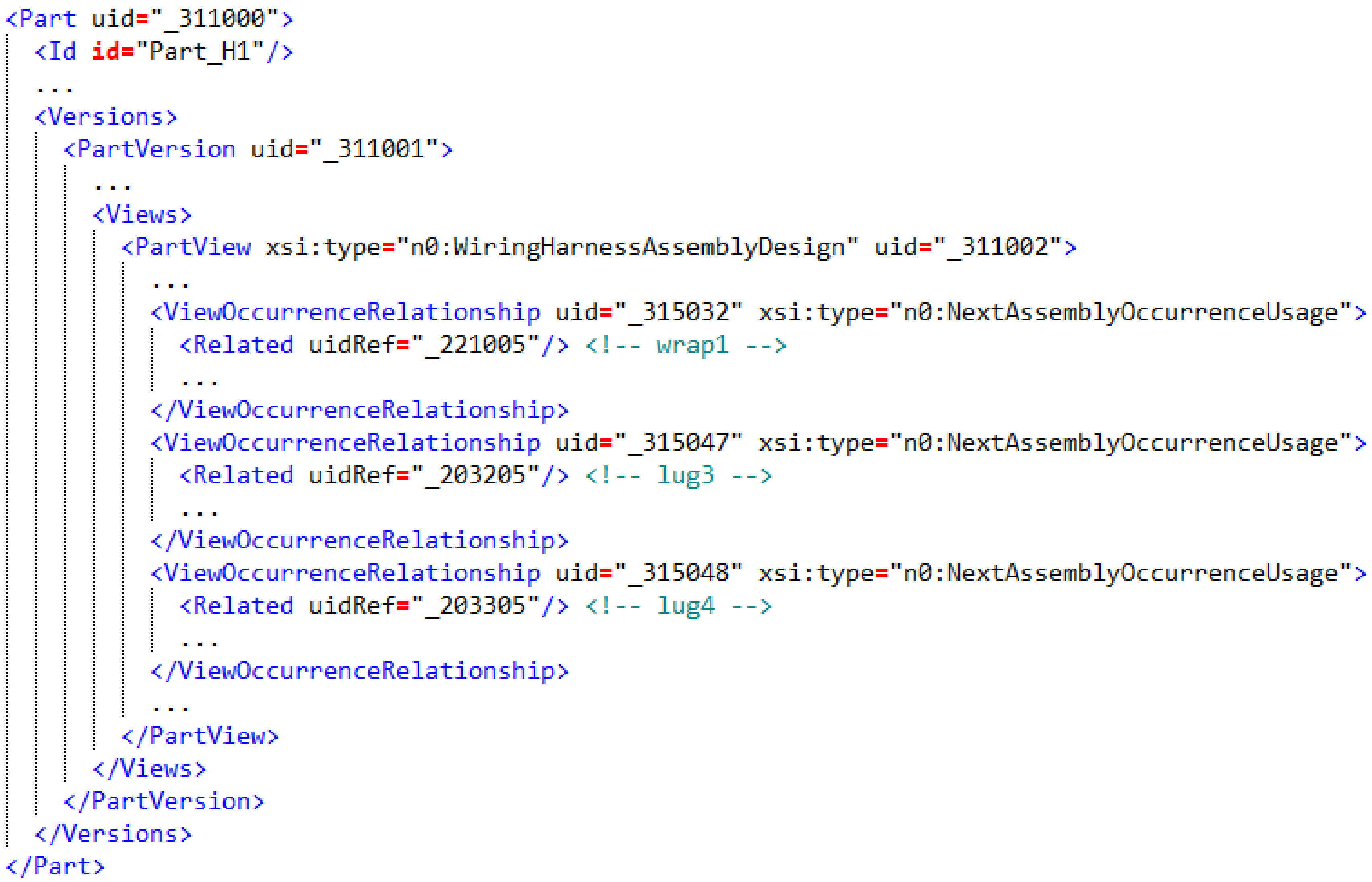
Domain Model: *AssemblyDefinition*

- An *AssemblyDefinition* is a type of *PartView* that contains *Occurrences* (single or quantified) through either
  - *NextAssemblyOccurrenceUsage* (default) or
  - *PromissoryAssemblyOccurrenceUsage* (special, e.g. “Poke Home”)
- Focus for this tutorial is subtype *WiringHarnessAssemblyDesign*


ViewOccurrenceRelationship
PartView
AssemblyOccurrenceRelationship
Relating
0..1
1
{redefines Relating}
AssemblyDefinition
references
AssemblyType : ClassSelect [0..1]
0..1
Placement
0..*
«Auxiliary»
TransformationSelect
WiringHarnessAssemblyDesign
PromissoryAssemblyOccurrenceUsage
values
RelationType : String [1] = promissory assembly occurrence {redefines RelationType}
MatingDefinition
NextAssemblyOccurrenceUsage
values
RelationType : String [1] = next assembly occurrence {redefines RelationType}

Example: Simple/Flat Assembly Structure

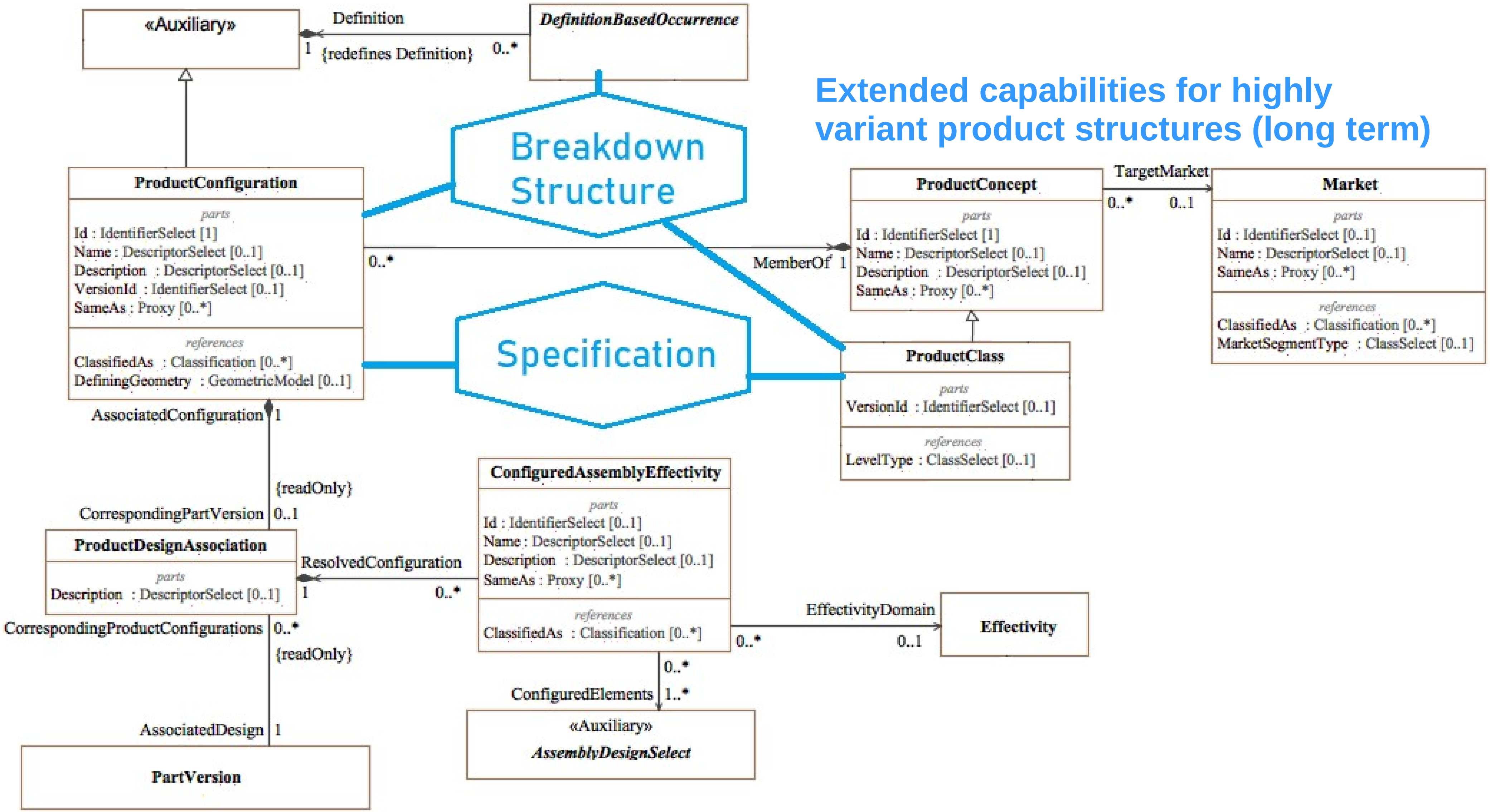
- The *QuantifiedOccurrence* “wrap1” and the *SingleOccurrences* “lug3” and “lug4” become members of the “Part_H1” assembly that is a *WiringHarnessAssemblDesign*



```xml
<Part uid="_311000">
  <Id id="Part_H1"/>
  ...
  <Versions>
    <PartVersion uid="_311001">
      ...
      <Views>
        <PartView xsi:type="n0:WiringHarnessAssemblyDesign" uid="_311002">
          ...
          <ViewOccurrenceRelationship uid="_315032" xsi:type="n0:NextAssemblyOccurrenceUsage">
            <Related uidRef="_221005"/> <!-- wrap1 -->
            ...
          </ViewOccurrenceRelationship>
          <ViewOccurrenceRelationship uid="_315047" xsi:type="n0:NextAssemblyOccurrenceUsage">
            <Related uidRef="_203205"/> <!-- lug3 -->
            ...
          </ViewOccurrenceRelationship>
          <ViewOccurrenceRelationship uid="_315048" xsi:type="n0:NextAssemblyOccurrenceUsage">
            <Related uidRef="_203305"/> <!-- lug4 -->
            ...
          </ViewOccurrenceRelationship>
          ...
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

Domain Model: *ProductConfiguration*

- An *Occurrence* can also be defined by a *ProductConfiguration* that is a member of a *ProductConcept*/*ProductClass*. This shall be used when a generic product is used without having a particular Part-Number (near term)
- *ProductConfiguration* & *ProductDesignAssociation* are also used for variant product structures together with *ConfiguredAssemblyEffectivity* (mid term)

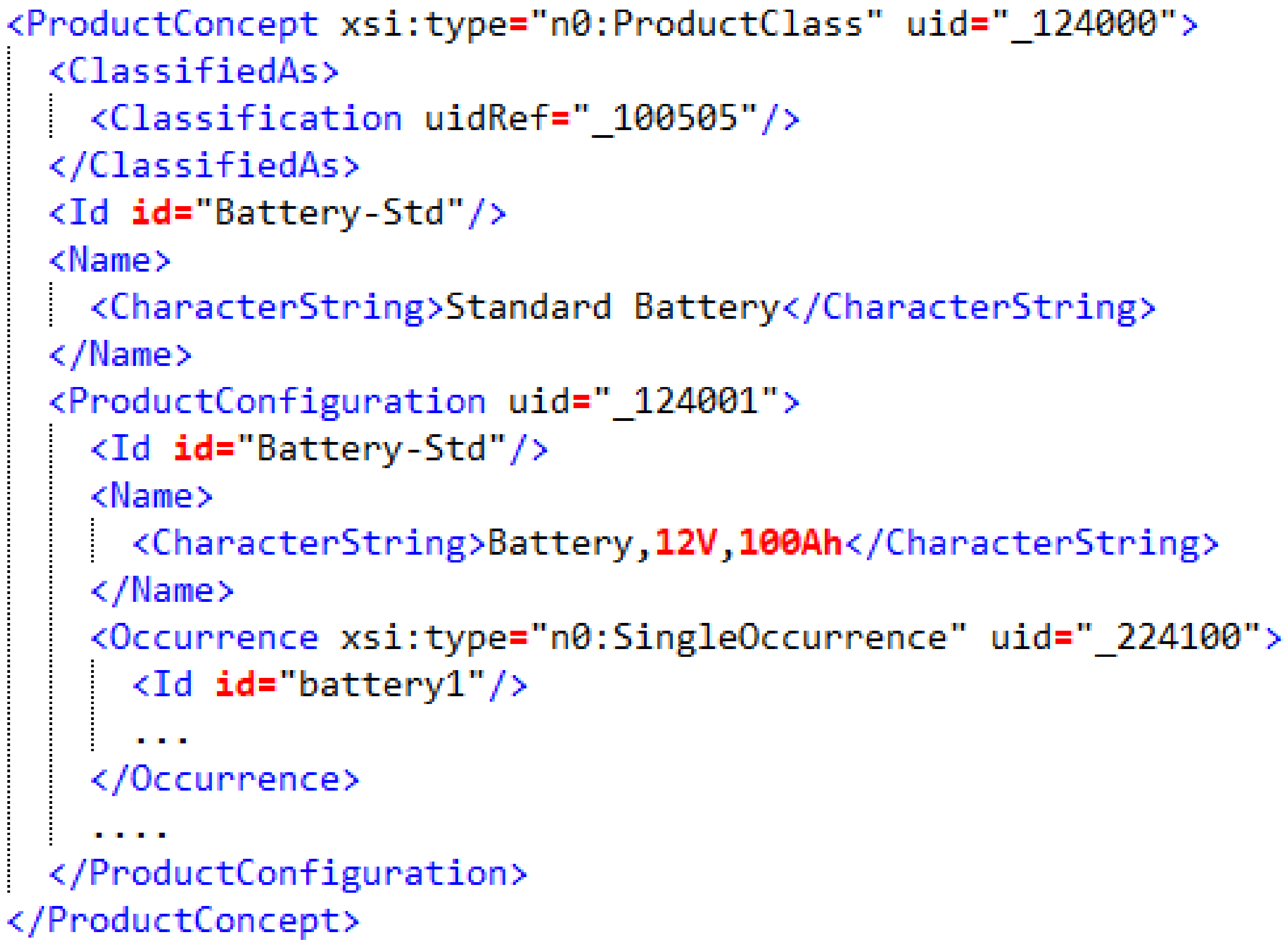


Extended capabilities for highly variant product structures (long term)

«Auxiliary»
Definition
1 {redefines Definition}
0..*
DefinitionBasedOccurrence
Breakdown Structure
ProductConfiguration
parts
Id : IdentifierSelect [1]
Name : DescriptorSelect [0..1]
Description : DescriptorSelect [0..1]
VersionId : IdentifierSelect [0..1]
SameAs : Proxy [0..*]
references
ClassifiedAs : Classification [0..*]
DefiningGeometry : GeometricModel [0..1]
0..*
MemberOf 1
ProductConcept
parts
Id : IdentifierSelect [1]
Name : DescriptorSelect [0..1]
Description : DescriptorSelect [0..1]
SameAs : Proxy [0..*]
TargetMarket
0..*
0..1
Market
parts
Id : IdentifierSelect [0..1]
Name : DescriptorSelect [0..1]
SameAs : Proxy [0..*]
references
ClassifiedAs : Classification [0..*]
MarketSegmentType : ClassSelect [0..1]
Specification
ProductClass
parts
VersionId : IdentifierSelect [0..1]
references
LevelType : ClassSelect [0..1]
AssociatedConfiguration 1
{readOnly}
CorrespondingPartVersion 0..1
ProductDesignAssociation
parts
Description : DescriptorSelect [0..1]
ResolvedConfiguration
1
0..*
ConfiguredAssemblyEffectivity
parts
Id : IdentifierSelect [0..1]
Name : DescriptorSelect [0..1]
Description : DescriptorSelect [0..1]
SameAs : Proxy [0..*]
references
ClassifiedAs : Classification [0..*]
EffectivityDomain
0..*
0..1
Effectivity
CorrespondingProductConfigurations 0..*
{readOnly}
0..*
ConfiguredElements 1..*
AssociatedDesign 1
PartVersion
«Auxiliary»
AssemblyDesignSelect

Example: *SingleOccurrence* defined by a *ProductConfiguration*

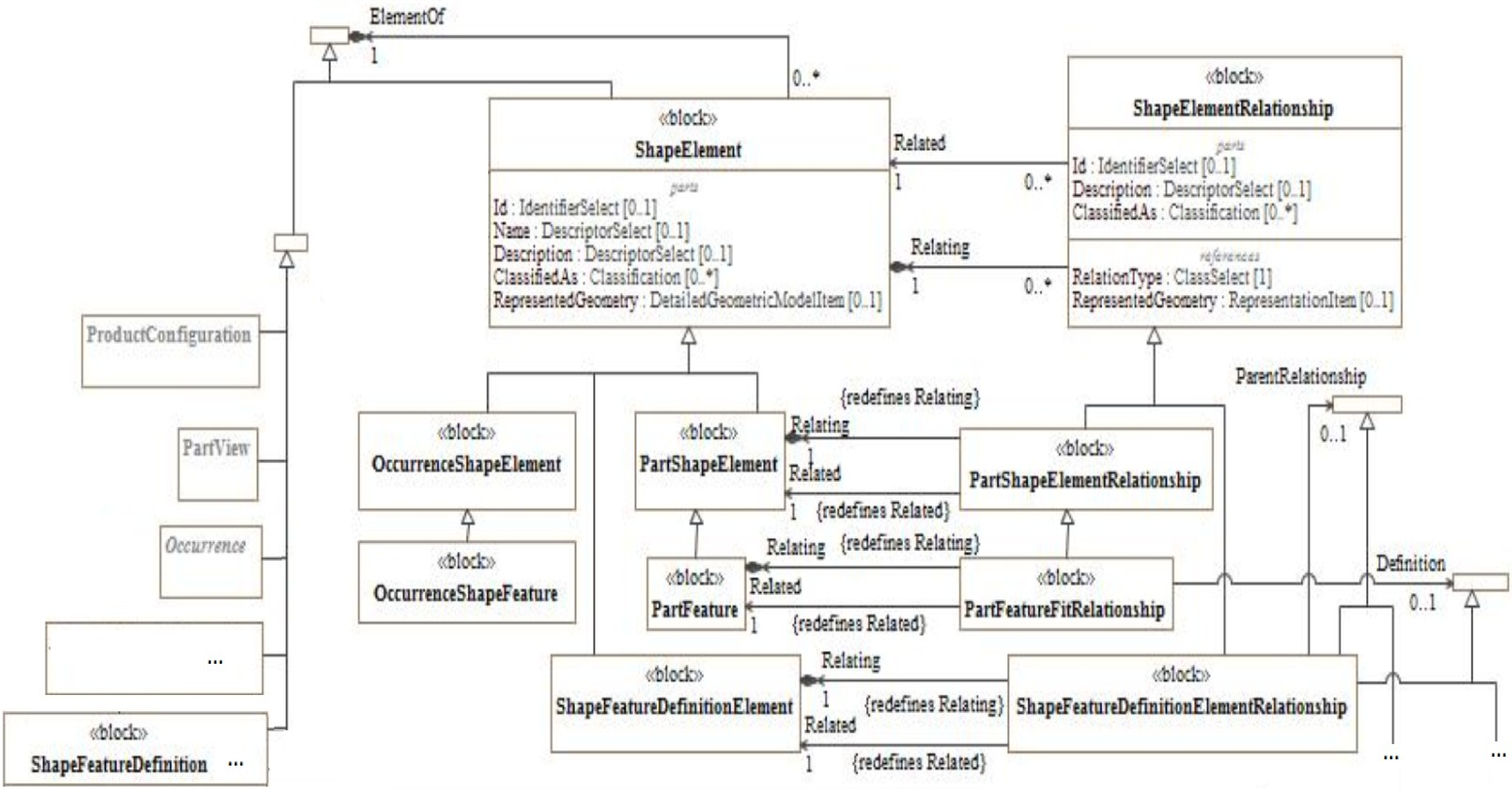
- A *SingleOccurrence* “battery1” for a standard battery without knowing a particular part number.



```xml
<ProductConcept xsi:type="n0:ProductClass" uid="_124000">
  <ClassifiedAs>
    <Classification uidRef="_100505"/>
  </ClassifiedAs>
  <Id id="Battery-Std"/>
  <Name>
    <CharacterString>Standard Battery</CharacterString>
  </Name>
  <ProductConfiguration uid="_124001">
    <Id id="Battery-Std"/>
    <Name>
      <CharacterString>Battery,12V,100Ah</CharacterString>
    </Name>
    <Occurrence xsi:type="n0:SingleOccurrence" uid="_224100">
      <Id id="battery1"/>
      ...
    </Occurrence>
    ....
  </ProductConfiguration>
</ProductConcept>
```

Domain Model: *ShapeElement*

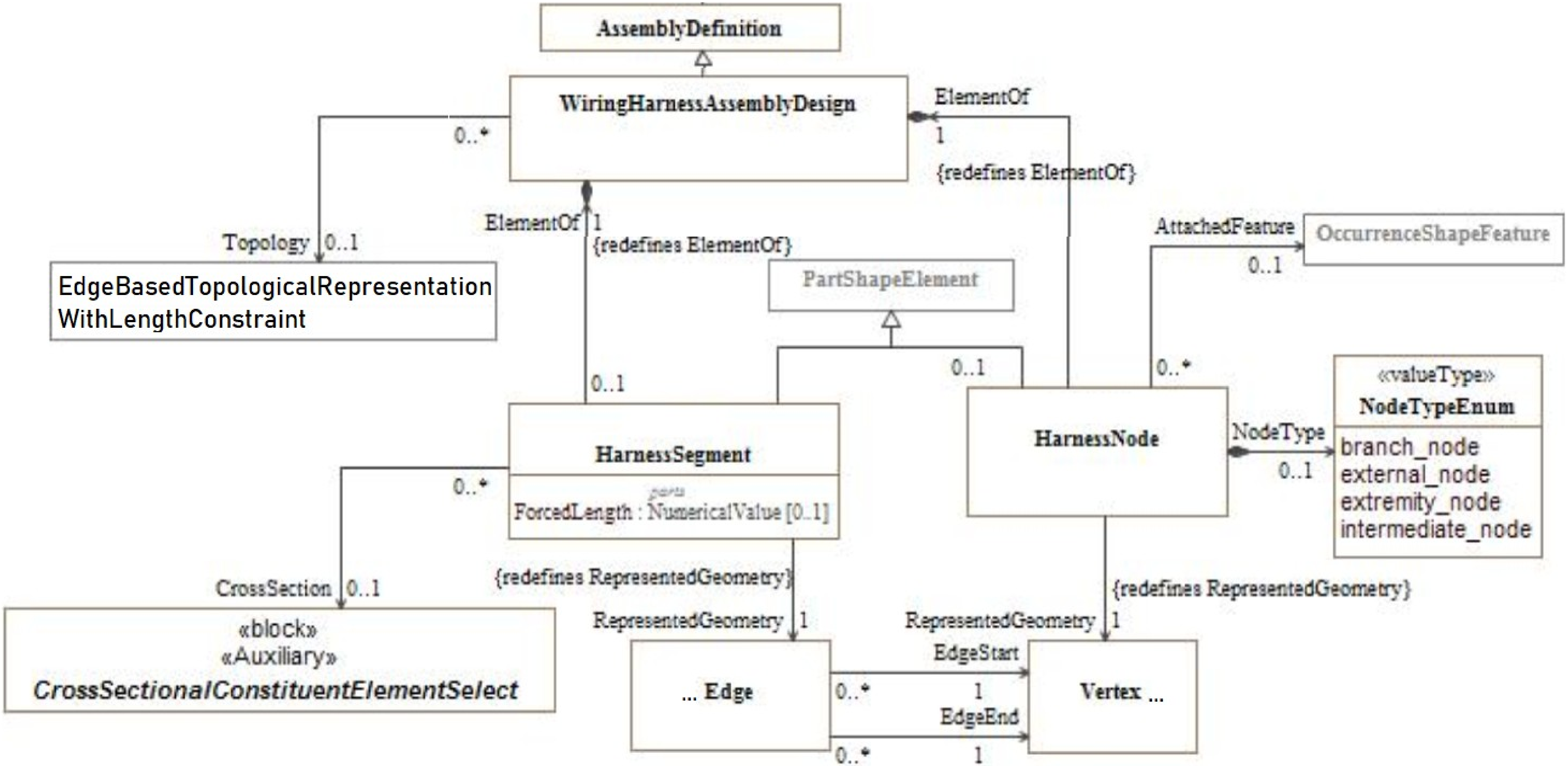
- A *ShapeElement* is the identification of an element of the shape of a *ProductConfiguration*, *PartView*, *Occurrence*, *ShapeFeatureDefinition* or of another *ShapeElement* or ...
- some subtypes of *ShapeElement* might be defined by a *ShapeFeatureDefinition* or another *ShapeElement*
- A *XxxFeature*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



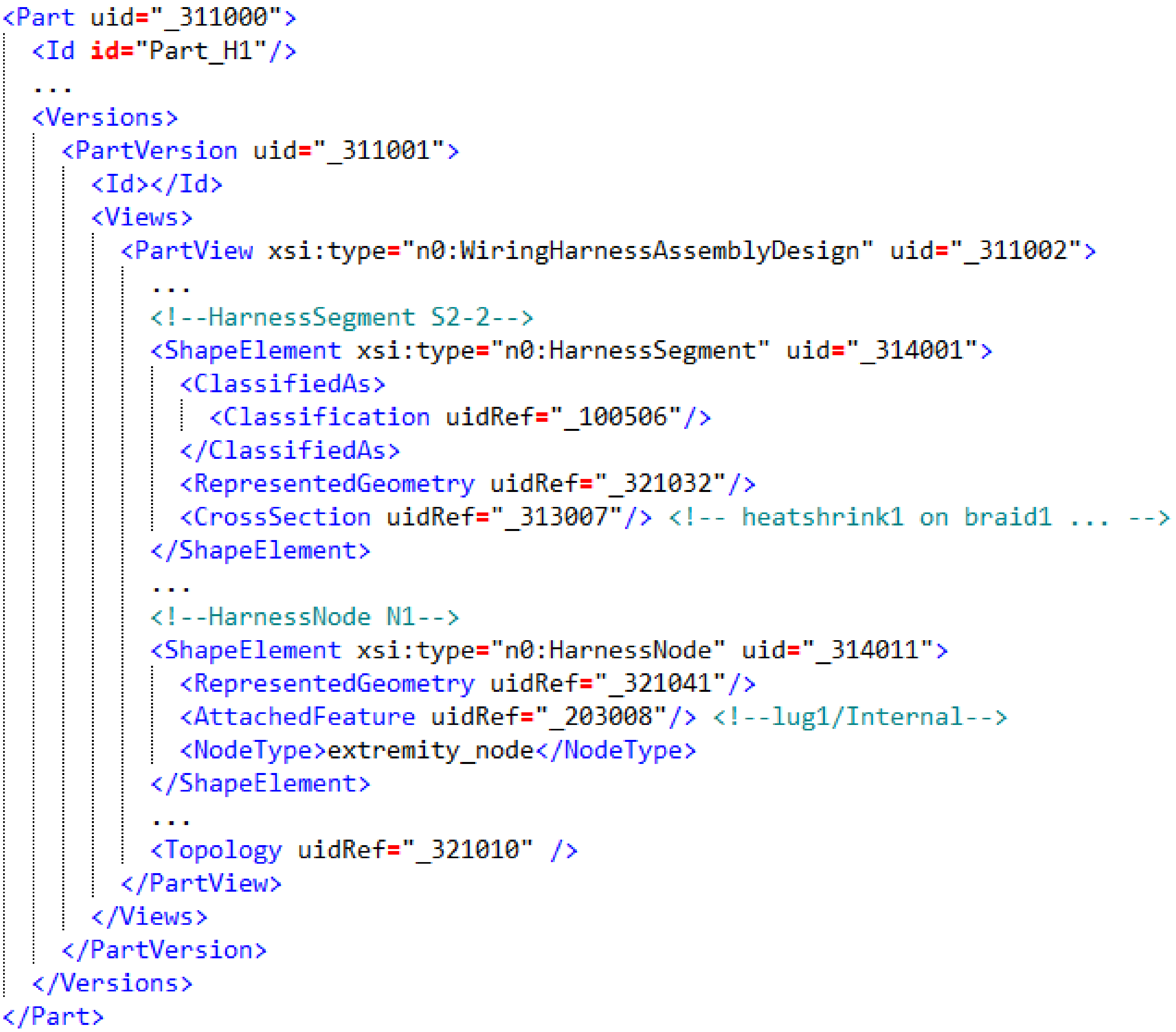
ElementOf
1
0..*
«block»
ShapeElement
parts
Id : IdentifierSelect [0..1]
Name : DescriptorSelect [0..1]
Description : DescriptorSelect [0..1]
ClassifiedAs : Classification [0..*]
RepresentedGeometry : DetailedGeometricModelItem [0..1]
Related
1
0..*
Relating
1
0..*
«block»
ShapeElementRelationship
parts
Id : IdentifierSelect [0..1]
Description : DescriptorSelect [0..1]
ClassifiedAs : Classification [0..*]
references
RelationType : ClassSelect [1]
RepresentedGeometry : RepresentationItem [0..1]
ProductConfiguration
PartView
Occurrence
...
«block»
ShapeFeatureDefinition ...
ParentRelationship
0..1
{redefines Relating}
«block»
OccurrenceShapeElement
«block»
PartShapeElement
Relating
1
Related
1
{redefines Related}
«block»
PartShapeElementRelationship
«block»
OccurrenceShapeFeature
Relating
{redefines Relating}
«block»
PartFeature
Related
1
{redefines Related}
«block»
PartFeatureFitRelationship
Definition
0..1
«block»
ShapeFeatureDefinitionElement
Relating
1
{redefines Relating}
Related
1
{redefines Related}
«block»
ShapeFeatureDefinitionElementRelationship
...
...

Domain Model: *WiringHarnessAssemblyDesign*

- A *WiringHarnessAssemblyDesign* is an electrical assembly with a topological representation defined by a *EdgeBasedTopologicalRepresentationWithLengthConstraint*
- *PartShapeElements* of type *HarnessSegment* and *HarnessNode* of an EWH are represented by subtypes of the topological elements *Edge* and *Vertex* respectively
- An *OccurrenceShapeFeature* (e.g. of a connector Occurrence) can be attached to a *HarnessNode*
- The detailed arrangement of the wires, cables, coverings ... in a *HarnessSegment* can be defined by a *CrossSectionalConstituentElementSelect*



Example: *WiringHarnessAssemblyDesign* with *Topology, HarnessSegment* and *HarnessNode*



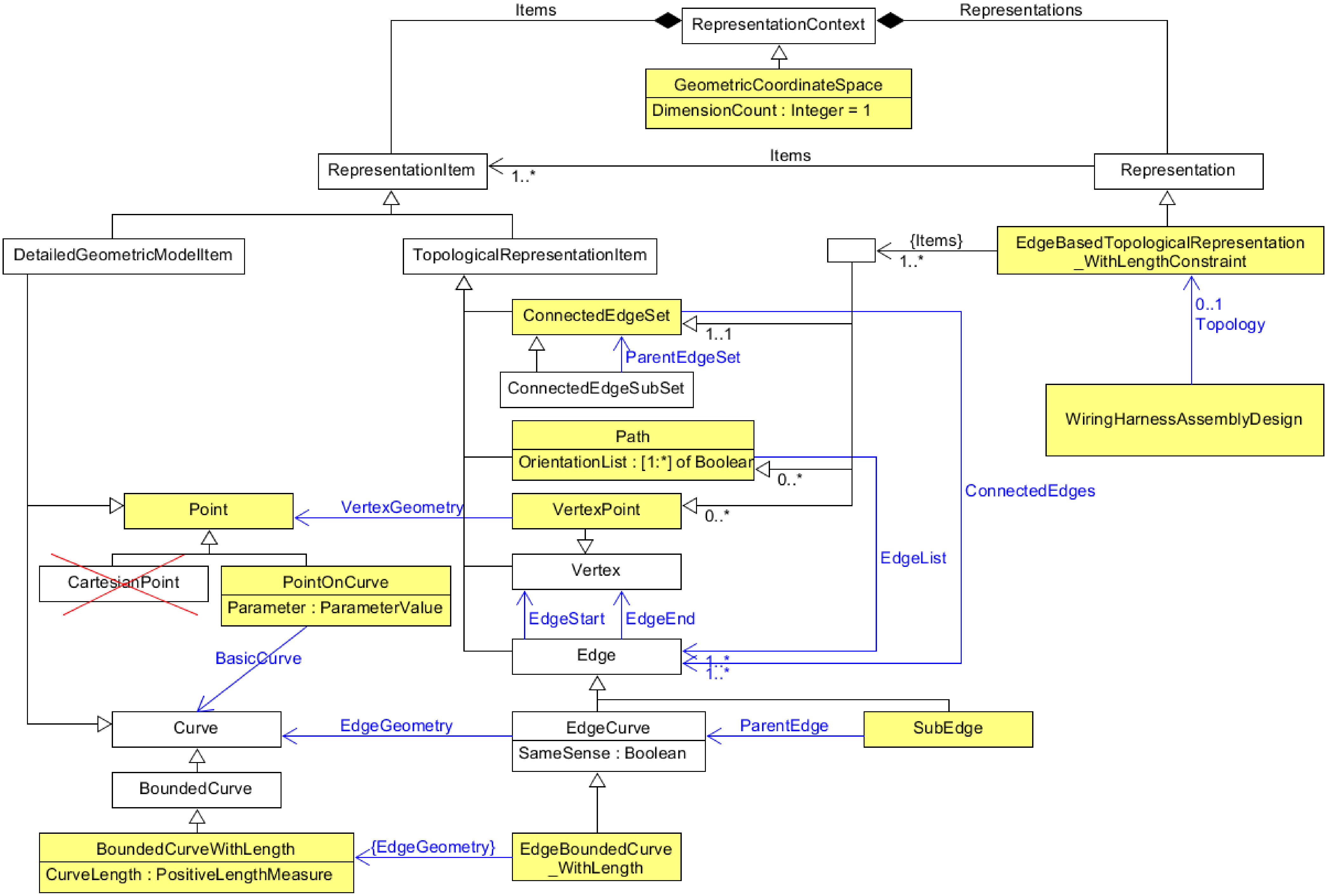
```xml
<Part uid="_311000">
  <Id id="Part_H1"/>
  ...
  <Versions>
    <PartVersion uid="_311001">
      <Id></Id>
      <Views>
        <PartView xsi:type="n0:WiringHarnessAssemblyDesign" uid="_311002">
          ...
          <!--HarnessSegment S2-2-->
          <ShapeElement xsi:type="n0:HarnessSegment" uid="_314001">
            <ClassifiedAs>
              <Classification uidRef="_100506"/>
            </ClassifiedAs>
            <RepresentedGeometry uidRef="_321032"/>
            <CrossSection uidRef="_313007"/> <!-- heatshrink1 on braid1 ... -->
          </ShapeElement>
          ...
          <!--HarnessNode N1-->
          <ShapeElement xsi:type="n0:HarnessNode" uid="_314011">
            <RepresentedGeometry uidRef="_321041"/>
            <AttachedFeature uidRef="_203008"/> <!--lug1/Internal-->
            <NodeType>extremity_node</NodeType>
          </ShapeElement>
          ...
          <Topology uidRef="_321010" />
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

Changes on topology model for EWH

These changes on the published AP242 ed2 (2020) will be covered in an upcoming amendment:

- rename: ConnectedEdgeWithLengthSetRepresentation => EdgeBasedTopologicalRepresentationWithLengthConstraint
- use RepresentationContext subtype GeometricCoordinateSpace (as “1D”)
- replace: EdgeWithLength => EdgeBoundedCurveWithLength + BoundedCurveWithLength
- replace: Vertex => VertexPoint + Point (not CartesianPoint)
- replace: VertexOnEdge => VertexPoint + PointOnCurve (by parameter on BoundedCurveWithLength)

EWH Topology Model (1)

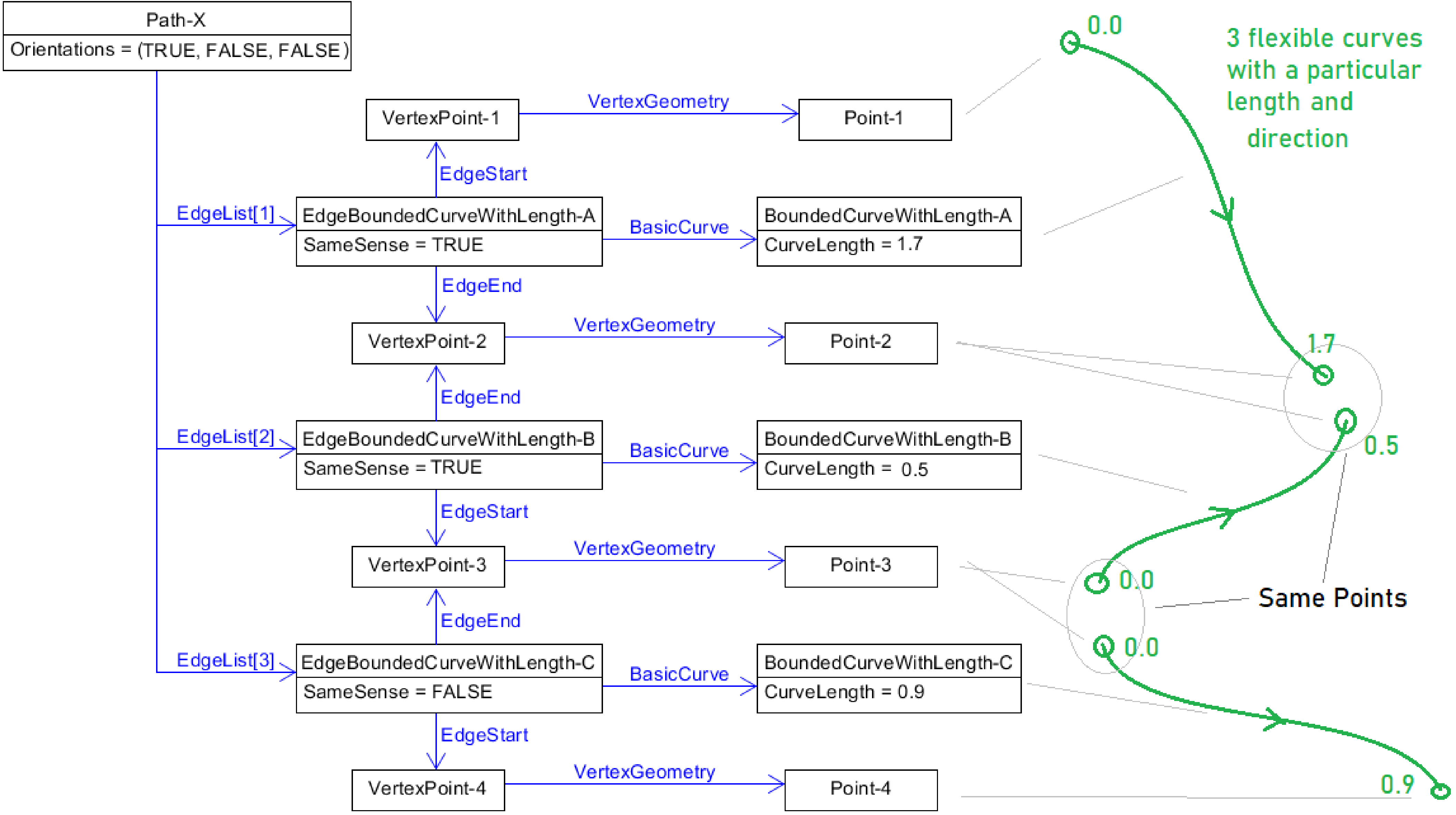


EWH Topology Model (2)

- An *EdgeBasedTopologicalRepresentationWithLengthConstraint* requires a context that is a *GeometricCoordinateSpace*.
- *GeometricCoordinateSpace* requires to specify a *DimensionCount*. As 2 (=2D) and 3 (=3D) are not suitable and the value must be greater 0, we recommend to use the value 1. But it must be clear that this does NOT mean that this is a 1D geometric coordinate space. There are e.g. no *CartesianPoints* with a single coordinate value (x-axis).
- An *EdgeBasedTopologicalRepresentationWithLengthConstraint* requires exactly one *ConnectedEdgeSet*.
- The *ConnectedEdgeSet* might be *ConnectedEdgeSubSet* of another *ConnectedEdgeSet* of another *EdgeBasedTopologicalRepresentationWithLengthConstraint* of another *WiringHarnessAssemblyDesign*. This extended capability supports the split of big projects, but is out of scope of current tests.
- The connected *Edges* of a *ConnectedEdgeSet* must all be of type *EdgeBoundedCurveWithLength* or in the case of a *ConnectedEdgeSubSet* might also be of type *SubEdge*.
- An *EdgeBoundedCurveWithLength* refers to a *BoundedCurveWithLength* that is a curve where only the length is known, but not any *CartesianPoint* or *Direction*. The start- and end-point of the *BoundedCurveWithLength* is identified by the *VertexPoints* that are referenced as *EdgeStart* and *EdgePoint*. For an *EdgeBoundedCurveWithLength* the *VertexGeometry* must be just *Point*. Subtypes such as *CartesianPoint* or *PointOnCurve* are not allowed.
- Other than the single *ConnectedEdgeSet*, an *EdgeBasedTopologicalRepresentationWithLengthConstraint* might have additional items of type *Path* or *VertexPoint*,
  - these additional *Paths* and *VertexPoints* must all be in the domain of the *ConnectedEdgeSet*.
  - *Paths* are used to define the flexible geometry of *QuantifiedOccurrences* with length such as wires, cables and protections
  - additional VertexPoints are used to define the position of clamps and splices,
    Note that additional VertexPoints that are needed to define SubEdges that are used by Paths (e.g. for protections) do not need to be added as they are founded already through the included Paths.
  - additional VertexPoints must refer to PointOnCurve where the underlying curve is one of the *BoundedCurveWithLength*
  - *Paths may be composed of complete EdgeBoundedCurveWithLengths or a parts of them by using SubEdge*

Direction control of Edges and Paths

- *the VertexPoints of an EdgeBoundedCurveWithLength corresponds to the start/end of the underlying BoundedCurveWithLengh that are given through its parametric definition*
- *common VertexPoints of several EdgeBoundedCurveWithLength constrain the underlying BoundedCurveWithLength to start/end at exactly the same Point*
- *the SameSense attribute is used to aligned the direction of an edge with the underlying curve*
- *the Orientations attribute of a path is used to align the orientation of the EdgeList elements*



Path-X
Orientations = (TRUE, FALSE, FALSE)

VertexPoint-1 —VertexGeometry→ Point-1

EdgeStart

EdgeList[1] → EdgeBoundedCurveWithLength-A
SameSense = TRUE
—BasicCurve→ BoundedCurveWithLength-A
CurveLength = 1.7

EdgeEnd

VertexPoint-2 —VertexGeometry→ Point-2

EdgeEnd

EdgeList[2] → EdgeBoundedCurveWithLength-B
SameSense = TRUE
—BasicCurve→ BoundedCurveWithLength-B
CurveLength = 0.5

EdgeStart

VertexPoint-3 —VertexGeometry→ Point-3

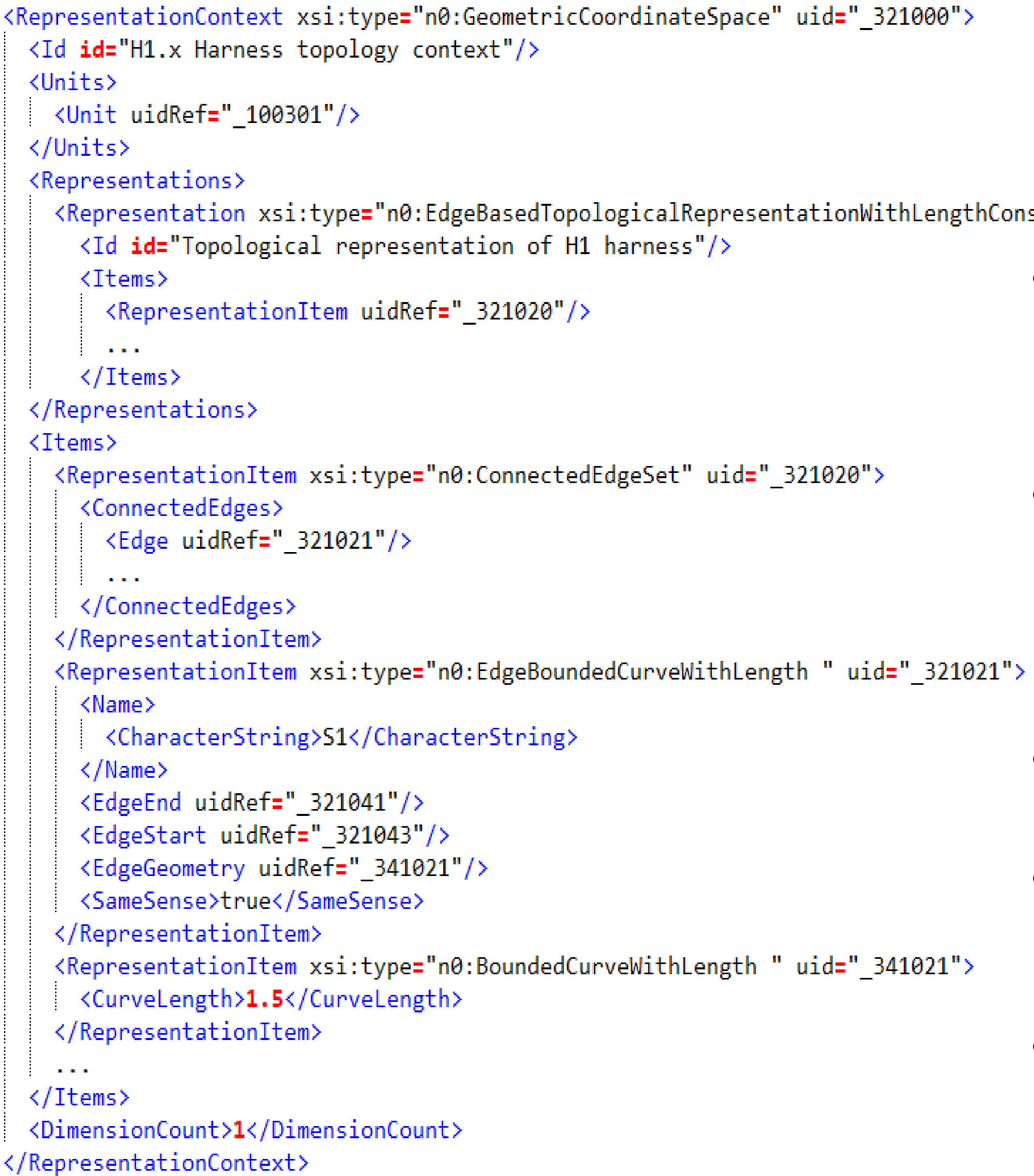
EdgeEnd

EdgeList[3] → EdgeBoundedCurveWithLength-C
SameSense = FALSE
—BasicCurve→ BoundedCurveWithLength-C
CurveLength = 0.9

EdgeStart

VertexPoint-4 —VertexGeometry→ Point-4

Example: *ConnectedEdgeWithLengthSetRepresentation*

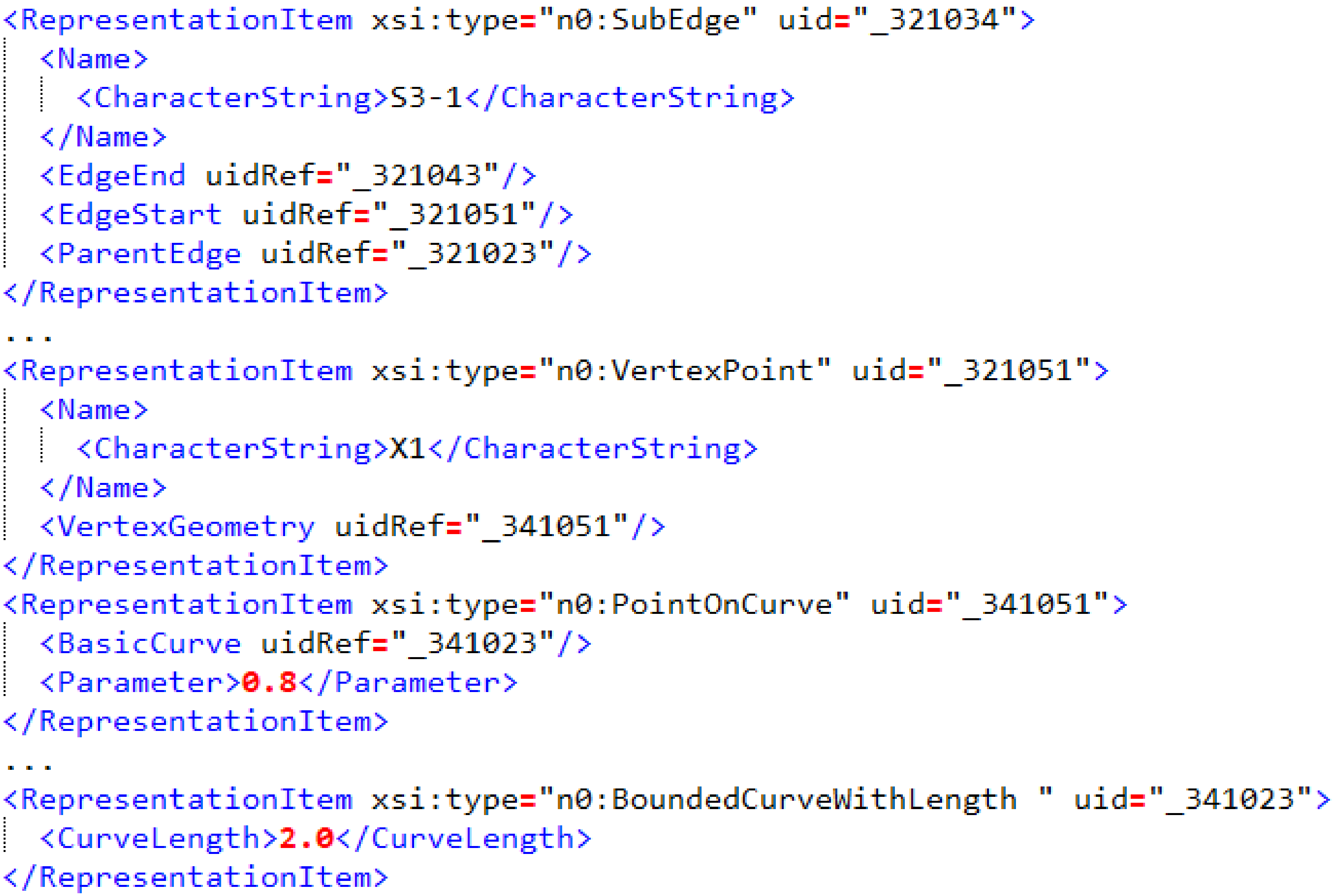


```xml
<RepresentationContext xsi:type="n0:GeometricCoordinateSpace" uid="_321000">
  <Id id="H1.x Harness topology context"/>
  <Units>
    <Unit uidRef="_100301"/>
  </Units>
  <Representations>
    <Representation xsi:type="n0:EdgeBasedTopologicalRepresentationWithLengthConstraint" uid="_321010">
      <Id id="Topological representation of H1 harness"/>
      <Items>
        <RepresentationItem uidRef="_321020"/>
        ...
      </Items>
  </Representations>
  <Items>
    <RepresentationItem xsi:type="n0:ConnectedEdgeSet" uid="_321020">
      <ConnectedEdges>
        <Edge uidRef="_321021"/>
        ...
      </ConnectedEdges>
    </RepresentationItem>
    <RepresentationItem xsi:type="n0:EdgeBoundedCurveWithLength " uid="_321021">
      <Name>
        <CharacterString>S1</CharacterString>
      </Name>
      <EdgeEnd uidRef="_321041"/>
      <EdgeStart uidRef="_321043"/>
      <EdgeGeometry uidRef="_341021"/>
      <SameSense>true</SameSense>
    </RepresentationItem>
    <RepresentationItem xsi:type="n0:BoundedCurveWithLength " uid="_341021">
      <CurveLength>1.5</CurveLength>
    </RepresentationItem>
    ...
  </Items>
  <DimensionCount>1</DimensionCount>
</RepresentationContext>
```

- a GeometricCoordinateSpace with DimensonCount 1 contains all *RepresentationItems* and *Representations*
- the *EdgeBasedTopologicalRepresentationWithLengthConstraint* references a *ConnectedEdgeSet*
- ... and that references an *EdgeBoundedCurveWithLength*
- ... that references a *BoundedCurveWithLength* that has a length of 1.5 m
- *VertexPoints* not shown here

Example: Defining SubEdges

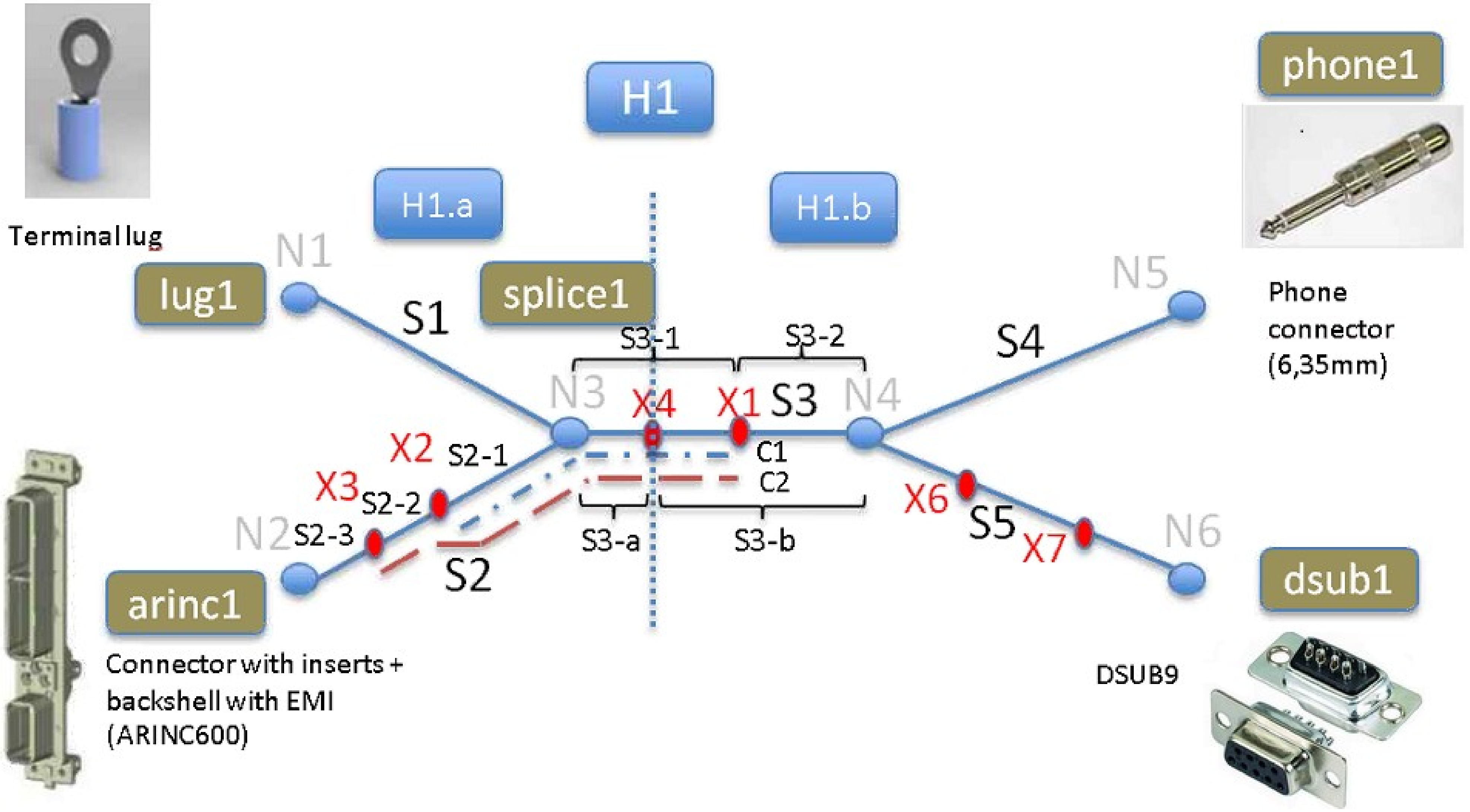
- A *SubEdge* references a *VertexPoint* whose geometry is defined by a *PointOnCurve* that is placed in 2m distance from the start of the underlying *BoundedCurveWithLength*
- Limitation: There is no way to define the distance from the end



```
<RepresentationItem xsi:type="n0:SubEdge" uid="_321034">
  <Name>
    <CharacterString>S3-1</CharacterString>
  </Name>
  <EdgeEnd uidRef="_321043"/>
  <EdgeStart uidRef="_321051"/>
  <ParentEdge uidRef="_321023"/>
</RepresentationItem>
...
<RepresentationItem xsi:type="n0:VertexPoint" uid="_321051">
  <Name>
    <CharacterString>X1</CharacterString>
  </Name>
  <VertexGeometry uidRef="_341051"/>
</RepresentationItem>
<RepresentationItem xsi:type="n0:PointOnCurve" uid="_341051">
  <BasicCurve uidRef="_341023"/>
  <Parameter>0.8</Parameter>
</RepresentationItem>
...
<RepresentationItem xsi:type="n0:BoundedCurveWithLength " uid="_341023">
  <CurveLength>2.0</CurveLength>
</RepresentationItem>
```

Example: H1 Topology with Sub Topologies H1.a and H1.b

- Main *EdgeBoundedCurveWithLength & HarnessSegment* S1 … S5
- Main *VertexPoints/Point & HarnessNodes* N1 … N6
- Additional *VertexPoints/PointOnCurve* X1 … X7
- Additional *SubEdges & HarnessSegments* S2-1 … S3-b



EWH topics for further trainings

- Hierarchical assemblies with SpecifiedOccurrence
- Terminals & Connectivity
  the “CFI Five-Box Model for Electrical & Mechanical Connectivity / Joints”
- Electrical and mechanical assembly joints
- Multi-terminals and joints
- Assembly features & feature templates (e.g. cavity and contact types)
- Cross Section of Cables or Harness Segments
- Linking the centreline of a Cable- / WireOccurrence to a Path of the topological model
- Linking a specific AxisPlacement of a SingleOccurrence (e.g. a connector) to a Vertex of the topological model
- Linking 2D or 3D geometry models with the topological model
- Markers
- Use of private and public classification systems,
  e.g. the IEC 61360 - Common Data Dictionary or eClass
- General Properties
- ...