



**Recommended Practices**  
for  
**STEP AP242 IS Business Object Model XML**  
**Product & Assembly Structure**

*Release 1.2.0*

June 30, 2017

**Contacts:**

**Organizational**

**Jochen Boy**  
PROSTEP AG  
Dolivostraße 11  
64293 Darmstadt  
Germany

[jochen.boy@prostep.com](mailto:jochen.boy@prostep.com)

**Phil Rosché**  
ACCR, LLC.  
125 King Charles Circle  
Summerville  
SC 29485 / USA

[phil.rosche@accr-llc.com](mailto:phil.rosche@accr-llc.com)

**Frédéric Darré**  
Cimpa  
4 Avenue Didier Daurat  
31700 Blagnac  
France

[frederic.darre.external@airbus.com](mailto:frederic.darre.external@airbus.com)

**Technical**

**Guillaume Hirel**  
T-Systems  
[guillaume.hirel@t-systems.com](mailto:guillaume.hirel@t-systems.com)

**Jochen Haenisch**  
Jotne EPM  
[jochen.haenisch@jotne.com](mailto:jochen.haenisch@jotne.com)



## Table of Contents

<b>1</b>	<b>Introduction.....</b>	<b>10</b>
1.1	Document Overview .....	10
1.1.1	Goal and Objectives .....	10
1.1.2	Scope.....	10
1.1.3	Intended Audience.....	10
1.1.4	Intended Use.....	10
1.1.5	Document Style.....	11
1.1.6	Document Structure.....	11
1.1.7	Instantiation Diagrams .....	11
1.2	Organizational Framework.....	15
1.2.1	Vendor Communities .....	15
1.2.2	User Communities .....	15
1.3	Maintenance of this Document .....	16
<b>2</b>	<b>Scope 16</b>	
2.1	In Scope .....	16
2.2	Out of scope .....	17
<b>3</b>	<b>Reference to Recommended Practices .....</b>	<b>18</b>
<b>4</b>	<b>Basic Concepts.....</b>	<b>18</b>
4.1	XML Format Specifics .....	18
4.1.1	Character Set.....	18
4.1.2	Containment vs. Referencing .....	19
4.1.3	Root Objects and DataContainer.....	20
4.1.4	Subtyping.....	20
4.1.5	Header object.....	20
4.1.6	XML context tagging .....	21
4.1.7	Schema version and document version identifications .....	22
4.1.8	Dates.....	22
4.1.9	STEP BO Model XML File Extensions .....	22
4.2	Rules for Attribute Cardinality .....	22
4.2.1	Entities and Attributes not supported by the Preprocessor .....	22
4.2.2	Entities and Attributes not supported by the Postprocessor.....	23
4.2.3	Unspecified and Optional Attribute Values .....	23
4.2.4	Derived Attributes .....	24
4.3	Uniqueness of Identifiers .....	24
4.4	Project Specific Values .....	24
4.5	Blanks in String Values.....	24
4.6	Basic Building Blocks .....	25
4.6.1	Template “ExchangeContext” .....	25
4.6.2	Template “Organization” .....	27
4.6.3	Template “Unit” .....	29
4.6.4	Template “Class”.....	30
4.6.5	Template “Classification” .....	33
4.6.6	Template “Identifier” .....	35
4.6.7	Template “Description” .....	38
4.6.8	Template “ViewContext” .....	38

4.6.9	Template “NumericalValue” .....	40
4.6.10	Template “StringValue” .....	42
4.6.11	Template “DateTime” .....	44
4.6.12	Template “Approval” .....	46
4.6.13	Template “Person” .....	48
4.6.14	Template “PersonInOrganization” .....	50
<b>5</b>	<b>Part Identification and Classification .....</b>	<b>55</b>
5.1	Templates “Part” and “Assembly” .....	55
5.1.1	Part .....	57
5.1.2	PartVersion .....	60
5.1.3	PartView .....	63
5.1.4	AssemblyDefinition .....	65
5.1.5	PartVersionRelationship .....	67
5.1.6	Identifiers at OEM and Supplier .....	70
<b>6</b>	<b>Part Properties .....</b>	<b>71</b>
6.1	Template “GeometricModel” .....	71
6.1.1	ExternalGeometricModel / ComposedGeometricModel .....	76
6.1.2	The subtype of ExternalGeometricModel .....	77
6.2	Template “PropertyAssignment” .....	78
6.3	Template “ShapeDependentProperty” .....	81
6.3.1	GeneralShapeDependentProperty .....	82
<b>7</b>	<b>Part Structure and Relationships .....</b>	<b>85</b>
7.1	Template “SingleOccurrence” .....	86
7.2	Template “SpecifiedOccurrence” .....	90
7.3	Full / Simplified Positioning Representation .....	91
7.3.1	Template “Simplified Positioning Representation” .....	92
7.3.2	Full Positioning Representations .....	96
7.4	Template “PartViewRelationship” .....	114
<b>8</b>	<b>Document Identification and Classification .....</b>	<b>120</b>
8.1	Template “Document” .....	121
8.1.1	Document .....	122
8.1.2	DocumentVersion .....	125
8.1.3	DocumentDefinition .....	127
8.2	Template “DocumentDefinitionRelationship” .....	130
8.3	Template “DocumentVersionRelationship” .....	133
<b>9</b>	<b>External Files .....</b>	<b>136</b>
9.1	Template “DigitalFile” .....	137
9.2	Template “FileRelationship” .....	142
9.3	File Structure (monolithic/nested) .....	145
<b>10</b>	<b>Document and File Properties .....</b>	<b>152</b>
10.1	Template “FormatProperty” .....	152
10.2	Template “ContentProperty” .....	155
10.3	Template “CreationProperty” .....	158
10.4	Template “SizeProperty” .....	159
10.5	Template “DocumentFileProperty” .....	161



<b>11 Document and File Association to Product Data</b> .....	<b>162</b>
11.1 Template "CAx Representation for DocumentAssignment" .....	164
11.2 Template "PDM Representation for DocumentAssignment" .....	169
<b>12 PDM Properties and CAD User-Defined Attributes</b> .....	<b>176</b>
12.1 Fundamental concepts .....	176
12.2 Template "PropertyDefinition" .....	177
12.3 Group of attributes and group of attributes values .....	180
12.4 Specifying the target for the attribute .....	182
12.4.1 Attributes at the part level .....	182
12.4.2 Attributes at component instances in an assembly .....	182
12.4.3 Attributes at the shape level .....	183
12.5 Definition of attribute value .....	183
12.5.1 Values with Unit .....	184
12.5.2 Values without Unit .....	184
12.6 Transfer of Meta-Data for the User Defined Attributes .....	185
12.6.1 Designation of the Attribute Type .....	186
<b>13 Validation Properties</b> .....	<b>186</b>
13.1 Assembly Validation Properties (Notional Solid, Number of Children).....	187
13.1.1 Number of Children.....	187
13.1.2 Notional Solids Centroid Position .....	188
13.2 Geometric Validation Properties (Repeated from referenced Parts) .....	190
13.2.1 Validation Properties for Solid Geometry .....	191
13.2.2 Validation Properties for Surface Geometry .....	194
13.2.3 Validation Properties for Curve / Wireframe Geometry .....	194
13.2.4 Bounding Box.....	195
<b>14 Outlook</b> .....	<b>195</b>
14.1 "Model-based" approach for future versions of this document.....	196
<b>15 List of Templates</b> .....	<b>196</b>
<b>Annex A XML Schema derivation from BO Model EXPRESS Schema</b> .....	<b>198</b>
A.1 General concepts .....	198
A.1.1 General concepts.....	198
A.1.2 Mapping of EXPRESS entity data types.....	198
A.1.3 Mapping of named data types .....	200
A.1.4 Mapping of SELECT data types .....	200
A.1.5 Mapping of EXPRESS attributes .....	200
A.1.6 Not mapped EXPRESS Constructs.....	204
A.1.7 Containment and referencing rules.....	204
A.1.8 Change of Direction for Associations .....	207
A.1.9 Representation of Id Attribute .....	209
A.1.10 Multilanguage Support.....	210
A.1.11 Representation of Date and Time.....	211
A.2 Unit of Serialization.....	211
A.3 XML configuration specification .....	211
<b>Annex B Known Issues</b> .....	<b>213</b>
<b>Annex C Reference Documents</b> .....	<b>215</b>

**Annex D Conversion from implicitly to explicitly defined transformation .....216**

**Annex E Recommendation for the Definition of Units .....218**

E.1 SI Base Unit Definitions .....218

E.2 SI Derived Units .....219

    E.2.1 Named units derived from SI base units ..... 219

    E.2.2 Derived quantities and units ..... 219

E.3 Unspecified Units .....221

    E.3.1 Byte ..... 221

    E.3.2 Each ..... 221

E.4 Imperial Units .....222

**Annex F AP214 AIM to AP242 BOM Mapping .....223**

**List of Figures**

Figure 1: Template "ExchangeContext" .....25

Figure 2: Template "Organization" .....27

Figure 3: Template "Unit" .....29

Figure 4: Template "Class" .....31

Figure 5: Template "Classification" .....33

Figure 6: Template "Identifier" .....35

Figure 7: Template "ViewContext" .....39

Figure 8: Template "NumericalValue" .....40

Figure 9: Template "StringValue" .....42

Figure 10: Template "DateTime" .....44

Figure 11: Template "Approval" .....46

Figure 12: Template "Person" .....49

Figure 13: Template "PersonInOrganization" .....51

Figure 14: Template "Part" .....56

Figure 15: Template "Assembly" .....57

Figure 16: Template "PartVersionRelationship" .....68

Figure 17: Template "GeometricModel" .....72

Figure 18: Template "PropertyAssignment" .....78

Figure 19: Template "ShapeDependentProperty" for either PartView or ShapeElement .....82

Figure 20: Template "SingleOccurrence" .....86

Figure 21: Template "Simplified Positioning Representation" .....93

Figure 22: Instantiation example if Part #2 has geometry .....95

Figure 23: Full Positioning Representation with Implicit Transformation .....98

Figure 24: Full Positioning Representation with Explicit Transformation ..... 103

Figure 25: Full Positioning Representation with Same Coordinate Space ..... 108

Figure 26: General Geometric Representation Relationship ..... 112

Figure 27: Template "PartViewRelationship" for properties ..... 115

Figure 28: Template "PartViewRelationship" for quantity (+ properties) ..... 116

Figure 29: Template "Document" ..... 122

Figure 30: Template "DocumentDefinitionRelationship" ..... 131

Figure 31: Template "DocumentVersionRelationship" ..... 134

Figure 32: Template "DigitalFile" ..... 137

Figure 33: Template "FileRelationship" .....	143
Figure 34: Example for Nested Structure.....	146
Figure 35: Example for Nested Structure with additional part-level XML files .....	146
Figure 36: Element Structure for Nested XML File.....	149
Figure 37: Template "FormatProperty" .....	152
Figure 38: Template "ContentProperty" .....	156
Figure 39: Template "CreationProperty" .....	159
Figure 40: Template "SizeProperty" .....	160
Figure 41: Template "DocumentFileProperty".....	161
Figure 42: Template "CAx Representation for DocumentAssignment".....	165
Figure 43: Model splitting for "CAx Representation for DocumentAssignment".....	167
Figure 44: Alternate models for "CAx Representation for DocumentAssignment".....	169
Figure 45: Template "PDM Representation for DocumentAssignment" .....	170
Figure 46: Model splitting for "PDM Representation for DocumentAssignment".....	173
Figure 47: Alternate models for "PDM Representation for DocumentAssignment".....	175
Figure 48: Definition of an attribute name and its usage.....	177
Figure 49: Template "PropertyDefinition".....	178
Figure 50: Defining a group of attributes.....	181
Figure 51: Defining a group of attribute values .....	182
Figure 52: User defined attribute at the part/assembly level .....	182
Figure 53: User defined attribute for a simple component instance in an assembly .....	183
Figure 54: User defined attribute for a shape or a portion of shape .....	183
Figure 55: Specification of meta data for an attribute and its value.....	186
Figure 56: Instantiation of AVP 'number of children' for 3 children.....	188
Figure 57: Instantiation of AVP 'notional solids centroid'.....	189
Figure 58: Geometric Validation Properties at the Part and Assembly level.....	191
Figure 59: GVP 'volume' of 3.4 cubicmetre assigned to a Part or Assembly.....	192
Figure 60: GVP 'surface area' of 2.3 squaremetre assigned to a Part or Assembly .....	193
Figure 61: GVP 'centroid' or CentreOfMass assigned to a Part or Assembly.....	194
Figure 62: Bounding Box defined by two opposing corner points.....	195

## List of Tables

Table 1: Instance Diagram Notation .....	14
Table 2: "ExchangeContext" Attributes .....	25
Table 3: "Organization" Attributes.....	28
Table 4: "Unit" Attributes .....	29
Table 5: "Class" Attributes.....	31
Table 6: "ExternalClassSystem" Attributes .....	32
Table 7: "Classification" Attributes.....	34
Table 8: "Identifier" Attributes .....	35
Table 9: "ViewContext" Attributes.....	39
Table 10: "NumericalValue" Attributes.....	41
Table 11: "StringValue" Attributes.....	43
Table 12: "DateTime" Attributes .....	45



Table 13: "ApprovalAssignment" Attributes .....	47
Table 14: "Approval" Attributes.....	48
Table 15: "Person" Attributes.....	50
Table 16: "OrganizationOrPersonInOrganizationAssignment" Attributes .....	52
Table 17: "PersonInOrganization" Attributes.....	54
Table 18: "Part" Attributes .....	59
Table 19: "PartVersion" Attributes .....	61
Table 20: "PartView" Attributes.....	64
Table 21: "AssemblyDefinition" Attributes, including attributes inherited from "PartView" .....	67
Table 22: "PartVersionRelationship" Attributes.....	70
Table 23: "GeometricModel" Attributes.....	73
Table 24: "GeometricCoordinateSpace" Attributes .....	73
Table 25: "AxisPlacement" Attributes .....	74
Table 26: "CartesianPoint" Attributes.....	75
Table 27: "ComposedGeometricModel" / "ExternalGeometricModel" Attributes.....	77
Table 28: "PropertyAssignment" Attributes .....	79
Table 29: "GeneralShapeDependentProperty" Attributes .....	83
Table 30: "ShapeElement" Attributes.....	83
Table 31: "NextAssemblyOccurrenceUsage" Attributes.....	88
Table 32: "SingleOccurrence" Attributes.....	90
Table 33: "CartesianTransformation" Attributes .....	94
Table 34: "GeometricRepresentationRelationshipWithPlacementTransformation" Attributes.....	102
Table 35: "GeometricRepresentationRelationshipWithCartesianTransformation" Attributes.....	107
Table 36: "GeometricRepresentationRelationshipWithSameCoordinateSpace" Attributes.....	111
Table 37: "General Geometric Representation Relationship" Attributes.....	113
Table 38: "PartViewRelationship" Attributes .....	118
Table 39: "NextAssemblyViewUsage" Attributes .....	119
Table 40: "Document" Attributes.....	124
Table 41: "DocumentVersion" Attributes.....	126
Table 42: "(Digital)DocumentDefinition" Attributes.....	129
Table 43: "DocumentDefinitionRelationship" Attributes.....	132
Table 44: "DocumentVersionRelationship" Attributes .....	135
Table 45: "DigitalFile" Attributes .....	140
Table 46: "ExternalItem" Attributes .....	141
Table 47: "FileRelationship" Attributes.....	144
Table 48: "FormatProperty" Attributes .....	153
Table 49: "ContentProperty" Attributes .....	156
Table 50: "CreationProperty" Attributes .....	159
Table 51: "SizeProperty" Attributes.....	160
Table 52: "DocumentAssignment" Attributes .....	163
Table 53: "PropertyDefinition" Attributes.....	178
Table 54: "CentreOfMass" Attributes .....	190

## Document History

Revision	Date	Change
1.0	2015-02-13	Initial Release
1.1	2016-05-20	Section 4: Updated descriptions of DataContainer, Subtyping and Header Object; updated Templates ExchangeContext, Class, Identifier Section 5: New section on PartVersionRelationship Section 6: Updated description for assignment of Units and use of Classification Section 7: Added descriptions for DefiningGeometry and “Part Overloading”; improved several descriptions Section 8: Improved definition of DocumentType; new section DocumentDefinitionRelationship Section 9: Improved descriptions; new section FileRelationship Section 10: Updated recommendations for CreationProperty Section 11: Updated descriptions to include Model Splitting / Sharing and Alternate Models, for CAx as well as PDM mapping Section 12: Improved description for attribute types Section 13: Updates description for classification of attributes



Revision	Date	Change
1.2	2017-06-30	<p>Overall: updated instantiation diagrams for Class / String; improved illustration of XML embedding</p> <p>Section 2: Refined Scope statements</p> <p>Section 4.1: Case handling in Attributes</p> <p>Section 4.2: Updated attributes for Version ID</p> <p>Section 4.6: Updated XML examples and attribute recommendations</p> <p>Section 5.1: Added PartViewRelationship; updated Property Value Assignment; added Multiple Part Versions and Relationships; added support for Supplier Part Number</p> <p>Section 6: Clarification for use of ShapeElement</p> <p>Section 6.1: Refined descriptions for CartesianPoint and GeometricCoordinateSpaces</p> <p>Section 6.2: Added Definition of Property Types; renamed PartProperty to PropertyAssignment</p> <p>Section 6.3: Added clarifications for Centroid and MomentsOfInertia</p> <p>Section 7: Added clarification for creation of occurrences; added PartViewRelationship</p> <p>Section 7.1: Updated recommendations for SingleOccurrence</p> <p>Section 7.3: Revised RotationMatrix attribute recommendations</p> <p>Section 7.4: Updated NextAssemblyOccurrenceUsage</p> <p>Section 8.1: Multiple DocumentVersions and -Definitions</p> <p>Section 8.3: Added DocumentVersionRelationship</p> <p>Section 9: File &amp; Document Structure updates</p> <p>Section 9.3: Added clarifications to the use of nested files and removal of PDM representation</p> <p>Section 10.1: Extended list of values for non-CAD documents; update FileFormat Property recommendations; added clarifications to the use of compression</p> <p>Section 10.4: Updated Property Assignment</p> <p>Section 12: Revision based on PDM-IF Discussions; Clarification for use of ShapeElement</p> <p>Section 12.2: Added 'general property' to PropertyDefinition.Id</p> <p>Section 12.5: Re-wrote attribute definition for values without unit</p> <p>Section 13: Added attribute table for CentreOfMass; clarifications for Centroid</p> <p>Annex A.1.5: Added explanation for order of elements</p> <p>Annex B: Updated list of BugZilla Issues</p> <p>Annex C: Updated Reference Document Versions</p> <p>Annex E.3: Unit 'each' for QuantifiedOccurrence</p>

# 1 Introduction

## 1.1 Document Overview

### 1.1.1 Goal and Objectives

The goal of this document is to describe the recommended structure and attribute population for particular instance models created from the entities and attributes defined by the STEP AP242 “Managed Model-based 3D Engineering” Business Object (BO) model and populated according to its XML Schema. The selected instance models illustrate how to encode data values that need to be exchanged in support of key industry requirements common across the mechanical design domain. The objectives of the usage guide are to:

- Support the short-term needs of the requirements of the Aerospace & Defense and the Automotive industries in the realm of mechanical design
- Prevent the emergence of “flavors”, i.e. diverging/conflicting implementations of the AP242 BO Model XML for different communities
- Ensure consistency with existing Recommended Practices for Basic Product Data Management (PDM), Assembly Structure, External References and Attributes.

### 1.1.2 Scope

This document describes the Recommended Practices for the exchange of Product and Assembly Structure data with external references to geometry files (regardless of file format). It is based on the STEP AP242 Business Object Model (ISO 10303-3001:2014) and the corresponding XML schemas, which can be found at:

- [http://standards.iso.org/iso/ts/10303/-3001/-ed-1/tech/xml-schema/bo\\_model/bom.xsd](http://standards.iso.org/iso/ts/10303/-3001/-ed-1/tech/xml-schema/bo_model/bom.xsd)  
– The BO Model XML Schema
- <http://standards.iso.org/iso/ts/10303/-3000/-ed-1/tech/xml-schema/common/common.xsd>  
– The XML Schema, which contains the structural definitions common to all STEP Business Object Models.

AP242 is the first STEP Business Object Model that is being implemented. During the development of this document and the related prototyping activities, several issues with the schemas and definitions have been identified. These have been documented in Bugzilla as official maintenance issues for AP242, and will be dealt with as part of the ISO maintenance procedures. The issues are documented throughout this document, and gathered in Annex B for reference.

### 1.1.3 Intended Audience

This document is intended to be an implementation guide for developers of CAD, PDM and file translation application systems that must use assembly structure, and exchange it with other systems and applications, in support of the design engineering and related downstream business processes.

### 1.1.4 Intended Use

This document is intended to be a manual and companion to the developer of STEP data exchange and translator software used by applications and information management systems that rely on product data. It provides guidelines for the consistent preprocessor instance model creation and requirement value encoding to enable meaningful information exchange between different systems and applications using the STEP AP242 BO model, and guidelines for the consistent interpretation by a postprocessor of the STEP AP242 BO model exchange file.

### 1.1.5 Document Style

The overall document proceeds in an incremental, step-by-step fashion to describe, and in parallel to illustrate the recommended instantiations of the XML elements in the STEP AP242 BO model.

The “template” concept is used in this document. Structures and sub-structures are defined in one section; they are then re-used in other sections of the documents. These templates are represented by the blue boxes in the diagrams.

The Instance Model diagram figures are presented using a graphical notation intended to illustrate the instance model.

Following each instance diagram, a table lists all the attributes of each displayed entity according to the XMLschema specification of ISO 10303-3001. The table includes not only the attributes of the EXPRESS schema of the AP242 BO Model, but also inverse attributes of all possible relations to the element in question. Attributes that are considered important for the scope of these Recommended Practices are in these tables written in black. Attributes that are written in grey are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

Below the table, all recommended attributes (written in black) are listed and recommendations are made for them.

Finally, a STEP AP242 BO model XML exchange structure example is included. The example exchange file corresponds directly to the instance model diagram and illustrates the very same thing using a different notation, i.e., STEP AP242 BO model XML syntax versus the graphical instance model notation.

### 1.1.6 Document Structure

The overall scope of requirements is partitioned into a set of major sections corresponding to the identified units of functionality. Within a major section, there may be sub-sections. These sub-sections further divide the scope into smaller components of coherent functionality (called “feature”) that interact with each other to realize the functionality of the entire unit.

There is generally a description of requirements and a corresponding instance diagram associated with each section and sub-section of this document. Each instance diagram is followed by a detailed explanation and specific recommendations for the entities used in the instantiation diagram example. The entity listing and explanation is in turn followed by the corresponding XML exchange structure example.

Within a section, diagrams corresponding to sub-sections incrementally build upon one another to finally achieve a complete instance model example that illustrates the entire scope of the unit of functionality.

### 1.1.7 Instantiation Diagrams

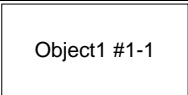
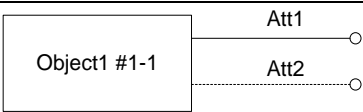
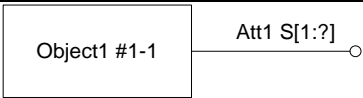
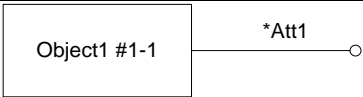
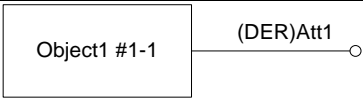
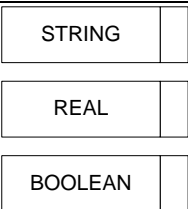
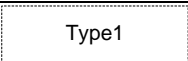
The diagrams are presented using a graphical notation intended to illustrate the instance model.

This notation is not EXPRESS-G and does not illustrate the XML schema; rather it is a graphical illustration of a specific population of a particular instance model of the schema. This notation supports:

- Illustration of entity instances and attribute values (both mapped as XML elements)
- Illustration and identification of referenced entity instances that are either fully illustrated in the current figure, or that refer to another template (if not fully illustrated in the current figure)
- Indication of optional attributes and optional reference entity instances (dashed lines),

- Illustration and identification of groups of functionally related instances (shaded bounding box), showing how XML elements are embedded into each other (the XML elements representing the entity instances placed below are embedded into the XML element representing the entity instance placed above), and
- Identification of specific attribute values (typically string values, may also be enumerated type values or numerical values).

A legend for the diagram notation is shown below:

	<p>Object (instance of an EXPRESS ENTITY)                  After the #, an instance number is given</p>
	<p>Att1: mandatory attribute                  Att2: optional attribute</p>
	<p>Aggregate type for the definition of the cardinality constraint:                  B: Bag (non-ordered and may contain duplicates)                  S: Set (non-ordered and may not contain duplicates)                  L: List (ordered)                  [x : y]: lower size : upper size                  ?: unconstrained                  A: Array (indexed)                  [x : y]: lower index : upper index</p>
	<p>Additional constraint on the object: the attribute(s) depicted with "*" have to contain unique values.  <i>Currently not used in this document</i></p>
	<p>Derived Information from another object or attribute  <i>Currently not used in this document</i></p>
	<p>Simple data types</p>
	<p>User-defined data type  <i>Currently not used in this document</i></p>

	<p>Enumeration Type</p> <p>(consists of a limited list of possible values defined for this type)</p>
	<p>Select Type</p> <p>(is used if a relationship from an object may apply either to Object1 or Object2)</p> <p>For a better readability, the members of a select type are displayed using the inheritance link (see below) and the mutually exclusive constraint. This has the same semantic as a select type.</p> <p><i>Currently not used in this document</i></p>
	<p>Attribute as relationship between two Objects (mandatory or optional),</p> <p>The circle at the end of the line gives the direction.</p> <p>Rel1: mandatory relationship        Rel2: optional relationship</p>
	<p>Rel2: inverse attribute (i.e. in the converse direction to Rel1) with cardinality constraint</p> <p><i>Currently not used in this document</i></p>
	<p>Inheritance from a Supertype to its Subtypes</p> <p>An Abstract Supertype (ABS) cannot be instantiated without one of its non-abstract Subtypes</p> <p>1: Only one subtype can be instantiated at a time (the subtypes are mutually exclusive). Per default, all the subtypes may be combined (not supported by XML)</p> <p>RT: an inherited attribute is re-typed, for example through restriction of its type, length, value range, cardinality, mandatory instead of optional or DERIVED</p> <p><i>Currently not used in this document</i></p>

	<p>Objects shown under each other within a blue colored square are embedded into each other in XML: here Object2 #2-1 is embedded into Object1 #1-1 as its XML element Rel1</p>
	<p>The templates defined in this document are re-used in other sections. This is the simple way to refer to a template (if the object referenced within the template is implicit, for example the object 'Classification' for the template 'Classification')</p>
	<p>If the template is more complex and the object referenced within the template is shown explicitly, portions of the reused template are displayed within a blue frame.</p>
	<p>Alternatively, a template may be reused through adding new XML containments to it</p>

Table 1: Instance Diagram Notation

## 1.2 Organizational Framework

These Recommended Practices for AP242 BO Model XML Product and Assembly Structure are jointly developed and supported by a number of “communities”, specifically the vendor and user communities, devoted to the development and implementation of AP242 and its associated Business Object Model. This section describes those communities’ roles and responsibilities.

### 1.2.1 Vendor Communities

The **PDM-IF Implementor Group (IG)** will be responsible for the overall organization and development of this document. The PDM-IF IG will:

- Coordinate the creation of the document
- Verify the approach of the recommended practices in PDM-IF Test Rounds
- Publish result summaries of testing AP242 BO Model product and assembly structure
- Ensure the consistency of other “AP242 BO Model XML Recommended Practices”

The **CAx Implementor Forum (CAx-IF)** will support the document development by:

- Sharing the recommended practices with CAx-IF participants
- Verify the approach of the recommended practices in CAx-IF Test Rounds
- Ensure the consistency with existing CAx-IF recommended practices

The **JT Implementor Forum (JT-IF)** will support the document development by:

- Sharing the recommended practices with JT-IF participants
- Verifying the approach of the recommended practices in JT-IF Test Rounds
- Ensuring the consistency with existing JT-IF recommended practices

### 1.2.2 User Communities

The **PDM-IF User Group (UG)** is a forum of PDM experts from the Aerospace and Defense as well as the automotive industries. The PDM-IF UG is responsible for development of the document and will:

- Support the development of the document
- Provide subject matter experts
- Provide industry requirements and ensure they are fulfilled
- Ensure the consistency with PDM standards spanning the complete product life cycle

**LOTAR** is the Aerospace and Defense user community supporting the development of the Long-term Archiving standards. LOTAR will:

- Support the development of the document
- Provide subject matter experts
- Provide A&D requirements and ensure they are fulfilled
- Ensure the consistency with LOTAR standards

**JT Workflow Forum (JT-WF)** is the Automotive user community supporting the development of the recommended practices for the ISO JT format. The JT-WF will:

- Support the development of the document
- Provide subject matter experts
- Provide Automotive requirements and ensure they are fulfilled

### **1.3 Maintenance of this Document**

This document describes the recommended practices to implement the core scope of the AP242 BO Model; that is the definition of products, structures, documents, and the directly related properties. This scope is of relevance to all three involved implementor forums (PDM, CAx and JT) and needs to be supported in a harmonized way across industries, domains, and software tools.

The scope of this document will be restricted to the core capabilities listed below in section 2. Domain-specific extensions, such as Configuration Management for PDM, or Kinematics for CAD, will be documented in separate documents of the respective Implementor Forums, with references to this core document.

Changes to the core document require consensus from all IFs, to ensure changes from one group do not create road blocks for another group. This consensus is assured by sharing working drafts for review with the involved communities and will be coordinated in a way that does not delay the publication of new versions of this document.

AFNeT, PDES, Inc., prostep ivip Association and VDA as the hosting organizations of the involved implementor forums will maintain and extend the document, as long as it provides utility to the vendor community.

## **2 Scope**

### **2.1 In Scope**

**The following are within the scope of this document:**

- Implementations based on the AP242 BO Model (ISO 10303-3001) Edition 1 (2014) for:
  - Basic PDM Capabilities
    - Identification concept [Part, Part Version, Part View]
    - User Defined Attributes (non-geometric properties)
    - Mechanical CAD Assembly Structures
    - Component Instances, Placement / Transformations
    - Classification
    - Document Management
  - Geometric and Assembly Validation Properties
  - AP242 IS BO Model XML File Structure
    - One XML file for the entire assembly structure (“monolithic” approach)
    - One XML file per assembly node and per leaf node part (“nested” / “fully shattered” approach)
  - External References to files
    - STEP Part 21 files (CAx-IF / LOTAR Scope)
    - AP242 XML files (for “nested” / “fully shattered” approach)
    - ISO 14306 JT files (JT-IF / JT-WF Scope)
    - CAD native files (e.g. Creo, NX, CATIA V5/V6...)
    - Office and general files (e.g., PDF, JPEG...)



**The following capabilities are included in this document for completeness of the definitions, but have not yet been fully tested by the involved Implementor Forums:**

- 5.1.2: Multiple PartVersions within one Part
- 5.1.3: Multiple PartViews within one PartVersion
- 5.1.5: PartVersionRelationship
- 5.1.6: Identifiers at OEM and Supplier
- 6.3.1: GeneralShapeDependentProperty
- 7.2: SpecifiedOccurrence
- 8.2: DocumentDefinitionRelationship
- 8.3: DocumentVersionRelationship
- 9.2: FileRelationship
- 10.5: DocumentFileProperty
- 11.1: Model splitting and alternative geometries (CAx mapping)
- 11.2: Model splitting and alternative geometries (PDM mapping)
- 12.3: Groups of attributes and attribute values
- 12.6: Transfer of Meta-Data for attributes
- 13.2: Repeating Part-level Geometric Validation Properties in PDM files
- A.1.10: Multilanguage Support

These sections are defined to the best knowledge of the authors; however, changes based on implementation feedback are possible once these capabilities are being tested.

## **2.2 Out of scope**

**The following are out of scope for this document because they will be covered in other documents:**

- CAx-specific Capabilities such as Kinematics, Composites, etc.
- Advanced PDM Capabilities:
  - Configuration, Effectivity, Activity and Work Management, Approval...
- External Element References (into Part 21 files or between BO Model XML files)
- Implementations based on any edition of the AP242 BO Model (ISO 10303-3001) other than Edition 1 (2014).

**The following are out of scope for this document at the moment, because the underlying use cases and requirements have not yet been fully described:**

- |   |   |
|---|---|
| • Address   | • InformationUsageRightAssignment                           |
| • Certification   | • MaterialPropertyAssignment                                |
| • Contract  | • ModelPropertyAssignment                                   |
| • ConstituentPart (including measured, calculated, and estimated / asserted weight for single parts and assemblies) | • OccurrenceRelationship (for example for substitute parts) |
| • FrozenAssignment  | • OrganizationRelationship                                  |
|   | • PartRelationship (for example for alternate parts)        |

- MakeFromRelationship
- QuantifiedOccurrence
- ReplacedPartViewRelationship
- ToolPartRelationship
- Project
- ProjectAssignment
- Validation Properties for Product Structures and External References
- ProjectRelationship
- PropertyDefinitionAssignment
- SecurityClassificationAssignment
- ShapeElementRelationship
- TimeIntervalAssignment

These will be added in later revisions of this document, based on users' needs and testing progress.

### 3 Reference to Recommended Practices

For validation purposes, STEP processors shall state which Recommended Practice document and version thereof have been used in the creation of the STEP file. This will not only indicate what information a consumer can expect to find in the file, but even more importantly where to find it in the file.

This shall be done by adding a pre-defined string to the first string element of the `Documentation` attribute of the `Header` element in the XML file (for details see section 4.1.5 below). The value follows a specific pattern well established in Part 21 files:

Document Type---Document Name---Document Version---Publication Date

The string corresponding to this version of this document is:

```
<Documentation>CAx-IF Rec.Pracs.---AP242 BO Model XML  
Assembly Structure---1.2.0---2017-06-30</Documentation>
```

#### **General Postprocessor Recommendation:**

If a postprocessor encounters attribute values, or object instantiations different from the ones recommended in this version of the document, a warning shall be recorded. In such case, an additional exchange agreement is supposed to be in place among the parties involved in the data exchange.

## 4 Basic Concepts

### 4.1 XML Format Specifics

Annex A describes the guiding principles used for the mapping from EXPRESS to XML. This section gives some additional hints on how to instantiate it.

#### 4.1.1 Character Set

Beside the use of the XML special characters `&amp;` (for `&`), `&apos;` (for `'`), `&gt;` (for `>`), `&lt;` (for `<`) and `&quot;` (for `"`) in elements of the kind `STRING`, any character (even special) can be used.

#### **Preprocessor Recommendations:**

Use of UTF-8 (stated in the first line of the XML file):

```
<?xml version="1.0" encoding="UTF-8"?>
```

➔ special characters like  $\frac{1}{4}$  are mapped with their decimal value (here `&#x172;`).

Concerning the instance identifiers (called 'uid') of type `xsd:ID`, they must start with either a letter or underscore (`_`) and may contain only letters, digits, underscores (`_`), hyphens (`-`), and periods (`.`). White space is not allowed.

- ➔ unlike the instance identifier of ISO STEP Part21, which are restricted to numeric integer values, it is possible to set the uids to some more readable values.

### **Preprocessor Recommendations:**

Use some (even proprietary) convention to ease the human interpretation of the uids, like:

<unique abbreviation of the object type>--<number unique within the XML file>

For example, for a PartView instance: pv--4711

Since some PDM systems are not case-sensitive and import for example 'Part4711' as 'PART4711', it is recommended to handle all attributes as case insensitive, except:

- the File Id, ExternalItem.Id and ExternalItem.Source (because all Unix-based operating systems are case-sensitive)
- PropertyValue.Name (because e.g. in CATIA, "name" and "NAME" are two different attributes)

### **4.1.2 Containment vs. Referencing**

Containment is preferred over Referencing as far as possible, since it enables the storing of the maximum number of aspects of an object at one single place in the XML file. For example here is a Part with its PartVersion(s), PartView(s), Occurrences(s), Document(s)...:

```
<Part uid="p--000000001E60C660">
  <Id>
    <Identifier uid="pid--000000001E60C660--id6" id="bolt" idContextRef="o-
-000000178"/>
  </Id>
  <Name>
    <CharacterString>bolt</CharacterString>
  </Name>
  <PartTypes>
    <ClassString>piece part</ClassString>
  </PartTypes>
  <Versions>
    <PartVersion uid="pv--000000001E60C660--id6">
      <Id id="/NULL"/>
      <Views>
        <PartView uid="pvv--000000001E60C660--id6">
          <InitialContext uidRef="ac--mechanicaldesign--design"/>
          <Occurrence xsi:type="n0:SingleOccurrence" uid="pi--
000000001E60C660--18">
            <Id id="bolt.1"/>
          </Occurrence>
          <DocumentAssignment xsi:type="n0:DocumentAssignment" uid="da--
000000001E60C660--id6">
            <AssignedDocument uidRef="df--000000001E60C660"/>
            <Role>
              <ClassString>mandatory</ClassString>
            </Role>
          </DocumentAssignment>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

On the other hand, referenced instances would be spread all over the XML file.

Containment is used for context dependent objects which cannot exist without the container object. Reference is used for objects that can exist on their own; they may be reference multiple times (reused), which avoids duplication of data.

The type of all referenced instances (defined as "cmn:Reference") and contained instances can be validated by the XSD.

### 4.1.3 Root Objects and DataContainer

All the entities not being declared as contained in any XML element are defined as a subtype of cmn:BaseRootObject. Otherwise they are defined as a subtype of cmn:BaseObject.

Each Root Object is defined as an element of the so-called AP242DataContainer (defined as a subtype of cmn:DataContainer). The DataContainer is like a schema instance (contains instances of the entities defined in the BO Model EXPRESS schema according to their definition in the XSD), just like the Data section in a Part21 STEP Physical File.

An XML file may have one or many AP242DataContainers.

#### ***Preprocessor Recommendations:***

Define all data into one single AP242DataContainer.

The order of the root objects is not critical (since defined as `xsd:choice minOccurs="0" maxOccurs="unbounded"`), but the order of the attributes and containments within each object is strictly defined in the BO Model XSD (as `xsd:sequence`). In case of inheritance, the ordering of the attributes is: first the attributes of the top level supertype, then the attributes of the next level supertype, etc... and at the end, the attributes of the instantiated subtype.

### 4.1.4 Subtyping

If a subtype shall be instantiated, the top-level supertype defined in the EXPRESS and XML schema shall be instantiated, followed by a subtyping clause `xsi:type`. For example to instantiate the `ComposedGeometricModel` subtype of `GeometricRepresentation`:

```
<GeometricRepresentation uid="egm--1" xsi:type="n0:ComposedGeometricModel">
```

### 4.1.5 Header object

#### ***Preprocessor Recommendations:***

The header is mandatory and shall contain at least the following information:

- Name: name of the XML file
- TimeStamp: creation (or last modification date) of the XML file
- Organization.Name: name of the sending organization. Use the same unique ID conventions as for the Id of the template "Organization"
- PreprocessorVersion: name and release of the Preprocessor
- OriginatingSystem: name and release of the originating system
- Documentation: version of the Recommended Practices used to implement the pre-processor + (optionally) some concatenated free texts explaining the contents of the exchange file

Here is an example:

```
<Header xmlns="">  
  <Name>as1.stpx</Name>  
  <TimeStamp>2017-07-02T09:54:06Z</TimeStamp>  
  <Organization>
```

```
<Name>mercedes-benz.com</Name>
</Organization>
<PreprocessorVersion>T-Systems International GmbH COM/FOX
V6.1.4</PreprocessorVersion>
<OriginatingSystem>CATIA V5 B25 SP0 HF0</OriginatingSystem>
<Documentation>CAx-IF Rec.Pracs.---AP242 BO Model XML
Assembly Structure---1.2.0---2017-06-30</Documentation>
</Header>
```

### Remarks:

- The organization mentioned in the header is not necessarily redundant with the one mentioned in ExchangeContext.IdentificationContext: for example, if a tier-1 supplier forwards data from its customer to an tier-2 supplier, forwarding the identifiers from the customer, the IdentificationContext will be the one of the customer while the organization in the header will be the tier-1 supplier
- The version of the AP242 BO Model XSD doesn't need to be defined here, since it is already defined in the top line of the XML file (see next section).
- With the upcoming Technical Corrigendum of the AP242 BO Model, "Documentation" will be changed to become a set, i.e. the following pattern will be recommended instead (see BugZilla #5558):
  - First occurrence of "Documentation": (mandatory) version of the Recommended Practices used to implement the preprocessor
  - Further occurrences of "Documentation": (optional) free texts explaining the contents of the exchange file

### 4.1.6 XML context tagging

Using the appropriate scoping, lots of XML context tagging may be avoided.

#### **Preprocessor Recommendations:**

Use the following definition to enclose all schema definitions needed by the BO Model XML file:

```
<n0:Uos
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:n0="http://standards.iso.org/iso/ts/10303/-3001/-ed-1/tech/xml-
schema/bo_model"
xmlns:cmn="http://standards.iso.org/iso/ts/10303/-3000/-ed-1/tech/xml-
schema/common"
xsi:schemaLocation="http://standards.iso.org/iso/ts/10303/-3001/-ed-
1/tech/xml-schema/bo_model http://standards.iso.org/iso/ts/10303/-3001/-ed-
1/tech/xml-schema/bo_model/bom.xsd">
...
<Header>
...
</Header>
<DataContainer xsi:type="n0:AP242DataContainer">
...
</DataContainer>
</n0:Uos>
```

Avoid defining namespaces in the DataContainer.

The only case within the data where context tagging is needed is ,n0' for the subtype clauses like xsi:type="n0:ComposedGeometricModel".

## 4.1.7 Schema version and document version identifications

The name of the schema that an AP242 BO Model XML file shall be compliant to shall be listed at the beginning of the uos element.

Example: To indicate AP242 BO model XML version 1.0, the following URL is used:

```
xsi:schemaLocation="http://standards.iso.org/iso/ts/10303/-3001/-ed-1/tech/xml-schema/bo_model http://standards.iso.org/iso/ts/10303/-3001/-ed-1/tech/xml-schema/bo_model/bom.xsd">
```

This has been issued in Bugzilla under #5088.

## 4.1.8 Dates

Unlike in the EXPRESS schema definition of DateTimeString, the time is not optional in XML using xsd:dateTime.

### ***Preprocessor Recommendations:***

If not available, set the time to T00:00:00, for example: 2010-08-19T00:00:00

## 4.1.9 STEP BO Model XML File Extensions

It was agreed by implementors and users alike that the default file extension “\*.xml” is too generic, since there are so many XML files for a wide variety of purposes and applications already using the extension. Since STEP BO Model XML files oftentimes are process-relevant, they should be easily identifiable, and it should be possible to associate a default handling application.

Since “\*.stp” shall remain reserved for STEP Part 21 files and the previously proposed “\*.stpxml” seemed to ungainly, the CAx-IF agreed to consistently use the following file extensions:

- “\*.stpx” – for STEP (AP209, AP242,...) BO Model XML files
- “\*.stpxZ” – for compressed STEP BO Model XML files

The compression of STEP files – Part 21 as well as BO Model XML – is defined in the CAx-IF Recommended Practices for STEP File Compression (see Annex C).

## 4.2 Rules for Attribute Cardinality

### 4.2.1 Entities and Attributes not supported by the Preprocessor

The guidance provided in this section reuses the “General Information” section of the PDM Schema Usage Guide V4.3, but adds recommendations for derived attributes and for mandatory numerical attributes.

For various reasons, there may be entities that cannot be completely exported by a preprocessor. For example, an application may not maintain all the information that is mandatory for data exchange according to this specification. Or, the information is maintained by a sending system, but it will for some reasons not be included in the data exchange file. The preprocessor shall provide values for all mandatory attributes in an exchange file.

For mandatory string-value attributes, special values shall be used to further indicate the reason why no real data is provided according to the following convention:

- For string-value attributes: empty string <tag></tag> or <tag/> indicates user data managed by the sending system, but having no value.
- For string-value attributes: string <tag>NULL</tag> indicates user data in a mandatory attribute in STEP, that is not managed by the sending system, or currently not known. Part.Id, Document.Id and File.Id are mandatory base identifiers that shall never be

mapped to 'NULL'.

If an object with version 'NULL' is encountered on import, it shall be mapped to the highest existing version (if not frozen), or a new version shall be created with the next-higher available version id.

- For string-value attributes: string <tag>/ANY</tag> indicates that the value – mandatory or optional - may be computed by the target system (for example PartVersion.Id if the assembly structure of the source system stores only the part number of the component and computes the right PartVersion at runtime).
- For string-value attributes: string <tag>/DUMMY</tag> indicates that an entity as a whole is not supported by a pre-processor, but is mandatory according to the XML Schema => all its mandatory string attributes are set to /DUMMY. This may apply for example to DateAndPersonOrganization.PersonOrOrganization.

Accordingly, it is not recommended to use the empty string or the default strings 'ANY', 'DUMMY' and 'NULL' as valid user data.

For mandatory INTEGER, REAL or NUMBER attributes, 2147483647 (MAX\_LONG) indicates user data in a mandatory attribute that is not managed by the sending system or currently not known.

For mandatory Date attributes, 1970-01-01T00:00:00 indicates user data in a mandatory attribute that is not managed by the sending system or currently not known.

Accordingly, it is not recommended to use 2147483647 or 1970-01-01T00:00:00 as valid user data. Dates older or newer than 1970-01-01T00:00:00 shall be interpreted as user data.

For further mandatory non-string value attributes, these recommended practices do not provide further guidance.

If an optional attribute is not instantiated, the corresponding element shall be completely removed from the physical file. Though not recommended, it is also valid to list the element start and end tags without providing any value.

#### 4.2.2 Entities and Attributes not supported by the Postprocessor

The guidance provided in this section corresponds with the "General Information" section of the PDM Schema Usage Guide V4.3.

For various reasons, there may be entities that cannot be completely imported by a postprocessor. The postprocessor translator implementation may not support the import of the entity. Or, the receiving system may not maintain the information that is carried by an entity or attribute, or it may require specific attribute values that are not present in the input data.

The names of entities and attributes not imported should be recorded in a history log file together with a reason. Entities and attributes not supported by the receiving system shall not cause a system failure. The minimum acceptable behavior shall be to ignore the unsupported constructs gracefully.

#### 4.2.3 Unspecified and Optional Attribute Values

The guidance provided in this section corresponds with the "General Information" section of the PDM Schema Usage Guide V4.3.

Optional attributes without specific recommended values, such as the description attribute, are available on many entities in the AP242 BO Model. The following general recommendation for the use of this type of attribute is given:

**Preprocessor** - First, follow the usage guide as much as is possible. If some specific common harmonized user requirement has been documented in the usage guide for the type of attribute, adapt this requirement to the attributes in question (i.e., map the standard into your

domain). If no specific common harmonized user requirement has been documented in the usage guide, in general, such an optional attribute should not be instantiated. However, these attributes may be used in some bilateral agreements between exchange partners.

**Postprocessor** - Any optional attribute with no specific mapping specified can, in general, not be specifically interpreted in an interoperable way. While these types of attributes are in general not recommended to be instantiated, the postprocessor should gracefully handle any data that is exchanged using these attributes. A robust, interoperable AP242 BO Model postprocessor will generally provide user access to also these values.

#### 4.2.4 Derived Attributes

The guidance provided in this section corresponds with the “General Information” section of the PDM Schema Usage Guide V4.3.

In general, derived attributes are not covered by this recommended practices document. This is consistent with the STEP part 21 and part 28 specifications where derived attributes are not represented in an exchange file. This document does not include cases of derived attributes where special attention is required.

#### 4.3 Uniqueness of Identifiers

Two types of identifiers are distinguished:

- 1) Uid-identifier, which is assigned to each element in the XML-file and which plays the same role as the instance identifier in ISO 10303-21 files. This identifier shall be unique within a single file; they are not unique across several physical files, even though such files may form a consistent data package.
- 2) User-defined identifiers that are provided as part of the product data. These recommended practices do not require such identifier strings to be unique, neither globally nor within a single physical file. Thus, in the concurrent management of internal and external identifiers in a database, duplicate identifiers may occur.

NOTE: To ensure uniqueness of identification the EXPRESS schema of the AP242 BO Model requires the combination of the values of the attributes id, role and identificationContext of instances of entity Identifier to be unique. See 4.6.6 for recommendations of instantiating entity Identifier in general; additional guidance may be given in the sections of entities that have attributes of type Identifier.

#### 4.4 Project Specific Values

The guidance provided in this section corresponds with the “General Information” section of the PDM Schema Usage Guide V4.3.

Attribute values recommended in this usage guide shall be supported by systems that conform to the AP242 BO Model. Other values negotiated between exchange partners in specific projects may be used where the interpretation of their meaning does not contradict definitions provided in this usage guide. However, these agreements will generally not lead to interoperable solutions

#### 4.5 Blanks in String Values

The guidance provided in this section corresponds with the “General Information” section of the PDM Schema Usage Guide V4.3.

All white space within the XML tag delimiters of a STRING value shall be considered valid user data, that is, also leading and trailing blanks are valid user data.



## 4.6 Basic Building Blocks

The objective of this chapter is to define the basic templates that will be reused in the representation of complex concepts (chapter 4.6.11 and following).

### 4.6.1 Template “ExchangeContext”

The `ExchangeContext` entity specifies a default context for the identifications and descriptions, a default language and a default length unit relevant for a defined context.

The `Description` provides the context of the exchange.

The `DefaultLanguage` sets the default language used in the exchanged file if no specific language information is provided.

The `DefaultUnit` sets the default length unit to be used for the exchanged file if no specific unit is provided.

The `IdentificationContext` sets the default organization managing the different id and description if no specific organization is provided.

### The Instance Model: AP242 BO Model XML entities and attributes

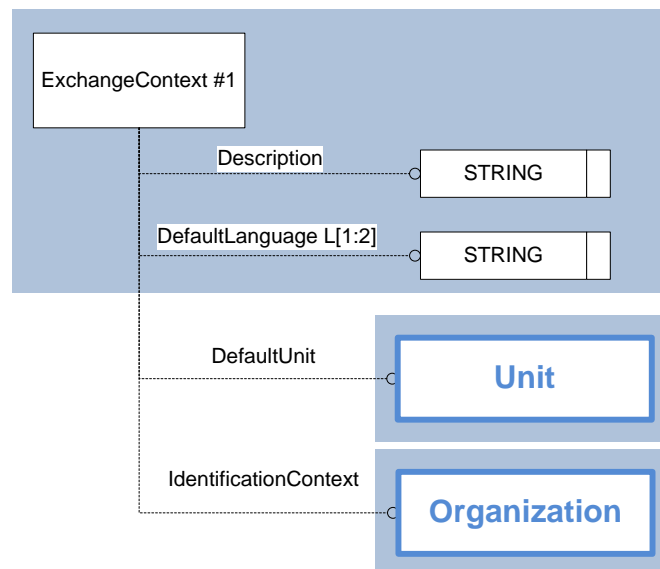


Figure 1: Template “ExchangeContext”

ENTITY ExchangeContext	Attribute Type
DefaultLanguage	OPTIONAL Language
DefaultUnit	OPTIONAL Unit
Description	OPTIONAL DescriptorSelect
IdentificationContext	OPTIONAL IdentificationContextSelect

Table 2: “ExchangeContext” Attributes

#### Attribute recommendation

- The **Description** attribute is the text providing information on the exchange context. The value of this attribute need not be specified. Use “Description” template (see 4.6.7).
- The **DefaultLanguage** attribute is the set of text by which the default language is known.

The value of this attribute need not be specified. Use `xsd:language` type. For more details, refer to chapter A.1.10.

- The **DefaultUnit** attribute characterizes a default length unit. The value of this attribute need not be specified. Use “Unit” template (see 4.6.3).
- The **IdentificationContext** attribute specifies a default organization. The value of this attribute need not be specified. Use “Organization” template (see 4.6.2).

Remark: Just as in the AP242 Specification, the XSD does not restrict the values in DefaultLanguage

#### **Preprocessor Recommendations:**

- All preprocessors should provide a unique ExchangeContext.
- For the language definition (for example ‘en-US’), even if the country code (here ‘US’) is optional, it is recommended to set it
- As specified in the EXPRESS data model via a WHERE rule, at least one of the attributes DefaultLanguage or IdentificationContext shall be specified

#### **Postprocessor Recommendations:**

- If no context is given for an Identifier the ExchangeContext.IdentificationContext should be used.
- If no length unit is given the ExchangeContext.DefaultUnit shall be used (like in simplified positioning representation of assembly structure, see chapter 7.3.1).
- If no language is given for names or descriptions, the DefaultLanguage should be used.

**Related Entities:** There are no specific related entities.

### **The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)**

```
<ExchangeContext uid="ec--000000001">
  <DefaultLanguage>en-US</DefaultLanguage>
  <DefaultUnit uidRef="u--000000002"/>
  <Description>
    <CharacterString>AP242 BO Model XML Assembly Structure exchange</CharacterString>
  </Description>
  <IdentificationContext uidRef="o--000000178"/>
</ExchangeContext>
<Unit uid="u--000000002">
  <Kind>
    <ClassString>SI system</ClassString>
  </Kind>
  <Name>
    <ClassString>metre</ClassString>
  </Name>
  <Prefix>
    <ClassString>milli</ClassString>
  </Prefix>
</Unit>
<Organization uid="o--000000178">
  <Id id="mercedes-benz.com"/>
  <Name>
    <CharacterString>Mercedes-Benz</CharacterString>
  </Name>
```

```
<OrganizationTypes>
  <ClassString>company</ClassString>
</OrganizationTypes>
</Organization>
```

#### 4.6.2 Template “Organization”

In the same way as in section 13.1.1 of the PDM Schema Usage Guide V4.3, the Organization entity represents a group of people (e.g., companies, countries, etc.).

The Id is very important providing unique identification to the organization or company; this attribute should be populated with unique data.

The Name attribute should contain the common nomenclature of the organization.

The OrganizationTypes attribute should contain a characterization of the type of the organization.

#### The Instance Model: AP242 BO Model XML entities and attributes

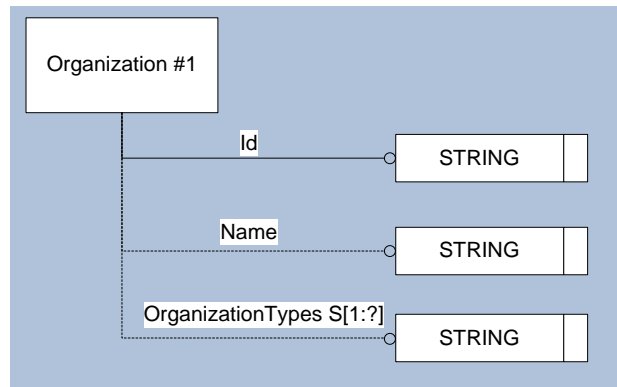


Figure 2: Template "Organization"

ENTITY Organization	Attribute Type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
Name	OPTIONAL DescriptorSelect
OrganizationTypes	OPTIONAL SET[1:?] of ClassSelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
AddressAssignment	OPTIONAL SET[1:?] of AddressAssignment
ContractAssignment	OPTIONAL SET[1:?] of ContractAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
OrganizationOrPersonInOrganiza-	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganiza-

ENTITY Organization	Attribute Type
tionAssignment	tionAssignment
OrganizationRelationship	OPTIONAL SET[1:?] of OrganizationRelationship
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssignment

Table 3: "Organization" Attributes

**Attribute recommendation**

- The **Id** attribute is the identifier that distinguishes the organization. Use IdentifierString type.
- The **Name** attribute is the label by which the organization is known. The value of this attribute need not be specified. Use "Description" template (see 4.6.7).
- The **OrganizationTypes** attribute characterizes the type of organization. The value of this attribute need not be specified. Use ClassString if one of the values below is used, otherwise use "Class" template (see 4.6.4). According to the ISO AP242 Specification, where applicable, the following values shall be used:

OrganizationTypes	Explanation
'company'	The organizationTypes specifies that the Organization is a company
'department'	The organizationTypes specifies that the Organization is a department
'plant'	The organizationTypes specifies that the Organization is a plant

- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:** All preprocessors should provide a unique organization id to eliminate ambiguities where organizations may have the same names. If the intended domain for the data is large, the reader is referred to ISO/IEC 8824-1, which can provide some guidance on creating unique identifiers. If appropriate, a URL-like convention for the organization identifier may be used, e.g., cax-if.org. A unique string obtained under ISO/IEC 8824-1 can be used as, or prefixed to, the organization identifier. For example, if the organization typically used an identifier of "93699" and the unique string were "USA", the unique value of the organization id would be "USA93699". If available and appropriate, the following values should be used to describe the organization type:

- 'company' to indicate a business entity;
- 'department' to indicate an organizational group within a company;
- 'plant' to indicate that the organization is a plant.

**Postprocessor Recommendations:** All postprocessors should make use of any provided information in the id attribute to eliminate ambiguities where organizations may have the same name.

**Related Entities:** There are no specific related entities.

## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<Organization uid="o--000000178">
  <Id id="mercedes-benz.com"/>
  <Name>
    <CharacterString>Mercedes-Benz</CharacterString>
  </Name>
  <OrganizationTypes>
    <ClassString>company</ClassString>
  </OrganizationTypes>
</Organization>
```

### 4.6.3 Template "Unit"

This entity is a quantity chosen as a standard in terms of which other quantities may be expressed. The types of units supported are SI units as well as derived or conversion based units as defined in ISO 10303-41. See Annex E for the recommendation of the Units definition.

The `Name` provides the type of the unit.

The `Kind` represents the type of system used.

The `Prefix` is the ratio of the unit.

## The Instance Model: AP242 BO Model XML entities and attributes

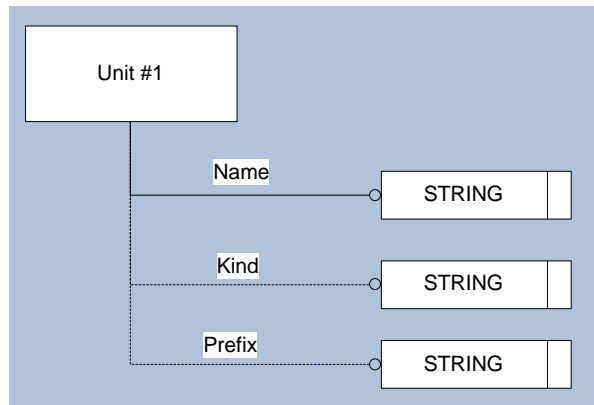


Figure 3: Template "Unit"

ENTITY Unit	Attribute Type
Kind	OPTIONAL ClassSelect
Name	ClassSelect
Prefix	OPTIONAL ClassSelect
Quantity	OPTIONAL ClassSelect

Table 4: "Unit" Attributes

### Attribute recommendations

- The **Name** attribute is the text defining the type of the unit. Use ClassString type.
- The **Kind** attribute is the type of system used. The value of this attribute need not be specified. Use ClassString if one of the values below is used, otherwise use "Class" template (see 4.6.4). According to the ISO AP242 Specification, when applicable the 'SI System' value should be used

- The **Prefix** attribute is the definition of the ratio. The value of this attribute need not be specified. Use `ClassString` if one of the values below is used, otherwise use “Class” template (see 4.6.4). According to the ISO AP242 Specification, when applicable the following values shall be used ‘exa’, ‘peta’, ‘tera’, ‘giga’, ‘mega’, ‘kilo’, ‘hecto’, ‘deca’, ‘deci’, ‘centi’, ‘milli’, ‘micro’, ‘nano’, ‘pico’, ‘femto’, ‘atto’.
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:** None specified.

**Postprocessor Recommendations:** None specified.

**Related Entities:** There are no specific related entities.

## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<Unit uid="u--000000002">  
  <Kind>  
    <ClassString>SI system</ClassString>  
  </Kind>  
  <Name>  
    <ClassString>metre</ClassString>  
  </Name>  
  <Prefix>  
    <ClassString>milli</ClassString>  
  </Prefix>  
</Unit>
```

### 4.6.4 Template “Class”

This entity is a classification which characterizes all objects of the same kind.

The `Id` provides a unique identification to the classification; this attribute must be populated with unique data.

The `Description` attribute should contain the textual information concerning the class.

The `DefinedIn` attribute should reference a specific externally defined set of value.

## The Instance Model: AP242 BO Model XML entities and attributes

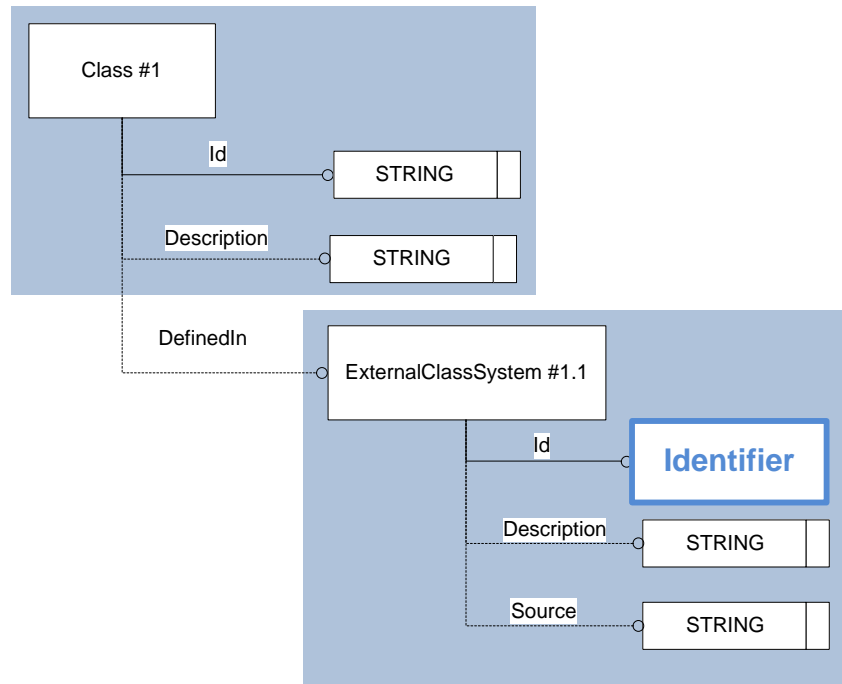


Figure 4: Template "Class"

ENTITY Class	Attribute Type
ClassAttribute	OPTIONAL SET[1:?] of ClassAttribute
DefinedIn	OPTIONAL ExternalClassSystem
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
SetMembership	OPTIONAL SET[1:?] of SetMembership
SubsetMember	OPTIONAL SET[1:?] of SubsetMember
VersionId	OPTIONAL IdentifierSelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 5: "Class" Attributes

**Attribute recommendation**

- The **Id** attribute is the identifier that distinguishes the class. Use IdentifierString type if the attribute DefinedIn is set, otherwise use “Identifier” template (see 4.6.6).
- The **Description** attribute is the text by which the class is described. The value of this attribute need not be specified. Use “Description” template (see 4.6.7).
- The **DefinedIn** attribute specifies where is defined the type represented by the Id. The value of this attribute need not be specified. Reference to an ExternalClassSystem element.
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

ENTITY ExternalClassSystem	Attribute Type
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
Source	OPTIONAL ExternalSourceSelect
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 6: "ExternalClassSystem" Attributes

**Attribute recommendation**

- The **Description** attribute is the text by which the external system is described. The value of this attribute need not be specified. "Description" template (see 4.6.7).
- The **Id** attribute is the identifier that distinguishes the external system. Use “Identifier” template (see 4.6.6).
- The **Source** attribute specifies where the external system is located. The value of this attribute need not be specified. Use IdentifierString type.
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:** All preprocessors should provide unique class ids.

The ExternalClassSystem should be used when the Class.id value is not commonly agreed by the AP242 specification or recommended practices.

Classes commonly agreed by the AP242 specification do not need an identification context. This is why in this case, the use of IdentifierString for Class.Id is recommended where ClassString is not possible (i.e. for Identifier.IdRoleRef, see “Identifier” template 4.6.6).



The entity allows specifying all the values supported by the preprocessor and among them, those referenced by the exchanged assembly structure.

**Postprocessor Recommendations:** None specified.

**Related Entities:** There are no specific related entities.

### The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<ExternalClassSystem uid="ecs--fdf">
  <Id>
    <Identifier uid="fdf--filedataformat-id1" id="file data format" idContextRef="o--000000178" />
  </Id>
</ExternalClassSystem>
<Class uid="fdf--CGR">
  <DefinedIn uidRef="ecs--fdf" />
  <Id id="CGR" />
</Class>

<FormatProperty uid="ffp--CGR">
...
  <DataFormat>
    <Class uidRef="fdf--CGR" />
  </DataFormat>
</FormatProperty>
```

#### 4.6.5 Template “Classification”

This entity permits the attachment of a Class to one or more objects.

The Role provides the meaning of the association.

The Class attribute provides the classification information.

### The Instance Model: AP242 BO Model XML entities and attributes

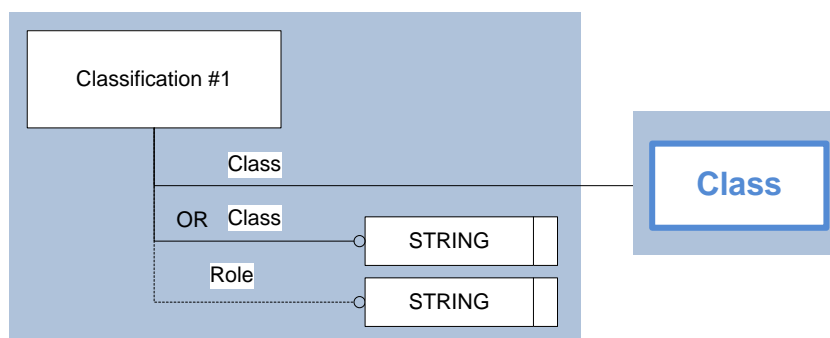


Figure 5: Template "Classification"

ENTITY Classification	Attribute Type
Class	ClassSelect

ENTITY Classification	Attribute Type
Role	OPTIONAL String
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
ClassificationRelationship	OPTIONAL SET[1:?] of ClassificationRelationship
ContractAssignment	OPTIONAL SET[1:?] of ContractAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 7: "Classification" Attributes

#### Attribute recommendation

- The **Class** attribute is the reference to the classification. Use ClassString if the value is recommended within this document (for example for Filecontent.GeometryTypes), otherwise use "Class" template (see 4.6.4).
- The **Role** attribute is the text that defines the role of the association of the class to an object. The value of this attribute need not be specified. Use "Description" template (see 4.6.7).
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:** When applicable, the following values shall be used for the attribute Role:

- 'electromagnetic compatibility': The classification categorizes the classified element in respect of its ability to comply with requirements concerning electromagnetic interference
- 'environmental conditions': The classification categorizes the classified element with respect to its ability to comply with environmental impact requirements.

**Postprocessor Recommendations:** None specified.

**Related Entities:** There are no specific related entities.

#### The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<Classification uid="vda--SpecifiedReference">
  <Class>
    <ClassString>specified reference</ClassString>
  </Class>
</Classification>
```

#### 4.6.6 Template “Identifier”

The identifier supports the ability to uniquely identify an object via a combination of three criteria: id, role and context.

The id is very important providing unique identification to the related object.

The idRoleRef attribute should refer a Class managing the role of the identification.

The idContextRef attribute should be set and refer the organization managing the id (in case of a root object) or refer the identifier of the root object (in case of an embedded object).

#### The Instance Model: AP242 BO Model XML entities and attributes

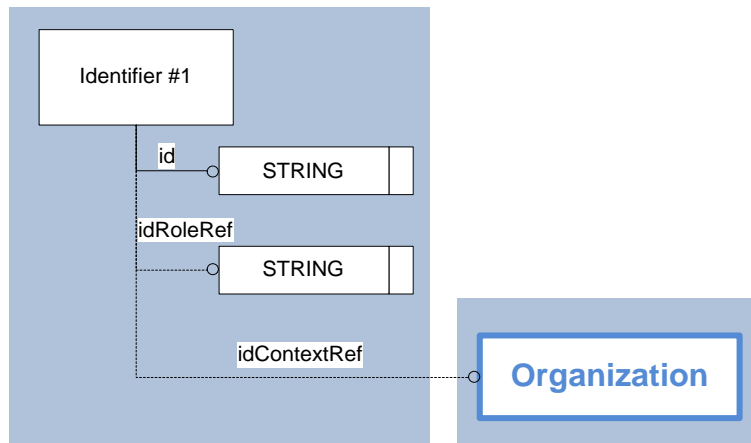


Figure 6: Template "Identifier"

ENTITY Identifier	Attribute Type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
ContractAssignment	OPTIONAL SET[1:?] of ContractAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
IdentifierRelationship	OPTIONAL SET[1:?] of IdentifierRelationship
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
id	OPTIONAL String
idRoleRef	OPTIONAL ClassSelect
idContextRef	OPTIONAL IdentificationContextSelect

Table 8: "Identifier" Attributes

## Attribute recommendation

- The **id** attribute is the text that represents an identifying name or code. Use IdentifierString type.
- The **idRoleRef** attribute is the text that defines the role of the identifier. Use “Class” template (see 4.6.4) and the following value:
  - ‘identification information’: the id identifies the object  
*Comment: since Identifier.idRoleRef is mandatory in the EXPRESS schema, but optional in the XSD, an issue in Bugzilla has been created under #5089. Until then, if unset, the value ‘identification information’ is assumed to be the default.*
- The **idContextRef** attribute is the context within which the **Identifier** has been created and is unique. The value of this attribute should be specified. Use “Organization” template (see 4.6.2) or “Identifier” template (this section).  
*Remark: there is one exception: Organization.Id shall not have a value for idContextRef.*
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Remark:** since the EXPRESS attributes ‘role’ and ‘identificationContext’ have been mapped to XML attributes (and not XML elements) for a compact representation as this object is frequently used, it is not possible to distinguish between ClassString and a reference to a Class/ExternalClass/ExternalOwlClass, resp. between IdentifierString and a reference to an Identifier/Organization. This is why they have been renamed to idRoleRef and idContextRef and shall not be mapped to a String (the XSD checks that they contain an uid).

**Preprocessor Recommendations:** All preprocessors should ensure the uniqueness of the combination of id, idRoleRef and idContextRef.

In the case where the root object and its embedded objects have all the same context and only one role, Identifier should only be used for root objects and idContextRef shall reference an Organization. Embedded objects are already identified through their root object => IdentifierString shall be used.

In other cases (i.e. multiple contexts or multiple roles), the embedded objects shall also use the Identifier template and their idContextRef shall reference one of the Identifiers of their root object.

In the XML format, it is possible to duplicate the identification in the Id.id attribute and in the Identifier.id attribute. This usage is not recommended since it is not possible in the EXPRESS definition of the AP242 BO Model. All preprocessors must avoid it. The Id.id shall only be used in special cases like Organization.Id

This has been documented in BugZilla #5884.

**Postprocessor Recommendations:** None specified.

**Related Entities:** There are no specific related entities.

## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

This example shows the simple case where only one context is involved:

```
<Class uid="rl--ii">
  <Id id="identification information"/>
</Class>
<Organization uid="o--000000178">
  <Id id="mercedes-benz.com"/>
```

```
<Name>
  <CharacterString>Mercedes-Benz</CharacterString>
</Name>
<OrganizationTypes>
  <ClassString>company</ClassString>
</OrganizationTypes>
</Organization>
<Part uid="p--000000001E60C660">
  <Id>
    <Identifier uid="pid--000000001E60C660--id6" id="bolt" idRoleRef=",,rl--
ii" idContextRef="o--000000178"/>
  </Id>
  ...
  <Versions>
    <PartVersion uid="pv--000000001E60C660">
      ...
      <Id id="A.1"/>
      ...
    </PartVersion>
    ...
  </Versions>
  ...
</Part>
```

This example shows that the same part (and its PartVersions) may have a different Identifier at an OEM and at its supplier:

```
<Organization uid="o--000000178">
  <Id id="mercedes-benz.com"/>
  <Name>
    <CharacterString>Mercedes-Benz</CharacterString>
  </Name>
  <OrganizationTypes>
    <ClassString>company</ClassString>
  </OrganizationTypes>
</Organization>
<Organization uid="o--000000179">
  <Id id="bosch.com"/>
  <Name>
    <CharacterString>Bosch</CharacterString>
  </Name>
  <OrganizationTypes>
    <ClassString>company</ClassString>
  </OrganizationTypes>
</Organization>
<Part uid="p--000000001E60C660">
  <Id>
    <Identifier uid="pid--000000001E60C660--id6" id="bolt" idContextRef="o-
-000000178"/>
    <Identifier uid="pid--000000001E60C660--id7" id="bolt4711" idContex-
tRef="o--000000179"/>
  </Id>
  ...
  <Versions>
    <PartVersion uid="pv--000000001E60C660">
      ...
      <Id>
        <Identifier uid="pid--000000001EAA8110--id6" id="A.1" idContex-
tRef="pid--000000001E60C660--id6"/>
      </Id>
    </PartVersion>
  </Versions>
</Part>
```

```
<Identifier uid="pid--000000001EAA8110--id7" id="001,3" idCon-
textRef="pid--000000001E60C660--id7"/>
</Id>
...
</PartVersion>
...
</Versions>
...
</Part>
```

*Comment: according to the XSD:*

```
<xsd:complexType name="Id">
  <xsd:sequence>
    <xsd:element name="Identifier" type="Identifier" minOccurs="0"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string" use="optional"/>
</xsd:complexType>
<xsd:complexType name="Identifier">
  <xsd:complexContent>
    <xsd:extension base="cmn:BaseObject">
      <xsd:sequence>
...
        <xsd:attribute name="id" type="xsd:string" use="optional"/>
...
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
```

*Both elements id (as xsd:string in Identifier and in Id) can be set in parallel, while according to the EXPRESS schema only one is possible (also see BugZilla #5884):*

```
TYPE SingleIdentifierSelect = SELECT(
  Identifier,
  IdentifierString
);
END_TYPE;
```

#### 4.6.7 Template “Description”

In the context of this recommended practice, all the descriptions shall be set with the CharacterString type.

To support PDM environments, multi-language support and other capabilities built on the description, the full definition of the “Description” template will have to be done.

#### 4.6.8 Template “ViewContext”

In the same way than in section 1.1.2.4 of the PDM Schema Usage Guide V4.3, the ViewContext entity identifies a universe suitable for the description of parts.

The Description provides further information about the type of view defined.

The ApplicationDomain attribute contains the application domain information.

The LifeCycleStage attribute contains the life cycle stage information.

### The Instance Model: AP242 BO Model XML entities and attributes

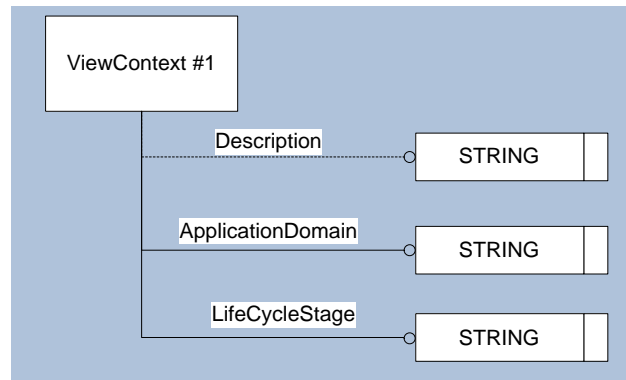


Figure 7: Template "ViewContext"

ENTITY ViewContext	Attribute Type
ApplicationDomain	ApplicationDomainSelect
Description	OPTIONAL DescriptorSelect
LifeCycleStage	LifeCycleStageSelect
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment

Table 9: "ViewContext" Attributes

**Attribute recommendations**

- The **Description** attribute is the text by which the type is described. The value of this attribute need not be specified. Use "Description" template (see 4.6.7).
- The **ApplicationDomain** attribute is the text representing the application domain. Use ProxyString type.
- The **LifeCycleStage** attribute is the text representing the life cycle stage. Use ProxyString type.
- Other attributes than these are not covered by the Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:** The Description attribute provides a distinction on the type of view on a part version ('part definition') from one of a document version ('digital document definition', 'physical document definition'). This attribute may also indicate other types of definitions: e.g., functional, or spatial and/or zonal.

Recommended values for ApplicationDomain include 'assembly study', 'digital mock-up', 'electrical design', 'mechanical design', 'preliminary design', 'process planning', 'product support' and 'not specific'.

Recommended values for LifeCycleStage include 'design', 'manufacturing', 'support', 'recycling' and 'not specific'.

All preprocessors should ensure that the combinations of the ApplicationDomain and the LifeCycleStage are unique.

**Postprocessor Recommendations:** Postprocessors should interpret the value of the description attribute as a type distinction between various definitions of parts and documents. The LifeCycleStage attribute value may be interpreted as the relevant viewpoint from which the data is valid.

**Related Entities:** There are no specific related entities.

## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<ViewContext uid="vc--000000781">
  <ApplicationDomain>
    <ProxyString>mechanical design</ProxyString>
  </ApplicationDomain>
  <LifeCycleStage>
    <ProxyString>design</ProxyString>
  </LifeCycleStage>
</ViewContext>
```

### 4.6.9 Template “NumericalValue”

The NumericalValue is a subtype of ValueWithUnit representing a textual definition and a numerical value associated to a Unit type.

The Definition provides textual information on the property.

The Name provides the identification of the property.

The Unit specifies in which the ValueComponent is expressed.

The ValueComponent is the quantity.

## The Instance Model: AP242 BO Model XML entities and attributes

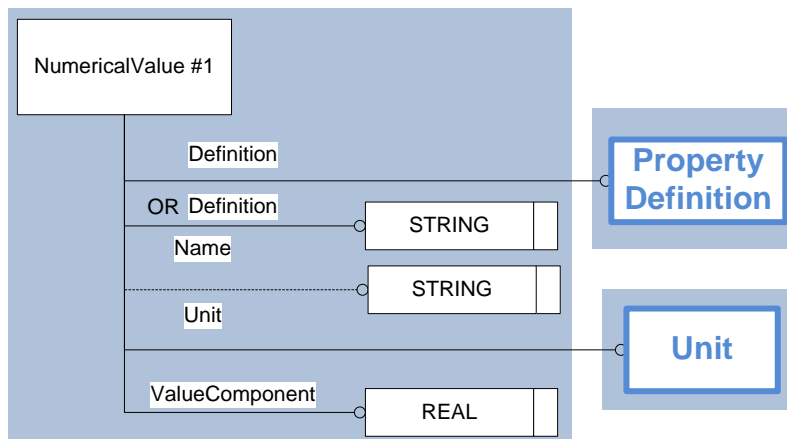


Figure 8: Template "NumericalValue"

ENTITY NumericalValue	Attribute Type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Definition	PropertyDefinitionSelect
DeterminationMethod	OPTIONAL ClassSelect
Name	OPTIONAL DescriptorSelect
Qualifications	OPTIONAL MeasureQualification
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment



ENTITY NumericalValue	Attribute Type
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
ProjectAssignment	OPTIONAL SET[1:?] of ProjectAssignment
PropertyValueRelationship	OPTIONAL SET[1:?] of PropertyValueRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
SignificantDigits	Optional Integer
Unit	UnitSelect
ValueComponent	Double
ValueContext	OPTIONAL NumericalContext

Table 10: "NumericalValue" Attributes

#### Attribute recommendations

- The **Definition** attribute is the text by which the property is described. In the case of system property, PDM property, User Defined attributes (see chapter 12), general shape definition properties (see section 6.3.1) or validation properties (see chapter 13), use "PropertyDefinition" template (see 12.2), otherwise (for example FileSize) use PropertyDefinitionString type.
- The **Name** attribute is the text by which the property is known. The value of this attribute need not be specified. Use "Description" template (see 4.6.7).
- The **Unit** attribute is the Unit of the expressed value. Use "Unit" template (see 4.6.3).
- The **ValueComponent** attribute is the Double representing the quantity.
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:** None specified.

**Postprocessor Recommendations:** None specified.

**Related Entities:** There are no specific related entities.

### The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<Unit uid="u--000000003">
  <Kind>
    <ClassString>SI system</ClassString>
  </Kind>
  <Name>
    <ClassString>byte</ClassString>
  </Name>
  <Prefix>
    <ClassString>kilo</ClassString>
  </Prefix>
</Unit>
```

```

</Unit>
<File xsi:type="n1:DigitalFile" uid="df--0000000088">
  <FileSize uid="fsp--1">
    <FileSize xsi:type="n0:NumericalValue">
      <Definition>
        <PropertyDefinitionString>file size property</PropertyDefinitionString>
      </Definition>
      <Name>
        <CharacterString>file size</CharacterString>
      </Name>
      <Unit uidRef="u--0000000003"></Unit>
      <ValueComponent>46</ValueComponent>
    </FileSize>
  </FileSize>
</File>
    
```

#### 4.6.10 Template “StringValue”

The StringValue is a subtype of PropertyValue representing a textual definition and a text value.

The Definition provides textual information on the property.

The Name provides the identification of the property.

The ValueComponent is the representation of the value.

#### The Instance Model: AP242 BO Model XML entities and attributes

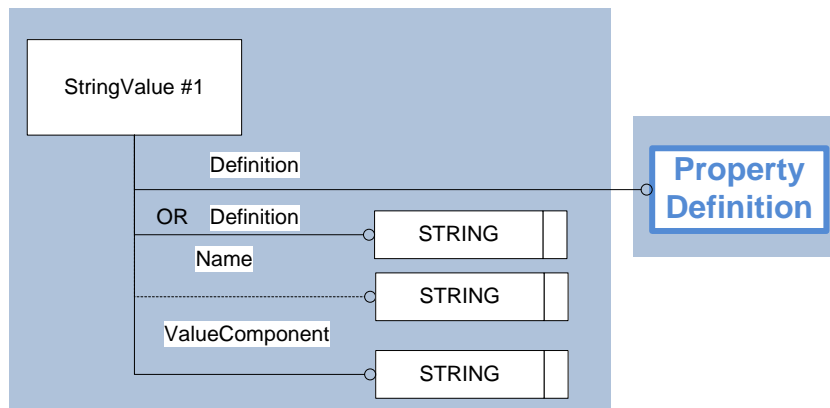


Figure 9: Template "StringValue"

ENTITY StringValue	Attribute Type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Definition	PropertyDefinitionSelect
DeterminationMethod	OPTIONAL ClassSelect
Name	OPTIONAL DescriptorSelect
Qualifications	OPTIONAL MeasureQualification
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment

ENTITY StringValue	Attribute Type
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
ProjectAssignment	OPTIONAL SET[1:?] of ProjectAssignment
PropertyValueRelationship	OPTIONAL SET[1:?] of PropertyValueRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
ValueComponent	MultiLingualStringSelect

Table 11: "StringValue" Attributes

**Attribute recommendations**

- The **Definition** attribute is the text by which the property is described. In the case of system property, PDM property, User Defined attributes (see chapter 12), general shape definition properties (see section 6.3.1) or validation properties (see chapter 13), use "PropertyDefinition" template (see 12.2), otherwise (for example FileSize) use PropertyDefinitionString type.
- The **Name** attribute is the text by which the property is known. The value of this attribute need not be specified. Use of "Description" template (see 4.6.7).
- The **ValueComponent** attribute is the text representing the value. Use of "Description" template (see 4.6.7).
- Other attributes than these are not covered by the Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:** None specified.

**Postprocessor Recommendations:** None specified.

**Related Entities:** There are no specific related entities.

**The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)**

```

...
  <PartView uid="pvv--000000453">
    ...
    <PropertyValue uid="pv--000000455" xsi:type="n0:StringValue">
      <Definition>
        <PropertyDefinition uidRef="pd--000000320"/>
      </Definition>
      <Name>
        <CharacterString>provenance</CharacterString>
      </Name>
      <ValueComponent>
        <CharacterString>CAx-IF</CharacterString>
      </ValueComponent>
    </PropertyValue>
    ...
  </PartView>

```

```

</PartView>

<PropertyDefinition uid="pd--000000320">
  <Id id="quality property"/>
  <PropertyType>
    <ClassString>PDM property</ClassString>
  </PropertyType>
</PropertyDefinition>
    
```

#### 4.6.11 Template “DateTime”

The entity DateTimeAssignment permits the attachment of a DateTimeString to one or more objects.

The Role provides the meaning of the assignment.

The AssignedDate attribute provides date and time information.

#### The Instance Model: AP242 BO Model XML entities and attributes

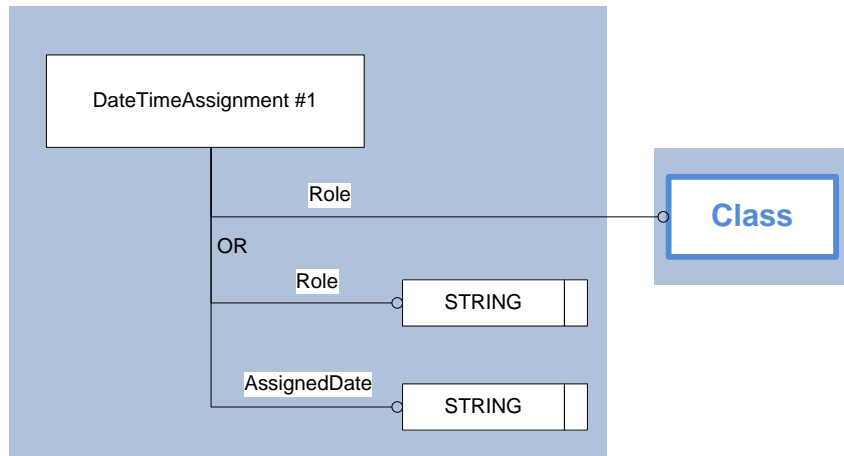


Figure 10: Template “DateTime”

ENTITY DateTimeAssignment	Attribute Type
AssignedDate	Xsd:dateTime
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
Role	ClassSelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
AssignmentObjectRelationship	OPTIONAL SET[1:?] of AssignmentObjectRelationship
ConditionAssignment	OPTIONAL SET[1:?] of ConditionAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment

ENTITY DateTimeAssignment	Attribute Type
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueRelationship	OPTIONAL SET[1:?] of PropertyValueRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 12: "DateTime" Attributes

**Attribute recommendations**

- The **Role** attribute is the text that defines the meaning of the association of the date and time to an object. Use ClassString if one of the values below is used, otherwise use "Class" template (see 4.6.4).
- The **AssignedDate** attribute is the text representing the date and time information. Use DateTimeString type.
- Other attributes than these are not covered by the Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:** The representation of DateTimeString shall respect ISO 8601. In the BO Model definition of DateTimeString type it is mandatory to represent the date, whereas the time is optional, but with the xsd:dateTime type both date and time are mandatory. It is recommended to set the time with the following default value: YYYY-MM-DDT00:00:00.

When applicable, the following values shall be used for the attribute Role:

Role	Explanation
'classification date'	the specified object is classified at the given date and time
'creation'	the referenced object was created at the given date and time
'installation'	the referenced object was mounted in a product at the given date and time
'lock'	the specified object is locked in the underlying legacy system since the given date and time
'production'	the referenced object was produced at the given date and time
'registration'	the referenced object was determined at the given date and time
'update'	the referenced object was altered at the given date and time

**Postprocessor Recommendations:** Postprocessors should interpret the value of the AssignedDate attribute according to ISO 8601

**Related Entities:** There are no specific related entities.

**The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)**

```
...
  <DateTimeAssignment uid="dta--0000000017D374A0--id1">
    <AssignedDate>2014-10-16T09:08:07</AssignedDate>
  </DateTimeAssignment>
```

```

    <Role>
      <ClassString>creation</ClassString>
    </Role>
  </DateTimeAssignment>
  ...
  
```

#### 4.6.12 Template “Approval”

The entity `ApprovalAssignment` allows the attachment of an `Approval` to one or more objects.

The entity `Approval` represents a statement made by technical personnel or management personnel whether certain requirements are met.

The `Description` provides further information about the approval.

The `Status` attribute provides a user interpretable designation of the level of acceptance.

#### The Instance Model: AP242 BO Model XML entities and attributes

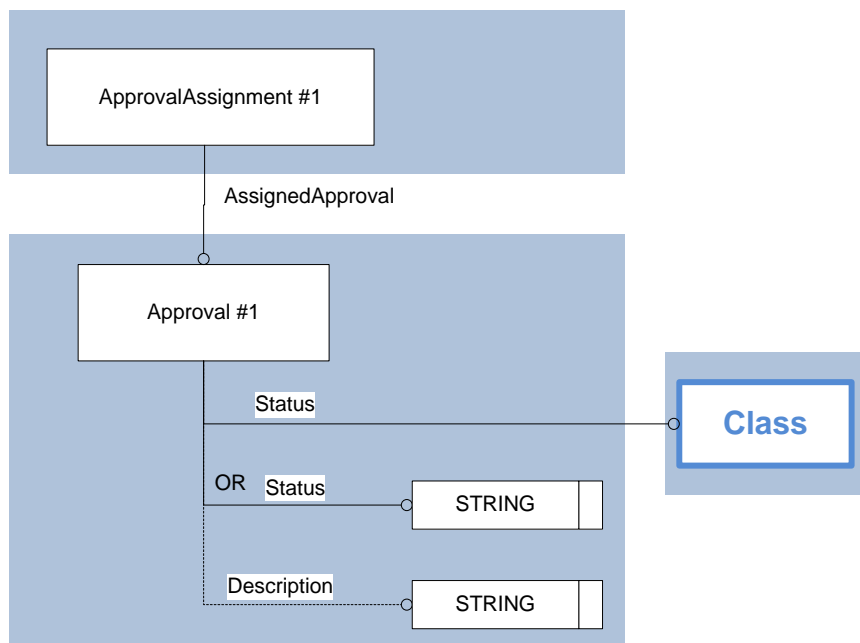


Figure 11: Template “Approval”

ENTITY ApprovalAssignment	Attribute Type
AssignedApproval	Approval
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
AssignmentObjectRelationship	OPTIONAL SET[1:?] of AssignmentObjectRelationship
ConditionAssignment	OPTIONAL SET[1:?] of ConditionAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment

ENTITY ApprovalAssignment	Attribute Type
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueRelationship	OPTIONAL SET[1:?] of PropertyValueRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 13: "ApprovalAssignment" Attributes

**Attribute recommendations**

- The **AssignedApproval** attribute is the reference to the Approval entity.
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:** None specified.

**Postprocessor Recommendations:** None specified.

**Related Entities:** There are no specific related entities.

ENTITY Approval	Attribute Type
ApprovalScope	OPTIONAL SET[1:?] of ApprovalScopeSelect
ApprovedBy	OPTIONAL SET[1:?] of ApprovingPersonOrganization
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
SameAs	OPTIONAL SET[1:?] of ProxySelect
Status	ClassSelect
ValidDate	Xsd:dateTime
ApprovalRelationship	OPTIONAL SET[1:?] of ApprovalRelationship
ConditionAssignment	OPTIONAL SET[1:?] of ConditionAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment

ENTITY Approval	Attribute Type
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueRelationship	OPTIONAL SET[1:?] of PropertyValueRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 14: "Approval" Attributes

#### Attribute recommendations

- The **Status** attribute is the text representing a user interpretable designation of the level of acceptance. Use ClassString if one of the values below is used, otherwise use "Class" template (see 4.6.4).
- The **Description** attribute is the text by which the approval is described. The value of this attribute need not be specified. Use "Description" template (see 4.6.7).
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:** When applicable, the following values shall be used for the attribute Status:

- 'in progress'
- 'approved'
- 'approved with comments'
- 'not approved'.

**Postprocessor Recommendations:** None specified.

**Related Entities:** There are no specific related entities.

### The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```

...
<ApprovalAssignment uid="appas--0000000017D374A0--id1">
  <AssignedApproval uidRef="app--inprogress"/>
</ApprovalAssignment>
...
<Approval uid="app--inprogress">
  <Description>
    <CharacterString>disposition</CharacterString>
  </Description>
  <Status>
    <ClassString>in progress</ClassString>
  </Status>
</Approval>

```

#### 4.6.13 Template "Person"

The entity `Person` represents an individual human being.

The `Id` attribute provides a unique identification of the person.

The `FirstName` attribute provides the first name of the person.

The `LastName` attribute provides the last name of the person.



## The Instance Model: AP242 BO Model XML entities and attributes

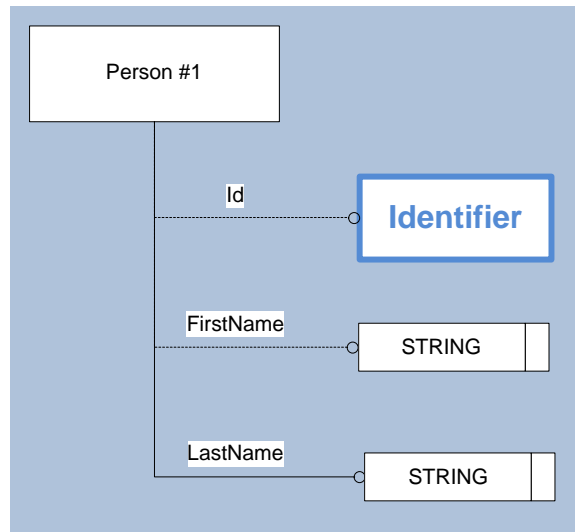


Figure 12: Template “Person”

ENTITY Person	Attribute Type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
FirstName	OPTIONAL String
Id	OPTIONAL IdentifierSelect
LastName	String
MiddleNames	OPTIONAL LIST[1:?] of String
PrefixTitles	OPTIONAL LIST[1:?] of String
SameAs	OPTIONAL SET[1:?] of ProxySelect
SuffixTitles	OPTIONAL LIST[1:?] of String
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueRelationship	OPTIONAL SET[1:?] of PropertyValueRelationship

ENTITY Person	Attribute Type
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssignment

Table 15: "Person" Attributes

### Attribute recommendations

- The **FirstName** attribute is the text by which the human being is known. The value of this attribute need not be specified.
- The **Id** attribute is the identifier that distinguishes the person. The value of this attribute need not be specified. Use "Identifier" template (see 4.6.6).
- The **LastName** attribute is the text by which the human being is known.
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:** None specified.

**Postprocessor Recommendations:** None specified.

**Related Entities:** There are no specific related entities.

### The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<Person uid="pers--Mustermann">
  <FirstName>Max</FirstName>
  <Id>
    <Identifier uid="pers--Mustermann--1" id="4711" idContextRef="o--
000000178"/>
  </Id>
  <LastName>Mustermann</LastName>
</Person>
```

#### 4.6.14 Template "PersonInOrganization"

The entity `OrganizationOrPersonInOrganizationAssignment` allows the attachment of a `PersonInOrganization` to one or more objects.

The `Role` attribute specifies the responsibility of the assigned person.

The entity `PersonInOrganization` represents the membership of a person in an organization with a specific role.

The `AssociatedPerson` is a reference to the person.

The `AssociatedOrganization` is a reference to the organization.

The `Id` provides a unique identification to the `PersonInOrganization`.

The `PersonRole` attribute specifies the role of the person inside the organization.

### The Instance Model: AP242 BO Model XML entities and attributes

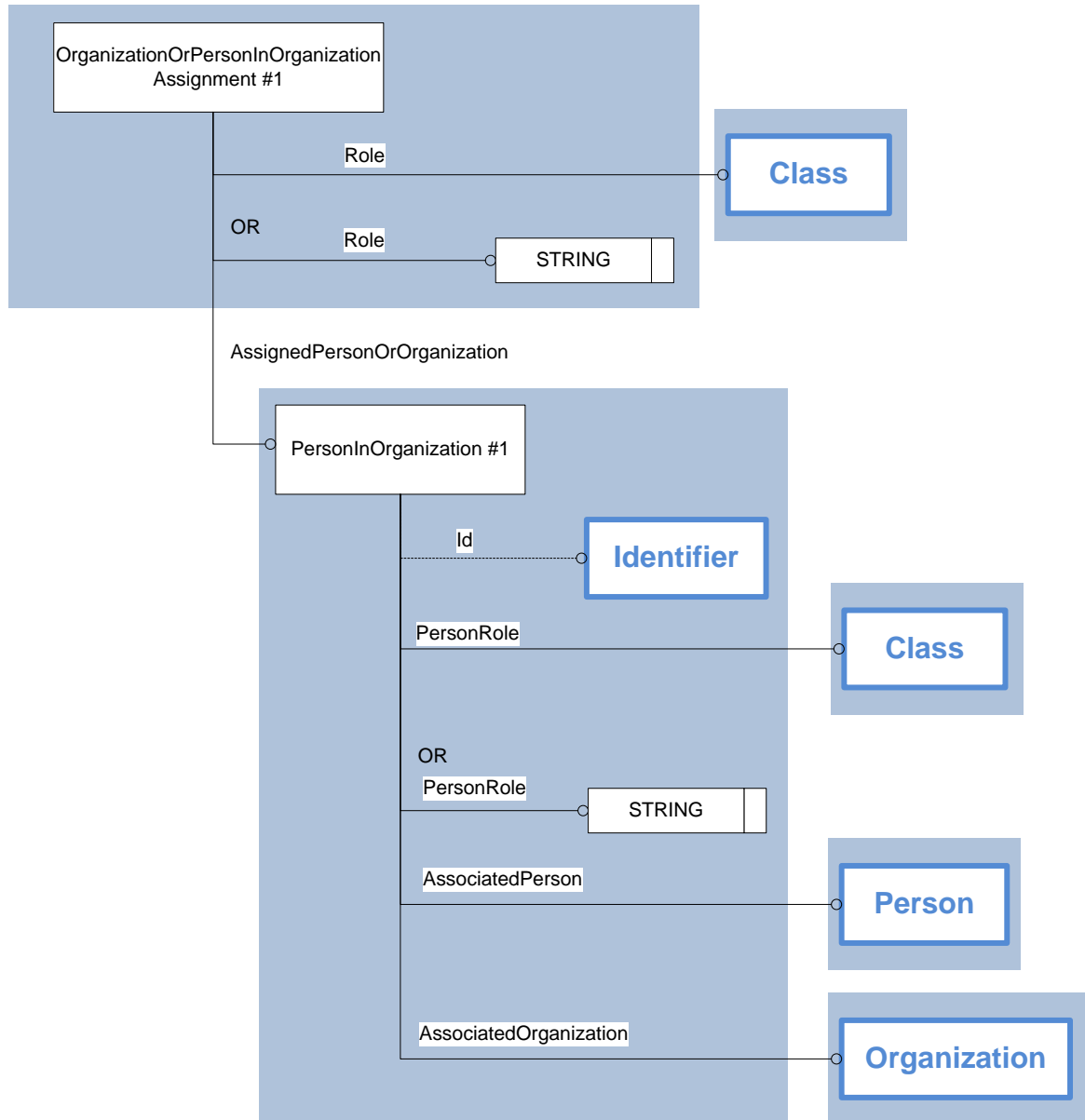


Figure 13: Template “PersonInOrganization”

ENTITY OrganizationOrPersonInOrganizationAssignment	Attribute Type
AssignedPersonOrOrganization	OrganizationOrPersonInOrganizationSelect
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
Role	ClassSelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
AssignmentObjectRelationship	OPTIONAL SET[1:?] of AssignmentObjectRelationship
ConditionAssignment	OPTIONAL SET[1:?] of ConditionAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment

ENTITY OrganizationOrPersonInOrganizationAssignment	Attribute Type
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueRelationship	OPTIONAL SET[1:?] of PropertyValueRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 16: "OrganizationOrPersonInOrganizationAssignment" Attributes

**Attribute recommendations**

- The **AssignedPersonOrOrganization** attribute defines the person inside an organization with a reference to the PersonInOrganization entity or an organization with a reference to the Organization entity.
- The **Role** attribute is the text describing the responsibility of the person. Use ClassString if one of the values below is used, otherwise use "Class" template (see 4.6.4).
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:** When applicable, the following values shall be used for the attribute Role:

Role	Explanation
'author'	The author holds the copyright
'classification officer'	The assigned person or organization is formally responsible for the classification of the referenced object
'creator'	The referenced object has been created by the assigned person or organization
'custodian'	The assigned person or organization is responsible for the existence and integrity of the referenced object
'customer'	The assigned person or organization acts as a purchaser or consumer of the referenced object
'design supplier'	The assigned person or organization is the one who delivers the data describing the referenced object
'editor'	One or more attributes have been modified by the assigned person or organization
'id owner'	The assigned person or organization is the one responsible for the designation of an identifier
'location'	The assigned organization is the place where the referenced object

Role	Explanation
	can be found or where it takes place
'locked by'	The assigned person that currently locks the associated object in the underlying legacy system
'manufacturer'	The assigned person or organization is the one who produces the actual (physical) object
'owner'	The assigned person or organization owns the referenced object, and has final say over its disposition and any changes to it
'read access'	The assigned person or organization neither has the right to modify any attributes of the referenced object, nor to modify, create or delete objects that are attached directly or indirectly to the referenced object
'supplier'	The assigned person or organization is the one who delivers the actual (physical) object (e.g., a dealer)
'wholesaler'	The assigned person or organization the one who is in the sales chain between the manufacturer and the supplier
'write access'	The assigned person or organization has the right to modify the attributes of the referenced object, as well as to modify, create, or delete objects that are attached directly or indirectly to the referenced object

**Postprocessor Recommendations:** None specified.

**Related Entities:** There are no specific related entities.

ENTITY PersonInOrganization	Attribute Type
AssociatedOrganization	Organization
AssociatedPerson	Person
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
PersonRole	ClassSelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
AddressAssignment	OPTIONAL SET[1:?] of AddressAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
ContractAssignment	OPTIONAL SET[1:?] of ContractAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment

ENTITY PersonInOrganization	Attribute Type
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PersonInOrganizationRelationship	OPTIONAL SET[1:?] of PersonInOrganizationRelationship
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueRelationship	OPTIONAL SET[1:?] of PropertyValueRelationship
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssignment

Table 17: "PersonInOrganization" Attributes

**Attribute recommendations**

- The **AssociatedPerson** attribute defines the person. Use "Person" template (see 4.6.13).
- The **AssociatedOrganization** attribute defines the organization. Use "Organization" template (see 4.6.2).
- The **Id** attribute is the identification of the entity. Use "Identifier" template (see 4.6.6).
- The **PersonRole** attribute is the text describing the role. Use ClassString.
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:** None specified.

**Postprocessor Recommendations:**

When applicable, the following values shall be used for the attribute Role:

- 'employee': The associated person is a member of kind 'employee' of the associated organization.

**Related Entities:** There are no specific related entities.

**The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)**

```

...
<OrganizationOrPersonInOrganizationAssignment uid="poa--0000000017D374A0-
-id1">
  <AssignedPersonOrOrganization uidRef="pio--005-TPEVD-Mustermann"/>
  <Role>
    <ClassString>creator</ClassString>
  </Role>
</OrganizationOrPersonInOrganizationAssignment>
...
<PersonInOrganization uid="pio--005-TPEVD-Mustermann">
  <AssociatedOrganization uidRef="org--005-TPEVD"/>
  <AssociatedPerson uidRef="pers--Mustermann"/>
  <Id>
    <Identifier uid="pio--005-TPEVD-Mustermann--1" id="005-TP/EVD-
Mustermann" idContextRef="o--000000178"/>
  </Id>
  <PersonRole>
    <ClassString>employee</ClassString>

```

```
</PersonRole>  
</PersonInOrganization>
```

## 5 Part Identification and Classification

The scope of this section corresponds to sections 1 and 2 of the PDM Schema Usage Guide V4.3.

The AP242 BO Model manages industrial products as Parts. An AP242 BO Model conformant data exchange shall include at least one element of type Part.

Identification of Parts in the AP242 BO Model uses three concepts:

- Part Master Identification,
- Context Information, and
- Type Classification.

Part master identification consists of the base part number, a unique part version identification, and – optionally – the identification of a view definition that describes application domain, lifecycle stage and property values. Details are specified in 5.1.

Context information provides scope and environment of interpretation of product identification information. Context information may be given locally, that is, for a single XML element, such as, for Parts using ViewContext (see PartView in 5.1.3 and the “ViewContext” template in 4.6.8), or globally for the entire physical file using the element ExchangeContext for stating the organization that owns all identifiers in the data set (see template in 4.6.1). For a summary of all context mechanisms, see 4.1.6.

For Part classification the AP242 BO Model distinguishes the following two approaches:

- Type classification
  - An identified Part may be placed into one or several of the following categories: 'piece part', 'product', 'software', 'assembly', 'tool', or 'raw material'. These values are set in the attribute Part.PartTypes; see 5.1.1 .
- General classification
  - Parts may need to be classified according to a classification system with explicit reference to classification criteria and related properties. For example, pumps may be classified according to their principle of working and their capacity. Such classification is enabled by the attribute Part.ClassifiedAs; see 5.1.1. Thus, a Part may be linked to an extensive and already existing classification system.

These three concepts are represented in a data exchange by attributes of the three main information elements in each of the two templates “Part” and “Assembly”.

### 5.1 Templates “Part” and “Assembly”

To enable independent use of Parts and Assemblies both a “Part” template and an “Assembly” template are specified here. They support the ability to uniquely identify Parts and Assemblies including their metadata and properties. This backbone of the AP242 information model consists in the AP242 BO Model of the following structurally distinct data types as also shown in Figure 14 and Figure 15:

- Part,
- PartVersion and,
- PartView respectively AssemblyDefinition.

The representations of the Part and PartVersion concepts are identical for both the “Part” and the “Assembly” templates; only on the third level of detail they differ, as shown in Figure 14 and Figure 15 below.

The Part maintains information common to all Part versions and disciplines and/or life-cycle views. It contains the base Part number and name. The base number should not be subject to any encoding of information into a single complex parseable string.

The version information may represent a design revision or iteration in a design cycle of a part. The Part version collects and, thus, relates all information among all associated disciplines and life-cycle view definitions.

Part, PartVersion and PartView, respectively AssemblyDefinition, shall be written to the XML-file using containment. The information elements in the white area on the left side of Figure 14 and Figure 15 are root elements and are, thus, outside of the containment blocks.

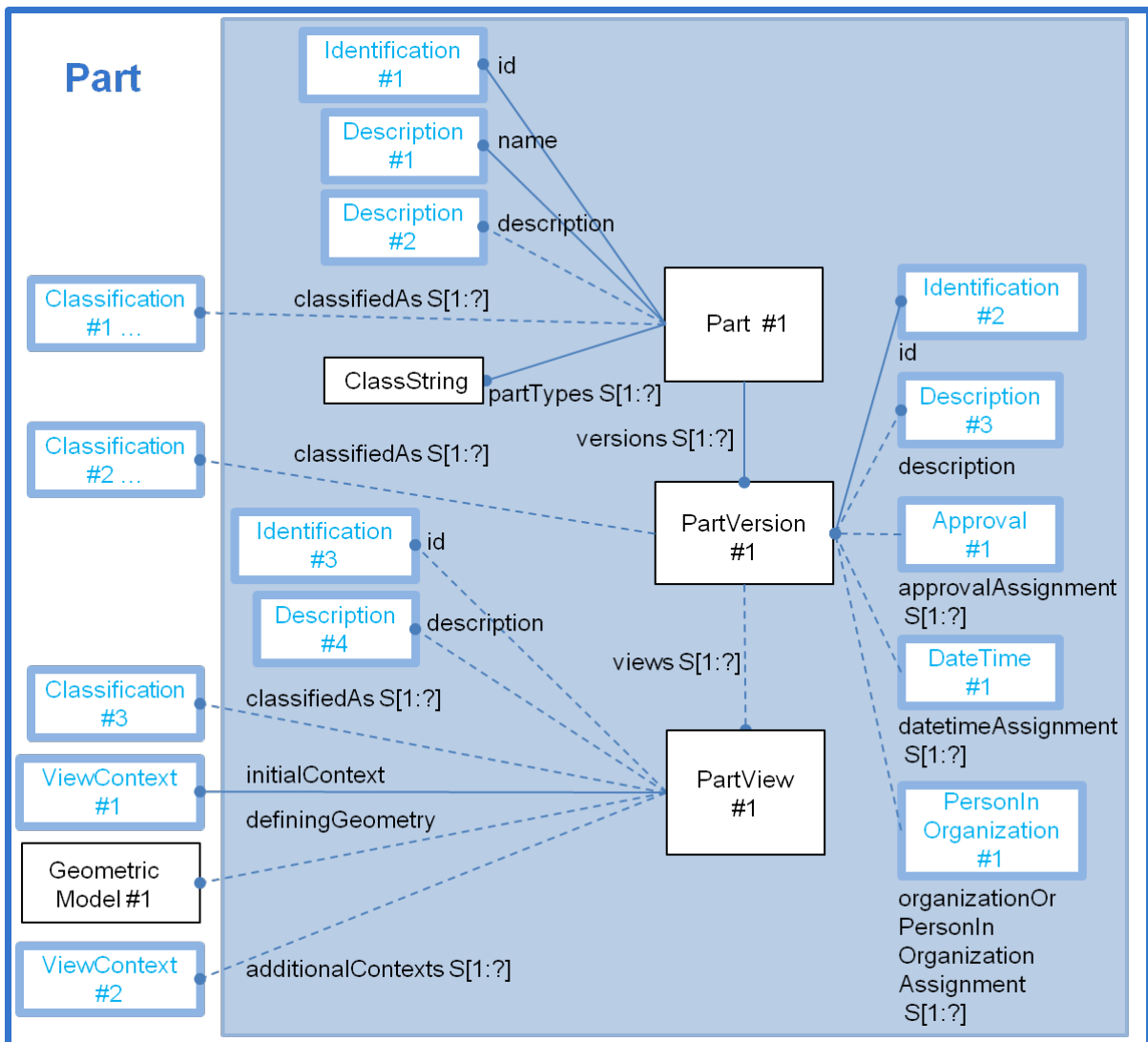


Figure 14: Template “Part”



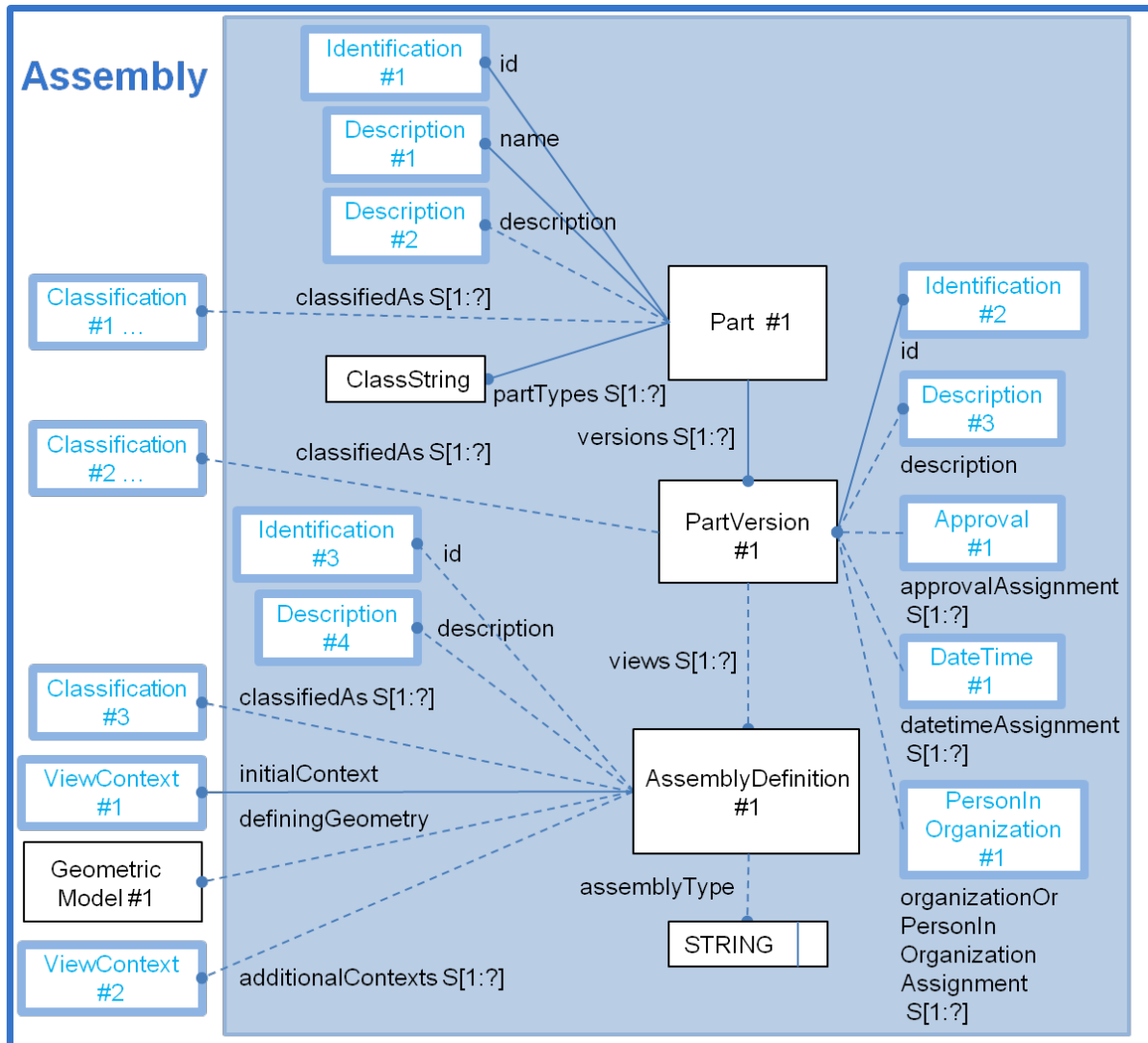


Figure 15: Template "Assembly"

### 5.1.1 Part

The Part entity represents the part master base information. This entity collects all information that is common among the different versions and views of the part. The part number is strictly an identifier. It should not be used as a 'smart string' with some parseable internal coding scheme, e.g., to identify version or classification information.

The Part number identifier shall be unique within the scope of the business process of the information exchange. This is typically not a problem when the product data is only used within a single company. If the data is being assembled for external use, the identification must be interpreted as unique within that broader domain. Processors may need to evaluate more than one string (i.e., more than only Part.id) to establish unique identification of the Part. The "Identifier" template provides a combination of parameters including Identifier.idRoleRef and Identifier.idContextRef that make Part identification unique.

The following XML-snippet is an example from a physical file that is in accordance to Figure 14.

## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```

<Part uid="p--0000000017086CB0">
  <Id>
    <Identifier uid="pid--0000000017086CB0--id1" id="as1" idContextRef="o--000000178"/>
  </Id>
  ...
  <Versions>
    <PartVersion uid="pv--0000000017086CB0--id1">
    ...
      <Views>
        <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--0000000017086CB0--id1">
        ...
          </PartView>
        </Views>
      </PartVersion>
    </Versions>
  </Part>
<Part uid="p--000000001E5A89F0">
  <Id>
    <Identifier uid="pid--000000001E5A89F0--id2" id="plate" idContextRef="o--000000178"/>
  </Id>
  ...
  <Versions>
    <PartVersion uid="pv--000000001E5A89F0--id2">
    ...
      <Views>
        <PartView uid="pvv--000000001E5A89F0--id2">
        ...
          </PartView>
        </Views>
      </PartVersion>
    </Versions>
  </Part>

```

Entity Part attributes	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
Name	DescriptorSelect
PartTypes	SET[1:?] of ClassSelect
SameAs	OPTIONAL SET[1:?] of ProxySelect
Versions	SET[1:?] of PartVersion

Entity Part attributes	Attribute type
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
CertificationAssignment	OPTIONAL SET[1:?] of CertificationAssignment
ContractAssignment	OPTIONAL SET[1:?] of ContractAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DatetimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PartRelationship	OPTIONAL SET[1:?] of PartRelationship
ProjectAssignment	OPTIONAL SET[1:?] of ProjectAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssignment

Table 18: "Part" Attributes

#### Attribute recommendations

- **ClassifiedAs:** the classifications of the Part. The value of this attribute need not be specified. Use "Classification" template (see 4.6.5).
- **Description:** an expanded name or text that provides further information about the Part. The value of this attribute need not be specified. Use "Description" template (see 4.6.7).
- **Id:** the identifier or set of identifiers for the Part, the part number. Use "Identifier" template (see 4.6.6).

- **Name:** the nomenclature or common name of the Part. Use “Description” template (see 4.6.7).
- **PartTypes:** the category of a Part. Use ClassString type, that is, PartTypes is a set of one or many strings. The value of this element shall be one or several of the following:

PartTypes	Explanation
'piece part'	a product that is not subject to decomposition from the perspective of a specific application; is also called component
'product'	a thing or substance produced by a natural or artificial process; may be a piece part, an assembly of piece parts, a tool, an assembly of tools, and raw material
'software'	a non-tangible product that is an organized collection of computer data and instructions for use by a computer
'tool'	a product used to manufacture products by applying various manufacturing technologies
'assembly'	a product that is decomposable into a set of piece parts or other assemblies from the perspective of a specific application
'raw material'	basic substance in its natural, modified, or semi-processed state, used as an input to a production process that shall result in piece parts and tools

- **Versions:** the related variants of the Part; a Part shall have at least one PartVersion.
- **PropertyValueAssignment:** to assign a PropertyValue to the Part. Use the “PropertyAssignment” template; see 6.2 for details.

**Preprocessor Recommendation:**

- Properties shall only be associated to the Part if they apply in the source PDM system to ALL PartVersions (i.e. if the source PDM system has two objects: a part master object and a part version object).

**Postprocessor Recommendation:**

- If the target PDM system also has a part master object, these properties shall be mapped to it and apply to all part versions. If not, they shall be mapped only to those part versions that are mentioned for this part in the XML file.

Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**5.1.2 PartVersion**

The PartVersion element represents the identification of a specific version of the base Part identification. A particular PartVersion is always related to exactly one Part. This is why, in XML it is embedded within a Part element.

**Preprocessor Recommendations:**

- For the purpose of the typical CAx data exchange use case of these recommended practices, only one view definition (PartView) shall be assigned to each PartVersion and only one version for each part.
- For the purpose of the typical PDM data exchange use case of these recommended practices, multiple versions of each part and multiple views of each version may be exchanged. This has, however, not yet been fully tested by the PDM-IF.

Examples of PartVersion instantiations are in the XML-snippet in section 5.1.1 .

<b>Entity PartVersion attributes</b>	<b>Attribute type</b>
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
SameAs	OPTIONAL SET[1:?] of ProxySelect
Views	OPTIONAL SET[1:?] of PartView
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
CertificationAssignment	OPTIONAL SET[1:?] of CertificationAssignment
ContractAssignment	OPTIONAL SET[1:?] of ContractAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DatetimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PartVersionRelationship	OPTIONAL SET[1:?] of PartVersionRelationship
ProjectAssignment	OPTIONAL SET[1:?] of ProjectAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssignment

Table 19: "PartVersion" Attributes

### **Attribute recommendations**

- **ApprovalAssignment:** the level of acceptance of the PartVersion. The value of this attribute need not be specified. Use "Approval" template (see 4.6.12).
- **ClassifiedAs:** the classifications of the PartVersion. The value of this attribute need not be specified. Use "Classification" template (see 4.6.5).
- **DatetimeAssignment:** the date and time of the creation or update of the PartVersion. The value of this attribute need not be specified. Use "DateTime" template (see 4.6.11).
- **Description:** the reason for the creation of the version. The value of this attribute need not be specified. Use "Description" template (see 4.6.7).
- **Id:** the identifier or set of identifiers for the PartVersion, the part version number. Use IdentifierString type or "Identifier" template (see 4.6.6).
  - **Preprocessor Recommendations:** If an organization does not version parts, it is recommended that the id attribute contains the string '/NULL' to indicate that no version information is relevant or intended. In this case only a single PartVersion shall be assigned to the Part. The id attribute shall be given the value /ANY if the assembly structure of the source system stores only the Part number and computes the identifier of the PartVersion at runtime based on parameters, such as, latest version and version valid at a given time.
    - Note: This technique may reduce the amount of data sent in change packages, but it also reduces the ability to track the actual contents of parts lists at a particular change level.
  - For the purpose of the typical CAx data exchange use case of these recommended practices, the use of '/ANY' is not recommended.
  - **Postprocessor Recommendations:** If the value of the id attribute for a PartVersion is the string '/NULL', postprocessors should use this as an indication that the sending system or business process does not support versioning of Parts. Postprocessors need to recognize an id value of '/ANY' as a generic revision of a Part that is involved as a component in an assembly. This is used to indicate that any existing revision of the component is valid for use in the parent assembly and that the right PartVersion identifier must be computed at runtime.
- **Views:** the set of PartView objects that are defined for the PartVersion.
  - Each PartVersion shall have at least one associated PartView. This PartView shall represent the mechanical view definition of the part. For this mandatory PartView, the SET-type attribute PartView.initialContext.applicationDomain.sameAs shall contain the ProxyString type value 'mechanical design'. No other instances of PartView of the same PartVersion shall contain this string.
  - Other instances of PartView may be associated to the same PartVersion, for example, a PartView of the composites representation of the part. A meaningful value of PartView.initialContext.applicationDomain.sameAs should be agreed between data exchange partners; the list of pre-defined values in 4.6.8 may be extended by user-defined values.
- **OrganizationOrPersonInOrganizationAssignment:** an organization or person in organization with a specific relation to the PartVersion according to the OrganizationOrPersonInOrganizationAssignment.role attribute. The value of this attribute need not be specified. Use "PersonInOrganization" template (see 4.6.14).

- **PartVersionRelationship:** a PartVersion of the same Part or of a different Part with a specific relation to the PartVersion according to the PartVersionRelationship.RelationType attribute. The value of this attribute need not be specified. Use "PartVersionRelationship" template (see 5.1.5).
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

### 5.1.3 PartView

The PartView entity represents the identification of a particular view on a version of the part base identification relevant for the requirements of particular life-cycle stages, application domains and user-defined properties. A PartView may be based on an application domain and/or a life-cycle stage (e.g., design, manufacturing). A PartView collects product data for a specific discipline and life-cycle. More than one PartView may be associated with a particular PartVersion, each representing a different view of the Part.

AssemblyDefinition is a subtype of PartView. It is used to associate subordinate components of the Part.

The PartView entity enables the establishment of many relationships between Parts and other product data concepts, such as, assembly structures, properties (including shape), and external descriptions of the product via documents (see chapter 7.4).

#### **Preprocessor Recommendations:**

- The use of PartView entities is not strictly required by rules in the AP242 BO Model, but it is strongly recommended. All PartVersion entities shall have at least one associated PartView.
- If a PDM system does not distinguish between PartVersion and PartView, only one PartView shall be mapped (having id as unset).

An example of a PartView instantiation is in the XML-snippet in section 5.1.1.

Entity PartView attributes	Attribute type
AdditionalContexts	OPTIONAL SET[1:?] of ViewContext
ClassifiedAs	OPTIONAL SET[1:?] of Classification
DefiningGeometry	OPTIONAL GeometricModel
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
InitialContext	ViewContext
InZone	OPTIONAL SET[1:?] of InZone
MaterialIdentification	OPTIONAL SET[1:?] of MaterialIdentification
Occurrence	OPTIONAL SET[1:?] of Occurrence
SameAs	OPTIONAL SET[1:?] of ProxySelect
ShapeDependentProperty	OPTIONAL SET[1:?] of ShapeDependentProperty
ShapeElement	OPTIONAL SET[1:?] of ShapeElement
SurfaceCondition	OPTIONAL SET[1:?] of SurfaceCondition
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment

Entity PartView attributes	Attribute type
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
BreakdownVersionAssignment	OPTIONAL SET[1:?] of BreakdownVersionAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DatetimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PartViewRelationship	OPTIONAL SET[1:?] of PartViewRelationship
ProjectAssignment	OPTIONAL SET[1:?] of ProjectAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
ViewOccurrenceRelationship	OPTIONAL SET[1:?] of ViewOccurrenceRelationship
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssignment

Table 20: "PartView" Attributes

### Attribute recommendations

- **AdditionalContexts:** the set of ViewContext objects that are relevant context descriptions for this PartView in addition to the InitialContext. The AdditionalContexts shall not contain the ViewContext that is referenced as the InitialContext. The value of this attribute need not be specified. Use "ViewContext" template (see 4.6.8).
- **ClassifiedAs:** the classifications of the PartView. The value of this attribute need not be specified. Use "Classification" template (see 4.6.5).



*Comment: currently the XSD doesn't detect if an instance of 'Class' instead of 'Classification' is referenced, an issue has been created in Bugzilla under #5153*

- **DefiningGeometry:** the GeometricModel that provides the shape for the PartView. See 6.1 for details of instantiating a GeometricModel and linking it to a PartView. The value of this attribute need not be specified.
- **Description:** text or the set of texts that provide further information about the PartView. The value of this attribute need not be specified. Use "Description" template (see 4.6.7).
- **Id:** the identifier or set of identifiers for the PartView. The value of this attribute need not be specified. Use IdentifierString type or "Identifier" template (see 4.6.6).
  - **Preprocessor Recommendations:** There is no standard mapping for the id attribute of PartView; however, the value should be unique relative to other PartViews related to the same PartVersion. The id attribute shall not be 'overloaded' to include, for example, life-cycle or organizational information; this is generally not recommended for the AP242 BO Model. This attribute should contain a unique identifier for the PartView - no additional semantics are associated with this attribute.
  - **Postprocessor Recommendations:** Postprocessors do not need to expect any semantics from the id attribute; it is a pure identifying string. The id value – possibly composed of several values according to the "Identifier" template - should be unique relative to other the identifiers of other PartViews related to the same PartVersion.
- **InitialContext:** the ViewContext in which this view of the PartVersion has been designed primarily. Use "ViewContext" template (see 4.6.8).
- **Occurrence:** the instantiations of the PartView in a product structure. The element Occurrence itself cannot be instantiated. For the purpose of these recommended practices only subtypes "SingleOccurrence" (see template in 7.1) and "SpecifiedOccurrence" (see template in 7.2) shall be used.
- **ShapeDependentProperty:** a characteristic of the shape, or of a portion of the shape of a PartView. The element ShapeDependentProperty itself cannot be instantiated. All three subtypes may be used, but these recommended practices provide guidance only for GeneralShapeDependentProperty (see template in 6.3).
- **DocumentAssignment:** to assign a DocumentVersion to the PartView. See 11.2 for details.
- **PartViewRelationship:** to assign an assembly link to the PartView. See the restrictions to the use of this construct in the section 7.4.
- **PropertyValueAssignment:** to assign a PropertyValue to the PartView. Use the "PropertyAssignment" template; see 6.2 for details.
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

#### 5.1.4 AssemblyDefinition

The AssemblyDefinition is a definition of a PartVersion that associates subordinate components to this PartVersion. It is a subtype of PartView and inherits, thus, all its attributes. As for PartViews, occurrences can be derived from AssemblyDefinition, properties, such as, shape, can be assigned to it and documents may be associated with it.

Components are added to an AssemblyDefinition by NextAssemblyOccurrenceUsage; see 7.1 and 7.2.

An example of an AssemblyDefinition instantiation is in the XML-snippet in section 5.1.1.

**Preprocessor Recommendations:** single parts (having no component parts beyond them), shall not be mapped as AssemblyDefinition, but rather as PartView.

Entity AssemblyDefinition attributes	Attribute type
AdditionalContexts	OPTIONAL SET[1:?] of ViewContext
ClassifiedAs	OPTIONAL SET[1:?] of Classification
DefiningGeometry	OPTIONAL GeometricModel
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
InitialContext	ViewContext
InZone	OPTIONAL SET[1:?] of InZone
MaterialIdentification	OPTIONAL SET[1:?] of MaterialIdentification
Occurrence	OPTIONAL SET[1:?] of Occurrence
SameAs	OPTIONAL SET[1:?] of ProxySelect
ShapeDependentProperty	OPTIONAL SET[1:?] of ShapeDependentProperty
ShapeElement	OPTIONAL SET[1:?] of ShapeElement
SurfaceCondition	OPTIONAL SET[1:?] of SurfaceCondition
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
BreakdownVersionAssignment	OPTIONAL SET[1:?] of BreakdownVersionAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DatetimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PartViewRelationship	OPTIONAL SET[1:?] of PartViewRelationship
ProjectAssignment	OPTIONAL SET[1:?] of ProjectAssignment

Entity AssemblyDefinition attributes	Attribute type
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
ViewOccurrenceRelationship	OPTIONAL SET[1:?] of ViewOccurrenceRelationship
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssignment
AssemblyType	OPTIONAL ClassSelect
KinematicMechanismAssociation (If the Bugzilla issue #5669 is accepted)	OPTIONAL SET OF KinematicMechanismAssociation

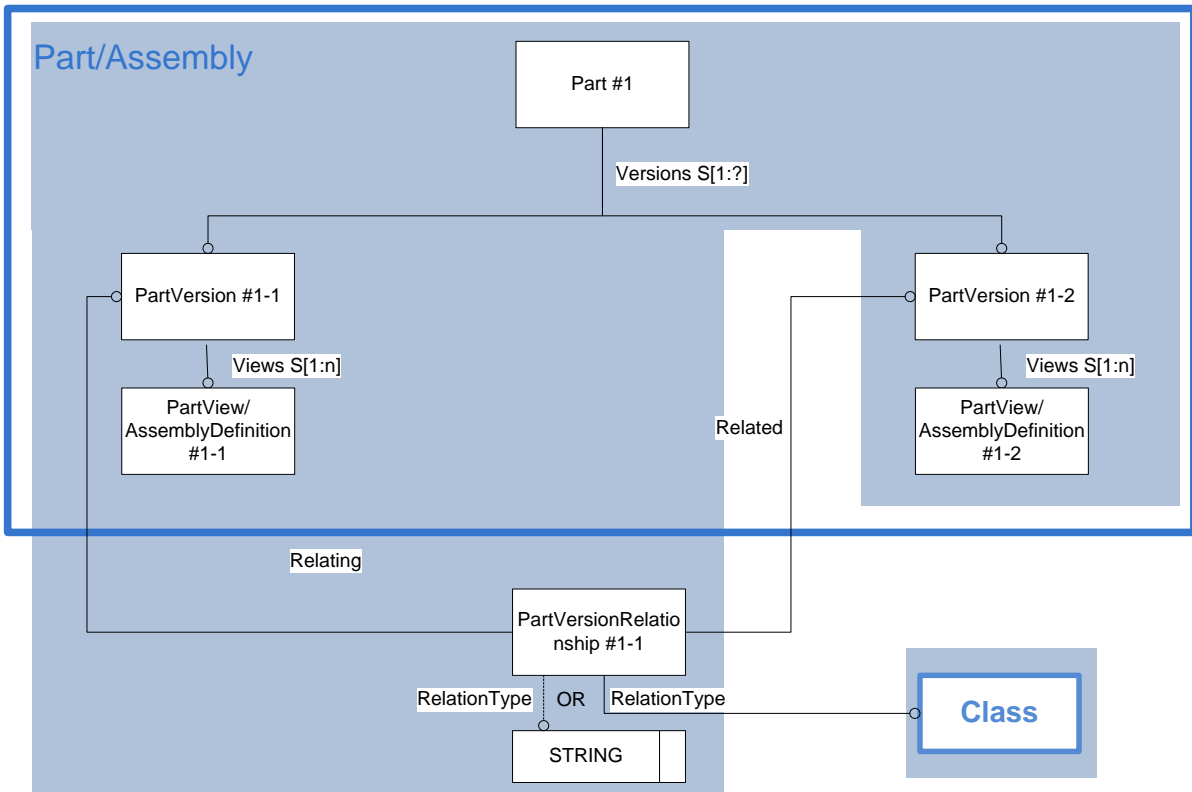
Table 21: "AssemblyDefinition" Attributes, including attributes inherited from "PartView"

#### Attribute recommendations

- **AssemblyType:** the kind of the AssemblyDefinition. The value of this attribute need not be specified. The following are examples of recommended AssemblyType values:
  - 'functional assembly',
  - 'manufacturing assembly',
  - 'design assembly' .
- **ViewOccurrenceRelationship:** to assign an assembly link to the PartView. Use the "SingleOccurrence" template (see 7.1 for details) or "SpecifiedOccurrence" template (see 7.2 for details).
- In addition, all attributes and attribute recommendations for PartView apply.
- **KinematicMechanismAssociation:** to assign (optionally) one or multiple Mechanisms to the AssemblyDefinition. Use the "Mechanism" template (see Recommended Practices for AP242 BO Model XML Kinematics for details).

#### 5.1.5 PartVersionRelationship

Used to relate several versions of the same part:



or of different parts.

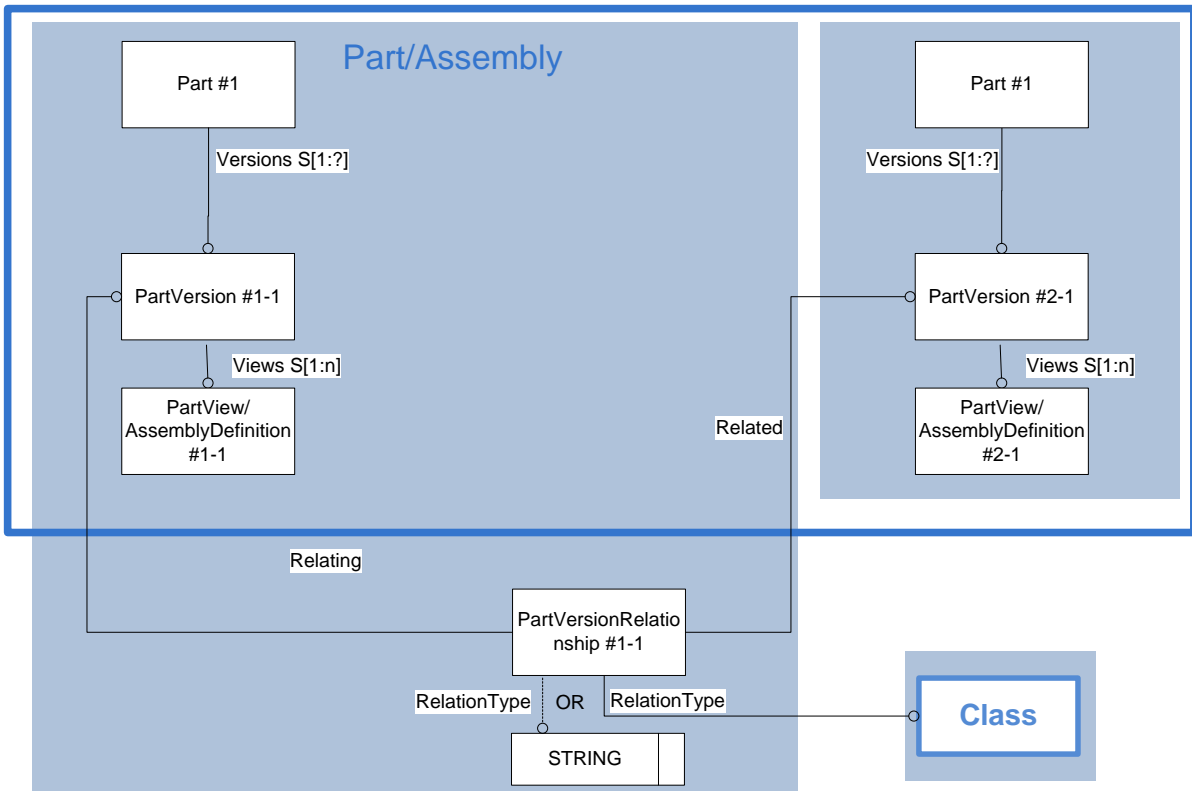


Figure 16: Template "PartVersionRelationship"

## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<Part uid="p--0000000017D374A0">
  <Id>
    <Identifier uid="pid--0000000017D374A0--id1" id="as1" idContextRef="o--000000178"/>
  </Id>
  <Name>
    <CharacterString>as1</CharacterString>
  </Name>
  ...
  <Versions>
    <PartVersion uid="pv--0000000017D374A0--id1">
      ...
      <Id id="A.1"/>
      ...
      <PartVersionRelationship uid="pvr--1">
        <Related uidRef="pv--0000000017D374A0--id2"/>
        <RelationType>
          <ClassString>sequence</ClassString>
        </RelationType>
      </PartVersionRelationship>
    </PartVersion>
    <PartVersion uid="pv--0000000017D374A0--id2">
      ...
      <Id id="A.2"/>
      ...
    </PartVersion>
  </Versions>
</Part>
```

Entity PartVersionRelationship attributes	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
Related	DocumentDefinition
RelationType	ClassSelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
CertificationAssignment	OPTIONAL SET[1:?] of CertificationAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DatetimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment

Entity PartVersionRelationship attributes	Attribute type
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 22: "PartVersionRelationship" Attributes

### Attribute recommendations

- **RelationType:** the meaning of the relationship. Use ClassString type if one of the values below is used, otherwise use "Class" template (see 4.6.4). According to the ISO AP242 Specification, where applicable, the following values shall be used:

RelationType	
'derivation'	The business object defines a deriving relationship where the related PartVersion is based on the relating PartVersion which is an earlier version of the same or of a different Part
'hierarchy'	The business object defines a hierarchical relationship where the related PartVersion is a subordinate version of the relating PartVersion
'sequence'	The business object defines a version sequence where the relating PartVersion is the preceding version of the related PartVersion that is the following version. For a given PartVersion there shall be at most one PartVersionRelationship of this relationType referring to this PartVersion as 'relating' and at most one PartVersionRelationship of this relationType referring as related
'supplied item'	The business object defines a relationship between two PartVersion objects (both exchanged in the same XML file) representing the same object in different organizational contexts, but having different contents (for example full assembly structure at the supplier and single part at the OEM)

- **Related:** the other object of [PartVersion](#) that is part of the relationship
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

### 5.1.6 Identifiers at OEM and Supplier

To map the part identifier at the OEM and at the suppliers, the Part.Id, PartVersion.Id and (if relevant) the PartView.Id shall be set to IdentifierSet:

```
<Part uid="p--0000000017D374A0">
  <Id>
    <Identifier uid="pid--0000000017D374A0--id1" id="as1" idRoleRef="r1--ii" idContextRef="o--000000178"/>
  </Id>
</Part>
```

```
<Identifier uid="pid--0000000017D374A0--id2" id="as1 bosch" idRol-
eRef="rl--ii" idContextRef="o--000000179"/>
<Identifier uid="pid--0000000017D374A0--id3" id="as1 hella" idRol-
eRef="rl--ii" idContextRef="o--000000180"/>
</Id>
...
<Versions>
  <PartVersion uid="pv--0000000017D374A0--id1">
...
  <Id>
    <Identifier uid="pid--0000000017D374A0--idv1" id="A.1" idRol-
eRef="rl--ii" idContextRef="o--000000178"/>
    <Identifier uid="pid--0000000017D374A0--idv2" id="023.8" idRol-
eRef="rl--ii" idContextRef="o--000000179"/>
    <Identifier uid="pid--0000000017D374A0--idv3" id="AB.9" idRol-
eRef="rl--ii" idContextRef="o--000000180"/>
  </Id>
...
  </PartVersion>
</Versions>
</Part>
```

The same applies to Document/DocumentVersion/DocumentDefinition and all other objects.

## 6 Part Properties

### 6.1 Template “GeometricModel”

The aim of this section is to specify the method for attaching a shape to a part and linking this shape to an external file.

The GeometricModel entity represents the shape of the Part through the PartView and the ExternalGeometricModel subtype entity allow a DigitalFile to be attached to the shape.

The different usage of the subtypes will be detailed in the chapter 6.1.1.

### The Instance Model: AP242 BO Model XML entities and attributes

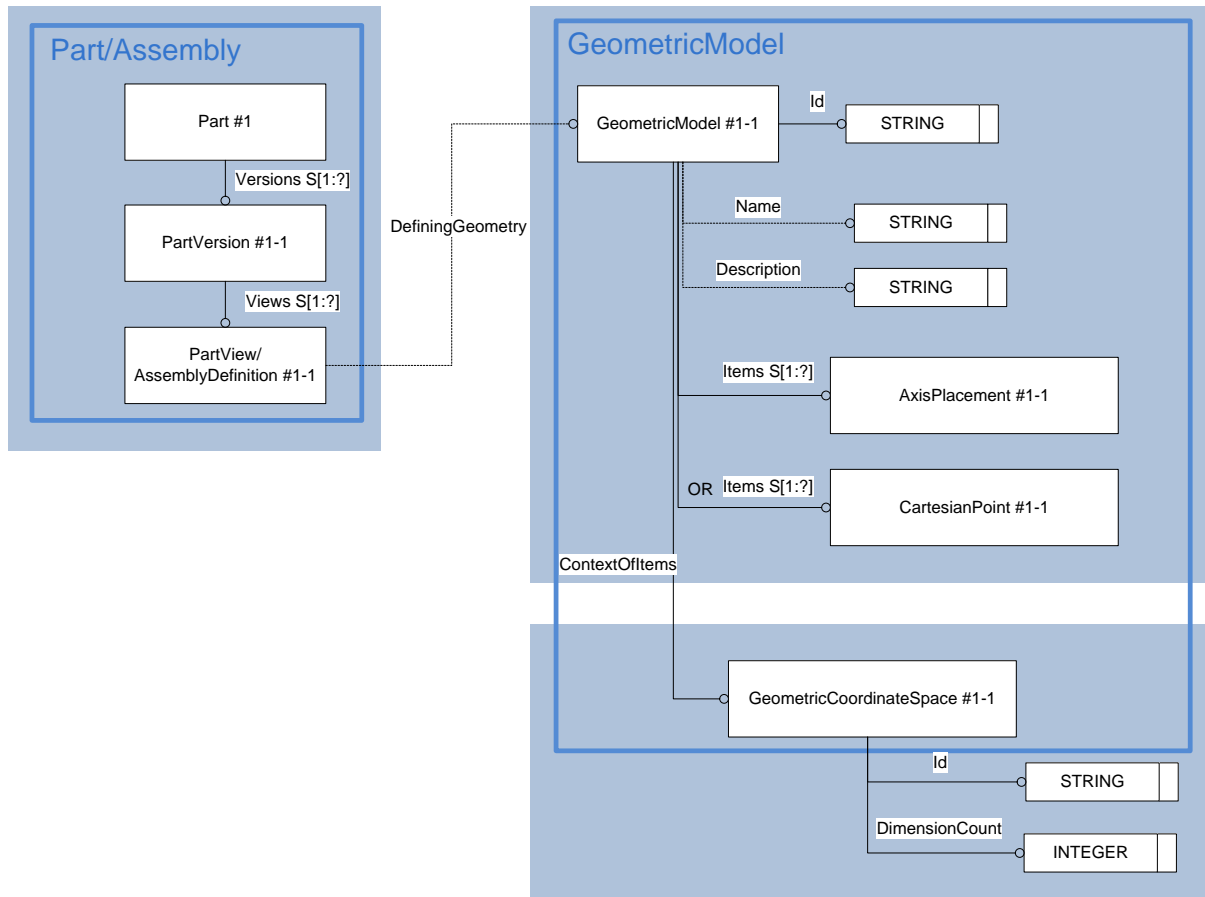


Figure 17: Template "GeometricModel"

ENTITY GeometricModel	Attribute Type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
ContextOfItems	GeometricCoordinateSpace
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
Items	SET[1:?] of RepresentationItem
Name	OPTIONAL DescriptorSelect
RepresentationTypes	OPTIONAL SET[1:?] of ClassSelect
VersionId	OPTIONAL IdentifierSelect
ModelExtent	OPTIONAL String
ModelProperty	OPTIONAL SET[1:?] of ModelProperty
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment



ENTITY GeometricModel	Attribute Type
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
GeometricRepresentationRelationship	OPTIONAL SET[1:?] of GeometricRepresentationRelationship
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 23: "GeometricModel" Attributes

**Attribute recommendations**

- The **ContextOfItems** attribute is the reference to the GeometricCoordinateSpace element.
- **Description:** the text or the set of texts that provides further information about the GeometricModel. The value of this attribute need not be specified. Use "Description" template (see 4.6.7).
- The **Id** attribute is the text that represents an identifying name or code. Use IdentifierString type if the all the usages of GeometricModel have the same context, otherwise use "Identifier" template (see 4.6.6).
- The **Items** attribute is the SET of elements representing the different kind of representation item attached to a shape. Reference to AxisPlacement (see below) or CartesianPoint element or KinematicPair element.
- **Name:** the words or set of words by which the GeometricModel is known. The value of this attribute need not be specified. Use "Description" template (see 4.6.7).
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

ENTITY GeometricCoordinateSpace	Attribute Type
Accuracies	OPTIONAL SET[1:?] of MeasureQualification
DimensionCount	Integer
Id	IdentifierSelect
Units	OPTIONAL SET[1:?] of Unit

Table 24: "GeometricCoordinateSpace" Attributes

**Attribute recommendations**

- The **DimensionCount** attribute specifies the dimensionality.
- The **Id** attribute is the text that represents an identifying name or code. Use IdentifierString type.
- **Units:** the various units in which any values are expressed. The same length unit is applied to each coordinate direction. Only one unit of a kind shall be specified. The

value of this attribute need not be specified except if the length unit deviates from the ExchangeContext.DefaultUnit.

- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Remark:** the recommendation of the AP242 ARM specification: “In the case where geometric elements are defined in the GeometricCoordinateSpace, there shall be at least two units specified, the length unit and the plane angle unit” doesn’t apply to the BO Model, since geometric elements are not covered by the BO Model.

**Preprocessor Recommendations:**

- The length unit used in the positioning (implicit/explicit transformation) and in the ExternalGeometricModel shall be the same.
- The use of GeometricCoordinateSpace.Units is only necessary if no default length unit is given in the ExchangeContext or if the length unit of a particular assembly node deviates from the default length unit (so-called multi unit assembly).
- If elements with different units are required, they have to be separated into different models with their own GeometricCoordinateSpace

**Postprocessor Recommendations:**

- The default length unit defined in ExchangeContext applies to all positioning information as well as within the ExternalGeometricModels, except if on the level of an assembly node another length unit is specified in ExternalGeometricModel.Units.

ENTITY AxisPlacement	Attribute Type
External	OPTIONAL ExternalItem
Name	OPTIONAL DescriptorSelect
Axis	OPTIONAL String
Position	String
RefDirection	OPTIONAL String

Table 25: "AxisPlacement" Attributes

**Attribute recommendations**

- **Name:** the words or set of words by which the AxisPlacement is known. The value of this attribute need not be specified. Use of CharacterString element.
- **Axis:** the relative x, y and z value specifying the direction of the local Z axis of the AxisPlacement. The value of this attribute need not be specified.
- **Position:** the absolute x, y and z value (in the GeometricCoordinateSpace they are defined in) specifying the origin position of the AxisPlacement.

Remark: The combination of two AxisPlacements in a GeometricRepresentationRelationshipWithPlacementTransformation provides a relative placement.

- **RefDirection:** the relative x, y and z value specifying the reference direction for the local X axis of the AxisPlacement. The value of this attribute need not be specified.
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:**

- The GeometricCoordinateSpace.DimensionCount must be greater than 0.
- If a RefDirection is given, it shall be specified so that it is orthogonal to the Axis.
- In case of relative positioning, each Geometry shall be associated to its own GeometricCoordinateSpace.

**Postprocessor Recommendations:** None specified.

**Related Entities:** There are no specific related entities.

ENTITY CartesianPoint	Attribute Type
External	OPTIONAL ExternalItem
Name	OPTIONAL DescriptorSelect
Coordinates	LIST [2:3] of LengthMeasure

Table 26: "CartesianPoint" Attributes

**Attribute recommendations**

- **Name:** the words or set of words by which the CartesianPoint is known. The value of this attribute need not be specified. Use of CharacterString element.
- **Coordinates:** is a list, the individual elements of this list are defined below:
  - coordinates[1]: The first coordinate of the CartesianPoint location.
  - coordinates[2]: The second coordinate of the CartesianPoint location.
  - coordinates[3]: The third coordinate of the CartesianPoint location.
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:**

- The third coordinate will not exist in the case of a two-dimensional point.
- The CartesianPoint is defined by its coordinates in a rectangular Cartesian coordinate system.

**Postprocessor Recommendations:** None specified.

**Related Entities:** There are no specific related entities.

**The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)**

```

...
  <PartView uid="pvv--000000001AA415B0--id6">
    <DefiningGeometry uidRef="egm--000000001AA415B0"/>
    ...
  </PartView>
...
<GeometricRepresentation uid="egm--000000001AA415B0"
xsi:type="n0:ExternalGeometricModel">
  <ContextOfItems uidRef="ccs--origin"/>
  <Id id="bolt.stp"/>
  <Items>
    <RepresentationItem xsi:type="n0:AxisPlacement" uid="repi--
000000001AA415B0--18">
      <Position>0.0 0.0 0.0</Position>
    </RepresentationItem>
  </Items>

```

```

</Items>
  <ExternalFile uidRef="df--000000001AA415B0" />
</GeometricRepresentation>
<GeometricCoordinateSpace uid="ccs--origin">
  <DimensionCount>3</DimensionCount>
  <Id id="/NULL"/>
</GeometricCoordinateSpace>
  
```

### 6.1.1 ExternalGeometricModel / ComposedGeometricModel

The ExternalGeometricModel / ComposedGeometricModel entities are subtype of the GeometricModel.

Entity ComposedGeometricModel / ExternalGeometricModel	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of ClassificationSelect
ContextOfItems	GeometricCoordinateSpace
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
Items	SET[1:?] of AxisPlacement
Name	OPTIONAL DescriptorSelect
RepresentationTypes	OPTIONAL SET[1:?] of ClassSelect
VersionId	OPTIONAL IdentifierSelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
GeometricRepresentationRelationship	OPTIONAL SET[1:?] of GeometricRepresentationRelationship
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
ModelExtent	OPTIONAL STRING
ModelProperty	OPTIONAL SET[1:?] of ModelProperty
(only for ExternalGeometricModel) ExternalFile	DigitalFile
(only for ExternalGeometricModel) Parameter-Values	OPTIONAL SET[1:?] OF PropertyValue

Table 27: "ComposedGeometricModel" / "ExternalGeometricModel" Attributes

### Attribute recommendations

- **GeometricRepresentationRelationship:** add for each element stored in 'Items' an instance of GeometricRepresentationRelationship to the ComposedGeometricModel. This attribute shall not be used in the case of an ExternalGeometricModel.
- **ExternalFile:** the DigitalFile that contains the externally defined geometry information. Use "DigitalFile" template (see 9.1)
- The other attributes are either not covered by these Recommended Practices, or it is not recommended to use them for the purpose of these Recommended Practices.
- In addition, all attributes and attribute recommendations for GeometricModel apply.

The attribute ExternalGeometricModel.ExternalModel is described in the chapter 8.2.

**Preprocessor Recommendations:** None specified.

**Postprocessor Recommendations:** None specified.

**Related Entities:** There are no specific related entities.

## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
...
<GeometricRepresentation uid="egm--000000001AA415B0"
xsi:type="n0:ExternalGeometricModel">
  <ContextOfItems uidRef="ccs--origin-2"/>
  <Id id="bolt.stp"/>
  <Items>
    <RepresentationItem xsi:type="n0:AxisPlacement" uid="repi--
000000001AA415B0--18">
      <Position>0.0 0.0 0.0</Position>
    </RepresentationItem>
  </Items>
</GeometricRepresentation>
...
```

### 6.1.2 The subtype of ExternalGeometricModel

The BO Model allows further specification of which type of geometry is contained in an external model. For this purpose, a number of subtypes of ExternalGeometricModel are defined. These subtypes do not add any additional attributes; they carry the additional information in their name. The subtypes of ExternalGeometricModel are:

- ExternalAdvancedBrepShapeRepresentation,
- ExternalCsgShapeRepresentation,
- ExternalCurveSweptSolidShapeRepresentation,
- ExternalEdgeBasedWireframeShapeRepresentation,
- ExternalElementaryBrepShapeRepresentation,
- ExternalFacetedBrepShapeRepresentation,
- ExternalGeometricallyBoundedSurfaceShapeRepresentation,
- ExternalGeometricallyBoundedWireframeShapeRepresentation,

- ExternalManifoldSurfaceShapeRepresentation,
- ExternalShellBasedWireframeShapeRepresentation,
- ExternalTessellatedShapeRepresentation

are optional (since ExternalGeometricModel is not defined as ABSTRACT) and mutually exclusive (ONEOF)

**Preprocessor Recommendations:**

- If a geometry file contains exact BREP geometry combined with tessellated geometry), ExternalAdvancedBrepShapeRepresentation shall be used

**Postprocessor Recommendations:**

- If some of the subtypes are not supported by the converter, the general behavior shall be 'only for information', and shall not cause the postprocessor to stop processing. The postprocessor shall load and import the file correctly.

**6.2 Template "PropertyAssignment"**

In the same way than in section 3.1 of the PDM Schema Usage Guide V4.3, the aim of this section is to specify how to attach a property to a part.

The PropertyValueAssignment entity represents the attachment of the PartView to the value represented via the "NumericalValue" (see 4.6.9) or "StringValue" templates (see 4.6.10).

**Preprocessor Recommendations:** It is recommended that all the part properties use the same PropertyValueAssignment.

**The Instance Model: AP242 BO Model XML entities and attributes**

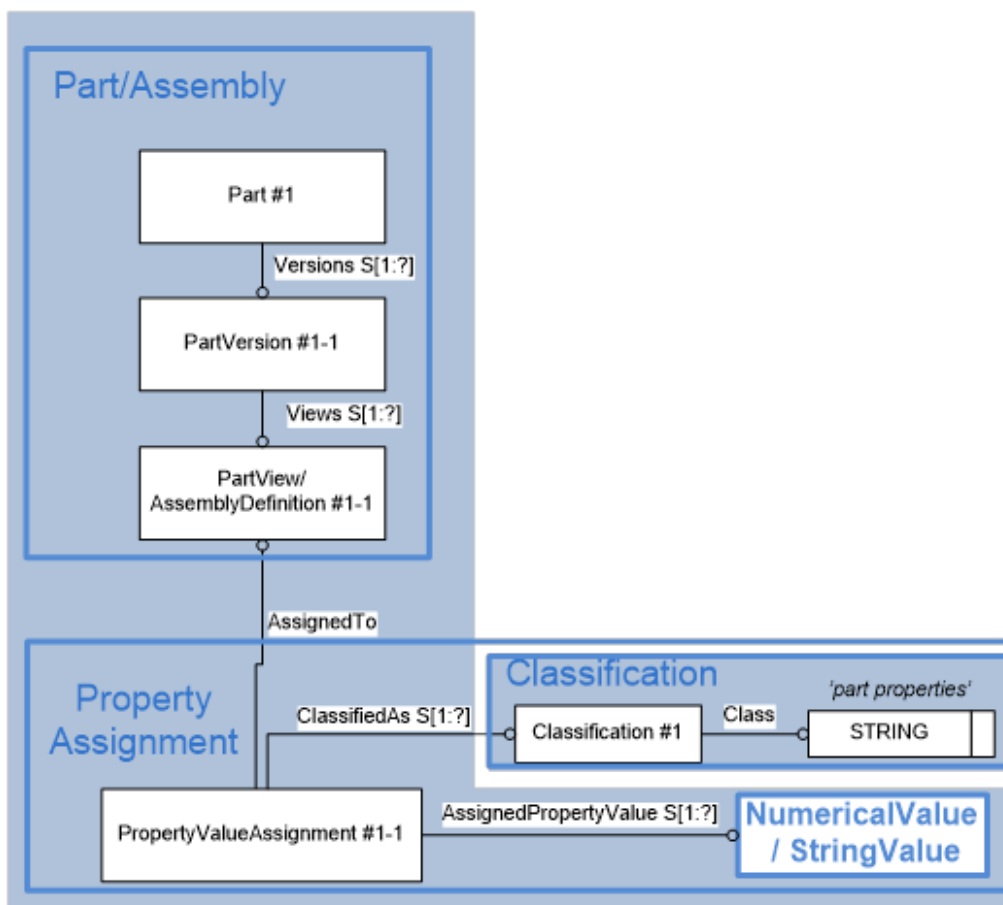


Figure 18: Template "PropertyAssignment"

ENTITY PropertyValueAssignment	Attribute Type
AssignedPropertyValues	SET[1:?] of PropertyValue
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
Role	OPTIONAL ClassSelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
AssignmentObjectRelationship	OPTIONAL SET[1:?] of AssignmentObjectRelationship
ConditionAssignment	OPTIONAL SET[1:?] of ConditionAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
PropertyValueAssignmentRelationship	OPTIONAL SET[1:?] of PropertyValueAssignmentRelationship
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
ValueAssignmentContext	OPTIONAL PropertyDefinitionAssignment

Table 28: "PropertyAssignment" Attributes

**Attribute recommendations**

- The **AssignedPropertyValues** attribute is the SET of element references representing the properties attached to the Part. Use "NumericalValue" (see 4.6.9) or "StringValue" templates (see 4.6.10).
- The ClassifiedAs attribute shall be represented by the template "Classification" (see 4.6.5). Use the following values:

ClassString	Explanation
'validation properties'	the AssignedPropertyValues are of type validation properties associated to a PartView according to section 13
'part master properties'	the AssignedPropertyValues are of type part properties associated to a Part according to section 5.1.1

'part properties'	the AssignedPropertyValues are of type part properties associated to a PartView, PartViewRelationship, or NextAssemblyOccurrenceUsage according to the current section, or GeneralShapeDependentProperty according to section 6.3
'document master properties'	the AssignedPropertyValues are of type document properties associated to a DocumentDefinition according to section 10.5. An issue has been created in Bugzilla under #6020 in order to enable Document (like Part) to have PropertyValueAssignments.
'document properties'	the AssignedPropertyValues are of type document properties associated to a DocumentDefinition or DigitalFile according to section 10.5

- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:** It is recommended that all the properties attached to a Part (i.e. its PartView or AssemblyDefinition) are spread over two instances of PropertyValueAssignment. One instance shall collect the properties that describe the Part; the other instance shall collect the properties that describe the validation properties of the same Part.

**Postprocessor Recommendations:** None specified.

**Related Entities:** There are no specific related entities.

### The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```

<Classification uid="cla--pp">
  <Class>
    <ClassString>part properties</ClassString>
  </Class>
</Classification>
<Part uid="p--0000000017086CB0">
  <Id>
    <Identifier uid="pid--0000000017086CB0--id1" id="as1" idContextRef="o--000000178"/>
  </Id>
  <Name>
    <CharacterString>as1</CharacterString>
  </Name>
  <PartTypes>
    <ClassString>assembly</ClassString>
  </PartTypes>
  <Versions>
    <PartVersion uid="pv--0000000017086CB0--id1">
      <Id id="/NULL"/>
      <Views>
        <PartView xsi:type="n0:AssemblyDefinition"
uid="pvv--0000000017086CB0--id1">
          <InitialContext uidRef="ac--mechanicaldesign--
design"/>
          <PropertyValueAssignment uid="pva--
0000000017086CB0--id1">
            <AssignedPropertyValues>
              <PropertyValue uid="pv--
0000000011E581B0--CATProdProp--cf--id1--PartNumber"
xsi:type="n0:StringValue">
                <Definition>

```



```

uidRef="pd--000000320" />
<Name>
<CharacterString>PartNumber</CharacterString>
</Name>
<ValueComponent>
<CharacterString>RollCurve_with_kin</CharacterString>
</ValueComponent>
</PropertyValue>
<PropertyValue uid="pv--000000017086CB0--CATProdProp--cf--id1--Provenance"
xsi:type="n0:StringValue">
<Definition>
<PropertyDefinition
uidRef="pd--000000320" />
</Definition>
<Name>
<CharacterString>provenance</CharacterString>
</Name>
<ValueComponent>
<CharacterString>CAx-IF</CharacterString>
</ValueComponent>
</PropertyValue>
</AssignedPropertyValues>
<ClassifiedAs>
<Classification uidRef="cla--pp" />
</ClassifiedAs>
</PropertyValueAssignment>
</PartView>
</Views>
</PartVersion>
</Versions>
</Part>
<PropertyDefinition uid="pd--000000320">
<Id id="quality property" />
<PropertyType>
<ClassString>customized PDM property</ClassString>
</PropertyType>
</PropertyDefinition>
<ViewContext uid="vc--000000781">
<ApplicationDomain>
<ProxyString>mechanical design</ProxyString>
</ApplicationDomain>
<LifeCycleStage>
<ProxyString>design</ProxyString>
</LifeCycleStage>
</ViewContext>

```

### 6.3 Template “ShapeDependentProperty”

In the same way as in section 3.2 of the PDM Schema Usage Guide V4.3, the aim of this section is to specify how to attach a property to part shape.

The ShapeDependentProperty entity represents the characteristic of the shape, or of a portion of the shape of a PartView.

It is recommended to use the subtypes of ShapeDependentProperty in the following way:

- CentreOfMass for the centre of the mass of a body.
  - Note that material properties, such as density and weight, are currently not in scope of this document. Hence, the terms “center of mass”, “center of geometry” and “centroid” are used synonymously.
- MomentsOfInertia to describe the matrix of inertia of a rigid body.
  - Note: MomentsOfInertia is currently not in scope of this document.
- GeneralShapeDependentProperty to define a user-defined characteristic of an object.

For the time being, this document describes only the subtypes GeneralShapeDependentProperty (in this chapter) and CentreOfMass (in section 13.1.2).

### 6.3.1 GeneralShapeDependentProperty

#### The Instance Model: AP242 BO Model XML entities and attributes

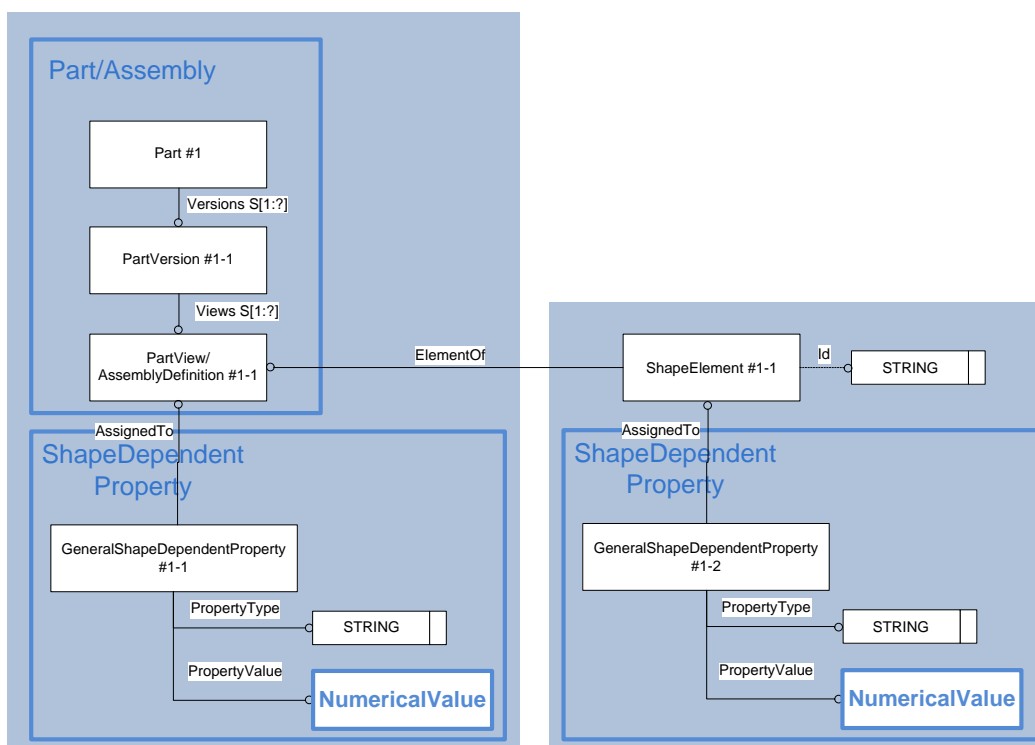


Figure 19: Template "ShapeDependentProperty" for either PartView or ShapeElement

ENTITY	Attribute Type
GeneralShapeDependentProperty	
ClassifiedAs	OPTIONAL SET[1:?] of Classification
DefinedIn	OPTIONAL GeometricCoordinateSpace
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL Id
Role	OPTIONAL ClassSelect
ValueDetermination	OPTIONAL ClassSelect
PropertyType	PropertyDefinitionSelect
PropertyValue	PropertyValueSelect

ENTITY	Attribute Type
GeneralShapeDependentProperty	
Unit	OPTIONAL Unit

Table 29: "GeneralShapeDependentProperty" Attributes

**Attribute recommendations**

- The **PropertyType** attribute is the text by which the type of the property is described. Use PropertyDefinitionString type. Where applicable, the same values shall be used as in chapter 12.2

Since the PropertyValue has also a PropertyType (PropertyValue.Definition.PropertyType), an issue in Bugzilla ([#6021](#)) has been created to clarify the added value of GeneralShapeDependentProperty.PropertyType.

- The **PropertyValue** attribute is the element reference representing the properties attached to the Part. Use "NumericalValue" templates (see 4.6.9).
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

ENTITY ShapeElement	Attribute Type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
MaterialIdentification	OPTIONAL SET[1:?] of MaterialIdentification
Name	OPTIONAL DescriptorSelect
RepresentedGeometry	OPTIONAL SET[1:?] of RepresentedGeometry
ShapeDependentProperty	OPTIONAL SET[1:?] of ShapeDependentProperty
SurfaceCondition	OPTIONAL SET[1:?] of SurfaceCondition
ThicknessSizeDimension	OPTIONAL SET[1:?] of ThicknessSizeDimension
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
ShapeElementRelationship	OPTIONAL SET[1:?] of ShapeElementRelationship

Table 30: "ShapeElement" Attributes

**Attribute recommendations**

- The **Id** attribute is the text that represents an identifying name or code for the portion of shape. It's recommended to set this attribute. Use IdentifierString type if the all the usages of ShapeElement have the same context, otherwise use "Identifier" template (see 4.6.6).
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:** When applicable, the following values shall be used for the attribute PropertyType:

- 'centroid': The PropertyValue identifies the centre point of the mass of a body. The relative position of this point within the body is an invariant datum relative to rotation and translation.
  - Example: GeneralShapeDependentProperty with propertyType 'centroid' is used instead of the object CentreOfMass: sometimes the GeneralShapeDependentProperty is calculated by some system at some point of time. But there are cases where the information, e.g., the GeneralShapeDependentProperty of a die, is transferred into a following stage, e.g., in order to perform a feasibility check, where there is only a simplified shape representation that does not allow the calculation of the exact GeneralShapeDependentProperty. Yet, in this stage the information about the GeneralShapeDependentProperty is needed in order to check whether the transport of a die with a crane is feasible.
- 'surface area': The PropertyValue specifies the overall surface of the bodies contained in the referenced shape.
- 'volume': The PropertyValue specifies the overall volume of the bodies contained in the referenced shape.

**Postprocessor Recommendations:** None specified.

**Related Entities:** There are no specific related entities.

### The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<Part uid="p--0000000017086CB0">
  <Id>
    <Identifier uid="pid--0000000017086CB0--id1" id="as1" idContextRef="o--000000178"/>
  </Id>
  <Name>
    <CharacterString>as1</CharacterString>
  </Name>
  <PartTypes>
    <ClassString>assembly</ClassString>
  </PartTypes>
  <Versions>
    <PartVersion uid="pv--0000000017086CB0--id1">
      <Id id="/NULL"/>
      <Views>
        <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--0000000017086CB0--id1">
          <InitialContext uidRef="ac--mechanicaldesign--design"/>
          <ShapeDependentProperty uid="sdp--000000782"
xsi:type="n1:GeneralShapeDependentProperty">
            <PropertyType>
              <PropertyDefinitionString>user defined attribute</PropertyDefinitionString>
            </PropertyType>
            <PropertyValue>
              <NumericalValue>
                <Definition>
                  <PropertyDefinition uidRef="pd--000000320"/>
                </Definition>
                <Name>
                  <CharacterString>volume</CharacterString>
                </Name>
                <Unit uidRef="u--00000000"></Unit>
                <ValueComponent>120</ValueComponent>
              </NumericalValue>
            </PropertyValue>
          </ShapeDependentProperty>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

```
        </PropertyValue>
        </ShapeDependentProperty>
        <ShapeElement uidRef="se--1"/>
    </PartView>
</Views>
</PartVersion>
</Versions>
</Part>
<ShapeElement uid="se--1">
    <Id>
        <Identifier uid="se--1--id1" id="/NULL" idContextRef="o--000000178"/>
    </Id>
    <ShapeDependentProperty uid="sdp--100000782"
xsi:type="n0:GeneralShapeDependentProperty">
        <PropertyType>
            <PropertyDefinitionString>user defined attrib-
ute</PropertyDefinitionString>
        </PropertyType>
        <PropertyValue>
            <NumericalValue>
                <Definition>
                    <PropertyDefinition uidRef="pd--000000320"/>
                </Definition>
                <Name>
                    <CharacterString>volume</CharacterString>
                </Name>
                <Unit uidRef="u--000000004"/>
                <ValueComponent>120</ValueComponent>
            </NumericalValue>
        </PropertyValue>
    </ShapeDependentProperty>
</ShapeElement>

<Unit uid="u--000000009">
    <Kind><ClassString>Unspecified SI derived unit</ClassString></Kind>
    <Name><ClassString>cubic metre</ClassString></Name>
    <Prefix><ClassString>milli</ClassString></Prefix>
    <Quantity><ClassString>volume</ClassString></Quantity>
</Unit>

<PropertyDefinition uid="pd--000000320">
    <Id id="quality property"/>
    <PropertyType>
        <ClassString>user defined attribute</ClassString>
    </PropertyType>
</PropertyDefinition>
```

## 7 Part Structure and Relationships

The aim of this section is to map a multi-level assembly, possibly containing multiple individual occurrences of the same component, and to position (orientation and location) each occurrence in 3D relatively to its usage in the next higher assembly. It does this in the same way that section 4.2 of the PDM Schema Usage Guide V4.3 accomplishes it.

For this reason, the use of PartViewRelationship is not recommended, since (according to AP242-ISO document in chapter 4.2.3 Assembly structure):

“In the part view based assembly structure concept, a specific part occurrence can be identified by a single PartViewRelationship object, i.e. part occurrences can exist only in the context of an assembly structure, whereas in the part occurrences based assembly structure

concept, part occurrences can exist independent of an assembly that uses the product occurrences as constituents.”

**Note** that occurrences are usually not created manually by themselves. A user creates a link (usage) between two part views or assembly definitions; the occurrence gets created automatically in the process. Occurrences not used by any other element shall not be exchanged.

From the four possible kinds of occurrences in AP242 BO Model (derived from the abstract supertype ‘Occurrence’), only these two are in scope of this document:

- SingleOccurrence: has no owned attributes, but enables to position (orientation and location) each occurrence in 3D
- SpecifiedOccurrence: enable to distinct between multiple individual occurrences of the same component. This may i.e. be used to map kinematic constraints or instance styling (which are both not yet in scope of this document).

The further two subtypes of Occurrence (QuantifiedOccurrence and SelectedOccurrence) are used in the area of BoM systems (not yet in scope of this document).

### 7.1 Template “SingleOccurrence”

This is the normal case, where the usages of a component are only documented within their next higher assembly.

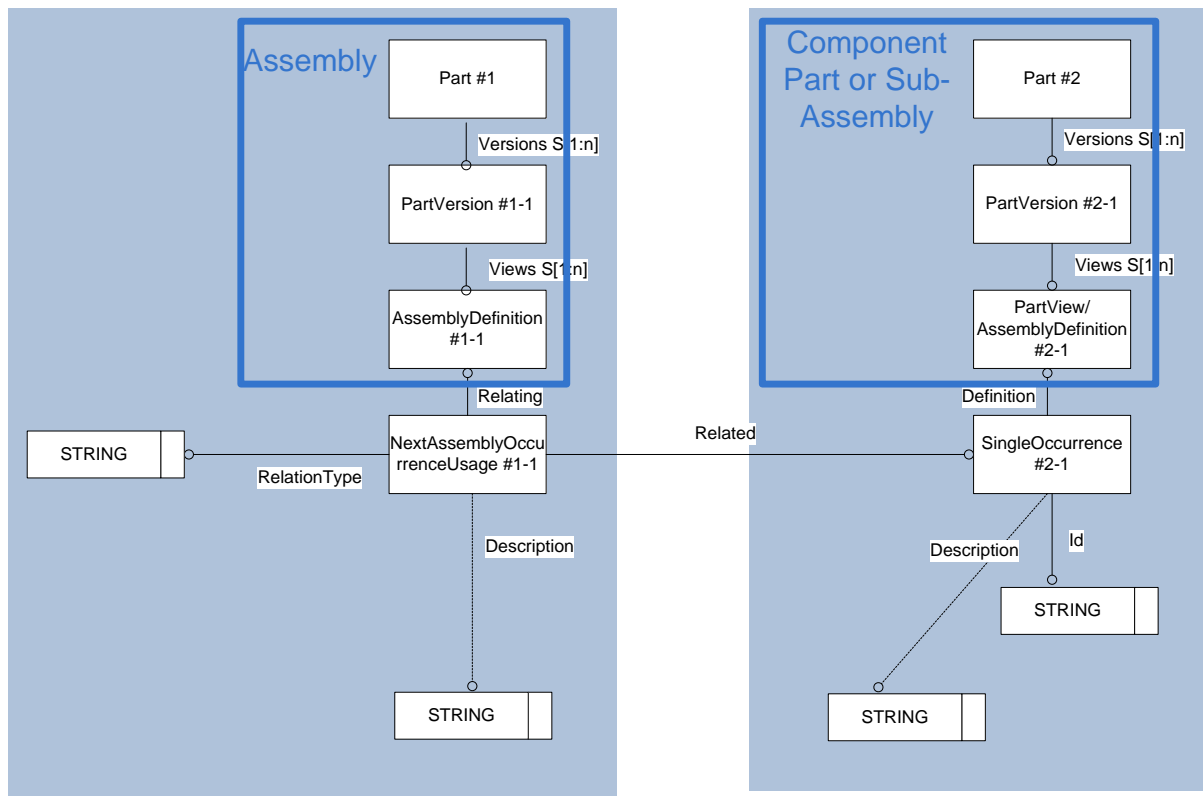


Figure 20: Template "SingleOccurrence"

### The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<Part uid="p--0000000017086CB0" >
  <Id>
```

```
<Identifier uid="pid--0000000017086CB0--id1" id="as1" idContextRef="o--
000000178"/>
</Id>
...
<Versions>
  <PartVersion uid="pv--0000000017086CB0--id1">
    ...
    <Views>
      <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--
0000000017086CB0--id1">
        ...
        <ViewOccurrenceRelationship uid="pvvid--000000001E5A89F0--10"
xsi:type="n0:NextAssemblyOccurrenceUsage">
          <Related uidRef="pi--000000001E5A89F0--10"/>
          <RelationType>
            <ClassString>next assembly occurrence</ClassString>
          </RelationType>
        ...
      </ViewOccurrenceRelationship>
    ...
  </PartView>
</Views>
</PartVersion>
</Versions>
</Part>
<Part uid="p--000000001E5A89F0">
  <Id>
    <Identifier uid="pid--000000001E5A89F0--id2" id="plate" idContex-
tRef="o--000000178"/>
  </Id>
  ...
  <Versions>
    <PartVersion uid="pv--000000001E5A89F0--id2">
      ...
      <Views>
        <PartView uid="pvv--000000001E5A89F0--id2">
          ...
          <Occurrence xsi:type="n0:SingleOccurrence" uid="pi--
000000001E5A89F0--10">
            <Id id="plate.1"/>
          </Occurrence>
        ...
      </PartView>
    </Views>
  </PartVersion>
</Versions>
</Part>
```

### **Preprocessor Recommendations:**

The usage of SingleOccurrence is necessary to position each occurrence in 3D.

QuantifiedOccurrence shall not be used in a CAx context. Where multiple occurrences are needed, use multiple instances of SingleOccurrence.

It is not recommended to use the supertypes of NextAssemblyOccurrenceUsage like ViewOccurrenceRelationship or AssemblyOccurrenceRelationship.

The Occurrence referenced by "Related" shall not belong to the part where the NextAssemblyOccurrenceUsage is defined, nor to any assembly that builds this part (so-called cycle in the product structure).

It is not recommended to instantiate SingleOccurrence without this Occurrence being referenced by a NextAssemblyOccurrenceUsage.

**Postprocessor Recommendations:**

If a SingleOccurrence is encountered which is not referenced by any NextAssemblyOccurrenceUsage, it shall be ignored.

Entity NextAssemblyOccurrenceUsage	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of ClassificationSelect
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
MaterialIdentification	OPTIONAL SET[1:?] of MaterialIdentification
Related	Occurrence
RelationType	ClassSelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PartRelationship	OPTIONAL SET[1:?] of PartRelationship
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
Placement	OPTIONAL TransformationSelect
AssemblyOccurrenceRelationshipSubstitution	OPTIONAL SET[1:?] of AssemblyOccurrenceRelationshipSubstitution
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment

Table 31: "NextAssemblyOccurrenceUsage" Attributes

**Attribute recommendations**



- **Description:** the text or the set of texts that provide further information about the Assembly structure link. The value of this attribute need not be specified. Use "Description" template.
- **Related:** Reference to a SingleOccurrence of the component part built into the assembly part.
- **RelationType:** the meaning of the relationship. Use ClassString type. Mandatory value: 'next assembly occurrence'.
- **PropertyValueAssignment:** adds the value of a property to the NextAssemblyOccurrenceUsage. Use "PropertyAssignment" template (see 6.2)
- **Placement:** specifies the transformation information which is used to locate and orient the constituent in the coordinate space of the AssemblyDefinition. Placement is a reference to a CartesianTransformation, GeometricRepresentationRelationshipWithPlacementTransformation, GeometricRepresentationRelationshipWithCartesianTransformation or GeometricRepresentationRelationshipWithSameCoordinateSpace. For more details, refer to 7.3.
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

Entity SingleOccurrence	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of ClassificationSelect
DefiningGeometry	OPTIONAL GeometricModel
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
MaterialIdentification	OPTIONAL SET[1:?] of MaterialIdentification
ShapeDependentProperty	OPTIONAL SET[1:?] of ShapeDependentProperty
ShapeElement	OPTIONAL SET[1:?] of ShapeElement
SurfaceCondition	OPTIONAL SET[1:?] of SurfaceCondition
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
BreakdownVersionAssignment	OPTIONAL SET[1:?] of BreakdownVersionAssignment
CertificationAssignment	OPTIONAL SET[1:?] of CertificationAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment

Entity SingleOccurrence	Attribute type
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
ProjectAssignment	OPTIONAL SET[1:?] of ProjectAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 32: "SingleOccurrence" Attributes

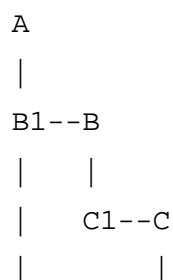
### Attribute recommendations

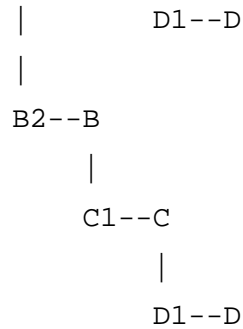
- **DefiningGeometry:** the **GeometricModel** that contains the shape information. The value of this attribute shall not be specified, except in case of a flexible part where the geometry of the part deviates from the one defined on PartView.DefiningGeometry level for this particular occurrence.
- **Description:** the text or the set of texts that provides further information about the SingleOccurrence. The value of this attribute need not be specified. Use "Description" template.
- **Id:** stores the Identifier for the SingleOccurrence (shall be unique over all Occurrences directly referenced via NextAssemblyOccurrenceUsage by an assembly part). Use IdentifierString type.
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

## 7.2 Template "SpecifiedOccurrence"

In order to distinguish a specific occurrence of a component in an assembly of more than two hierarchical levels, the SpecifiedOccurrence entity is used additionally to the SingleOccurrence mentioned above.

For example, a wheel-axle assembly (A) contains two wheels (B), the right (B1) and the left (B2). The wheel again contains a sub-assembly (C), which again contains a component part (D). The requirement to individually identify the left D, for example, is supported by this capability.





**Note:** using the current BO Model XSD, the instantiation of SpecifiedOccurrence is impossible (missing SpecifiedOccurrence.UpperAssemblyRelationship and not expected Occurrence.SpecifiedOccurrence): a proposal has been submitted to the ISO (Bugzilla issue see #5078 in Annex B). This will be document in version 1.2 of this document, based on the upcoming Technical Corrigendum of the AP242 BO Model..

### 7.3 Full / Simplified Positioning Representation

The scope of this section corresponds to section 3.4 of the PDM Schema Usage Guide V4.3

In addition to the usual (geometrical) way of mapping the 3D positioning within an assembly structure, a more compact (especially in XML) and simple way has been defined in AP242. This section describes all possible ways.

#### **Preprocessor Recommendations:**

As long as no critical precision issues occur (especially caused by the multiplication of several relative positioning matrices containing large numbers in deep assembly structures), either relative or absolute 3D positioning can be used.

Since large numbers cause computers to truncate decimal digits, the depth of such assembly structure shall be limited in the case of relative positioning, or absolute positioning should be used. In the latter case, all intermediate assembly nodes between the top node and the assemblies/components associated to an absolute positioning should be positioned via an identity matrix, or share the same coordinate space.

Absolute positioning of multiple usages of components requires the use of SpecifiedOccurrence (see previous section) and is therefore not recommended.

The use of mirroring in the 3D transformation from a component part in an assembly part is not allowed.

The vectors within the RotationMatrix of a CartesianTransformation and the Axis/RefDirection of an AxisPlacement shall be orthogonal to each other.

**Note:** Transformation matrices exchanged using this BO model are not guaranteed to be orthogonal, since compared to the definition of axis2\_placement\_3d in Part 42, the definition of AxisPlacement in the BO Model is missing one step in the calculation intended to ensure the orthogonality of axis and refDirection. Hence, special attention is needed to define them as orthogonal from the beginning. Otherwise, in the case of multi-level assembly the concatenation of the transformation matrices may result in inconsistent or incorrect results between exchange partners

The simplified positioning representation is recommended whenever the assembly nodes have no geometry. It is a shortcut to avoid to instantiate GeometricModel if there is none. If there is a GeometricModel, it has to be referenced by the GeometricRepresentationRelationship and to point to a GeometricCoordinateSpace => the simplified positioning representation is not usable.

The implicit or explicit representation is recommended whenever the assembly nodes have geometry, since from a pure CAD point of view, each Geometry has his own Geometric-CoordinateSpace.

This recommendation applies independently from the fact that nested or monolithic mapping is used (see section 0), since the reference to nested nodes ist not geometry but AP242 XML. The only point is: does the assembly node own a geometry file or not.

### 7.3.1 Template “Simplified Positioning Representation”

The only instance needed here is a CartesianTransformation.

#### ***Preprocessor Recommendations:***

Since this mapping does not support the explicit mapping of a unit (for the elements of the translation vector), a DefaultUnit shall be defined in ExchangeContext and all translation vectors shall be given according to this unit.

#### **The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)**

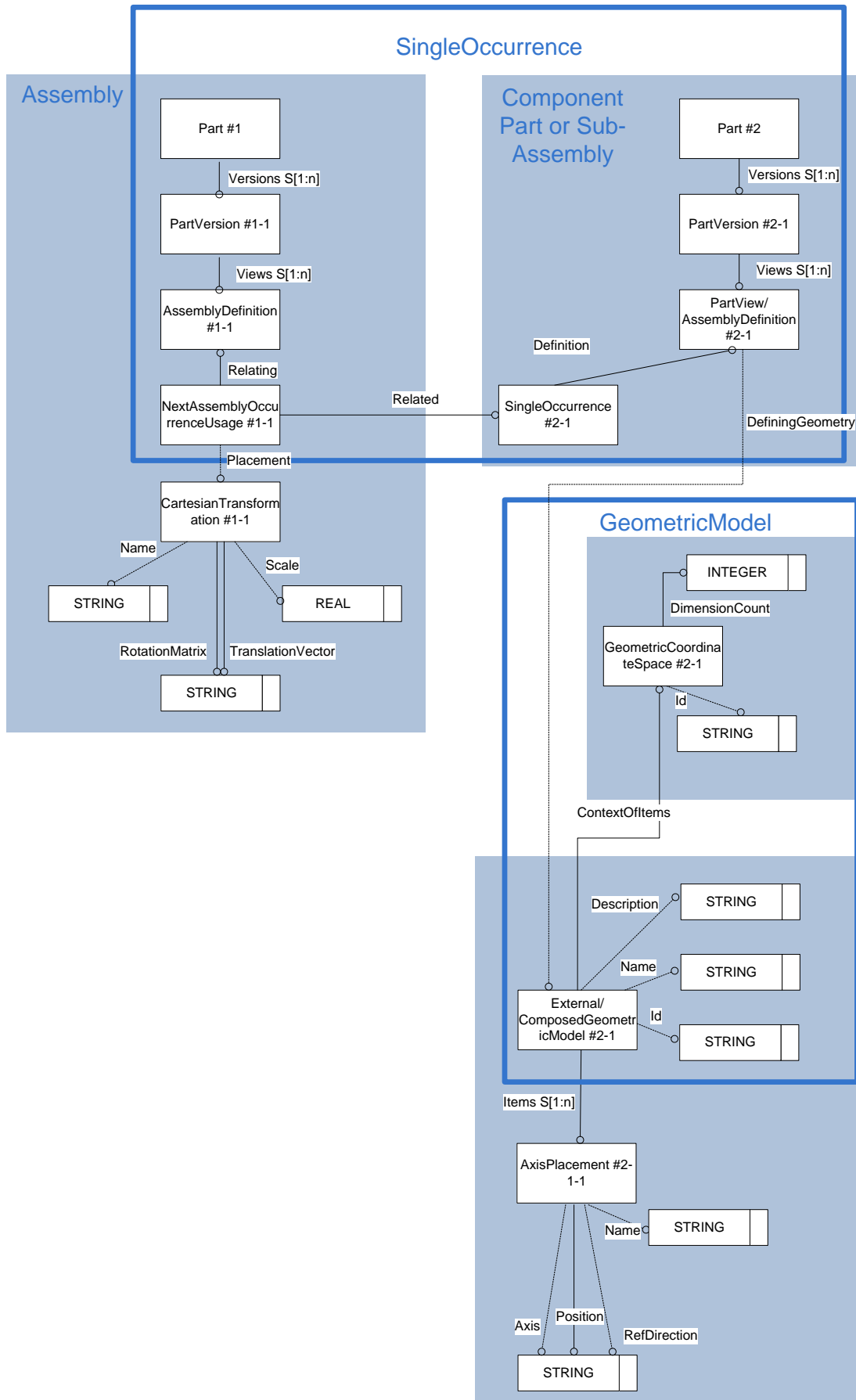


Figure 21: Template "Simplified Positioning Representation"

```

<Part uid="p--0000000017085F00">
  <Id>
    <Identifier uid="pid--0000000017085F00--id3" id="bracket_asm" idContextRef="o--000000178"/>
  </Id>
  ...
  <Versions>
    <PartVersion uid="pv--0000000017085F00--id3">
      ...
      <Views>
        <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--0000000017085F00--id3">
          ...
          <ViewOccurrenceRelationship uid="pvvid--000000001E6E14B0--14" xsi:type="n0:NextAssemblyOccurrenceUsage">
            <Related uidRef="pi--000000001E6E14B0--14"/>
            <RelationType>
              <ClassString>next assembly occurrence</ClassString>
            </RelationType>
            <Placement>
              <CartesianTransformation uid="cto--000000001E6E14B0--14">
                <RotationMatrix>2.83808309622E-16 -1.48711849984E-5
                0.999999999889 7.14623103897E-14 -0.999999999889 -1.48711849984E-5 1.0
                7.146231460233355E-14 -2.82745580352E-16</RotationMatrix>
                <TranslationVector>7451.5038 127.065 -
                443.85</TranslationVector>
              </CartesianTransformation>
            </Placement>
          </ViewOccurrenceRelationship>
          ...
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
  
```

Entity CartesianTransformation	Attribute type
External	OPTIONAL ExternalItem;
Name	OPTIONAL DescriptorSelect
RotationMatrix	String
Scale	Real
TranslationVector	String

Table 33: "CartesianTransformation" Attributes

**Attribute recommendations**

- **Name:** the words or set of words by which the CartesianTransformation is known. The value of this attribute need not be specified. Use "Description" template.
- **RotationMatrix:** 3x3 Matrix with the values: xx xy xz yx yy yz zx zy zz
  - xx xy xz represent the X axis direction of the transformation target.
  - yx yy yz represent the Y axis direction of the transformation target.
  - zx zy zz represent the Z axis direction of the transformation target.
- **Scale:** According to the AP242 specification, the scale factor shall be omitted or set to 1.0.
- **TranslationVector:** 3-dimensional vector with the following values: x y z

- No further attributes are covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

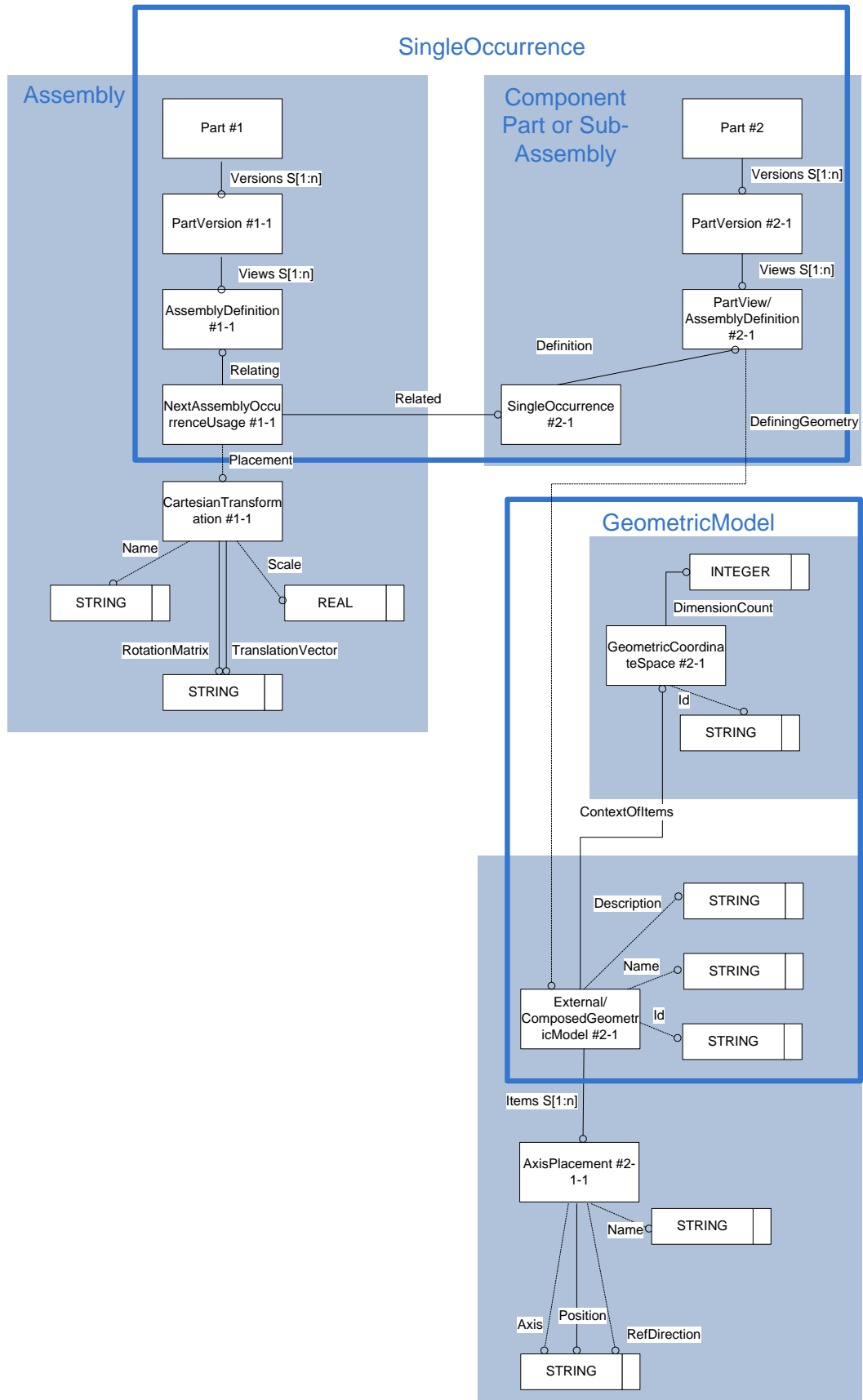


Figure 22: Instantiation example if Part #2 has geometry

### 7.3.2 Full Positioning Representations

In all these mapping alternatives, the assembly and the component node are associated to a subtype of `GeometricModel` and the 3D Positioning information is mapped in a subtype of `GeometricRepresentationRelationship` that references both `GeometricModels`.

Here some common recommendations and entities applying to all mapping alternatives:

#### ***Preprocessor Recommendations:***

The `GeometricModel` associated to the Assembly node shall be of kind `ComposedGeometricModel`, while

The `GeometricModel` associated to the Component node shall be of kind:

- `ExternalGeometricModel` if it is a simple part and has its own geometry, defined in a digital file (see section File Reference)
- `ComposedGeometricModel` if it is an assembly node

The use of `GeometricModel` itself (without a subtype) as well as the use of further subtypes of `GeometricModel` (i.e. `TransformedGeometricModel`) is not recommended.

The use of the subtypes of `ExternalGeometricModel` is optional (like in the EXPRESS schema, since `ExternalGeometricModel` is not an ABSTRACT SUPERTYPE), and shall be interpreted purely as ‘for information purpose only’ by the postprocessor. There are several reasons for this:

- Many file formats allow for the combination of different geometric representations of the same model, e.g. precise B-Rep and tessellated, or solid and surface model, within the same file.
  - the ONEOF constraint defined in the EXPRESS schema between the subtypes doesn’t apply all the time
- Some of the subtypes are not supported by the converters (for example `ExternalCurveSweptSolidShapeRepresentation`).
- Since currently most converters do not evaluate the `Creation_Property`, `Format_Property`, `File_Type_Property` during import, but rather try to load the file
  - it is likely they will also not evaluate the subtypes of `ExternalGeometricModel`
- It may be quite an effort to add this to the converters, with a rather low added value

In case of a relative 3D positioning, it is not recommended to reuse the same instance of `GeometricCoordinateSpace` for both geometric models, since each of them has its own coordinate space.

If necessary an adjustment to `refDirection` has to be made to maintain orthogonality to the axis direction. If axis or `refDirection` are omitted, these directions are taken from the geometric coordinate system.

Although `GeometricModel` is defined as XML `RootObject`, it should be always associated to one and only one `PartView` or `Occurrence` via `DefinedGeometry`.

If an `ExternalGeometricModel` is defined for an `Occurrence` (flexible part), this `ExternalGeometricModel` shall be involved in the `GeometricRepresentationRelationshipWithPlacementTransformation` or the `GeometricRepresentationRelationshipWithCartesianTransformation` that involves this `Occurrence`, and not the `ExternalGeometricModel` defined via `PartView.DefiningGeometry`.

The `GeometricModel` referenced by “Related” shall not belong to the part where the `GeometricModelRelationship` is defined, nor to any assembly that builds this part (so-called cycle in the product structure).



### ***Postprocessor Recommendations:***

To derive the Y vector from the Axis (Z) and RefDirection (X) of an AxisPlacement, please refer to the Annex D (Conversion from Implicit to Explicit Transformation Information) taken over from the PDM Usage Guide.

The subtypes of ExternalGeometricModel shall be interpreted purely as 'for information purpose only'. Do not rely on them for processing the file and do not stop processing in case the given subtype is not supported (the file shall be loaded anyway and an error produced only if it couldn't be processed).

#### **7.3.2.1 Implicit Transformation**

In this case, the GeometricRepresentationRelationship is of kind GeometricRepresentationRelationshipWithPlacementTransformation.

### ***Preprocessor Recommendations:***

- The AxisPlacement of the Component (Origin) shall contain '0 0 0' for Position and no value. for Axis and RefDirection.
- The use of further subtypes of RepresentationItem (apart of AxisPlacement) is not recommended.
- In case of relative 3D positioning, each GeometricModel should reference its own instance of GeometricCartesianSpace.

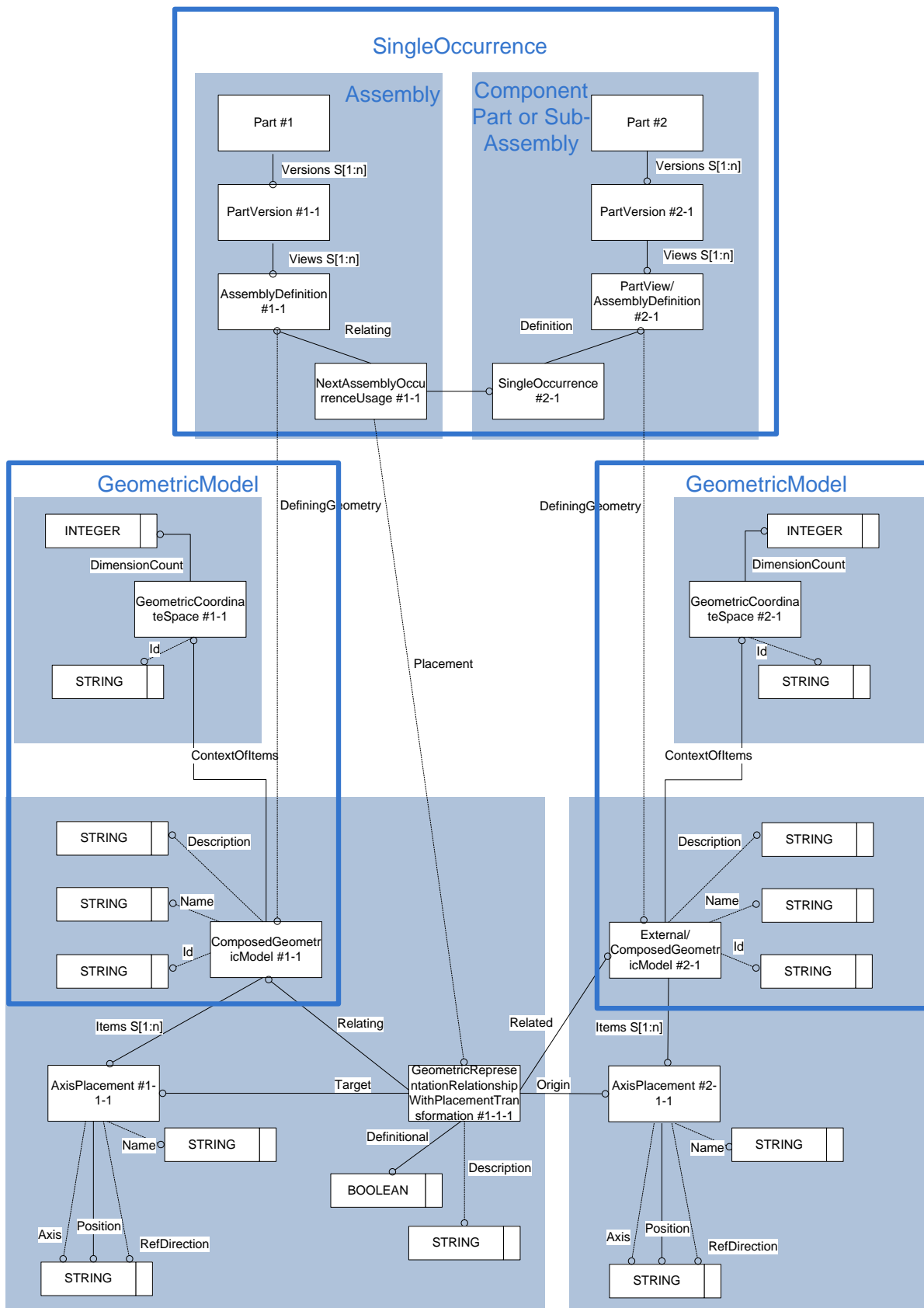


Figure 23: Full Positioning Representation with Implicit Transformation

## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<GeometricCoordinateSpace uid="ccs--origin-1">
  <DimensionCount>3</DimensionCount>
  <Id id="/NULL"/>
</GeometricCoordinateSpace>
<GeometricCoordinateSpace uid="ccs--origin-2">
  <DimensionCount>3</DimensionCount>
  <Id id="/NULL"/>
</GeometricCoordinateSpace>
<GeometricCoordinateSpace uid="ccs--origin-3">
  <DimensionCount>3</DimensionCount>
  <Id id="/NULL"/>
</GeometricCoordinateSpace>
<Part uid="p--0000000020AB6290">
  <Id>
    <Identifier uid="pid--0000000020AB6290--id5" id="nut and bolt"
idContextRef="o--000000178"/>
  </Id>
  ...
  <Versions>
    <PartVersion uid="pv--0000000020AB6290--id5">
      ...
      <Views>
        <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--
0000000020AB6290--id5">
          ...
          <ViewOccurrenceRelationship uid="pvvid--000000001AA415B0--18"
xsi:type="n0:NextAssemblyOccurrenceUsage">
            <Related uidRef="pi--000000001AA415B0--18"/>
            <RelationType>
              <ClassString>next assembly occurrence</ClassString>
            </RelationType>
            <Placement>
              <GeometricRepresentationRelationship uidRef="ctrafo--
0000000013DF75A0--18"/>
            </Placement>
          </ViewOccurrenceRelationship>
          <ViewOccurrenceRelationship uid="pvvid--000000001AA41A00--19"
xsi:type="n0:NextAssemblyOccurrenceUsage">
            <Related uidRef="pi--000000001AA41A00--19"/>
            <RelationType>
              <ClassString>next assembly occurrence</ClassString>
            </RelationType>
            <Placement>
              <GeometricRepresentationRelationship uidRef="ctrafo--
0000000013DF75A0--19"/>
            </Placement>
          </ViewOccurrenceRelationship>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

```
<GeometricRepresentation uid="egm--0000000020AB6290 "  
xsi:type="n0:ComposedGeometricModel">  
  <ContextOfItems uidRef="ccs--origin-1"/>  
  <Id id="nut and bolt"/>  
  <Items>  
    <RepresentationItem uid="repi--000000001AA41A00--18--2"  
xsi:type="n0:AxisPlacement">  
      <Axis>0.999999999889,-1.48711849984E-005,-2.82745580352E-016</Axis>  
      <Position>7451.5038,127.065,-443.85</Position>  
      <RefDirection>2.83808309622E-016,7.14623103897E-014,1.</RefDirection>  
    </RepresentationItem>  
    <RepresentationItem uid="repi--000000001AA41A00--19--2"  
xsi:type="n0:AxisPlacement">  
      <Axis>1.000000 0.000000 0.000000</Axis>  
      <Position>-33.000000 0.000000 0.000000</Position>  
      <RefDirection>0.0 0.0 0.0</RefDirection>  
    </RepresentationItem>  
  </Items>  
  <GeometricRepresentationRelationship uid="ctrafo--0000000013DF75A0--18 "  
xsi:type="n0:GeometricRepresentationRelationshipWithPlacementTransformation  
">  
    <Definitional>>true</Definitional>  
    <Related uidRef="egm--000000001AA415B0"/>  
    <Origin uidRef="repi--000000001AA415B0--18"/>  
    <Target uidRef="repi--000000001AA41A00--18--2"/>  
  </GeometricRepresentationRelationship>  
  <GeometricRepresentationRelationship uid="ctrafo--0000000013DF75A0--19 "  
xsi:type="n0:GeometricRepresentationRelationshipWithPlacementTransformation  
">  
    <Definitional>>true</Definitional>  
    <Related uidRef="egm--000000001AA41A00"/>  
    <Origin uidRef="repi--000000001AA41A00--19"/>  
    <Target uidRef="repi--000000001AA41A00--19--2"/>  
  </GeometricRepresentationRelationship>  
</GeometricRepresentation>  
<Part uid="p--000000001AA415B0">  
  <Id>  
    <Identifier uid="pid--000000001AA415B0--id6" id="bolt" idContextRef="o-  
-000000178"/>  
  </Id>  
  ...  
  <Versions>  
    <PartVersion uid="pv--000000001AA415B0--id6">  
  ...  
    <Views>  
      <PartView uid="pvv--000000001AA415B0--id6">  
        <DefiningGeometry uidRef="egm--000000001AA415B0"/>  
  ...  
        <Occurrence xsi:type="n0:SingleOccurrence" uid="pi--  
000000001AA415B0--18">  
          <Id id="bolt.1"/>  
        </Occurrence>  
  ...  
      </PartView>
```

```
        </Views>
    </PartVersion>
</Versions>
</Part>
<GeometricRepresentation uid="egm--000000001AA415B0"
xsi:type="n0:ExternalGeometricModel">
    <ContextOfItems uidRef="ccs--origin-2"/>
    <Id id="bolt.stp"/>
    <Items>
        <RepresentationItem xsi:type="n0:AxisPlacement" uid="repi--
000000001AA415B0--18">
            <Position>0.0 0.0 0.0</Position>
        </RepresentationItem>
    </Items>
...
</GeometricRepresentation>
<Part uid="p--000000001AA41A00">
    <Id>
        <Identifier uid="pid--000000001AA41A00--id7" id="nut" idContextRef="o--
000000178"/>
    </Id>
...
    <Versions>
        <PartVersion uid="pv--000000001AA41A00--id7">
...
            <Views>
                <PartView uid="pvv--000000001AA41A00--id7">
                    <DefiningGeometry uidRef="egm--000000001AA41A00"/>
...
                    <Occurrence xsi:type="n0:SingleOccurrence" uid="pi--
000000001AA41A00--19">
                        <Id id="nut.1"/>
                    </Occurrence>
...
                </PartView>
            </Views>
        </PartVersion>
    </Versions>
</Part>
<GeometricRepresentation uid="egm--000000001AA41A00"
xsi:type="n0:ExternalGeometricModel">
    <ContextOfItems uidRef="ccs--origin-3"/>
    <Id id="nut.stp"/>
    <Items>
        <RepresentationItem xsi:type="n0:AxisPlacement" uid="repi--
000000001AA41A00--19">
            <Position>0.0 0.0 0.0</Position>
        </RepresentationItem>
    </Items>
...
</GeometricRepresentation>
```

Entity GeometricRepresentationRelationshipWith-PlacementTransformation	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification-Select
Definitional	BOOLEAN
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
Related	ComposedGeometricModel or ExternalGeometricModel
Origin	AxisPlacement
Target	AxisPlacement

Table 34: "GeometricRepresentationRelationshipWithPlacementTransformation" Attributes

### Attribute recommendations

- **Definitional:** according to a WHERE rule in the EXPRESS model, shall be always TRUE (makes the related GeometricModel part of the definition of the relating GeometricModel).
- **Description:** the text or the set of texts that provides further information about the GeometricRepresentationRelationship. The value of this attribute need not be specified. Use "Description" template.
- **Related:** Reference to the ComposedGeometricModel or ExternalGeometricModel of the component part built into the assembly part
- **Origin:** Reference to the corresponding instance of AxisPlacement associated to the related ComposedGeometricModel or ExternalGeometricModel of the component part built into the assembly part. It is recommended to define an identity matrix within this AxisPlacement
- **Target:** Reference to the corresponding instance of AxisPlacement associated to the relating ComposedGeometricModel of the assembly part
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

### 7.3.2.2 Explicit Transformation

In this case, the GeometricRepresentationRelationship is of kind GeometricRepresentationRelationshipWithCartesianTransformation.

#### Preprocessor Recommendations:

The AxisPlacement of the Component (Related) shall contain '0 0 0' for Position and no value for Axis and RefDirection.

The use of further subtypes of RepresentationItem (apart of AxisPlacement for the component part and CartesianTransformation for the assembly part) is not recommended.

As specified in the EXPRESS data model via a WHERE rule, each GeometricModel shall reference its own instance of GeometricCartesianSpace.

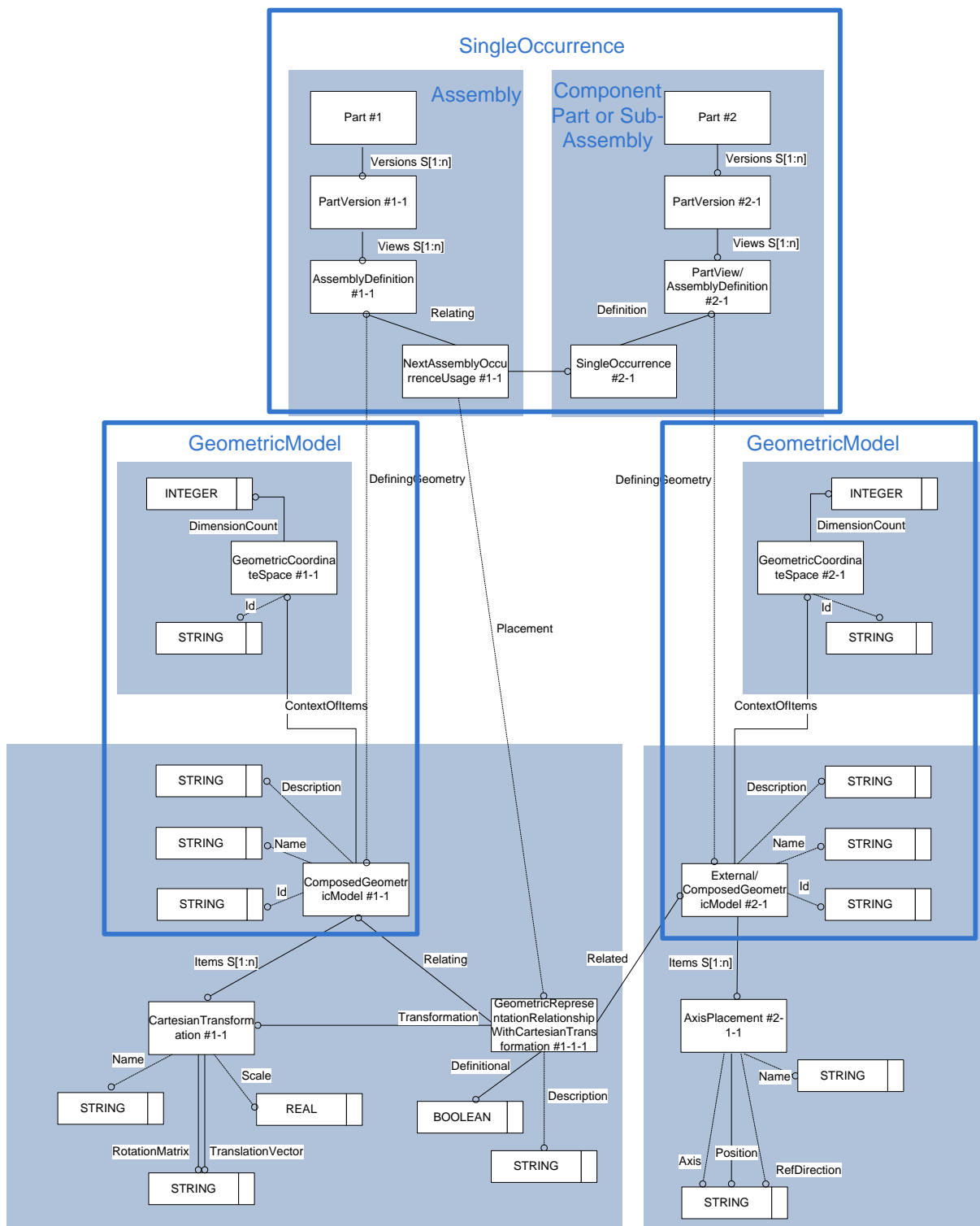


Figure 24: Full Positioning Representation with Explicit Transformation

## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<GeometricCoordinateSpace uid="ccs--origin-1">
  <DimensionCount>3</DimensionCount>
  <Id id="/NULL"/>
</GeometricCoordinateSpace>
<GeometricCoordinateSpace uid="ccs--origin-2">
  <DimensionCount>3</DimensionCount>
  <Id id="/NULL"/>
</GeometricCoordinateSpace>
<GeometricCoordinateSpace uid="ccs--origin-3">
  <DimensionCount>3</DimensionCount>
  <Id id="/NULL"/>
</GeometricCoordinateSpace>
<Part uid="p--0000000020AB6290">
  <Id>
    <Identifier uid="pid--0000000020AB6290--id5" id="nut and bolt"
idContextRef="o--000000178"/>
  </Id>
  ...
  <Versions>
    <PartVersion uid="pv--0000000020AB6290--id5">
      ...
      <Views>
        <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--
0000000020AB6290--id5">
          ...
          <ViewOccurrenceRelationship uid="pvvid--000000001AA415B0--18"
xsi:type="n0:NextAssemblyOccurrenceUsage">
            <Related uidRef="pi--000000001AA415B0--18"/>
            <RelationType>
              <ClassString>next assembly occurrence</ClassString>
            </RelationType>
            <Placement>
              <GeometricRepresentationRelationship uidRef="ctrafo--
0000000013DF75A0--18"/>
            </Placement>
          </ViewOccurrenceRelationship>
          <ViewOccurrenceRelationship uid="pvvid--000000001AA41A00--19"
xsi:type="n0:NextAssemblyOccurrenceUsage">
            <Related uidRef="pi--000000001AA41A00--19"/>
            <RelationType>
              <ClassString>next assembly occurrence</ClassString>
            </RelationType>
            <Placement>
              <GeometricRepresentationRelationship uidRef="ctrafo--
0000000013DF75A0--19"/>
            </Placement>
          </ViewOccurrenceRelationship>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```



```
<GeometricRepresentation uid="egm--0000000020AB6290 "  
xsi:type="n0:ComposedGeometricModel">  
  <ContextOfItems uidRef="ccs--origin-1"/>  
  <Id id="nut and bolt"/>  
  <Items>  
    <RepresentationItem uid="repi--000000001AA41A00--18--2 "  
xsi:type="n0:CartesianTransformation">  
      <RotationMatrix>2.83808309622E-16 -1.48711849984E-5 0.999999999889  
7.14623103897E-14 -0.999999999889 -1.48711849984E-5 1.0 7.146231460233355E-  
14 -2.82745580352E-16</RotationMatrix>  
      <TranslationVector>7451.5038 127.065 -443.85</TranslationVector>  
    </RepresentationItem>  
    <RepresentationItem uid="repi--000000001AA41A00--19--2 "  
xsi:type="n0:CartesianTransformation">  
      <RotationMatrix>1.000000 0.000000 0.000000 0.000000 1.000000 0.000000  
0.000000 0.000000 1.000000</RotationMatrix>  
      <TranslationVector>-33.000000 0.000000 0.000000</TranslationVector>  
    </RepresentationItem>  
  </Items>  
  <GeometricRepresentationRelationship uid="ctrafo--0000000013DF75A0--18 "  
xsi:type="n0:GeometricRepresentationRelationshipWithCartesianTransformation  
>  
    <Definitional>true</Definitional>  
    <Related uidRef="egm--000000001AA415B0"/>  
    <Transformation uidRef="repi--000000001AA41A00--18--2"/>  
  </GeometricRepresentationRelationship>  
  <GeometricRepresentationRelationship uid="ctrafo--0000000013DF75A0--19 "  
xsi:type="n0:GeometricRepresentationRelationshipWithCartesianTransformation  
>  
    <Definitional>true</Definitional>  
    <Related uidRef="egm--000000001AA41A00"/>  
    <Transformation uidRef="repi--000000001AA41A00--19--2"/>  
  </GeometricRepresentationRelationship>  
</GeometricRepresentation>  
<Part uid="p--000000001AA415B0">  
  <Id>  
    <Identifier uid="pid--000000001AA415B0--id6" id="bolt" idContextRef="o-  
-000000178"/>  
  </Id>  
  ...  
  <Versions>  
    <PartVersion uid="pv--000000001AA415B0--id6">  
    ...  
    <Views>  
      <PartView uid="pvv--000000001AA415B0--id6">  
        <DefiningGeometry uidRef="egm--000000001AA415B0"/>  
        ...  
        <Occurrence xsi:type="n0:SingleOccurrence" uid="pi--  
000000001AA415B0--18">  
          <Id id="bolt.1"/>  
        </Occurrence>  
        ...  
      </PartView>  
    </Views>  
  </PartVersion>
```

```
</Versions>
</Part>
<GeometricRepresentation uid="egm--000000001AA415B0"
xsi:type="n0:ExternalGeometricModel">
  <ContextOfItems uidRef="ccs--origin-2"/>
  <Id id="bolt.stp"/>
  <Items>
    <RepresentationItem xsi:type="n0:AxisPlacement" uid="repi--
000000001AA415B0--18">
      <Position>0.0 0.0 0.0</Position>
    </RepresentationItem>
  </Items>
  ...
</GeometricRepresentation>
<Part uid="p--000000001AA41A00">
  <Id>
    <Identifier uid="pid--000000001AA41A00--id7" id="nut" idContextRef="o--
000000178"/>
  </Id>
  ...
  <Versions>
    <PartVersion uid="pv--000000001AA41A00--id7">
      ...
      <Views>
        <PartView uid="pvv--000000001AA41A00--id7">
          <DefiningGeometry uidRef="egm--000000001AA41A00"/>
          ...
          <Occurrence xsi:type="n0:SingleOccurrence" uid="pi--
000000001AA41A00--19">
            <Id id="nut.1"/>
          </Occurrence>
          ...
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
<GeometricRepresentation uid="egm--000000001AA41A00"
xsi:type="n0:ExternalGeometricModel">
  <ContextOfItems uidRef="ccs--origin-3"/>
  <Id id="nut.stp"/>
  <Items>
    <RepresentationItem xsi:type="n0:AxisPlacement" uid="repi--
000000001AA41A00--19">
      <Position>0.0 0.0 0.0</Position>
    </RepresentationItem>
  </Items>
  ...
</GeometricRepresentation>
```

Entity GeometricRepresentationRelationship- WithCartesianTransformation	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of ClassificationSelect
Definitional	BOOLEAN
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
Related	ComposedGeometricModel or ExternalGeometricModel
Transformation	CartesianTransformation

Table 35: "GeometricRepresentationRelationshipWithCartesianTransformation" Attributes

#### Attribute recommendations

- **Definitional:** always TRUE (makes the related GeometricModel part of the definition of the relating GeometricModel).
- **Description:** the text or the set of texts that provides further information about the GeometricRepresentationRelationship. The value of this attribute need not be specified. Use "Description" template.
- **Related:** Reference to the ComposedGeometricModel or ExternalGeometricModel of the component part built into the assembly part
- **Transformation:** Reference to the corresponding instance of CartesianTransformation associated to the relating ComposedGeometricModel of the assembly part
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

#### 7.3.2.3 Same Coordinate Space

In this case, the GeometricRepresentationRelationship is of kind GeometricRepresentationRelationshipWithSameCoordinateSpace.

It can only apply to identify 3D transformations (identity matrix).

#### Preprocessor Recommendations:

The AxisPlacement of the Component (Related) and of the Assembly (Relating) shall contain '0 0 0' for Position and no value. for Axis and RefDirection.

The use of further subtypes of RepresentationItem (apart of AxisPlacement for the component part and for the assembly part) is not recommended.

As specified in the EXPRESS data model via a WHERE rule, unlike the two previous mapping alternatives, in this case the upper and lower GeometricModel shall share the same instance of GeometricCartesianSpace.

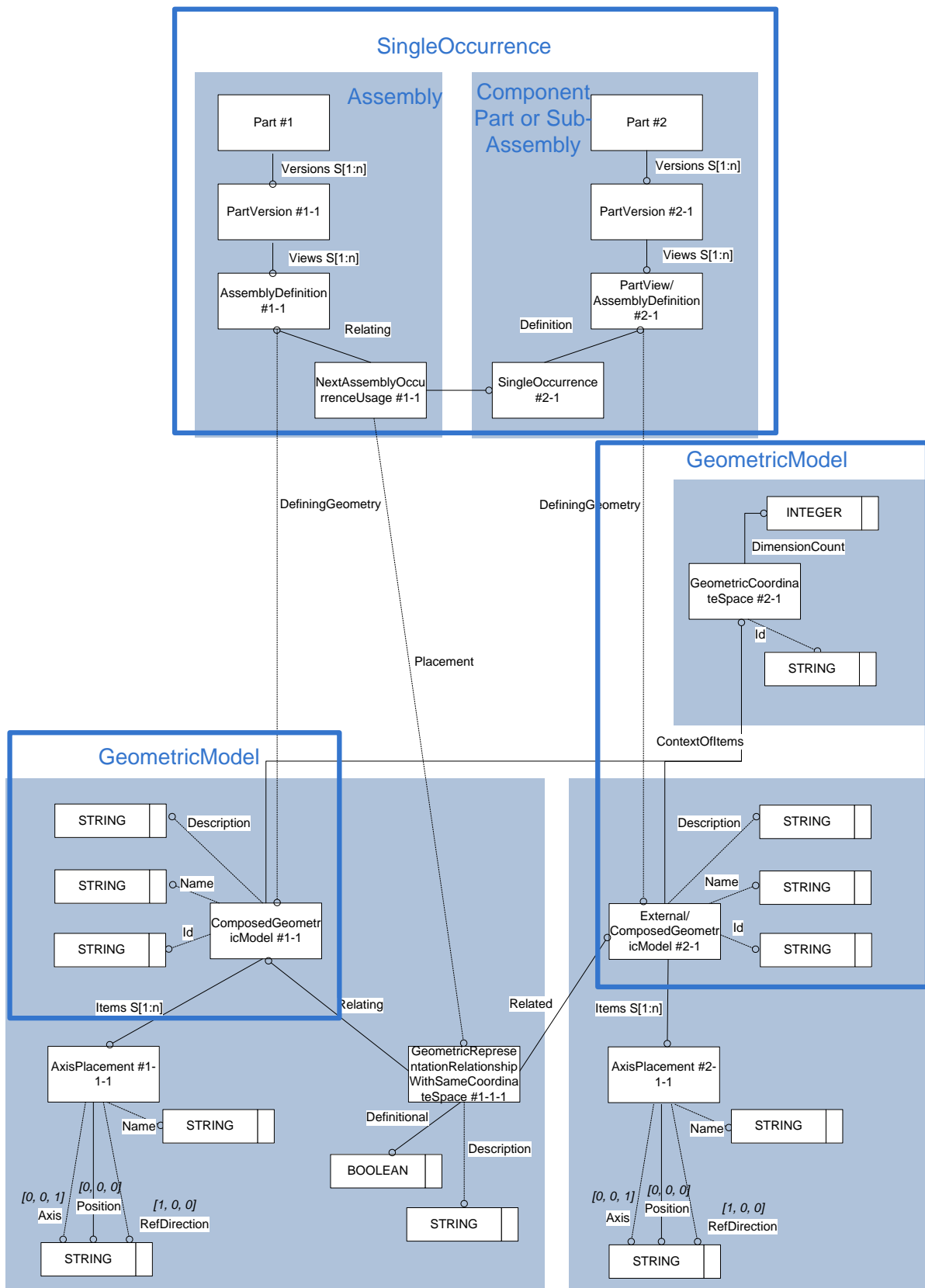


Figure 25: Full Positioning Representation with Same Coordinate Space

## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<GeometricCoordinateSpace uid="ccs--origin-1">
  <DimensionCount>3</DimensionCount>
  <Id id="/NULL"/>
</GeometricCoordinateSpace>
<Part uid="p--0000000020AB6290">
  <Id>
    <Identifier uid="pid--0000000020AB6290--id5" id="nut and bolt"
idContextRef="o--000000178"/>
  </Id>
  ...
  <Versions>
    <PartVersion uid="pv--0000000020AB6290--id5">
  ...
    <Views>
      <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--
0000000020AB6290--id5">
  ...
        <ViewOccurrenceRelationship uid="pvvid--00000001AA415B0--18"
xsi:type="n0:NextAssemblyOccurrenceUsage">
          <Related uidRef="pi--00000001AA415B0--18"/>
          <RelationType>
            <ClassString>next assembly occurrence</ClassString>
          </RelationType>
          <Placement>
            <GeometricRepresentationRelationship uidRef="ctrafo--
0000000013DF75A0--18"/>
          </Placement>
        </ViewOccurrenceRelationship>
        <ViewOccurrenceRelationship uid="pvvid--00000001AA41A00--19"
xsi:type="n0:NextAssemblyOccurrenceUsage">
          <Related uidRef="pi--00000001AA41A00--19"/>
          <RelationType>
            <ClassString>next assembly occurrence</ClassString>
          </RelationType>
          <Placement>
            <GeometricRepresentationRelationship uidRef="ctrafo--
0000000013DF75A0--19"/>
          </Placement>
        </ViewOccurrenceRelationship>
      </PartView>
    </Views>
  </PartVersion>
</Versions>
</Part>
<GeometricRepresentation uid="egm--0000000020AB6290"
xsi:type="n0:ComposedGeometricModel">
  <ContextOfItems uidRef="ccs--origin-1"/>
  <Id id="nut and bolt"/>
  <Items>
    <RepresentationItem uid="repi--00000001AA41A00--18--2"
xsi:type="n0:AxisPlacement">
      <Position>0.0 0.0 0.0</Position>
```

```
</RepresentationItem>
  <RepresentationItem uid="repi--000000001AA41A00--19--2"
xsi:type="n0:AxisPlacement">
    <Position>0.0 0.0 0.0</Position>
  </RepresentationItem>
</Items>
  <GeometricRepresentationRelationship uid="ctrafo--0000000013DF75A0--18"
xsi:type="n0:GeometricRepresentationRelationshipWithSameCoordinateSpace">
    <Definitional>true</Definitional>
    <Related uidRef="egm--000000001AA415B0"/>
  </GeometricRepresentationRelationship>
  <GeometricRepresentationRelationship uid="ctrafo--0000000013DF75A0--19"
xsi:type="n0:GeometricRepresentationRelationshipWithSameCoordinateSpace">
    <Definitional>true</Definitional>
    <Related uidRef="egm--000000001AA41A00"/>
  </GeometricRepresentationRelationship>
</GeometricRepresentation>
<Part uid="p--000000001AA415B0">
  <Id>
    <Identifier uid="pid--000000001AA415B0--id6" id="bolt" idContextRef="o-
-000000178"/>
  </Id>
  ...
  <Versions>
    <PartVersion uid="pv--000000001AA415B0--id6">
  ...
    <Views>
      <PartView uid="pvv--000000001AA415B0--id6">
        <DefiningGeometry uidRef="egm--000000001AA415B0"/>
  ...
        <Occurrence xsi:type="n0:SingleOccurrence" uid="pi--
000000001AA415B0--18">
          <Id id="bolt.1"/>
        </Occurrence>
  ...
      </PartView>
    </Views>
  </PartVersion>
</Versions>
</Part>
<GeometricRepresentation uid="egm--000000001AA415B0"
xsi:type="n0:ExternalGeometricModel">
  <ContextOfItems uidRef="ccs--origin-1"/>
  <Id id="bolt.stp"/>
  <Items>
    <RepresentationItem xsi:type="n0:AxisPlacement" uid="repi--
000000001AA415B0--18">
      <Position>0.0 0.0 0.0</Position>
    </RepresentationItem>
  </Items>
  ...
</GeometricRepresentation>
<Part uid="p--000000001AA41A00">
  <Id>
```

```

    <Identifier uid="pid--000000001AA41A00--id7" id="nut" idContextRef="o--
000000178"/>
  </Id>
  ...
  <Versions>
    <PartVersion uid="pv--000000001AA41A00--id7">
  ...
    <Views>
      <PartView uid="pvv--000000001AA41A00--id7">
        <DefiningGeometry uidRef="egm--000000001AA41A00"/>
  ...
        <Occurrence xsi:type="n0:SingleOccurrence" uid="pi--
000000001AA41A00--19">
          <Id id="nut.1"/>
        </Occurrence>
  ...
      </PartView>
    </Views>
  </PartVersion>
</Versions>
</Part>
<GeometricRepresentation uid="egm--000000001AA41A00"
xsi:type="n0:ExternalGeometricModel">
  <ContextOfItems uidRef="ccs--origin-1"/>
  <Id id="nut.stp"/>
  <Items>
    <RepresentationItem xsi:type="n0:AxisPlacement" uid="repi--
000000001AA41A00--19">
      <Position>0.0 0.0 0.0</Position>
    </RepresentationItem>
  </Items>
  ...
</GeometricRepresentation>

```

Entity GeometricRepresentationRelationshipWith-SameCoordinateSpace	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of ClassificationSelect
Definitional	BOOLEAN
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
Related	ComposedGeometricModel or ExternalGeometricModel

Table 36: "GeometricRepresentationRelationshipWithSameCoordinateSpace" Attributes

### Attribute recommendations

- **Definitional:** always TRUE (makes the related GeometricModel part of the definition of the relating GeometricModel).
- **Description:** the text or the set of texts that provides further information about the GeometricRepresentationRelationship. The value of this attribute need not be specified. Use “Description” template.
- **Related:** Reference to the ComposedGeometricModel or ExternalGeometricModel of the component part built into the assembly part
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

### 7.3.3 General Geometric Representation Relationship

This entity is introduced within the TC (bugzilla issue #5619).

Beside the existing subtypes of GeometricRepresentationRelationship defined in the previous sections, GeneralGeometricRepresentationRelationship shall support any other relationships between GeometricRepresentations.

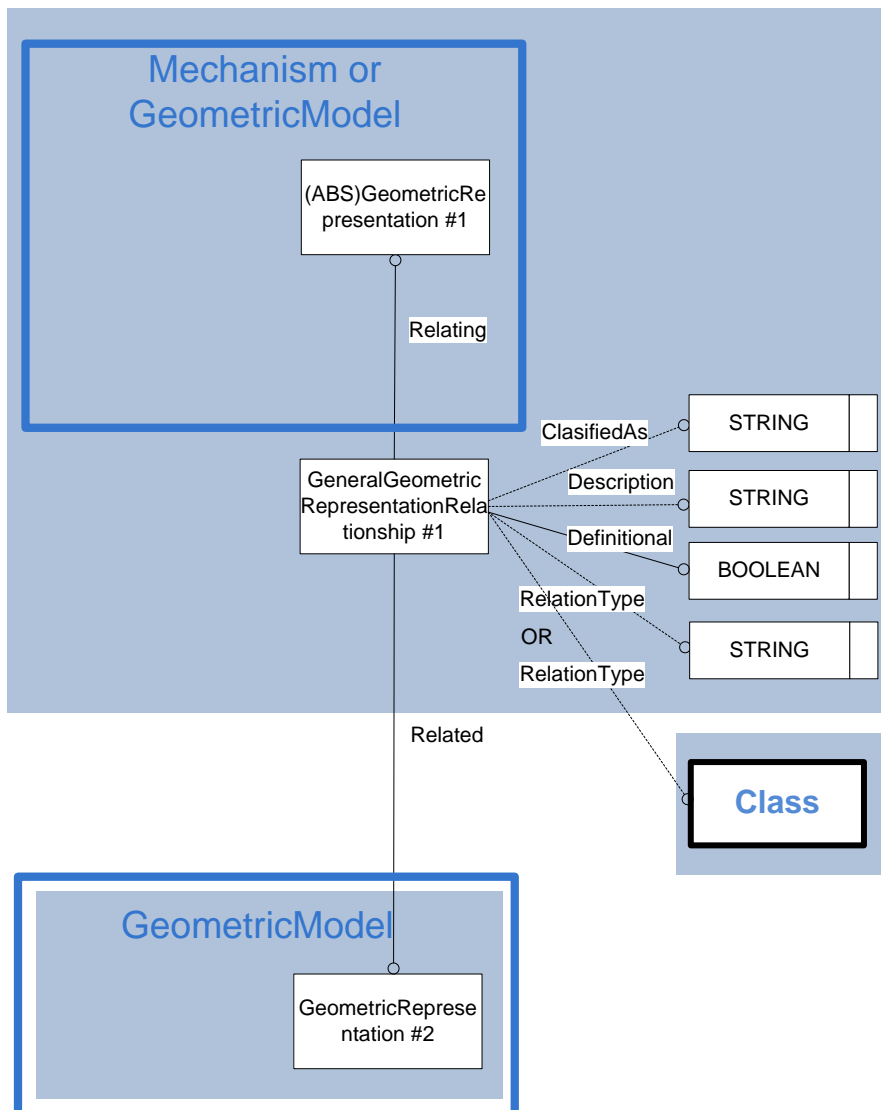


Figure 26: General Geometric Representation Relationship



## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<GeometricRepresentation xsi:type="n0:Mechanism" uid="kin--
0000000006039F00--m">
...
  <Id id="kin--0000000006039F00--m"/>
...
  <GeometricRepresentationRelationship uid="kin--0000000006039F00--mrel"
xsi:type="n0:GeneralGeometricRepresentationRelationship">
    <Definitional>false</Definitional>
    <Related uidRef="kin--0000000006039F00--m2"/>
    <RelationType>import</RelationType>
  </GeometricRepresentationRelationship>
</GeometricRepresentation>
```

Entity GeneralGeometricRepresentationRelationship	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of ClassificationSelect
Definitional	BOOLEAN
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
Related	ComposedGeometricModel or ExternalGeometricModel
RelationType	ClassSelect

Table 37: "General Geometric Representation Relationship" Attributes

### Attribute recommendations

- **Definitional:** always FALSE in case of RelationType='import' or 'local' (the imported or local dressup Mechanism is not part of the definition of the master Mechanism) and always TRUE in case of RelationType='decomposition' (the single GeometricModels are part of the definition of the overall GeometricModel).
- **Description:** the text or the set of texts that provides further information about the GeometricRepresentationRelationship. The value of this attribute need not be specified. Use "Description" template.
- **Related:** Reference to the decomposed GeometricModel (in case of RelationType='decomposition' or to the imported/local dressup Mechanism (in case of RelationType='import' or 'local')
- **RelationType:** the meaning of the relationship. Use ClassString type if one of the values below is used, otherwise use "Class" template (see 4.6.4). According to the ISO AP242 Specification, where applicable, the following values shall be used:

RelationType
--------------

'decomposition'	The business object defines a relationship where the related GeometricModel is one of potentially more sub models of the relating GeometricModel
'import'	The business object defines a relationship where the related dressup Mechanism imports all aspects of the relating master Mechanism
'local'	The business object defines a relationship where the related dressup Mechanism is locally defined on the relating master Mechanism

- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

#### 7.4 Template “PartViewRelationship”

This is the special case, where some properties need to be exchanged on PartViewRelationship or where the quantity to be stored in NextAssemblyViewUsage deviates from the number of Occurrences given by NextAssemblyOccurrenceUsage.

The use of the other subtypes of PartViewRelationship, is not recommended (not in scope of this document).

##### **Preprocessor Recommendation:**

Some PDM systems have an object to aggregate all the usages of a component in the next assembly node (with quantity N) and another object to map each usage of the component (with an occurrence.id and a 3D position matrix). The quantity n can deviate from the sum of the occurrences, for example expressing the intent to use that many occurrences but not all occurrences are established yet.

To map such an aggregation object, the use of NextAssemblyViewUsage as subtype of PartViewRelationship (with quantity N) is appropriate.

If some Properties attached to the aggregation object are exported by the source PDM system, the PartViewRelationship (with relationType='only for properties') or NextAssemblyViewUsage (with relationType='next assembly view') can be created and those properties attached to it.

It is not recommended to use multiple PartViewRelationships between the same parent and child, since this is not supported by all PDM systems.

##### **Postprocessor Recommendation:**

If the target PDM system has also an aggregation object as defined above, it may import the properties associated to the PartViewRelationship objects (but only the properties and not the assembly structure defined below it) as well as the NextAssemblyViewUsage.quantity (as 'target' quantity, independently of the number of NextAssemblyOccurrenceUsages attached to the is part.

If such an aggregation object is not provided in the target system, or if it gets created automatically out of the sum of all NextAssemblyOccurrenceUsages, the postprocessor shall ignore the PartViewRelationship/NextAssemblyViewUsage.

In both cases, the PartViewRelationship/NextAssemblyViewUsages and the NextAssemblyOccurrenceUsages shall not be interpreted in addition to each other; otherwise an assembly with too many components would be imported.

The postprocessor shall recognize multiple PartViewRelationships between the same parent and child, and (if the target system doesn't support it) merge all of them to one during import.

To enable to combine PartViewRelationship/NextAssemblyViewUsages and NextAssemblyOccurrenceUsages with the appropriate semantic, a Bugzilla issue #6022 has been submitted.

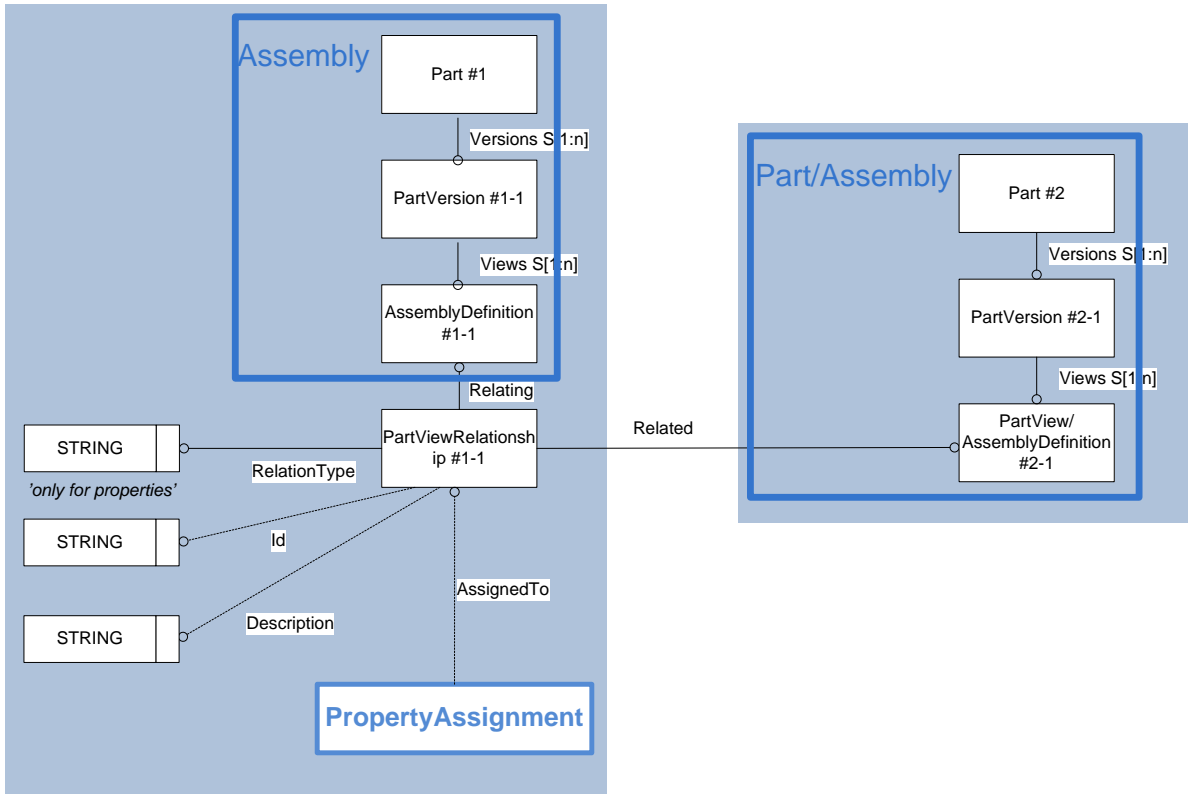


Figure 27: Template "PartViewRelationship" for properties

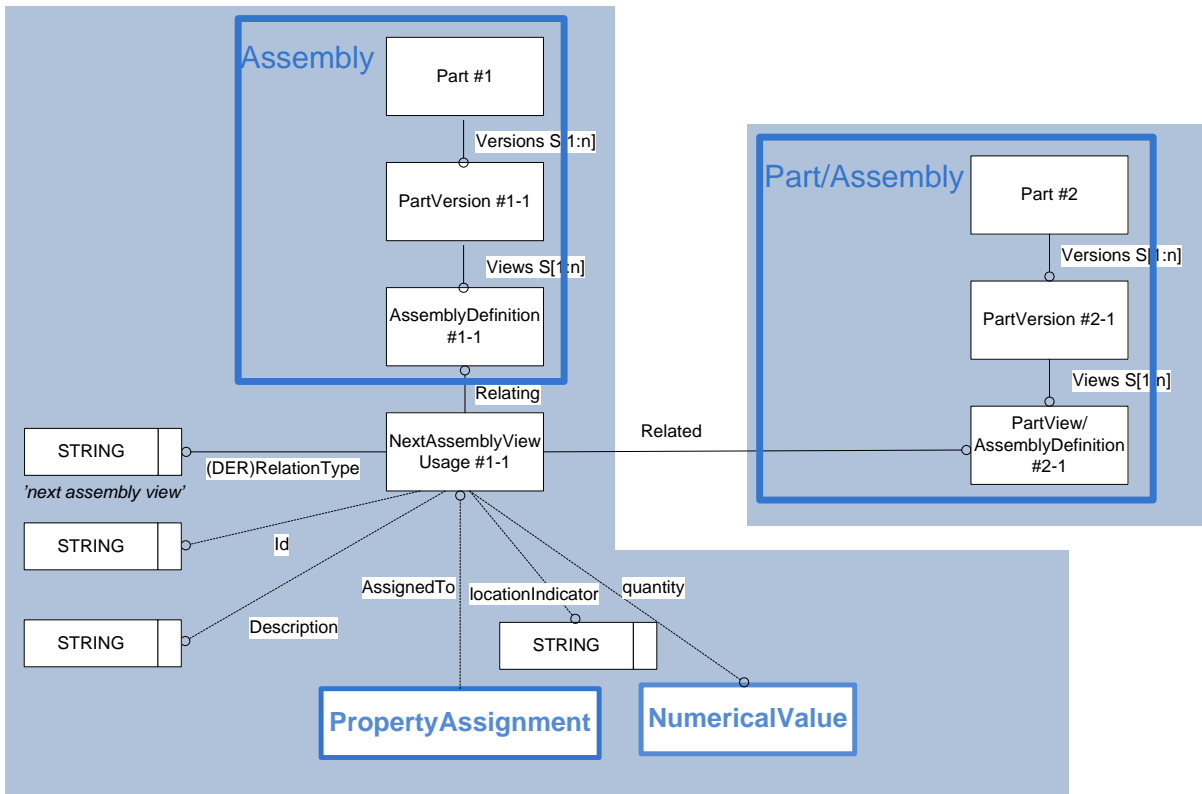


Figure 28: Template "PartViewRelationship" for quantity (+ properties)

## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```

<PropertyDefinition uid="i1067">
  <Id id="quality property"/>
  <PropertyType>
    <ClassString>PDM property</ClassString>
  </PropertyType>
</PropertyDefinition>
<Part uid="p--0000000017086CB0">
  <Id>
    <Identifier uid="pid--0000000017086CB0--id1" id="as1" idContextRef="o--000000178"/>
  </Id>
  ...
  <Versions>
    <PartVersion uid="pv--0000000017086CB0--id1">
      ...
      <Views>
        <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--0000000017086CB0--id1">
          ...
          <PartViewRelationship uid="pvrid--000000001E5A89F0--10">
            <Related uidRef="pvv--000000001E5A89F0--id2"/>
            <RelationType>
              <ClassString>only for properties</ClassString>
            </RelationType>
            <PropertyValueAssignment uid="i1769">
              <AssignedPropertyValues>
                <PropertyValue uid="i1771" xsi:type="n0:StringValue">
                  <Definition>
                    <PropertyDefinition uidRef="i1067">
                    </PropertyDefinition>
                  </Definition>
                  <Name>
                    <CharacterString>lineNumber</CharacterString>
                  </Name>
                  <ValueComponent>
                    <CharacterString>40</CharacterString>
                  </ValueComponent>
                </AssignedPropertyValues>
              </PropertyValueAssignment>
            </PartViewRelationship>
          </PartView>
        </Views>
      </PartVersion>
    </Versions>
  </Part>
  <Part uid="p--000000001E5A89F0">
    <Id>
      <Identifier uid="pid--000000001E5A89F0--id2" id="plate" idContextRef="o--000000178"/>
    </Id>
    ...
    <Versions>
      <PartVersion uid="pv--000000001E5A89F0--id2">

```

```

...
    <Views>
      <PartView uid="pvv--000000001E5A89F0--id2">
...
      </PartView>
    </Views>
  </PartVersion>
</Versions>
</Part>

```

Entity PartViewRelationship	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of ClassificationSelect
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
MaterialIdentification	OPTIONAL SET[1:?] of MaterialIdentification
Related	PartView
RelationType	ClassSelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
CertificationAssignment	OPTIONAL SET[1:?] of CertificationAssignment
ConditionAssignment	OPTIONAL SET[1:?] of ConditionAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssignment

Table 38: "PartViewRelationship" Attributes

**Attribute recommendations**

- **Description:** the text or the set of texts that provide further information about the Assembly structure link. The value of this attribute need not be specified. Use "Description" template.
- **Id:** stores the Identifier for the PartViewRelationship (if set, shall be unique over all PartViewRelationship directly referenced by an assembly part). The value of this attribute need not be specified. Use IdentifierString type.
- **Related:** Reference to a PartView of the component part built into the assembly part.
- **RelationType:** the meaning of the relationship. Use ClassString type. Mandatory value: 'only for properties'.
- **PropertyValueAssignment:** adds the value of a property to the PartViewRelationship. Use "PropertyAssignment" template (see 6.2)
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)**

```
<Part uid="p--0000000017086CB0">
  <Id>
    <Identifier uid="pid--0000000017086CB0--id1" id="as1" idContextRef="o--000000178"/>
  </Id>
  ...
  <Versions>
    <PartVersion uid="pv--0000000017086CB0--id1">
      ...
      <Views>
        <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--0000000017086CB0--id1">
          ...
          <PartViewRelationship xsi:type="n0:NextAssemblyViewUsage" uid="pvrid--000000001E5A89F0--10">
            <Related uidRef="pvv--000000001E5A89F0--id2"/>
            <RelationType>
              <ClassString>next assembly view</ClassString>
            </RelationType>
          ...
          <Quantity uid="i1770" xsi:type="n0:NumericalValue">
            <Definition>
              <PropertyDefinitionString>quantity</PropertyDefinitionString>
            </Definition>
            <Unit uidRef="i1231"></Unit>
            <ValueComponent>1.</ValueComponent>
          </Quantity>
        </PartViewRelationship>
      ...
    </PartView>
  </Views>
</PartVersion>
</Versions>
```

```

</Part>

<Unit uid="i1231">
  <Name>
    <ClassString>each</ClassString>
  </Name>
</Unit>

<Part uid="p--000000001E5A89F0">
  <Id>
    <Identifier uid="pid--000000001E5A89F0--id2" id="plate" idContextRef="o--000000178"/>
  </Id>
  ...
  <Versions>
    <PartVersion uid="pv--000000001E5A89F0--id2">
    ...
      <Views>
        <PartView uid="pvv--000000001E5A89F0--id2">
        ...
          </PartView>
        </Views>
      </PartVersion>
    </Versions>
  </Part>

```

Entity NextAssemblyViewUsage	Attribute type (additionaly to PartViewRelationship)
DefiningGeometry	OPTIONAL GeometricModel
LocationIndicator	OPTIONAL IdentifierSelect
Quantity	OPTIONAL ValueWithUnit
ShapeDependentProperty	OPTIONAL SET[1:?] of ShapeDependentProperty
ShapeElement	OPTIONAL SET[1:?] of ShapeElement
SurfaceCondition	OPTIONAL SET[1:?] of SurfaceCondition
AssemblyViewRelationshipSubstitution	OPTIONAL SET[1:?] of AssemblyViewRelationshipSubstitution
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment

Table 39: "NextAssemblyViewUsage" Attributes

**Attribute recommendations**

- **LocationIndicator:** the text that identifies this usage of the component in the assembly in a diagram, list, chart, or on a physical piece of equipment. The value of this attribute need not be specified Use IdentifierString type.
- **Quantity:** the [ValueWithUnit](#) that defines the amount of this usage of the component in the assembly. The value of this attribute need not be specified. Use NumericalValue template.
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

## 8 Document Identification and Classification

The scope of this section corresponds to sections 5 and 6 of the PDM Schema Usage Guide V4.3.

A Document in the context of ISO 10303-3001 is a managed document. This means, it is under revision control, and various representations of a document version may be distinguished. The DocumentVersion represents the minimum identification of a managed document under revision control. A document representation definition may optionally be associated with one or more constituent external files that make up the contents of the document.

Similar to Parts, the identification of Documents in the AP242 BO Model consists of three concepts:

- Document Master Identification:
  - Document identification has specific requirements to assign documents to other product data, and to optionally associate with the constituent external file(s) that make up a specific document representation view definition;
- Context Information,
  - Document identification has different context information than Part identification;
- Type Classification
  - Document identification has a different type classification than Part identification.

These three concepts are represented by attributes of the three elements of the Document template; see 8.1 for details.

An external file is not managed independently by the system - there is usually no revision control or any representation definitions of external files. Version identification may optionally be associated with an external file, but this is for information only and is not used for managed revision control.

If a file is under configuration control, it shall be represented as a constituent of a document definition view/representation. Thus, it is actually the managed document that is under direct configuration control; the file is only indirectly under configuration control. A change to the file results in a change to the managed document (i.e., a new version). The changed file becomes a constituent of a view/representation definition of the new document version. A simple external reference alone is not configuration controlled; it is just an external file reference to product data. See also 11.1 for association of unmanaged files.

Documents may be associated with product data in a specified role using DocumentAssignment to represent some relationship between a document and other elements of product data. Constraints may be specified on this association, in order to distinguish an applicable portion of an entire document or file in the association. This linkage may be made at the level of the base identification of the document, the document version, or the document representation view definition. The recommended level from which a document master should reference other product data is the document version. See chapter 11 for details.

The following types of data may in general be assigned to a Document in the context of ISO 10303-3001 to characterize it further:

- ActivityMethod, Approval, Certification, Contract, DateTimeString, Document, DocumentVersion, DocumentDefinition, File, Effectivity, InformationUsageRight, Project, RequirementView, SecurityClassification, and WorkRequest.

These recommended practices for assembly structures, however, only cover relationships to the following concepts:

- DateTimeString, Document, DocumentVersion, DocumentDefinition, File, and SecurityClassification.



For document classification the AP242 BO Model distinguishes – as for Parts - the following two approaches:

- Type classification
  - An identified document may be placed into one or several of the following categories: 'catalogue', 'manual' or 'specification'. These values are set in the attribute Document.DocumentTypes; see 8.1.1 .
- General classification
  - Documents may need to be classified according to a classification system with explicit reference to classification criteria and related properties. For example, design documents may be classified according to level of design and to type of product. Such classification is enabled by the attribute Document.ClassifiedAs; see 8.1.1 . Thus, a Document may be linked to an extensive and already existing classification system.

### **8.1 Template “Document”**

The Document template supports – similar to the Part template (see 5.1) – the ability to uniquely identify a Document, its meta data and its properties. It consists in the AP242 BO Model of three structurally distinct data types as also shown in Figure 29:

- Document,
- DocumentVersion and,
- DocumentDefinition.

The general recommendations given for Part identification apply also to the Document identification, except where differences are noted.

Base document identification is always associated with at least one document version. Multiple document versions of a base document identification may be related together to represent document version history.

DocumentDefinition is used to define a view of a particular representation of a document version. A document version does not have to have an associated document representation definition.

The view definition of a document version is used for association of document properties, to build document structures, and to associate a document with the set of constituent external files that make it up.

Document, DocumentVersion and DocumentDefinition shall be written to the XML-file using containment. The information elements in the white area on the left side of Figure 29 are root elements and are, thus, outside of this containment block.

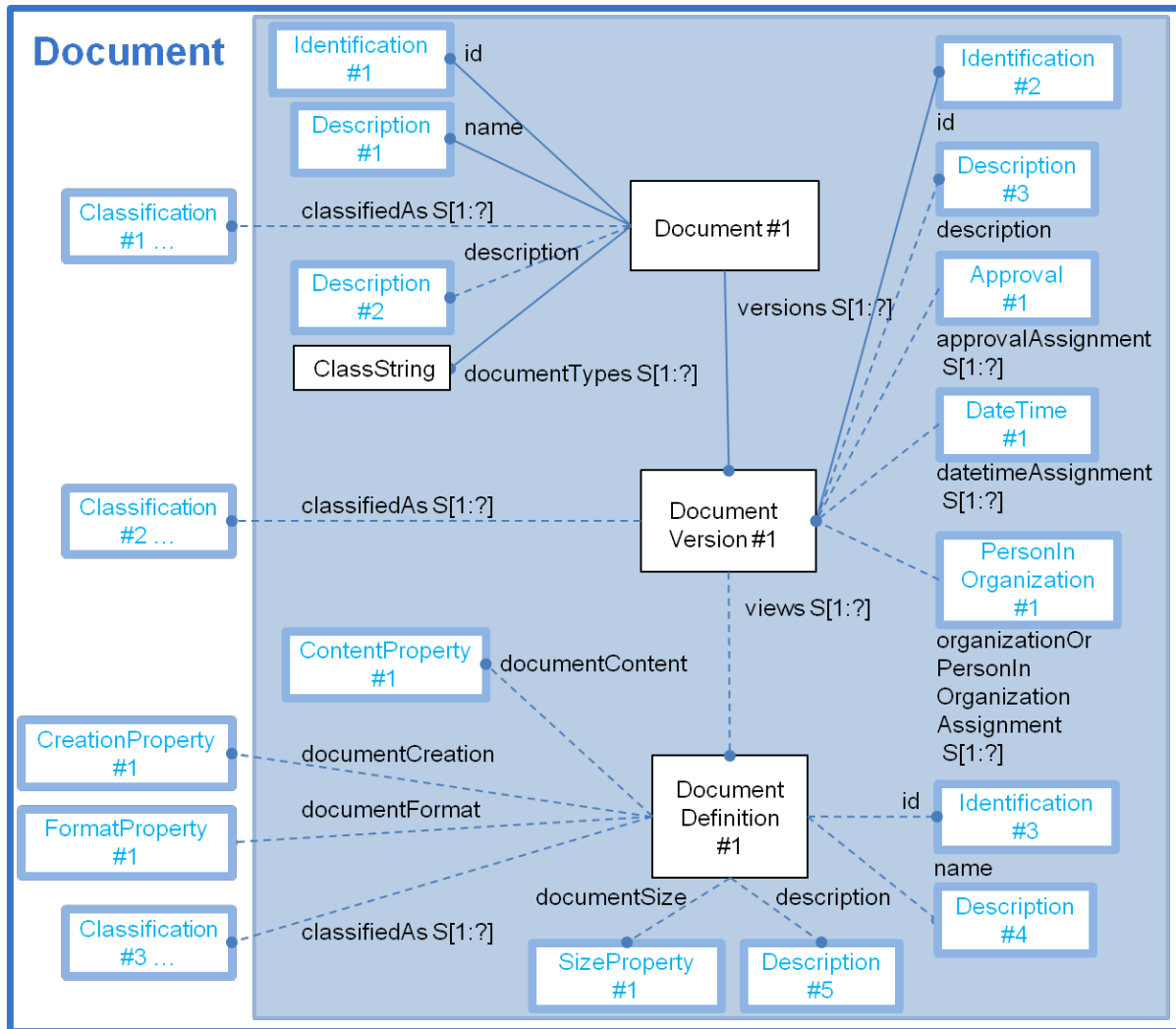


Figure 29: Template "Document"

### 8.1.1 Document

The Document entity represents the document master base information. This entity collects all information that is common among the different versions and views of the document. The document number is strictly an identifier. It should not be used as a 'smart string' with some parseable internal coding scheme, e.g., to identify version or classification information.

The Document number identifier shall be unique within the scope of the business process of the information exchange. This is typically not a problem when the Document is only used within a single company. For external use, the identification must be interpreted as unique within that broader domain. Processors may need to evaluate more than one string (i.e., more than only Document.id) to establish unique identification of the Document. The "Identifier" template provides a combination of parameters including Identifier.idRoleRef and Identifier.idContextRef that make Document identification unique.

The following XML-snippet is an example from a physical file that is in accordance to Figure 29.

### The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<Document uid="doc--000000001EAAE870">
  <DocumentTypes>
    <ClassString>geometry</ClassString>
```

```

</DocumentTypes>
<Id>
  <Identifier uid="docid--000000001EAAE870--id7" id="nut" idContextRef="o--000000178"/>
</Id>
<Name>
  <CharacterString>/NULL</CharacterString>
</Name>
<Versions>
  <DocumentVersion uid="dv--000000001EAAE870">
    <Id id="/NULL"/>
    <Views>
      <DocumentDefinition uid="ddd--000000001EAAE870"
xsi:type="n0:DigitalDocumentDefinition">
        <Id id="/NULL"/>
        <Files>
          <DigitalFile uidRef="df--000000001EAAE870"/>
        </Files>
      </DocumentDefinition>
    </Views>
  </DocumentVersion>
</Versions>
</Document>
  
```

Entity Document attributes	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
DocumentTypes	SET[1:?] of ClassSelect
Id	IdentifierSelect
Name	DescriptorSelect
SameAs	OPTIONAL SET[1:?] of ProxySelect
Versions	SET[1:?] of DocumentVersion
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
CertificationAssignment	OPTIONAL SET[1:?] of CertificationAssignment
ContractAssignment	OPTIONAL SET[1:?] of ContractAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DatetimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
DocumentRelationship	OPTIONAL SET[1:?] of DocumentRelationship
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment

Entity Document attributes	Attribute type
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
ProjectAssignment	OPTIONAL SET[1:?] of ProjectAssignment
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssignment

Table 40: "Document" Attributes

### Attribute recommendations

- **ClassifiedAs:** the classifications of the Document. The value of this attribute need not be specified. Use "Classification" template (see 4.6.5 ).
- **Description:** an expanded name or text that provides further information about the Document. The value of this attribute need not be specified. Use Description template (see 4.6.7).
- **DocumentTypes:** the category of a Document. Use ClassString type if one of the values below is used, otherwise use "Class" template (see 4.6.4). As defined in the ISO AP242 specification. When applicable, the value of this element shall be one or several of the following:

DocumentTypes	Explanation
'catalogue'	the Document is the catalogue in which the associated object is listed
'manual'	the Document is the handbook that is supplied for the associated object
'specification'	the Document specifies the considerations that lead to the design finally chosen for the associated object
'primary geometry'	The document file represents one or many shape models in the highest quality available for this part
'secondary geometry'	The document file represents one of many shape models in a format derived from the primary geometry, for example a standard format or a tessellated format or only the external shape (without inner geometry)
'NC data'	The document file represents numerical control data
'FE data'	The document file represents finite element data
'sample data'	The document file represents measured data
'process plan'	The document file represents process planning data
'check plan'	The document file represents quality control planning data
'drawing'	The document file represents a technical drawing
'structured product data'	The document file contains product meta data and data

DocumentTypes	Explanation
	related to product structure. This value shall be used for nested external references, when the referenced document relates to another BO Model XML file (see section 9.3)

- **Id:** the identifier or set of identifiers for the Document, the document number. The referenced Identifier element shall have valid values for elements Identifier.idRoleRef and Identifier.idContextRef. Use “Identifier” template (see 4.6.6).
- **Name:** the words or set of words by which the Document is known. Use “Description” template (see 4.6.7).
- **Versions:** the related releases of the Document; a Document shall have at least one DocumentVersion.
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendation:**

A part shall have at most one Document of type ‘primary geometry’. In most cases, this will be the native CAD geometry. A part may have none or many Documents of type ‘secondary geometry’; for example one containing a standard format, one containing the tessellated geometry, one containing only the external shape, etc...

What ‘primary’ and ‘secondary’ actually means is defined by the originating PDM system. There are exchange scenarios where only ‘secondary geometry’ will be exchanged, e.g. when for a Part where the native (‘primary’) is available in the sending system, only a derived format is sent to a supplier or customer.

**Postprocessor Recommendation:**

Documents of type ‘primary geometry’ and ‘secondary geometry’ are alternates to define the part, while all the other document types (‘drawing’, ‘specification’...) are complementary to the geometry.

### 8.1.2 DocumentVersion

A DocumentVersion is a release of a Document. It represents the identification of a specific version of the base Document identification. A particular DocumentVersion is always related to exactly one Document. This is why, in XML it is embedded within a Document element.

**Preprocessor Recommendations:**

- Though not required, it is recommended to assign at least one view definition to each document version. A valid exception to this general rule is the exchange of versions that represent an entire version history; in this case only the most recent version is required to have an associated view definition.
- For the purpose of the typical PDM data exchange use case of these recommended practices, multiple versions of each document and multiple DigitalDocumentDefinitions of each version may be exchanged.

An example of DocumentVersion instantiation is in the XML-snippet in section 8.1.1 .

Entity DocumentVersion attributes	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification

Entity DocumentVersion attributes	Attribute type
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
SameAs	OPTIONAL SET[1:?] of ProxySelect
Views	OPTIONAL SET[1:?] of DocumentDefinition
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
CertificationAssignment	OPTIONAL SET[1:?] of CertificationAssignment
ContractAssignment	OPTIONAL SET[1:?] of ContractAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DatetimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
DocumentVersionRelationship	OPTIONAL SET[1:?] of DocumentVersionRelationship
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
ProjectAssignment	OPTIONAL SET[1:?] of ProjectAssignment
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssignment

Table 41: "DocumentVersion" Attributes

### Attribute recommendations

- **ApprovalAssignment:** the level of acceptance of the DocumentVersion. The value of this attribute need not be specified. Use "Approval" template (see 4.6.12).
- **ClassifiedAs:** the classifications of the DocumentVersion. The value of this attribute need not be specified. Use "Classification" template (see 4.6.5).

- **DatetimeAssignment:** the date and time of the creation or update of the DocumentVersion. The value of this attribute need not be specified. Use "DateTime" template (see 4.6.11).
- **Description:** the reason for the creation of the version. The value of this attribute need not be specified. Use "Description" template (see 4.6.7).
- **Id:** the identifier or set of identifiers for the DocumentVersion, the document version number. Use IdentifierString type or "Identifier" template (see 4.6.6).
  - **Preprocessor Recommendations:** If an organization does not version documents, it is recommended that the id attribute contains the string '/NULL' to indicate that no version information is relevant or intended. In this case only a single DocumentVersion shall be assigned to the Document.
  - **Postprocessor Recommendations:** If the value of the id attribute for a DocumentVersion is the string '/NULL', postprocessors should use this as an indication that the sending system or business process does not support versioning of Documents.
- **OrganizationOrPersonInOrganizationAssignment:** an organization or person in organization with a specific relation to the DocumentVersion according to the OrganizationOrPersonInOrganizationAssignment.role attribute. The value of this attribute need not be specified. Use "PersonInOrganization" template (see 4.6.14).
- **Views:** the set of DocumentDefinition objects that are defined for the DocumentVersion.
  - In general, each DocumentVersion is recommended to have an associated DocumentDefinition representing one of its view definitions. In restricted cases, a DocumentVersion without a definition may be used to enhance information about another related, fully defined version. In the following specific case a DocumentVersion may be exchanged without an associated DocumentDefinition:
    - When version history (sequence relationship) is represented - only the most recent version is required to have an assigned DocumentDefinition. If there is no DocumentDefinition associated with the previous versions, only basic information about the sequence of previous versions is exchanged as additional information about the current DocumentVersion that is the focus of the data exchange.
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

### 8.1.3 DocumentDefinition

The DocumentDefinition entity denotes the definition of a particular view of a representation of a DocumentVersion. There may be more than one document representation definition associated with a single document version. The representation view definition of a document version is used for association of document properties, to build document structures, and to associate a document with the set of constituent external files that make it up. The entity DocumentDefinition supports property association and document structure. The subtype DigitalDocumentDefinition is used to associate a representation of a document version with the set of constituent files that make it up. See chapter 8.2 for identification of external files and for associating external files to documents.

#### **Preprocessor Recommendations:**

- The use of DocumentDefinition entities is not strictly required by rules in the AP242 BO Model, but it is strongly recommended. All DocumentVersion entities should always have at least one associated DocumentDefinition, except in the case of the exchange of pure version history information.
- If a PDM system does not distinguish between DocumentVersion and DocumentDefinition, only one DocumentDefinition shall be mapped (having id as unset).

**Postprocessor Recommendations:**

- the general behavior for evaluating ContentProperty, CreationProperty, FormatProperty and SizeProperty shall be 'only for information', and shall not cause the postprocessor to stop processing if the given content, creation system, format or size is not supported by the postprocessor. The postprocessor shall load and import the files correctly.

An example of a DocumentDefinition instantiation is in the XML-snippet in section 8.1.1 .

Entity (Digital)DocumentDefinition attributes	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
DocumentContent	OPTIONAL ContentProperty
DocumentCreation	OPTIONAL CreationProperty
DocumentFormat	OPTIONAL SET[1:?] OF FormatProperty
DocumentSize	OPTIONAL SizeProperty
Id	OPTIONAL IdentifierSelect
InZone	OPTIONAL SET[1:?] of InZone
Name	OPTIONAL DescriptorSelect
SameAs	OPTIONAL SET[1:?] of ProxySelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
BreakdownVersionAssignment	OPTIONAL SET[1:?] of BreakdownVersionAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DatetimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
DocumentDefinitionRelationship	OPTIONAL SET[1:?] of DocumentDefinitionRelationship
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment



Entity (Digital)DocumentDefinition attributes	Attribute type
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssignment
(only for DigitalDocumentDefinition) Files	OPTIONAL SET[1:?] of DigitalFiles

Table 42: "(Digital)DocumentDefinition" Attributes

#### Attribute recommendations

- **ClassifiedAs:** the classifications of the DocumentDefinition. The value of this attribute does not need to be specified. Use "Classification" template (see 4.6.5).
- **Description:** text or the set of texts that provide further information about the DocumentDefinition. The value of this attribute does not need to be specified. Use "Description" template (see 4.6.7).
- **DocumentContent:** the characteristics of the content of the document represented by DocumentDefinition. The value of this attribute does not need to be specified. Use "ContentProperty" template (see 10.2).
- **DocumentCreation:** further details of the creation of the document represented by DocumentDefinition. The value of this attribute does not need to be specified. Use "CreationProperty" template (see 10.3).
- **DocumentFormat:** the format of the document represented by DocumentDefinition. The value of this attribute does not need to be specified. Use "FormatProperty" template (see 10.1).
- **DocumentSize:** the size of the document represented by DocumentDefinition. The value of this attribute need not be specified. Use "SizeProperty" template (see 10.4).
- **Id:** the identifier or set of identifiers for the DocumentDefinition. The value of this attribute need not be specified. Use IdentifierString type or "Identifier" template (see 4.6.6).
  - **Preprocessor Recommendations:** There is no standard mapping for the id attribute of DocumentDefinition; however, the value should be unique relative to other DocumentDefinitions related to the same DocumentVersion. The id attribute shall not be 'overloaded' to include, for example, life-cycle or organizational information; this is generally not recommended for the AP242 BO Model. This attribute should contain a unique identifier for the DocumentDefinition - no additional semantics are associated with this attribute.
  - **Postprocessor Recommendations:** Postprocessors do not need to expect any semantics from the id attribute; it is a pure identifying string. The id value

– possibly composed of several values according to the “Identifier” template - should be unique relative to other the identifiers of other DocumentDefinition related to the same DocumentVersion.

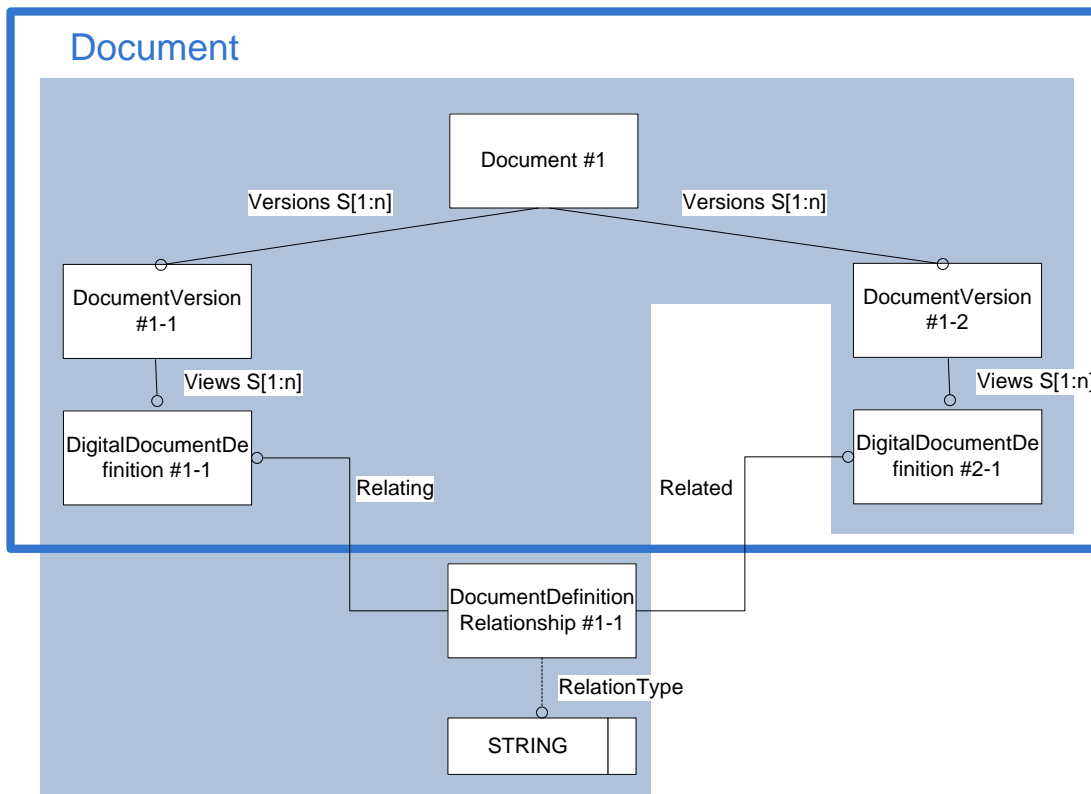
- **Name:** the words or set of words by which the DocumentDefinition is known. The value of this attribute need not be specified. Use “Description” template (see 4.6.7).
- **DocumentDefinitionRelationship:** to relate to another document. Use the “DocumentDefinitionRelationship” template (see 8.2).
- **PropertyValueAssignment:** to assign a PropertyValue to the DocumentDefinition. Use the “DocumentFileProperty” template (see 10.5).
- **Files** (in case of DigitalDocumentDefinition): to assign one or many DigitalFiles to the DocumentDefinition. Use the “DigitalFile” template (see 9.1).

**Remark: it is recommended to assign only one file here, in order to manage the versioning of each of them as a distinct document within the PDM system. In case of model splitting or alternate models, please refer to the special cases explained in 11.2.**

- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

## 8.2 Template “DocumentDefinitionRelationship”

This relationship enables to relate two DocumentDefinitions of the same document:



or of different documents:

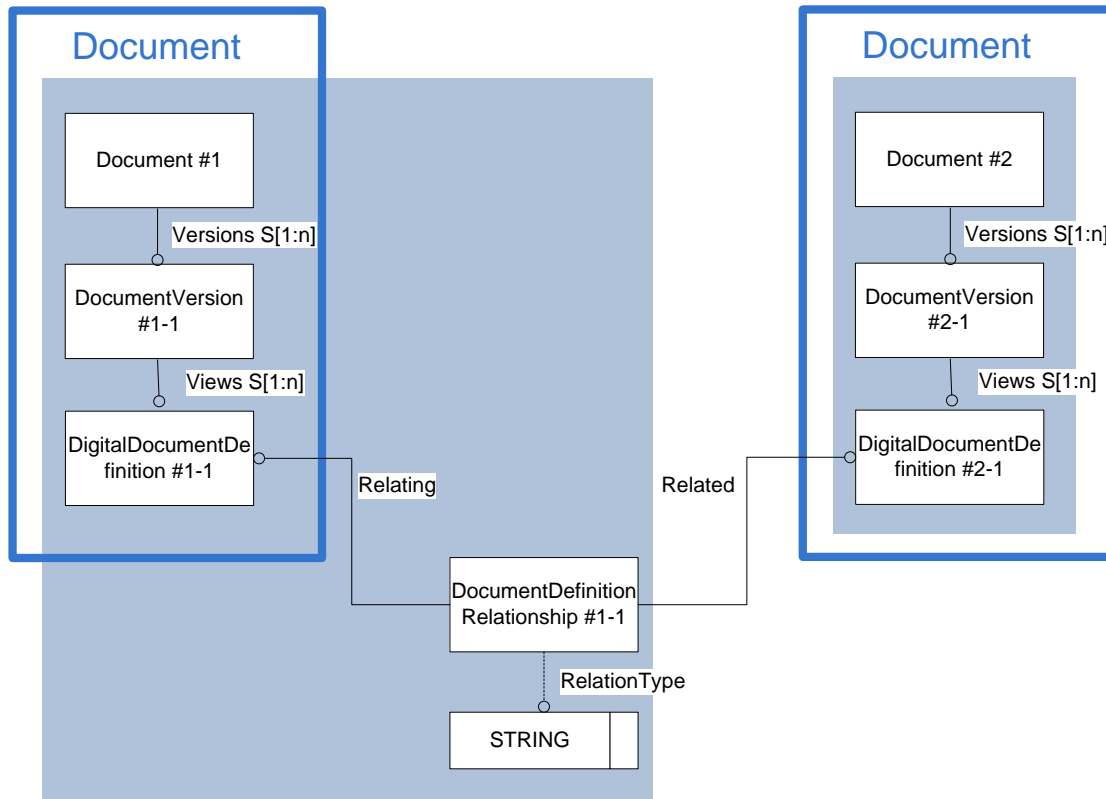


Figure 30: Template "DocumentDefinitionRelationship"

### The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```

<Document uid="doc--000000001EB04CF0">
...
  <Id>
    <Identifier uid="docid--000000001EB04CF0--id2" id="plate" idContextRef="o--000000178"/>
  </Id>
...
  <Versions>
    <DocumentVersion uid="dv--000000001EB04CF0">
...
      <Id id="A.1"/>
      <Views>
        <DocumentDefinition uid="ddd--000000001EB04CF0"
xsi:type="n0:DigitalDocumentDefinition">
          <Id id="/NULL"/>
          <DocumentDefinitionRelationship uid="ddr--1">
            <Related uidRef="ddd--000000001EB04CF0--2"/>
            <RelationType>
              <ClassString>sequence</ClassString>
            </RelationType>
          </DocumentDefinitionRelationship>
...
        </DocumentDefinition>
        <DocumentDefinition uid="ddd--000000001EB04CF0--2"
xsi:type="n0:DigitalDocumentDefinition">
          <Id id="/NULL"/>
...
        </DocumentDefinition>
      </Views>
    </DocumentVersion>
  </Versions>
</Document>

```

```
...
  </DocumentVersion>
</Versions>
</Document>
```

Entity DocumentDefinitionRelationship attributes	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
Related	DocumentDefinition
RelationType	ClassSelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
CertificationAssignment	OPTIONAL SET[1:?] of CertificationAssignment
ConditionAssignment	OPTIONAL SET[1:?] of ConditionAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DatetimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssignment

Table 43: "DocumentDefinitionRelationship" Attributes

**Attribute recommendations**

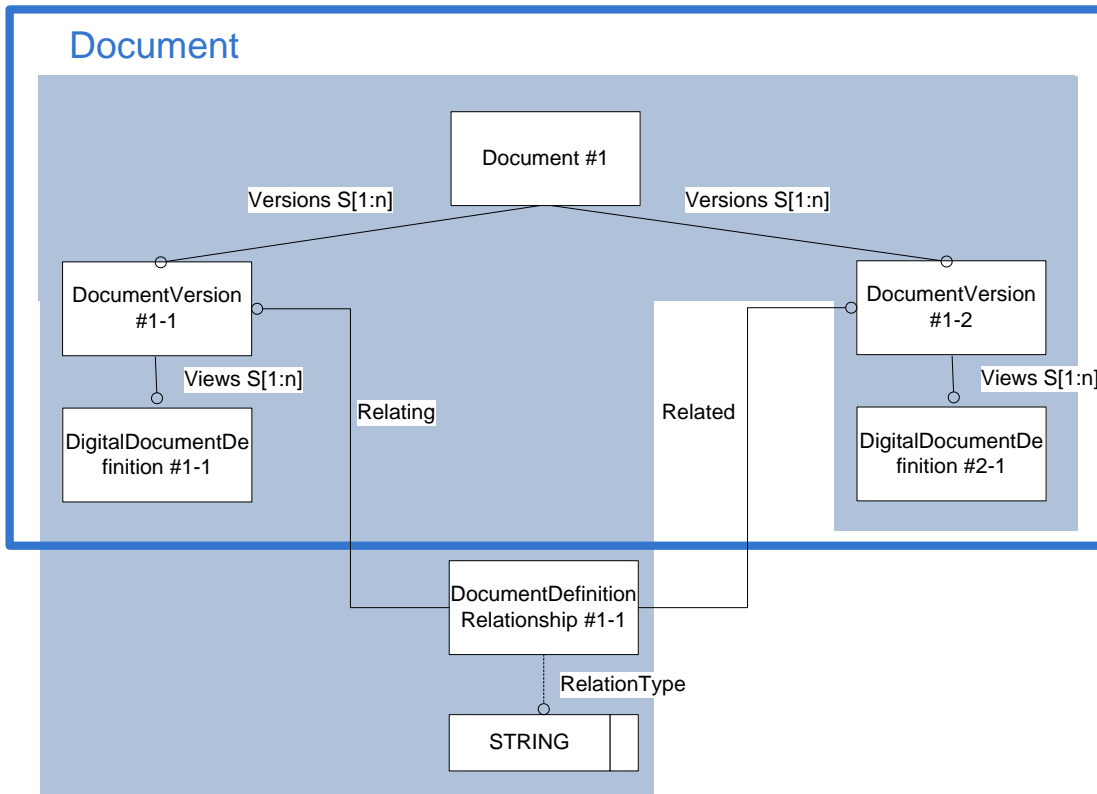
- **RelationType:** the meaning of the relationship. Use ClassString type if one of the values below is used, otherwise use "Class" template (see 4.6.4). According to the ISO AP242 Specification, where applicable, the following values shall be used:

RelationType	
'addition'	The business object specifies that the related document provides supplementary or collateral information with regard to the information provided by the relating document
'copy'	The business object defines a relationship where the related DocumentDefinition is a copy of the relating DocumentDefinition
'decomposition'	The business object defines a relationship where the related DocumentDefinition is one of potentially more sub documents of the relating DocumentDefinition
'derivation'	The business object defines a relationship where the related DocumentDefinition is derived from the relating DocumentDefinition
'peer'	The business object specifies that the related document provides required information with regard to that provided by the relating document. The peer document is essential for a complete understanding
'reference'	The business object defines a relationship where the related document is referenced from the relating
'sequence'	The business object defines a logical sequence where the related DocumentDefinition comes after the relating DocumentDefinition
'substitution'	The business object defines a relationship where the related DocumentDefinition replaces the relating DocumentDefinition
'translation'	The DocumentDefinitionRelationship specifies that the related document is generated through a translation process from the relating document

- **Related:** the other object of [DocumentDefinition](#) that is part of the relationship
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

### 8.3 Template “DocumentVersionRelationship”

Used to relate several versions of the same document:



or of different documents.

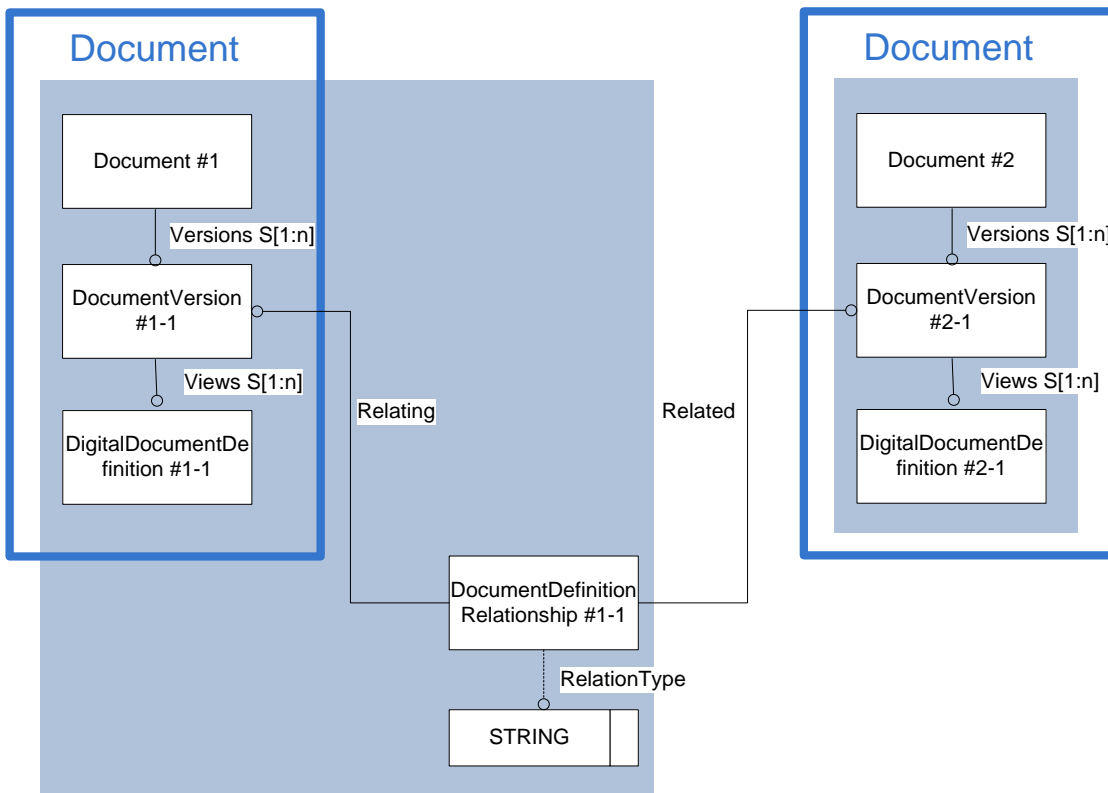


Figure 31: Template "DocumentVersionRelationship"

## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<Document uid="doc--0000000017D374A0">
...
  <Versions>
    <DocumentVersion uid="dv--0000000017D374A0">
...
      <DocumentVersionRelationship uid="dvr--1234">
        <Related uidRef="dv--000000001EB04CF0"/>
        <RelationType>
          <ClassString>sequence</ClassString>
        </RelationType>
      </DocumentVersionRelationship>
...
    </DocumentVersion>
  </Versions>
</Document>
```

Entity DocumentVersionRelationship attributes	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
Related	DocumentVersion
RelationType	ClassSelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
CertificationAssignment	OPTIONAL SET[1:?] of CertificationAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DatetimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 44: "DocumentVersionRelationship" Attributes

### Attribute recommendations

- **RelationType:** the meaning of the relationship. Use ClassString type if one of the values below is used, otherwise use “Class” template (see 4.6.4). According to the ISO AP242 Specification, where applicable, the following values shall be used:

RelationType	
'derivation'	The business object defines a deriving relationship where the related DocumentVersion is based on the relating DocumentVersion which is an earlier version of the same or of a different Document
'hierarchy'	The business object defines a hierarchical relationship where the related DocumentVersion is a subordinate version of the relating DocumentVersion
'sequence'	The business object defines a version sequence where the relating DocumentVersion is the preceding version of the related DocumentVersion that is the following version. For a given DocumentVersion there shall be at most one DocumentVersionRelationship of this relationType referring to this DocumentVersion as 'relating' and at most one DocumentVersionRelationship of this relationType referring as 'related'
'supplied document'	The business object defines a relationship between two DocumentVersion objects (both exchanged in the same XML file) representing the same object in different organizational contexts

- **Related:** the other object of [DocumentVersion](#) that is part of the relationship

Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

## 9 External Files

In the same way than in section 7 of the PDM Schema Usage Guide V4.3, the aim of this section is to map simple external references to a named file.

Referencing a specific element within the file (External Element References or EER) is not in scope of this document.

Depending on the business use case, this can be (reusing the terminology defined in the *CAx-IF External References Rec. Pracs.*):

- so-called 'Classic' or 'Basic' references:
  - an ISO STEP Part 21 (AP214, AP242) file containing the geometry of a part
  - an ISO JT file containing the light weight visualization of a part
  - any further standard geometry format (VDAFS, IGES, ...)
  - any proprietary geometry format (CATIA V4,V5,V6, ProEngineer, NX, ...)
- so-called 'Extended' or 'Nested' references:
  - another AP242 XML file (see „nested“ / „fully shattered“ section 0)

The referencing of further, non geometrical files (like MSWord, PDF, ...) as well as the referencing of Hardcopies is not in scope if this Recommended Practices document



## 9.1 Template “DigitalFile”

### Preprocessor Recommendations:

The referenced FormatProperties and CreationProperties can be reused within the XML file by all DigitalFiles to which they apply. Ditto for the Units referenced by the SizeProperties.

If the DigitalFiles are mapped as Documents, the Content, Creation and Format Properties may be applied to the DigitalDocumentDefinition instead of the DigitalFile, if these values apply to all files associated to the same Document.

If the external files are exchanged in the same directory than the assembly XML file (for example within a zip file), ExternalItem.Source can be left unset.

If the PDM representation is used, it is not recommended to reuse the same DigitalFile in several Documents via DigitalDocumentDefinition.Files.

### Postprocessor Recommendations:

Analogous to the mechanism described in the *CAx-IF External References Rec. Pracs.:*

- the name of the target file is to be expected in ExternalItem.Id
- if ExternalItem does not exist, evaluate DigitalFile.Id (defined as OPTIONAL in the schema, it becomes mandatory in this case)
- the general behavior for evaluating ContentProperty, CreationProperty, FormatProperty and SizeProperty shall be ‘only for information’, and shall not cause the postprocessor to stop processing if the given content, creation system, format or size is not supported by the postprocessor. The postprocessor shall anyway load and import the file correctly.

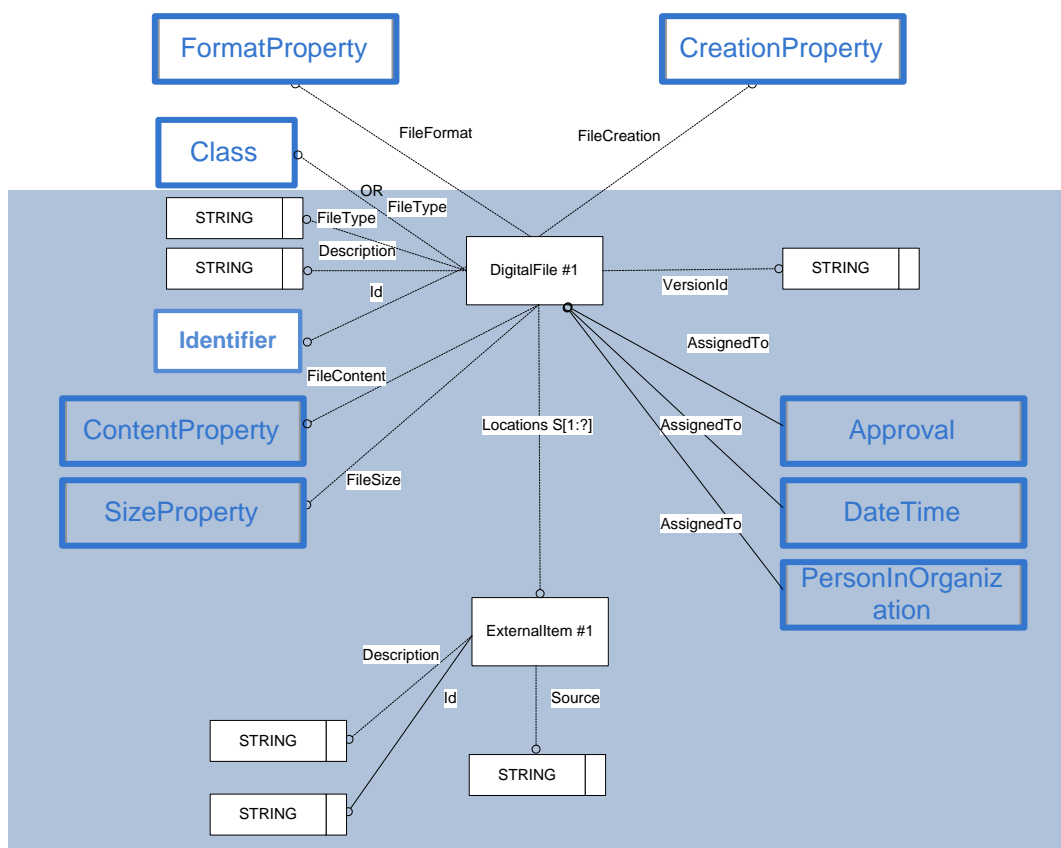


Figure 32: Template "DigitalFile"

## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<CreationProperty uid="fcp--V5">
  <CreatingInterface>COM/FOX V6.1.4</CreatingInterface>
  <CreatingSystem>CATIA V5 B25 SP0 HF0</CreatingSystem>
</CreationProperty>

<FormatProperty uid="ffp--STEP">
  <CharacterCode>
    <ClassString>ISO 8859-1</ClassString>
  </CharacterCode>
  <DataFormat>
    <ClassString>ISO 10303-242</ClassString>
  </DataFormat>
</FormatProperty>

<Unit uid="u--000000003">
  <Kind>
    <ClassString>SI system</ClassString>
  </Kind>
  <Name>
    <ClassString>byte</ClassString>
  </Name>
  <Prefix>
    <ClassString>kilo</ClassString>
  </Prefix>
</Unit>

<Classification uid="gtc--3">
  <Class>
    <ClassString>solid geometry</ClassString>
  </Class>
</Classification>

<File xsi:type="n0:DigitalFile" uid="df--000000001E60C660">
  <FileContent uid="fc--3">
    <DetailLevel>
      <CharacterString>production level</CharacterString>
    </DetailLevel>
    <GeometryTypes>
      <Classification uidRef="gtc--3"/>
    </GeometryTypes>
  </FileContent>
  <FileCreation uidRef="fcp--V5"/>
  <FileFormat uidRef="ffp--STEP"/>
  <FileSize uid="fsp--3">
    <FileSize uid="fspp--3" xsi:type="n0:NumericalValue">
      <Definition>
        <PropertyDefinitionString>file size
property</PropertyDefinitionString>
      </Definition>
      <Name>
        <CharacterString>file size</CharacterString>
      </Name>
      <Unit uidRef=" u--000000003"/>
      <ValueComponent>2.3</ValueComponent>
    </FileSize>
  </FileSize>
  <FileType>
```

```

    <ClassString>geometry</ClassString>
  </FileType>
  <Id>
    <Identifier uid="dfid--000000001E60C660--18" id="bolt.stp" idCon-
    tex"tRef="o--000000178"/>
  </Id>
  <Locations>
    <ExternalItem uid="idal--000000001E60C660--ei">
      <Id id="bolt.stp"/>
    </ExternalItem>
  </Locations>
</File>

```

Entity DigitalFile	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of ClassificationSelect
Description	OPTIONAL DescriptorSelect
FileContent	OPTIONAL ContentProperty
FileCreation	OPTIONAL CreationProperty
FileFormat	OPTIONAL FormatProperty
FileSize	OPTIONAL SizeProperty
FileType	OPTIONAL ClassSelect
Id	OPTIONAL IdentifierSelect
Locations	OPTIONAL SET[1:?] of ExternalItem
VersionId	OPTIONAL IdentifierSelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssign- ment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssign- ment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FileRelationship	OPTIONAL SET[1:?] of FileRelationship
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUs- ageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssign- ment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPer- sonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefini- tionAssignment

Entity DigitalFile	Attribute type
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssignment

Table 45: "DigitalFile" Attributes

### Attribute recommendations

- **Description:** the text or the set of texts that provides further information about the DigitalFile. The value of this attribute need not be specified. Use "Description" template.
- **FileContent:** the kind of geometric data stored into the DigitalFile. The value of this attribute need not be specified.
- **FileCreation:** details of the context of the creation of the DigitalFile => Reference to a CreationProperty. The value of this attribute need not be specified.
- **FileFormat:** data format of the DigitalFile => Reference to a FormatProperty. According to the CAX-IF recommendation the value of this attribute is mandatory, use "FormatProperty" Template (see 10.1) to fulfil it.
- **FileSize:** details of the size of the DigitalFile. The value of this attribute need not be specified.
- **FileType:** type of the DigitalFile => Use ClassString if one of the values below is used; otherwise use "Class" Template (see 4.6.4).

According to the ISO AP214 Specification of document\_type\_property, where applicable, the following values shall be used:

File Type	
'catalogue'	the File is the catalogue in which the associated object is listed
'manual'	the File is the handbook that is supplied for the associated object
'specification'	the File specifies the considerations that lead to the design finally chosen for the associated object
'geometry'	The file represents a shape model
'NC data'	The file represents numerical control data
'FE data'	The file represents finite element data
'sample data'	The file represents measured data
'process plan'	The file represents process planning data
'check plan'	The file represents quality control planning data

'drawing'	The file represents a technical drawing
'structured product data'	The document file contains product meta data and data related to product structure. This value shall be used for nested external references, when the referenced document relates to another BO Model XML file (see section 9.3)

*Comment: since this value list is not yet documented in the BO Model specification, an issue has been created in Bugzilla under #5111.*

- **Id:** the identifier for the DigitalFile. Although optional in the schema, this attribute shall be specified. Use “Identifier” template (see 4.6.6).
- **Locations:** location of the DigitalFile. If empty, the file shall be located in the same directory as the BO Model XML file referencing to it.
- **VersionId:** the identifier or set of identifiers for the version of the DigitalFile, the file version number. Use IdentifierString type. The value of this attribute need not be specified.
- **ApprovalAssignment:** to assign an Approval to the DigitalFile. Use the “Approval” template; see 4.6.12 for details. The value of this attribute need not be specified.
- **DateTimeAssignment:** to assign a DateTime to the DigitalFile. Use the “DateTime” template; see 4.6.11 for details. The value of this attribute need not be specified.
- **FileRelationship:** to relate to another DigitalFile. Use the “FileRelationship” template, see 0.
- **OrganizationOrPersonInOrganizationAssignment:** to assign an Organization or a PersonInOrganization to the DigitalFile. Use the “PersonInOrganization” template; see 4.6.14 for details. The value of this attribute need not be specified.
- **PropertyValueAssignment:** to assign a PropertyValue to the File. Use the “DocumentFileProperty” template; see 10.5 for details.
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

Entity ExternalItem	Attribute type
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
Source	OPTIONAL ExternalSourceSelect
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment

Table 46: "ExternalItem" Attributes

#### Attribute recommendations

- **Id:** the identifier for the ExternalItem (redundant to DigitalFile.Id). Use of IdentifierString type.

- **Description:** the text or the set of texts that provides further information about the ExternalItem. The value of this attribute need not be specified. Use of “Description” template.
- **Source:** the relative path to the file, or an absolute path (for example in the case of an URL). Use IdentifierString type. The following symbols shall be used in combination with directory names (if needed):
  - ‘/’ or ‘\’ to depict the directory structure
  - ‘.’ To depict the current directory (‘./’ and ‘.\’ are also allowed)
  - ‘..’ to move up to the next higher directory

The use of non-URL absolute paths (like [\\servername\.](#) or c:\... on Windows, or /... on Unix/Linux shall be agreed on project basis, since it prerequires that both sender and receiver have access to the same file system, which is not a typical use case).

- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

#### ***Preprocessor Recommendation:***

ExternalItem.Id shall not contain any path information. The path information shall be mapped only to ExternalItem.Source.

#### ***Postprocessor Recommendation:***

The path shall be extracted from ExternalItem.Source.

## **9.2 Template “FileRelationship”**

This relationship enables to relate two files.

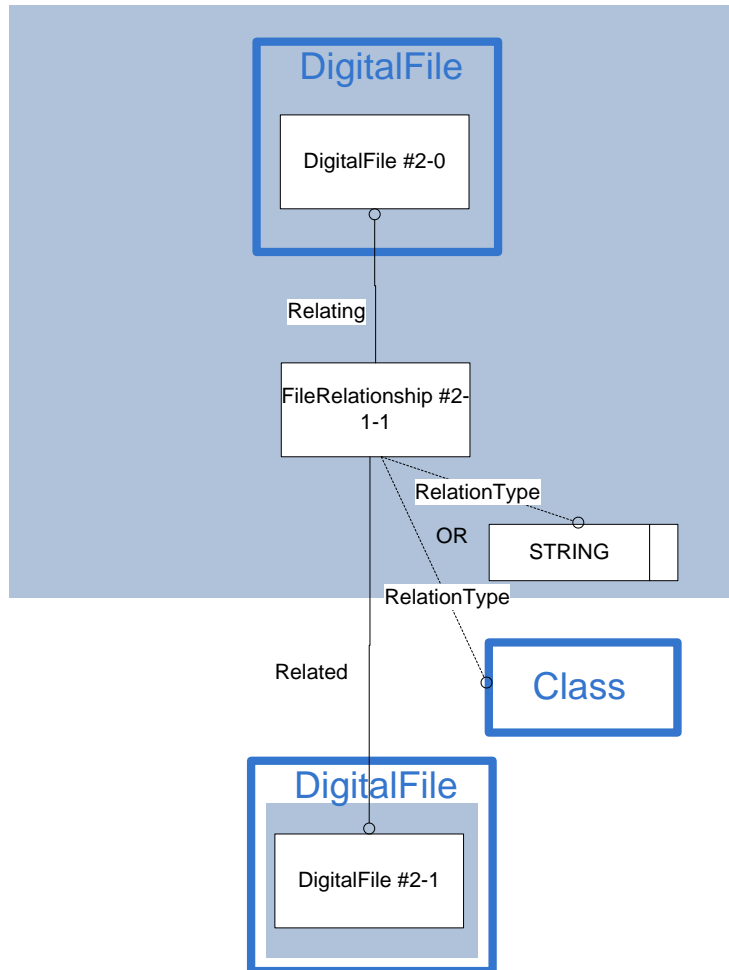


Figure 33: Template "FileRelationship"

### The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```

<File xsi:type="n0:DigitalFile" uid="df--000000001EAA8110">
...
  <Id>
    <Identifier uid="dfid--000000001EAA8110--14" id="l-bracket.stp" idContextRef="o--000000178"/>
  </Id>
...
  <FileRelationship uid="fr--1">
    <Related uidRef="df--000000001EAE2660"/>
    <RelationType>
      <ClassString>sequence</ClassString>
    </RelationType>
  </FileRelationship>
...
</File>
<File xsi:type="n0:DigitalFile" uid="df--000000001EAE2660">
...
  <Id>
    <Identifier uid="dfid--000000001EAE2660--18" id="bolt.stp" idContextRef="o--000000178"/>
  </Id>
...
</File>
  
```

Entity FileRelationship attributes	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
Related	DocumentDefinition
RelationType	ClassSelect
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment

Table 47: "FileRelationship" Attributes

**Attribute recommendations**

- **RelationType**: the meaning of the relationship. Use ClassString type if one of the values below is used, otherwise use "Class" template (see 4.6.4). According to the ISO AP242 Specification, where applicable, the following values shall be used:

RelationType	
'addition'	The business object specifies that the related file provides supplementary or collateral information with regard to the information provided by the relating file
'copy'	The business object defines a relationship where the related file is a copy of the relating file
'decomposition'	The business object defines a relationship where the related file is one of potentially more sub documents of the relating file
'derivation'	The business object defines a relationship where the related file is derived from the relating file
'peer'	The business object specifies that the related file provides required information with regard to that provided by the relating file. The peer document is essential for a complete understanding
'reference'	The business object defines a relationship where the related file is referenced from the relating
'sequence'	The business object defines a logical sequence where the related file comes after the relating file
'substitution'	The business object defines a relationship where the related file replaces the relating file



'translation'	The FileRelationship specifies that the related file is generated through a translation process from the relating file
---------------	--

- **Related:** the other object of [File](#) that is part of the relationship
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

### 9.3 File Structure (monolithic/nested)

Analogous to the mechanism described in the CAx-IF Recommended Practices for External References, it is possible to exchange the multiple level structure either in one XML file (so-called monolithic structure), or to split each part node into a dedicated XML file (so-called nested structure).

The so-called '*specified Reference*' Mechanism is taken over from the Chapter 5.5 and Annex D of VDA-Empfehlung 4956 "Product Data Exchange - Part 1: Assembly Data Exchange" 1.1 from Nov 2002 (see Annex C).

**Note:** this mechanism is not meant to be used for incremental data exchange: a nested structure shall be exchanged with all its components. It may also be used for the purpose of long term archiving, since each product structure node (and its history) can be archived independently from the other product structure nodes. Incremental data exchange is supported by using the so-called 'Reference' Mechanism of the above VDA recommendation. This mechanism is not in scope of this version of the Recommended Practices.

Special attention is needed in the case that the set of information contained in the XML files concerning the component parts goes beyond the minimum set of entities and attributes needed to define the external reference – for instance when user defined attributes are given as well. In the structure as illustrated in Figure 34 below, information concerning the Nut part would be stored redundantly in two XML files: the one for the Nut-Bolt Assembly and the one for the Rod Assembly. This opens the door for inconsistencies, especially since the information for the part is only complete when access the assembly file(s) referencing it.

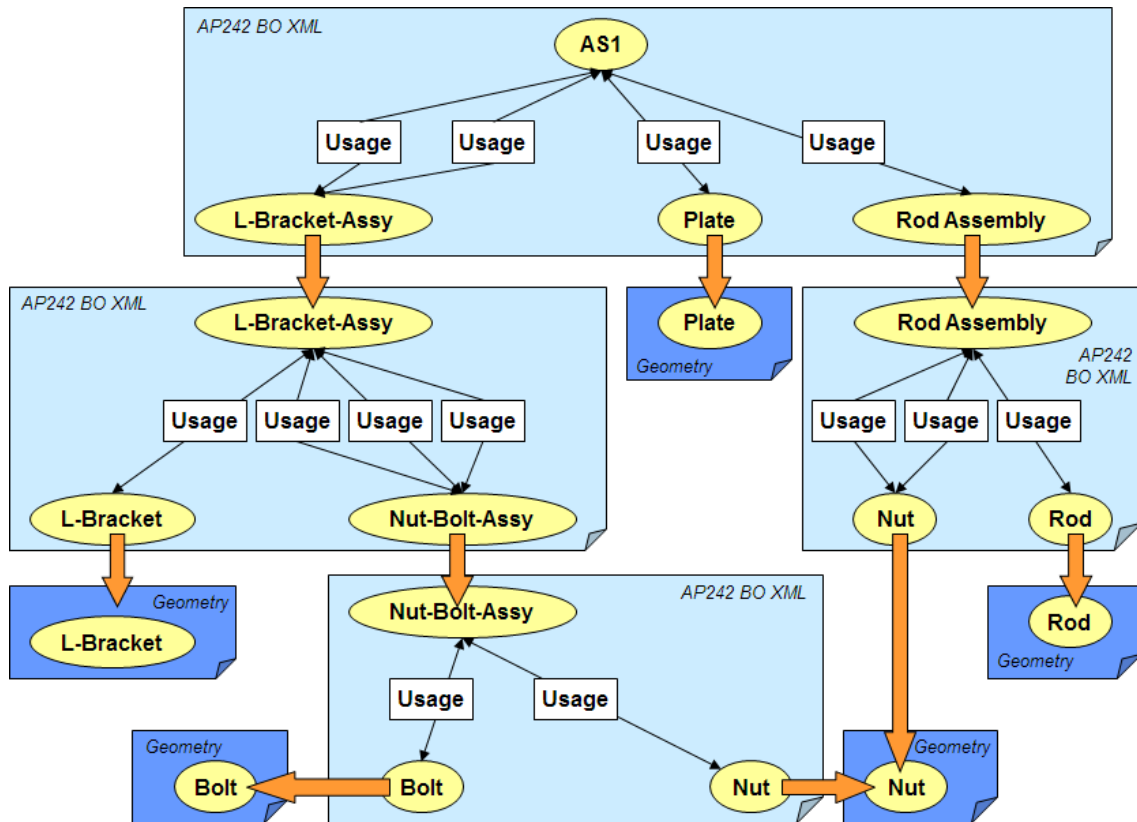


Figure 34: Example for Nested Structure

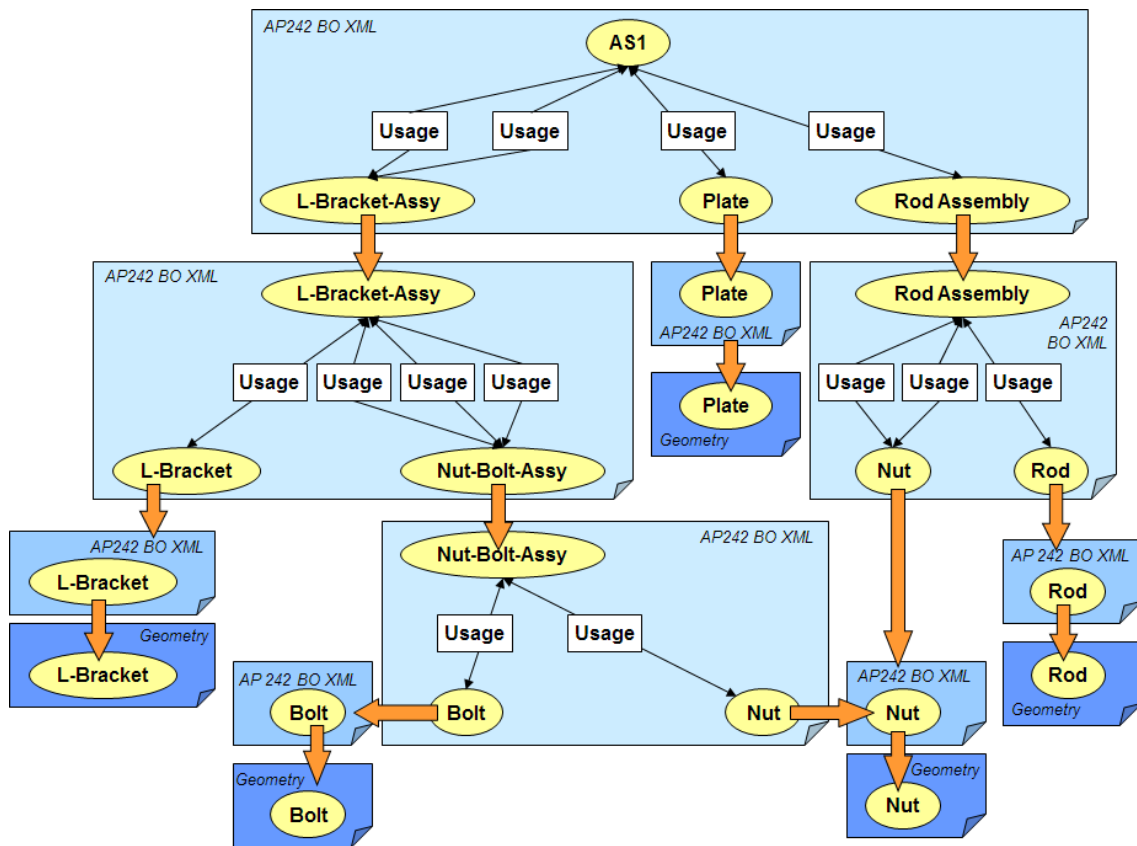


Figure 35: Example for Nested Structure with additional part-level XML files

Hence it is recommended in this case to create an additional XML file for every part file, which carries all PDM-relevant data for this part in one place. The superordinate assembly XML files will reference the part's XML file, which will in turn reference the actual part geometry file. Figure 35 illustrates the extended file structure.

This mechanism also provides the correct input for Long Term Archiving.

A referenced component is mapped in the XML File of its assembly(s) by:

1. Mapping a minimum set of entities and attributes (subset of those mapped in the component XML file or in the monolithic mapping):
  - Part.id with idRoleRef, idContextRef and PartType,
  - PartVersion.id (or '/ANY' if the right version get computed at runtime by the PDM application),
  - PartView.id (only if multiple views are handled),
2. Mapping a reference to the component file (geometry file or intermediate XML file) between the PartView and the DigitalFile
3. In case the full positioning representation defined in chapter 7.3.2 is used: mapping a reference to the component file (geometry file or intermediate XML file) between the ExternalGeometricModel and the DigitalFile, and
4. Mapping a dedicated Classification
5. SingleOccurrences, and all their properties, shall be defined in the superordinate assembly file, rather than in the part file. Doing so, the part file is independent from where and how often it is built into product structures.

*Remark:* if a SingleOccurrence is defined in the structure, but not used anywhere, it will not be mapped into any superordinate assembly file and thus will get lost.

The following mappings still need to be specified. This will done be as testing of nested product structures progresses in the PDM-IF. The definitions will be updated accordingly in future releases of this document.

1. SpecifiedOccurrences: the XML-embedded SpecifiedOccurrences via the attribute 'Definition' shall also be defined in the superordinate assembly file. Management of references via the attribute 'UpperUsage' needs to be defined. In addition, any SpecifiedOccurrence that is defined in the structure, but not used anywhere, will get lost entirely due to the splitting of the information.
2. Multiple PartViews: shall all Views of a PartVersion always be exchanged into one single file, or could each PartView be exchanged in a separate file? Both approaches are possible. In the latter case, a view identifier must be added to the file name. Requirements and practicality need to be discussed.
3. Documents: Since a PDM document can be shared by multiple parts, shall it be exchanged as a separate file? This scenario is similar to a part file being shared by multiple sub-assemblies as described above, but could lead to an exponential increase of references. Requirements and practicality need to be discussed.
4. Kinematics: This will be described in the upcoming Recommended Practices for AP242 BO Model XML Kinematics. Definitions applicable to the product structure in general made in this context will be reflected here. This is currently being discussed in the CAx-IF and JT-IF.
5. ProductConfiguration: This will be described in the upcoming Recommended Practices for AP242 BO Model XML Configuration Management. Definitions applicable to the product structure in general made in this context will be reflected here. Configuration management in context of nested assemblies has not yet been discussed.

### ***Preprocessor Recommendations:***

The part-level XML file describes the component part with all its master data. This master data should not be included in the assembly XML files placed above to avoid inconsistencies. Whether the part-level XML file is needed depends on the use case: in the area of Long Term Archiving, each part needs to be fully defined on its own, with its master data and geometry, which requires the additional XML file. For the exchange of an assembly structure with plain references to the component parts and no additional PDM information, it is optional.

To follow a reference from one XML file to another, the uniqueness of the parts is not ensured via the uids of the XML elements in the different XML files (the same part version could have a different uid in each XML file where it is defined or referenced), but via the **Identifier** elements.

Since ExternalGeometricModel that references to the XML shall have the same value in Id than the DigitalFile.Id and ExternalItem.Id (see recommendation in chapter 9.3.2), it doesn't reference the geometry of the component, but the XML file where the component is described (nut.stpx).

For the purpose of the typical CAx-IF data exchange use case of these recommended practices, the use of '/ANY' is not recommended.

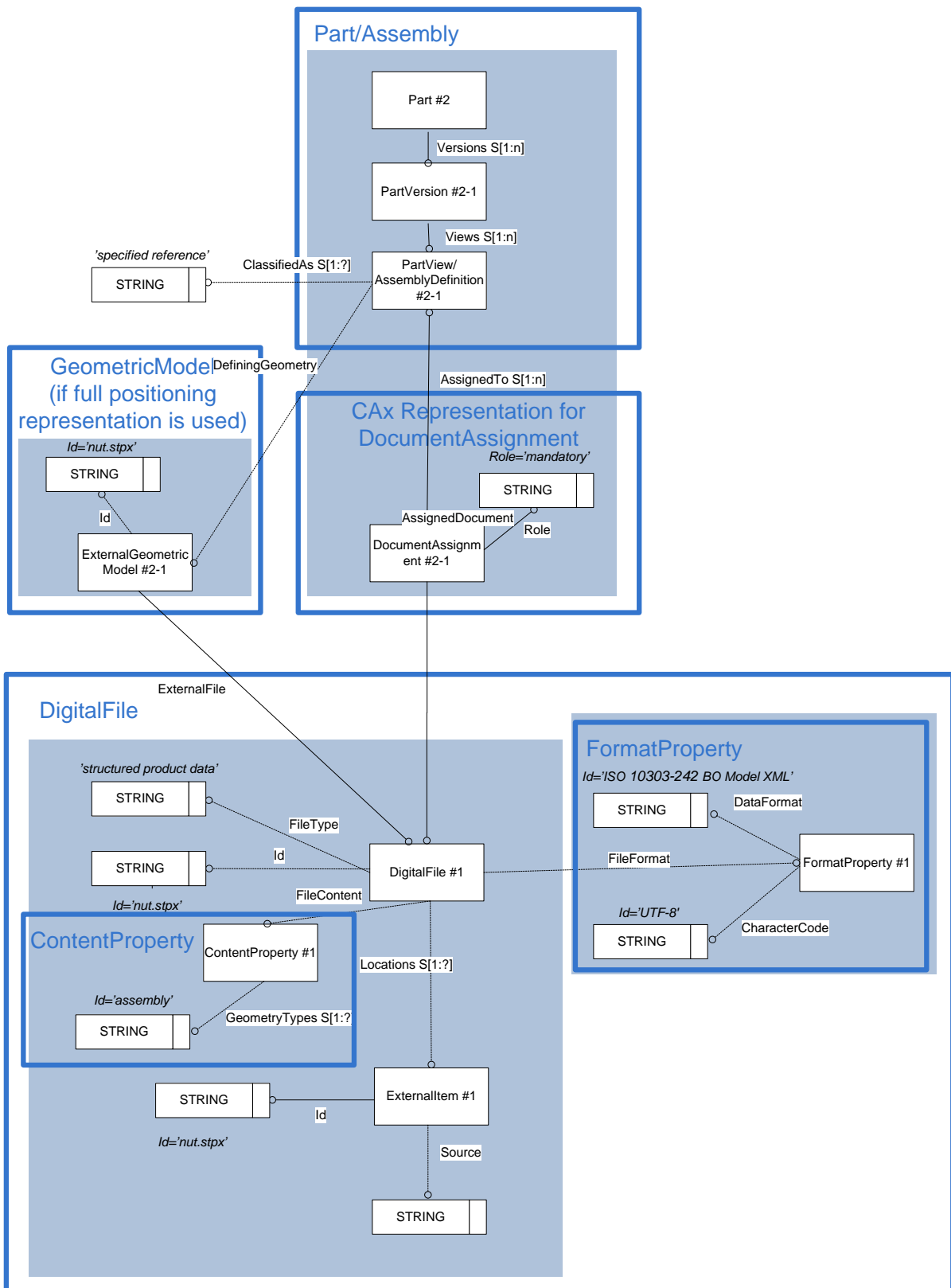


Figure 36: Element Structure for Nested XML File

## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<Classification uid="vda--SpecifiedReference">
  <Class>
    <ClassString>specified reference</ClassString>
  </Class>
</Classification>
<Part uid=" p--000000001E720B30">
  <Id>
    <Identifier uid="pid--000000001E720B30--id7" id="nut" idContextRef="o--
000000178"/>
  </Id>
  ...
  <Versions>
    <PartVersion uid="pv--000000001E720B30--id7">
      <Id id="/NULL"/>
      <Views>
        <PartView uid="pvv--000000001E720B30--id7">
          <ClassifiedAs>
            <Classification uidRef="vda--SpecifiedReference"/>
          </ClassifiedAs>
          ...
          <DefiningGeometry uidRef="egm--000000001E720B30"/>
          <DocumentAssignment xsi:type="n0:DocumentAssignment" uid="da--
000000001A304330--id7">
            <AssignedDocument uidRef="df--000000001E720B30"/>
            <Role>
              <ClassString>mandatory</ClassString>
            </Role>
          </DocumentAssignment>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
<GeometricRepresentation uid="egm--000000001E720B30"
xsi:type="n0:ExternalGeometricModel">
  ...
  <Id id="nut.stpx"/>
  ...
  <ExternalFile uidRef="df--000000001E720B30"/>
</GeometricRepresentation>
<FormatProperty uid="ffp--AP242BOMODEL">
  <CharacterCode>
    <ClassString>UTF-8</ClassString>
  </CharacterCode>
  <DataFormat>
    <ClassString>ISO 10303-242 BO Model XML</ClassString>
  </DataFormat>
</FormatProperty>
<Classification uid="gtc--2">
  <Class>
    <ClassString>assembly</ClassString>
  </Class>
</Classification>
```

```
<CreationProperty uid="fcp--V5">
  <CreatingInterface>COM/FOX V6.1.4</CreatingInterface>
  <CreatingSystem>CATIA V5 B25 SP0 HF0</CreatingSystem>
</CreationProperty>
<File xsi:type="n0:DigitalFile" uid="df--000000001E720B30">
  <FileContent uid="fc--4">
    <GeometryTypes>
      <Classification uidRef="gtc--2"/>
    </GeometryTypes>
  </FileContent>
  <FileCreation uidRef="fcp--V5"/>
  <FileFormat uidRef="ffp--AP242BOMODEL"/>
  <FileType>
    <ClassString>structured product data</ClassString>
  </FileType>
  <Id>
    <Identifier uid="dfid--000000001E720B30--19" id="nut.stpx"
idContextRef="o--000000178"/>
  </Id>
  <Locations>
    <ExternalItem uid="idal--000000001E720B30--ei">
      <Id id="nut.stpx"/>
    </ExternalItem>
  </Locations>
</File>
```

Unlike in section 11.1 and 11.2 (CAx vs. PDM representation of DocumentAssignment, the references to the intermediate XML files shall not be mapped as a managed document, since these are not stored as managed documents in the PDM system, but are only created during the export process.

**Preprocessor Recommendations:**

- If multiple PartVersions of a Part are provided, each PartVersion shall be mapped to a separate file => the Part.Id and the PartVersion.Id shall be used to name the files.
- DigitalFile.VersionId shall not be set
- If all the nested stpx files and the document files are exchanged within an overall archive, the compression algorithm used to build the archive shall be compliant to the “Recommended Practices for STEP File Compression” from the CAx IF (see [Annex C](#)).
- If the product structure mapped in the nested stpx file has one single top level assembly node, the name of the overall archive shall be the same as the stpx file of the top-level assembly node. This simplifies the analysis for the import tool which doesn’t have to go through all stpx files to find the top node.
- The use of compression for single stpx files (exchanged as stpxZ files) is optional. If used, it shall comply to the “Recommended Practices for STEP File Compression” from the CAx IF.

**Postprocessor Recommendations:**

- If the data is exchanged within an overall archive, and if there is one stpx(Z) file within the archive having the same name as the archive itself, this stpx(Z) can be assumed to be the only top-level assembly node of the data. If not, the top-level assembly node(s) have to be found out through scanning all the stpx(Z) files.

## 10 Document and File Properties

### 10.1 Template “FormatProperty”

The `FormatProperty` entity is the specification of characteristics of a File or of a DocumentDefinition that specify the format of the object.

**The Instance Model: AP242 BO Model XML entities and attributes**

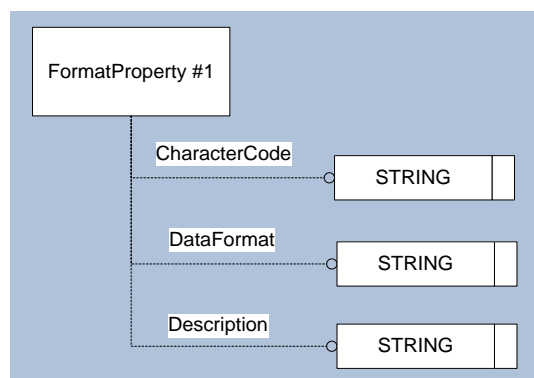


Figure 37: Template “FormatProperty”

Entity FormatProperty	Attribute type
CharacterCode	OPTIONAL ClassSelect
DataFormat	OPTIONAL ClassSelect



Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
SizeFormat	OPTIONAL RectangularSize
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship

Table 48: “FormatProperty” Attributes

**Attribute recommendations**

- **CharacterCode:** the computer application used to create the DigitalFile. The value of this attribute need not be specified. Use ClassString type if one of the values below is used, otherwise use “Class” template (see 4.6.4). According to the ISO AP242 Specification, where applicable, the following values shall be used:

CharacterCode	Explanation
'binary'	The document contains data in binary format
'IEC 61286'	The coded character set used to encode the document data according to IEC 61286
'ISO 646'	The coded character set used to encode the document data according to ISO 646; NOTE: The character set in ISO 646 is identical to the character set commonly known as ASCII
'ISO 6937'	The coded character set used to encode the document data is according to ISO/IEC 6937
'ISO 8859-1'	The coded character set used to encode the document data according to ISO 8859-1; NOTE: The character set in ISO 8859-1 is identical to the character set commonly known as LATIN-1. This is the default for STEP Part 21 files.
'compressed ISO 8859-1'	The coded character set used to encode the document data according to ISO 8859-1, where the file was compressed using the PKZip 2.04g format <sup>1</sup>
'UTF-8'	The coded character set used to encode the document data according to to UTF-8. NOTE: The character set in UTF-8 is the default encoding for XML files, including STEP BO Model XML files.
'compressed UTF-8'	The coded character set used to encode the document data according to UTF-8, where the file was compressed using the PKZip 2.04g format <sup>1</sup>
'ISO 10646'	The coded character set used to encode the document data according to ISO/IEC 10646.

<sup>1</sup> **Note:** If the respective compressed files are STEP files per the Recommended Practices for STEP File compression (see Annex C), the file reference shall always point to the uncompressed file. Thus, the FormatProperty should also state the original format, and not the compressed version. The PKZip format definition is available at <https://www.pkware.com/documents/APPNOTE/APPNOTE-6.2.0.txt>.

- **DataFormat:** the convention that was used to structure the information in the characterized object. The value of this attribute need not be specified. Use ClassString type if one of the values below is used, otherwise use “Class” template (see 4.6.4). According to the ISO AP242 Specification, where applicable, the following values shall be used:

DataFormat	Explanation
'DXF'	The document contains data in Drawing Exchange File format
'IGES'	The document contains data in Initial Graphics Exchange Specification format
'ISO 10303-203', 'STEP AP203'	Eventually followed by the release number (E2, E3): The document contains data in ISO 10303-203 Part21 format
'ISO 10303-214', 'STEP AP214'	Eventually followed by the release number (E2,E3): The document contains data in ISO 10303-214 Part21 format
'STEP AP214 CC06'	The document contains data in ISO 10303-214 Part21 format according to Conformance Class 06 (product structure only, the file contains no geometry, but references to external geometry files)
'TIFF CCITT GR4'	The document contains data in TIFF CCITT GR4 format
'VDAFS'	The document contains data in VDAFS format
'VOXEL'	The document contains data in VOXEL format
'CAD'	The document contains native CAD data. When used, the Document Creation Property (see 10.3) shall be used to convey specifics on the originating CAD system

### **Preprocessor Recommendation:**

In case of a non-native proprietary format (for example CATIA CGR), the DataFormat shall refer to a Class and ExternalClassificationSystem (see “Class” template 4.6.4).

Additionally, the following values are recommended, where applicable:

DataFormat	Explanation
'ISO 10303-242', 'STEP AP242'	The document contains data in ISO 10303-242 Part 21 format
'ISO 10303-242 BO Model XML'	The document contains data in ISO 10303-242 XML format
'ISO 14306 JT'	The document contains data in ISO 14306 JT format
MS Word	Microsoft Word
MS_Excel	Microsoft Excel
MS Powerpoint	Microsoft Powerpoint

DataFormat	Explanation
MS Outlook	Microsoft Outlook
MS Access	Microsoft Access
MS Project	Microsoft Project
full text	ASCII text
GIF	Graphics Interchange Format
HPGL	Hewlett-Packard Graphics Language
HTML	ISO/IEC 15445 HyperText Markup Language
JPEG	ISO/IEC 10918 JPEG
PDF	ISO 32000-1 Portable Document Format
TIFF	ISO 12639 Tagged Image File Format

- **Description:** the text or the set of texts that provides further information about the FormatProperty. The value of this attribute need not be specified. Use “Description” template (see 4.6.7).
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

### The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<FormatProperty uid="ffp--AP242BOMODEL">
  <CharacterCode>
    <ClassString>UTF-8</ClassString>
  </CharacterCode>
  <DataFormat>
    <ClassString>ISO 10303-242 BO Model XML</ClassString>
  </DataFormat>
</FormatProperty>
```

#### 10.2 Template “ContentProperty”

The ContentProperty entity is the specification of characteristics precisising the content of a File or of a DocumentDefinition.

### The Instance Model: AP242 BO Model XML entities and attributes

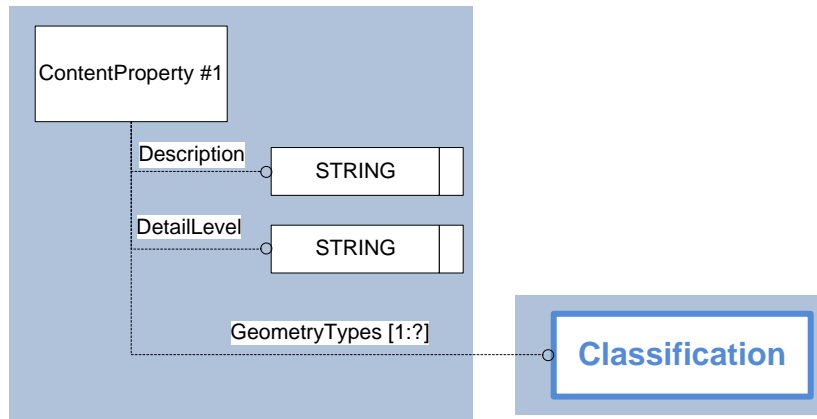


Figure 38: Template “ContentProperty”

Entity ContentProperty	Attribute type
Description	OPTIONAL DescriptorSelect
DetailLevel	OPTIONAL DescriptorSelect
GeometryTypes	OPTIONAL SET[1:?] of ClassificationSelect
Languages	OPTIONAL SET[1:?] of Language
RealWorldScale	OPTIONAL NumericalValue

Table 49: “ContentProperty” Attributes

**Attribute recommendations**

- **Description:** the text or the set of texts that provides further information about the ContentProperty. The value of this attribute need not be specified. Use “Description” template (see 4.6.7).
- **DetailLevel:** the level of detail that the DigitalFile provides. The value of this attribute need not be specified. Use “Description” template (see 4.6.7). The following recommended values for this attribute are derived from MIL-STD-31000A (see reference in Annex C):

DetailLevel	Explanation
‘conceptual level’	Conceptual level data relates to elements defining design concepts in graphic form, and includes appropriate information required for analysis and evaluation of those concepts. The data may consist of simple sketches/models, artist renderings and/or basic textual data
‘developmental level’	Developmental level data relates to elements providing sufficient data to support the analysis of a specific design approach, the fabrication of prototype material for test or experimentation, and limited production by the original design organization or with assistance from the original design organization
‘production level’	Production level data relates to elements providing the design, engineering, manufacturing, inspection, packaging and quality assurance provisions information enabling the procurement or manufacture of an item. The level of detail shall be sufficient for a competent manufacturer to produce an item, which duplicates the physical, interface, and functional

	characteristics of the original product, without additional design engineering effort or recourse to the current design activity. Production data shall reflect the approved, tested, and accepted configuration of the defined delivered item
--	--

- **GeometryTypes:** details of the context of the creation of the DigitalFile. The value of this attribute need not be specified. As far as applicable, one or several of the values given below can be used. Use ClassString type if one of the values below is used, otherwise use “Classification” template (see 4.6.5):

**If the DigitalFile contains the geometry of a part:**

GeometryTypes	Explanation
'wireframe geometry'	The document contains a three-dimensional model with precise definitions of wireframes or independent curves, meaning these curves are not edge curves of higher topological elements
'surface geometry'	The document contains a three-dimensional shape with with precise definitions of independent surfaces, meaning these surfaces are not faces of solids
'solid geometry'	The document contains a three-dimensional shape model in advanced boundary representation
'tessellated geometry'	The document contains a simplified shape representation that may consist of curves, surfaces and/or solids
'2D drawing'	The document contains a technical drawing. The drawing may have been derived from a 3D model
'PMI presentation'	The document contains Product and Manufacturing Information in a human-readable form, e.g. as 3D annotations
'PMI representation'	The document contains Product and Manufacturing Information in a semantic, machine-interpretable form
'implicit composite'	The document contains the implicit definition of a composite part as zero-thickness faces (plies) with boundaries and stacking order (laminar table)
'explicit composite'	The document contains the explicit representation of a solid composite part. This value is typically used together with 'tessellated geometry'.

**If the DigitalFile contains another AP242 BO Model XML file (see „nested“ / „fully shattered“ section below):**

GeometryTypes	Explanation
'assembly'	The document contains an assembly structure with reference to the assembled components and their transformation matrices
'assembly with mating elements'	The document contains an assembly structure including the mating components only, such as screws or rivets, with exact positioning information. This assembly representation is intended to be overlaid with the assembly structure for the main components

- Note that the AP242 Standard gives a different list of recommended values for DetailLevel and GeometryTypes. These have, however, been taken over from earlier versions of AP214 and AP203 without further review and are deemed outdated. Hence it was agreed by the CAx-IF to include an updated list that better describes the characteristics of information typically exchanged. This list is currently being reviewed, and after final agreement, an AP242 maintenance issue will be created to update the textual definition in the standard accordingly.

## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<Classification uid="gtc--1">
  <Class>
    <ClassString>surface geometry</ClassString>
  </Class>
</Classification>
<Classification uid="gtc--3">
  <Class>
    <ClassString>solid geometry</ClassString>
  </Class>
</Classification>
<File xsi:type="n0:DigitalFile" uid="df-000000001E720B30">
  <FileContent uid="fc-1">
    <DetailLevel>
      <CharacterString>development level</CharacterString>
    </DetailLevel>
    <GeometryTypes>
      <Classification uidRef="gtc--1"/>
      <Classification uidRef="gtc--3"/>
    </GeometryTypes>
  </FileContent>
  ...
</File>
```

### 10.3 Template “CreationProperty”

The CreationProperty entity is the specification of characteristics of a File or of a DocumentDefinition. It specifies the context of the creation of the object.

#### **Postprocessor Recommendation:**

A CreationProperty shall be created if the file extension is not unique (for example ‘.prt’ may be a Creo file or an NX file)

#### **Postprocessor Recommendation:**

If no CreationProperty is available, the file extension within File.Id or ExternalItem.Id shall be evaluated.

## The Instance Model: AP242 BO Model XML entities and attributes

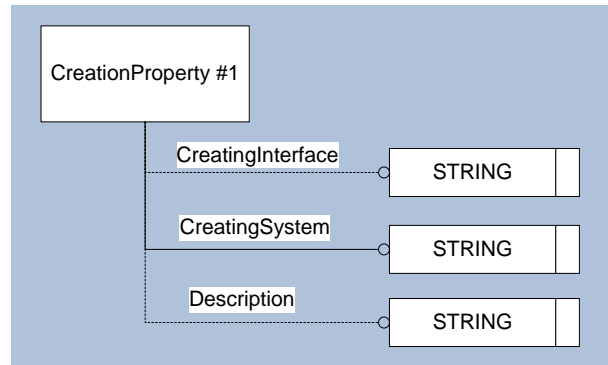


Figure 39: Template “CreationProperty”

Entity CreationProperty	Attribute type
CreatingInterface	OPTIONAL STRING
CreatingSystem	STRING
Description	OPTIONAL DescriptorSelect
OperatingSystem	OPTIONAL STRING

Table 50: “CreationProperty” Attributes

**Attribute recommendations**

- **CreatingInterface:** the computer application used to create the DigitalFile. The value of this attribute need not be specified. If the same preprocessor is used to create the AP242 XML data and the referenced file, CreatingInterface will be redundant to Header.PreprocessorVersion. In case of a native format, this attribute shall be left empty.
- **CreatingSystem:** the computer application or the machine that was used to generate the DocumentDefinition or File (for example CATIA V5R19).

Remark: if the data is mapped to AP242 XML and the referenced file are coming from the same system, CreatingSystem will be redundant to Header.OriginatingSystem.

- **Description:** the text or the set of texts that provides further information about the CreationProperty. The value of this attribute need not be specified. Use “Description” template (see 4.6.7).
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)**

```
<CreationProperty uid="fcp-V5">
  <CreatingInterface>COM/FOX V6.1.4</CreatingInterface>
  <CreatingSystem>CATIA V5 B25 SP0 HF0</CreatingSystem>
</CreationProperty>
```

**10.4 Template “SizeProperty”**

The SizeProperty entity is the specification of the size of a File or of a DocumentDefinition that specify the format of the object.

**The Instance Model: AP242 BO Model XML entities and attributes**

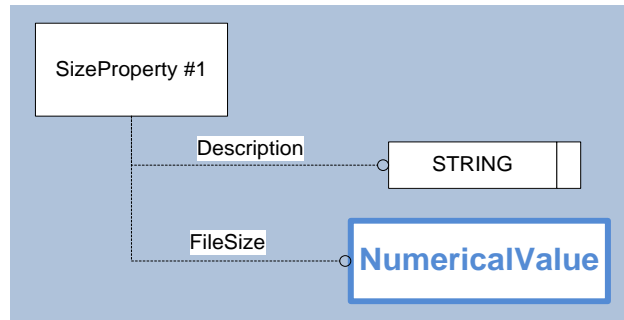


Figure 40: Template “SizeProperty”

Entity SizeProperty	Attribute type
Description	OPTIONAL DescriptorSelect
FileSize	OPTIONAL ValueWithUnit
PageCount	OPTIONAL ValueWithUnit

Table 51: “SizeProperty” Attributes

### Attribute recommendations

- **Description:** the text or the set of texts that provides further information about the SizeProperty. The value of this attribute need not be specified. Use “Description” template (see 4.6.7).
- **FileSize:** the size of a digitally stored document. The value of this attribute need not be specified. Use “NumericalValue” template (see 4.6.9).

Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

### The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<Unit uid="u--000000003">
  <Kind>
    <ClassString>SI system</ClassString>
  </Kind>
  <Name>
    <ClassString>byte</ClassString>
  </Name>
  <Prefix>
    <ClassString>kilo</ClassString>
  </Prefix>
</Unit>
<File xsi:type="n0:DigitalFile" uid="df--000000001E720B30">
...
  <FileSize uid="fsp--1">
    <FileSize uid="fspp--1" xsi:type="n0:NumericalValue">
      <Definition>
        <PropertyDefinitionString>file size
property</PropertyDefinitionString>
      </Definition>
      <Name>
        <CharacterString>file size</CharacterString>
      </Name>
      <Unit uidRef=" u--000000003"/>
      <ValueComponent>2.3</ValueComponent>
    </FileSize>
  </FileSize>
</File>
```



</FileSize>

...

### 10.5 Template “DocumentFileProperty”

In the same way that in section 9 and 9.7 of the PDM Schema Usage Guide V4.3, the aim of this section is to specify how to attach a property to a document or a file.

The PropertyValueAssignment entity represents the attachment of the DocumentDefinition or File to the value represented via the “NumericalValue” (see 4.6.9) or “StringValue” templates (see 4.6.10).

#### The Instance Model: AP242 BO Model XML entities and attributes

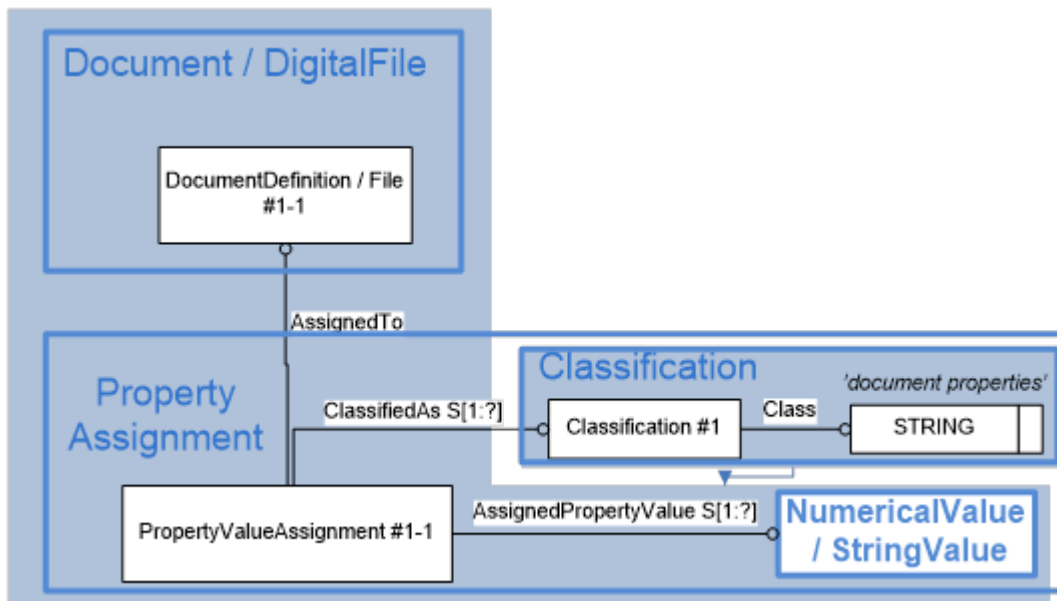


Figure 41: Template "DocumentFileProperty"

List of attributes and recommendation are similar to the PropertyAssignment template defined in chapter 6.2.

**Preprocessor Recommendations:** It is recommended that all the document properties use the same PropertyValueAssignment. The value “document properties” shall be used for ClassString in attribute PropertyValueAssignment.classifiedAs.class.

**Postprocessor Recommendations:** None specified.

**Related Entities:** There are no specific related entities.

#### The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<File xsi:type="n1:DigitalFile" uid="df--000000001E5A89F0">
...
  <PropertyValueAssignment uid="pva--000000001E5A89F0-id1">
    <AssignedPropertyValues>
      <PropertyValue uid="pv--000000001E5A89F0-id1"
xsi:type="n1:StringValue">
        <Definition>
          <PropertyDefinition uidRef="pd--000000320"/>
        </Definition>
        <Name>
```

```

    <CharacterString>checksum</CharacterString>
  </Name>
  <ValueComponent>
    <CharacterString>1582054665</CharacterString>
  </ValueComponent>
</PropertyValue>
</AssignedPropertyValues>
<ClassifiedAs>
  <Classification uidRef="cla--pp"/>
</ClassifiedAs>
</PropertyValueAssignment>
</File>

<PropertyDefinition uid="pd--000000320">
  <Id id="quality property"/>
  <PropertyType>
    <ClassString>system property</ClassString>
  </PropertyType>
</PropertyDefinition>

  <Classification uid="cla--pp">
    <Class>
      <ClassString>document properties</ClassString>
    </Class>
  </Classification>

```

## 11 Document and File Association to Product Data

The scope of this section corresponds to section 10 of the PDM Schema Usage Guide V4.3.

Two alternatives are described in this chapter. Both are based on the entity DocumentAssignment (see below) and ExternalGeometricModel (see chapter 6.1).

Entity DocumentAssignment	Attribute type
AssignedDocument	AssignedDocumentSelect
ClassifiedAs	OPTIONAL SET[1:?] of ClassificationSelect
Description	OPTIONAL DescriptorSelect
DocumentPortion	OPTIONAL MultiLingualStringSelect
Id	OPTIONAL IdentifierSelect
Role	ClassSelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
AssignmentObjectRelationship	OPTIONAL SET[1:?] of AssignmentObjectRelationship
ConditionAssignment	OPTIONAL SET[1:?] of ConditionAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment

EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 52: "DocumentAssignment" Attributes

**Attribute recommendations**

- **AssignedDocument:** the assigned DocumentVersion or DigitalFile
- **Description:** the text or the set of texts that provides further information about the DocumentAssignment. The value of this attribute need not be specified. Use "Description" template.
- **Id:** the identifier for the DocumentAssignment. The value of this attribute need not be specified. Use IdentifierString type.
- **Role:** the meaning of the assignment. Use ClassString if one of the values below is used, otherwise use "Class" template (see 4.6.4). Where applicable, the following values shall be used:

Role	Explanation
'additional information'	The assigned document provides information that is relevant for the associated object, but is not a description of the associated object itself
'behaviour'	The assigned document specifies information about the behaviour of the associated object
'description'	The assigned document provides textual information for the associated object itself
'informative'	The assigned document may or may not be considered
'mandatory'	The associated object shall conform to the content of the assigned document. <b>This value shall be used for the file that contains the geometry of the part.</b>
'mathematical description'	The assigned document specifies the associated object by providing the algorithmic specification of its behaviour
'dimensioning standard'	The assigned document specifies the dimensioning standard

- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

As opposed to a managed 'Document as Product', an external file is not managed by the system - there is no capability for managed revision control or any document representation definitions for an external file.

If a file is under configuration control, it should be represented as a constituent of a document definition view/representation according to 'Document as Product'. In this case, it is actually the managed document that is under direct configuration control; the file is, in this way, indirectly under configuration control. A change to the file results in a change to the managed document (i.e., a new version) - the changed file would be mapped as a constituent of a view/representation definition of the new document version. A simple external reference alone is not configuration controlled; it is just an external file reference to product data.

### ***11.1 Template "CAx Representation for DocumentAssignment"***

This section is relevant when the files are not under configuration control.

In this case, the DocumentAssignment shall refer directly to the DigitalFile.

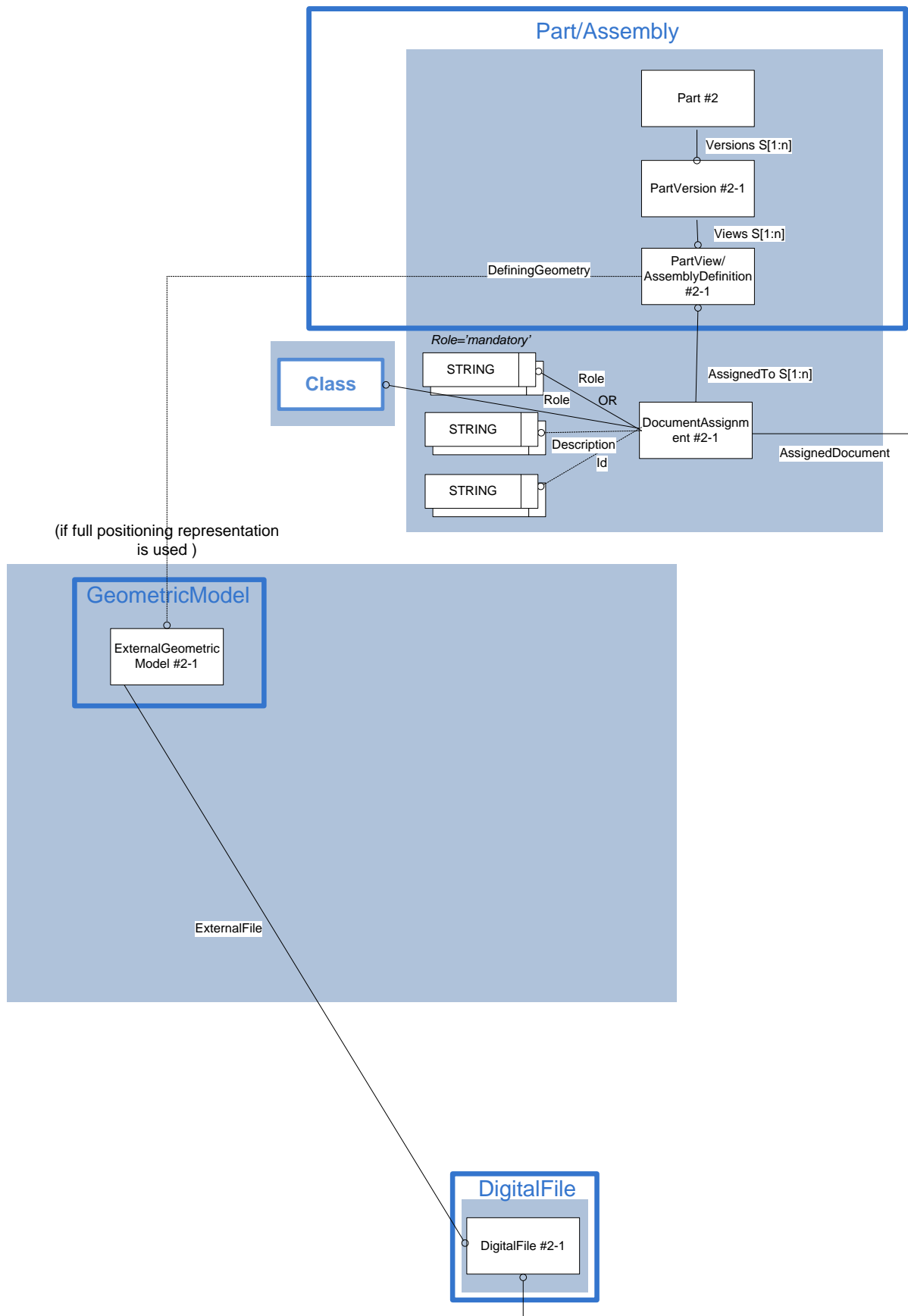


Figure 42: Template "CAx Representation for DocumentAssignment"

## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<Part uid="p--000000001E720B30">
  <Id>
    <Identifier uid="pid--000000001E720B30--id7" id="nut" idContextRef="o--000000178"/>
  </Id>
  ...
  <Versions>
    <PartVersion uid="pv--000000001E720B30--id7">
      <Id id="/NULL"/>
      <Views>
        <PartView xsiuid="pvv--000000001E720B30--id7">
          <DefiningGeometry uidRef="egm--000000001E720B30"/>
          ...
          <DocumentAssignment xsi:type="n0:DocumentAssignment" uid="da--000000001E720B30--id7">
            <AssignedDocument uidRef="df--000000001E720B30"/>
            <Role>
              <ClassString>mandatory</ClassString>
            </Role>
          </DocumentAssignment>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
<GeometricRepresentation uid="egm--000000001E720B30"
xsi:type="n0:ExternalGeometricModel">
  <ContextOfItems uidRef="ccs--origin-2"/>
  <Id id="nut.stp"/>
  ...
  <ExternalFile uidRef="df--000000001E720B30"/>
</GeometricRepresentation>
<File xsi:type="n0:DigitalFile" uid="df--000000001E720B30">
  ...
  <Id>
    <Identifier uid="dfid--000000001E720B30--19" id="nut.stp"
idContextRef="o--000000178"/>
  </Id>
  ...
</File>
```

### **Preprocessor Recommendations:**

Special cases: mapped compliantly to the recommendations of the PDM Usage Guide

#### 1. Model splitting:

In case the geometry of a part is splitted into multiple DigitalFiles:

- The DigitalFile referenced by the Part has no own geometry, but references the DigitalFiles via a FileRelationship of kind 'decomposition'.
- The ExternalGeometry referenced by the Part has no own geometry, but references the GeometricModels via a GeneralGeometricRepresentationRelationship of kind 'decomposition'.

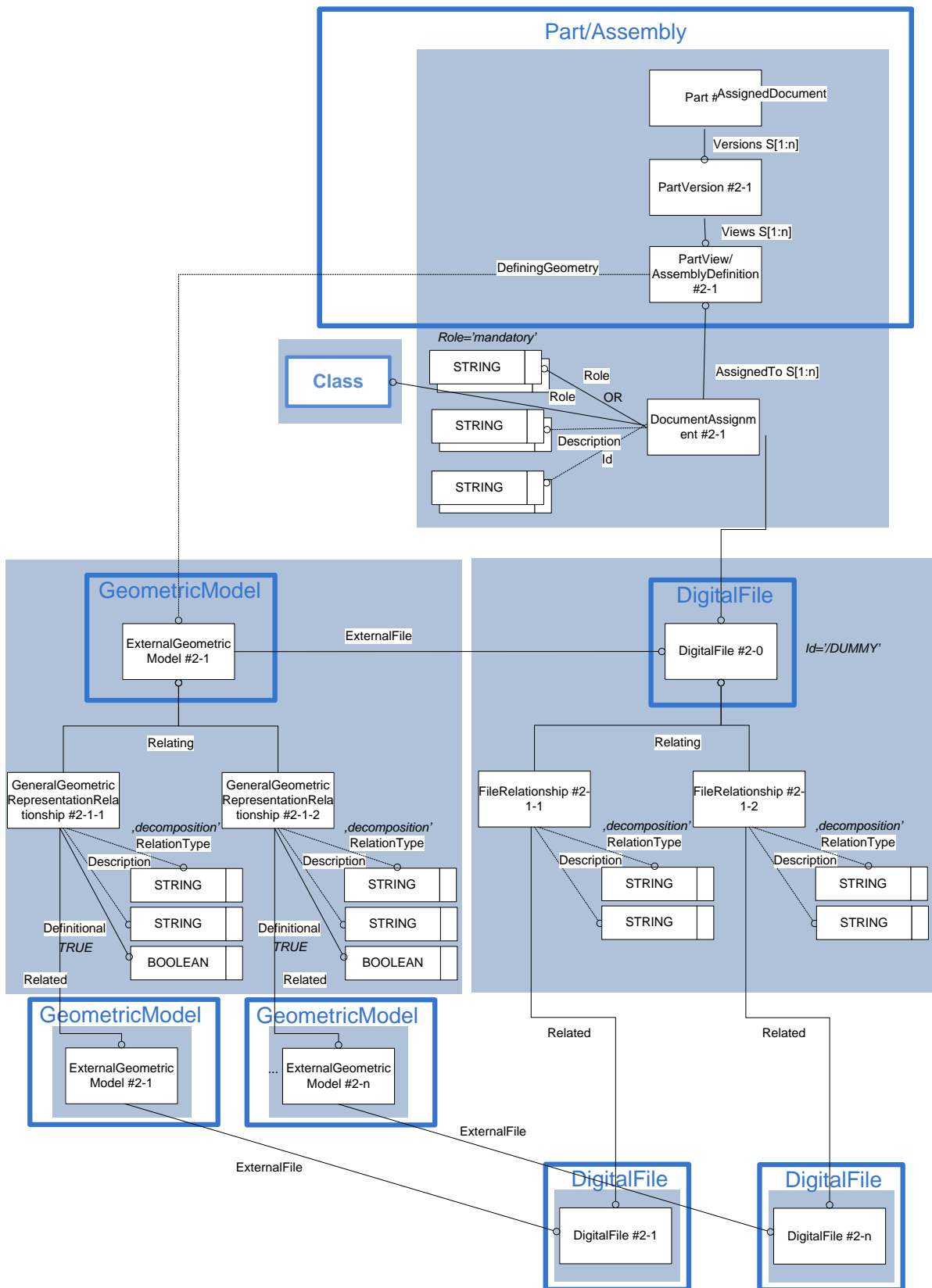


Figure 43: Model splitting for "CAx Representation for DocumentAssignment"

Remark: the use of multiple ShapeElements attached to the same PartView via 'ShapeElement.ElementOf' is not recommended, since according to the STEP resources, there is no semantic telling that all these ShapeElements describe the complete geometry of the part. ShapeElement shall be only used to attach properties to some of the shapes of the geometry (see 6.3.1).

## 2. Model sharing (shared geometry):

In case a geometry is shared by several parts, the DigitalFile and the GeometricModel "containing" this geometry shall be referenced from each part.

## 3. Alternate Models:

In case a part has several alternative GeometricModels (DigitalFiles) each of them shall be directly connected to the Part: the first one (master geometry) via DefiningGeometry and the further ones via AuxiliaryGeometry (introduced within the TC).

For flexible parts having several geometries, depending on their occurrences, AuxiliaryGeometry shall not be used, but rather Occurrence.DefiningGeometry



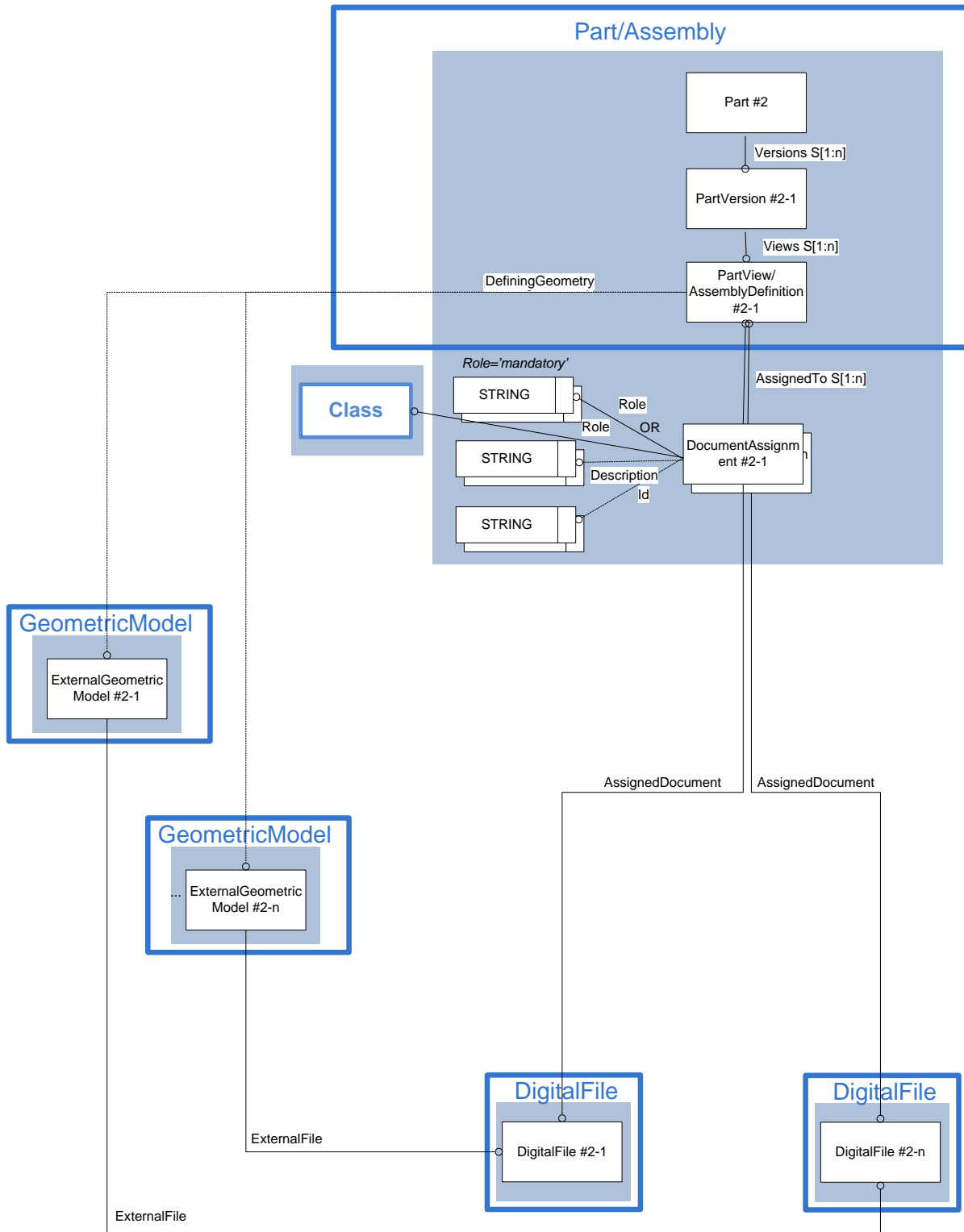


Figure 44: Alternate models for "CAx Representation for DocumentAssignment"

### 11.2 Template "PDM Representation for DocumentAssignment"

This section is relevant when the files are under configuration control.

Depending on the originating PDM System and its respective user settings, DocumentVersion and PartVersion may be handled independently, or in a linked manner. In an exchange scenario, the involved parties have to agree on what a new DocumentVersion means, whether it triggers a new PartVersion or not, and vice versa.

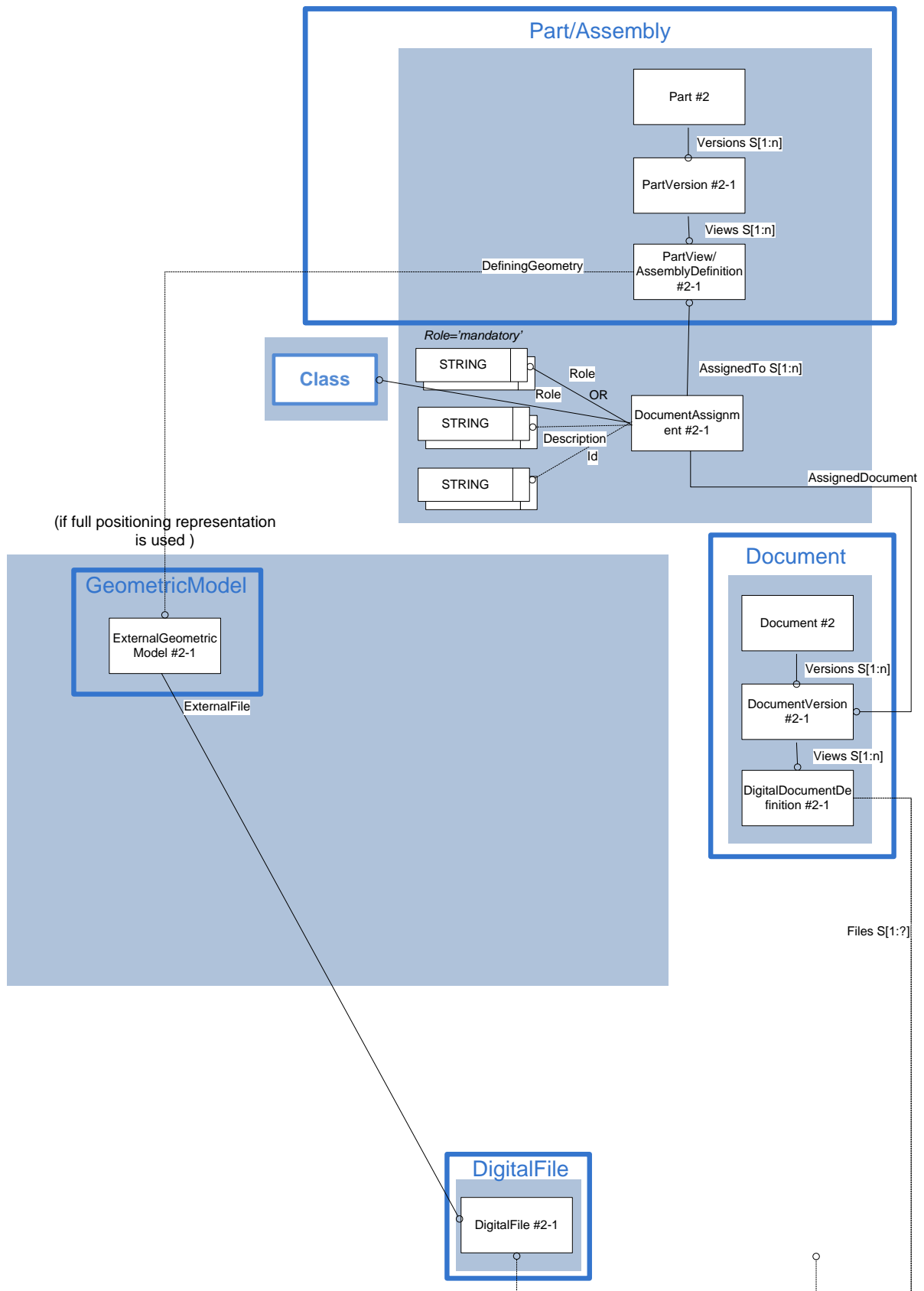


Figure 45: Template "PDM Representation for DocumentAssignment"

## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<Part uid="p--000000001EAAE870">
  <Id>
    <Identifier uid="pid--000000001EAAE870--id7" id="nut" idContextRef="o--
000000178"/>
  </Id>
  ...
  <Versions>
    <PartVersion uid="pv--000000001EAAE870--id7">
      <Id id="/NULL"/>
      <Views>
        <PartView uid="pvv--000000001EAAE870--id7">
          <DefiningGeometry uidRef="egm--000000001AA415B0"/>
          ...
          <DocumentAssignment xsi:type="n0:DocumentAssignment" uid="da--
000000001EAAE870--id7">
            <AssignedDocument uidRef="dv--000000001EAAE870"/>
            <Role>
              <ClassString>mandatory</ClassString>
            </Role>
          </DocumentAssignment>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
<GeometricRepresentation uid="egm--000000001AA415B0"
xsi:type="n0:ExternalGeometricModel">
  <ContextOfItems uidRef="ccs--origin-2"/>
  <Id id="nut.stp"/>
  ...
  <ExternalFile uidRef="df--000000001EAAE870"/>
</GeometricRepresentation>
<Document uid="doc--000000001EAAE870">
  ...
  <Id>
    <Identifier uid="docid--000000001EAAE870--id7" id="nut"
idContextRef="o--000000178"/>
  </Id>
  ...
  <Versions>
    <DocumentVersion uid="dv--000000001EAAE870">
      <Id id="/NULL"/>
      <Views>
        <DocumentDefinition uid="ddd--000000001EAAE870"
xsi:type="n0:DigitalDocumentDefinition">
          <Id id="/NULL"/>
          <Files>
            <DigitalFile uidRef="df--000000001EAAE870"/>
          </Files>
        </DocumentDefinition>
      </Views>
    </DocumentVersion>
  </Versions>
</Document>

```

```
</Views>
</DocumentVersion>
</Versions>
</Document>
<File xsi:type="n0:DigitalFile" uid="df--000000001EAAE870">
...
  <Id>
    <Identifier uid="dfid--000000001EAAE870--19" id="nut.stp"
idContextRef="o--000000178"/>
  </Id>
...
</File>
```

### ***Preprocessor Recommendations:***

- The DocumentAssignment shall refer to the DocumentVersion, rather than Document or DigitalDocumentDefinition.
  - If all DigitalFiles associated to a DocumentDefinition have the same value for their Type, Content, Creation and Format properties, these can be stored in DocumentContent, DocumentCreation and DocumentFormat rather than redundantly in each DigitalFile as FileContent, FileCreation and FileFormat.
  - Special cases: mapped compliantly to the recommendations of the PDM Usage Guide
1. Model splitting:
    - The Document referenced by the Part has no own geometry, but references the Documents via a DocumentDefinitionRelationship of kind 'decomposition'.
    - The ExternalGeometry referenced by the Part has no own geometry, but references the GeometricModels via a GeneralGeometricRepresentationRelationship of kind 'decomposition'.

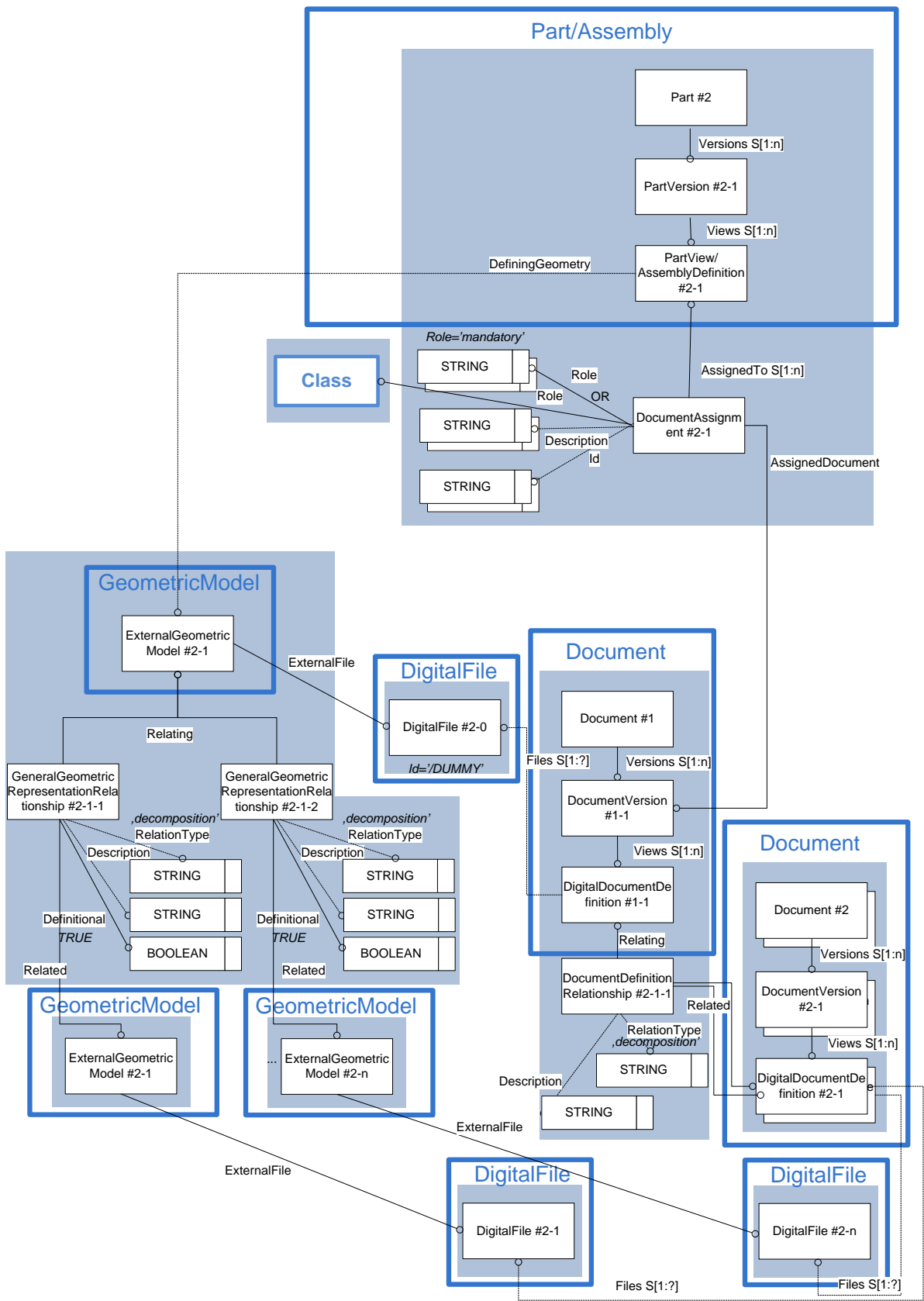


Figure 46: Model splitting for "PDM Representation for DocumentAssignment"

Remark: the use of multiple ShapeElements attached to the same PartView via 'ShapeElement.ElementOf' is not recommended, since according to the STEP resources, there is no semantic telling that all these ShapeElements describe the complete geometry of the part. ShapeElement shall be only used to attach properties to some of the shapes of the geometry (see 6.3.1).

2. Document sharing (shared geometry):

In case a document is shared by several parts, the Document and the Geometric-Model "containing" this geometry shall be referenced from each part.

3. Alternate Models:

In case a part has several alternative GeometricModels (Documents) each of them shall be directly connected to the Part: the first one (master geometry) via DefiningGeometry and the further ones via AuxiliaryGeometry (introduced within the TC).

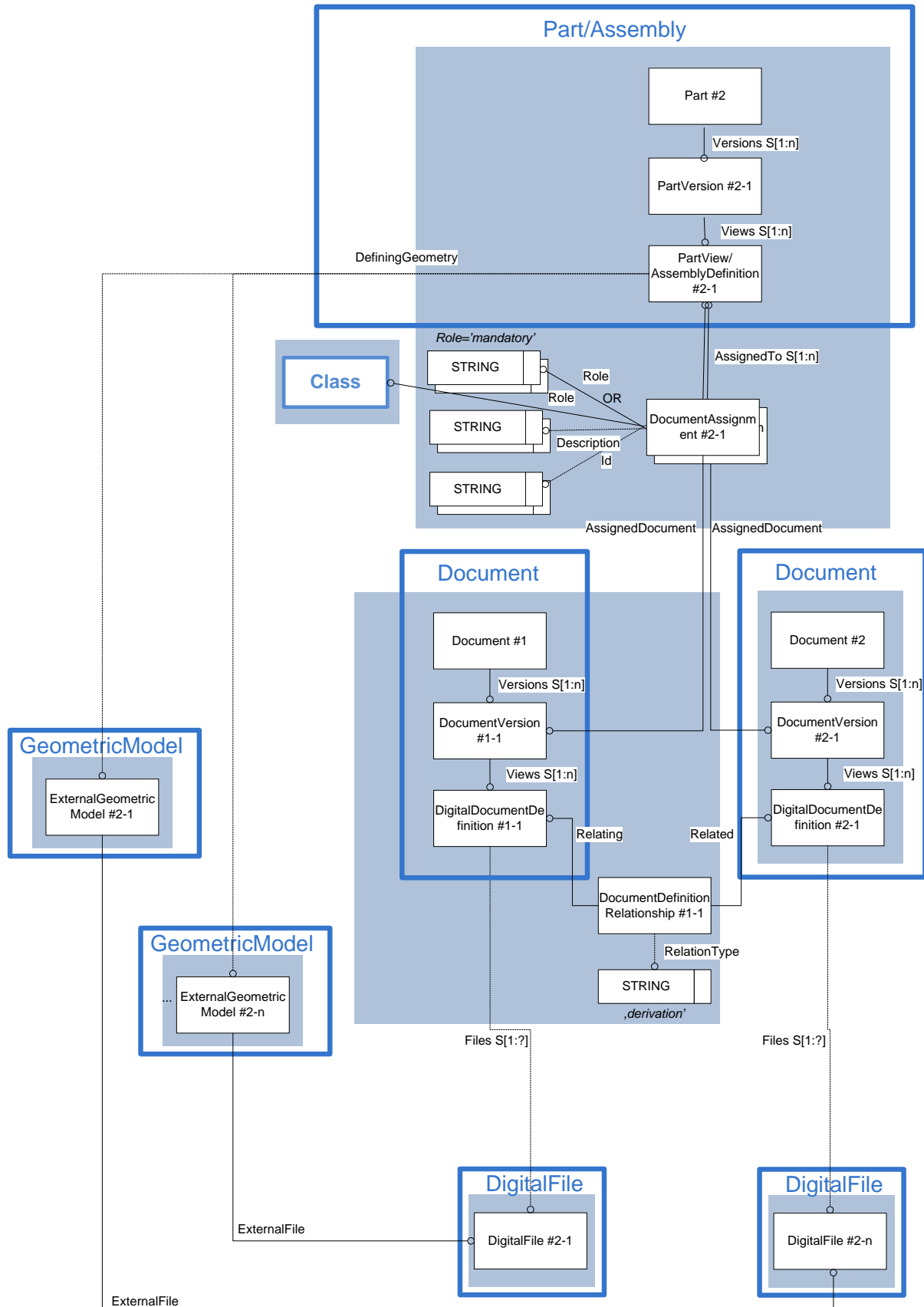


Figure 47: Alternate models for "PDM Representation for DocumentAssignment"

## 12 PDM Properties and CAD User-Defined Attributes

In the same way that in the CAx-IF Recommended Practices for User Defined Attributes V1.2, the aim of this section is to specify how to transfer:

- PDM type properties
- PDM system properties
- user defined attributes (UDA's) in Computer Aided Design (CAD) systems

### 12.1 Fundamental concepts

The approach used to transfer PDM properties and user defined attributes is the “general property” approach introduced in Part 41. It is based on the concept that an attribute (the key in a key-value pair) is defined once as a placeholder, and is then used to assign the actual values to the respective target elements as often as needed.

The reference points in a STEP file for which such an attribute shall be defined in the given context are:

- the entire part (`Part`, `AssemblyDefinition` or `PartView`) using “Property-ValueAssignment” template (see 6.2)
- an instance of the part in an assembly (`NextAssemblyOccurrenceUsage`) using `PropertyValueAssignment` template (see 6.2)
- a part view relationship in an assembly (`PartViewRelationship`) using `PropertyValueAssignment` template (see 6.2)
- a portion of the shape defining the part (`Part`, `AssemblyDefinition` or `PartView`) using `GeneralShapeDependentProperty` template (see 6.3)
- the entire document (`DocumentDefinition`) according to section 10.5
- a single file (`DigitalFile`) according to section 10.5

We will refer to these reference points as “Model Element” in figures below.

#### **Preprocessor Recommendations:**

- Even if `PropertyValueAssignments` may be assigned to further related objects (like `Occurrence`, `PartVersion`, `DocumentVersion`, ...), in order to reduce the complexity of the postprocessor implementation, only assignments to the objects mentioned above are recommended.
- If the exchange of `Properties` to further objects (like `Approval`, `Organization`, `PersonInOrganization`, ...) is needed, this shall be agreed between the exchange partners. In any case, they shall conform to the “`PropertyDefinition`” template (see 14.2) and to the “`PropertyValueAssignment`” template (see 6.2), possibly by using other values for `ClassifiedAs` like ‘approval properties’, ‘person properties’, ..).

There are a number of pre-defined property types in STEP that may be used to store a PDM property or a user-defined attribute. In the context of this document, this includes:

- descriptive attributes (« `StringValue` » template see 4.6.10)
  - name and description
- measure values (“`NumericalValue`” template see 4.6.9)
  - name and value
  - name, value and unit



## 12.2 Template “PropertyDefinition”

The `PropertyDefinition` defines a PDM property or a user defined attribute. This can then be used by one or several `PropertyValues`.

To assign a `PropertyDefinition`:

- create a `PropertyValue` with the “NumericalValue” (see 4.6.9) or “StringValue” templates (see 4.6.10).
- in the “NumericalValue” or “StringValue” template, link the `PropertyValue` to the `PropertyDefinition` with `PropertyValue.Definition` attribute.

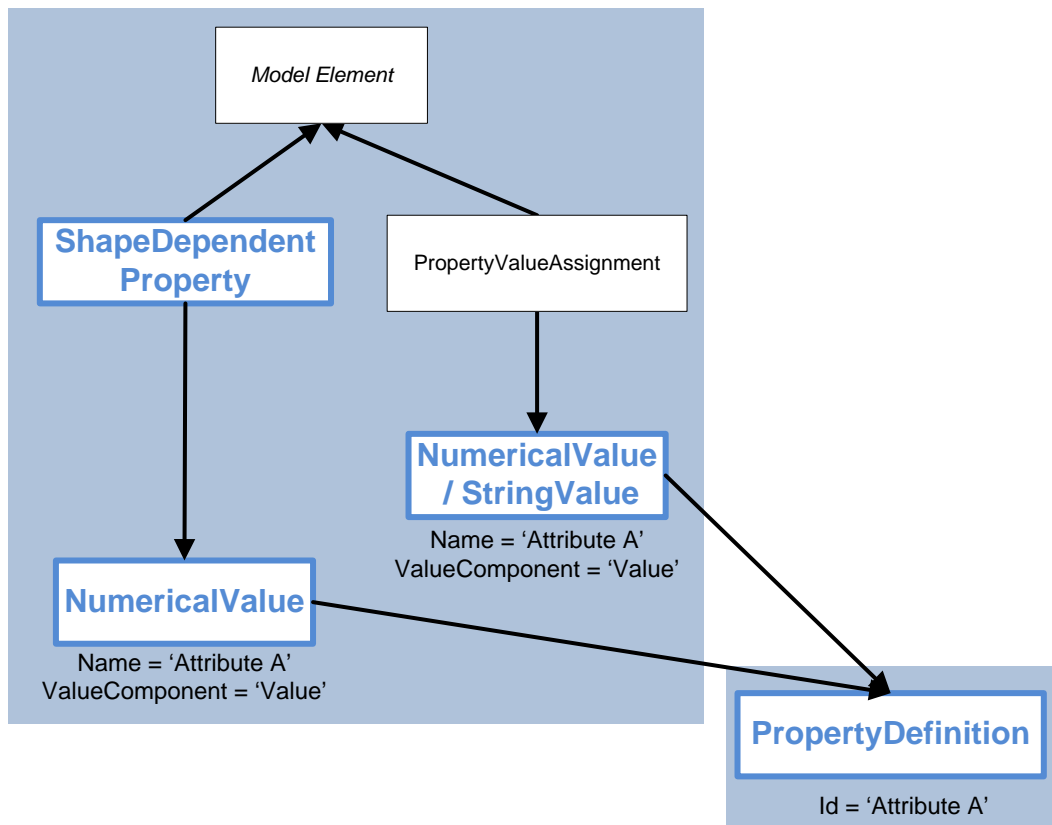


Figure 48: Definition of an attribute name and its usage

## The Instance Model: AP242 BO Model XML entities and attributes

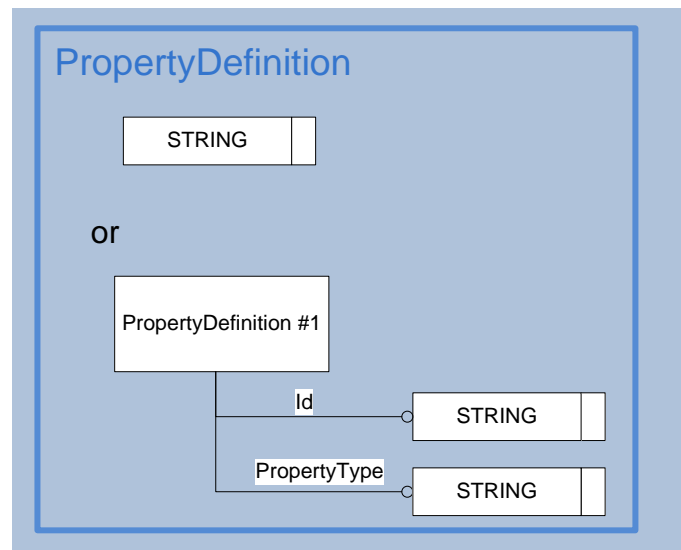


Figure 49: Template "PropertyDefinition"

ENTITY PropertyDefinition	Attribute Type
AllowedUnits	OPTIONAL SET[1:?] of UnitSelect
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
PropertyType	ClassSelect
VersionId	OPTIONAL IdentifierSelect
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DateTimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PropertyDefinitionRelationship	OPTIONAL SET[1:?] of PropertyDefinitionRelationship
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 53: "PropertyDefinition" Attributes

**Attribute recommendations**

- The **Id** attribute is the text that represents the general property. Use IdentifierString if one of the values below is used, or if the PropertyValue.name is repeated here, otherwise use "Identifier" template (see 4.6.6).

When applicable, the following values shall be used:

Id	Explanation
'recyclability property'	A recyclability property is information concerning the ability to re-use objects or components of objects after their primarily intended usage
'mass property'	a mass property is a quantity of matter of which an object consists
'quality property'	A quality property is a property that provides information about the level of quality of products or processes
'cost property'	A cost property is a property that specifies costs
'duration property'	A duration property is a property that specifies a period of time during which a given object is used or will last
'general property'	A general property is a property for which the exact purpose is not known at the time. This is the recommended value for the case an exporting system does not provide such a classification of properties by purpose.

- The **PropertyType** attribute is the kind of property the **PropertyDefinition** defines. Use "Class" template (see 4.6.4). When applicable, the following values shall be used. If one of these values apply, use ClassString:

PropertyType	Explanation
'system property'	The property is internal to the source PDM system and may be mostly ignored by the target PDM system, except in scenarios to maintain the full functionality of the transferred data Example: PDM vault, lifecycle template, ...
'PDM property'	The property is defined with COTS solution. Relevant for PDM data exchange. Example: calculated weight, estimated weight, material standard, ...
'customized PDM property'	The property is present only on particular categories of part (for example standard part, software part, ...) or documents customized in the underlying PDM system. The part/document category itself is mapped in PartTypes/DocumentTypes (see 5.1.1/8.1.1). Relevant for PDM data exchange. Example: software characteristics, ...
'user defined attribute'	The property is defined by a particular user on a specific object and does not exist for all instances of the PDM object. Relevant for the CAx data exchange, and (if the PDM system can handle them) for PDM data exchange Example: surface finish

- For the properties defined in section 13 (validation properties), the following values shall be used:

PropertyType	Explanation
assembly validation property	According to section 13.1.1
geometric validation property	According to section 13.2

- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:** If the Preprocessor does not support grouping of properties (see section 12.3) nor metadata to the user defined attributes (see section 12.6), rather use PropertyDefinitionString and set it with the PropertyType.

**Postprocessor Recommendations:** None specified.

**Related Entities:** There are no specific related entities.

## The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```
<PropertyValue uid="id1" xsi:type="ap:NumericalValue">
  <Definition>
    <PropertyDefinition uidRef="pd--000000320"/>
  </Definition>
  <Name>
    <CharacterString>angle</CharacterString>
  </Name>
  <Unit uidRef="u--000000002"/>
  <ValueComponent>45</ValueComponent>
</PropertyValue>
```

```
<PropertyDefinition uid="pd--000000320">
  <Id id="quality property"/>
  <PropertyType>
    <ClassString>user defined attribute</ClassString>
  </PropertyType>
</PropertyDefinition>
```

OR:

```
<PropertyValue uid="id1" xsi:type="ap:NumericalValue">
  <Definition>
    <PropertyDefinitionString>user defined attribute</PropertyDefinitionString>
  </Definition>
  <Name>
    <CharacterString>angle</CharacterString>
  </Name>
  <Unit uidRef="u--000000002"/>
  <ValueComponent>45</ValueComponent>
</PropertyValue>
```

### 12.3 Group of attributes and group of attributes values

In the same way that in section 5.2 of the CAX-IF Recommended Practices for User Defined Attributes V1.2, the aim of this section is to group user defined attributes on two semantic levels:

- **Groups of attributes**

**Note** that systems handling user attributes in a way that all attributes with the same name have the same meaning may also define groups on this level. This level of grouping is listed here for completeness. In the context of the CAX-IF, groups of attributes will always be defined on the attribute level, see section below.

To define a group of attributes in the sense that this grouping shall also be applied to all values of the respective attributes (e.g. the calculated weight, nominal weight and actual weight of a part), a `PropertyDefinition` will be created for that group, carrying the name of the group, and relating all `PropertyDefinition` instances, which are members of the group.

The following rules apply to the `PropertyDefinitionRelationship` attribute values:

- o `Relating` : the `PropertyDefinition` that defines the group of attributes
- o `Related` : the `PropertyDefinition` that defines an attribute in that group
- o `RelationType` : 'decomposition'

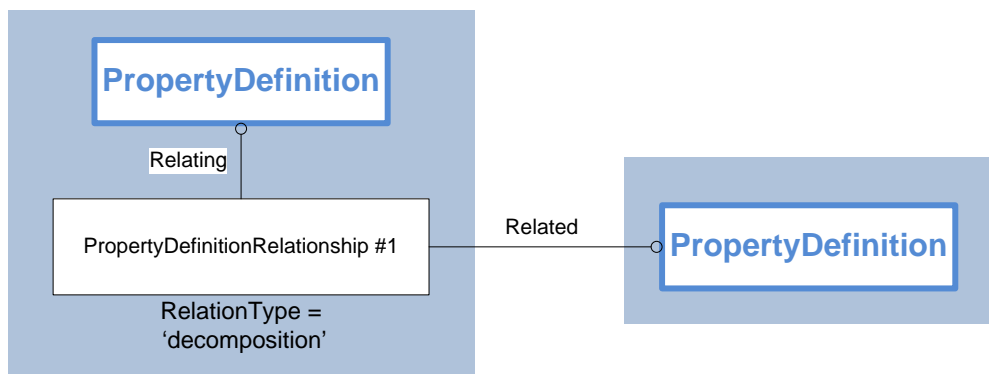


Figure 50: Defining a group of attributes

#### **Preprocessor Recommendations:**

- As defined in the EXPRESS data model via a WHERE rule, the related `PropertyDefinition` and the relating `PropertyDefinition` shall be different objects.
- **Groups of attribute values**

The approach to define a group of attribute values – i.e. which apply to the specific use of the respective attributes – is quite similar to the grouping of attributes themselves, only it will now happen on the `PropertyValue` level.

All `PropertyValue` – the one defining the group and all of the ones defining the attribute values – need to reference the same model element.

This means that only values for the same model element can be grouped.

The following rules apply to the `PropertyValueRelationship` attributes:

- o `Relating` : the `PropertyValue` that defines the group of attribute values
- o `Related` : the `PropertyValue` that defines an attribute value in that group
- o `RelationType` : 'decomposition'

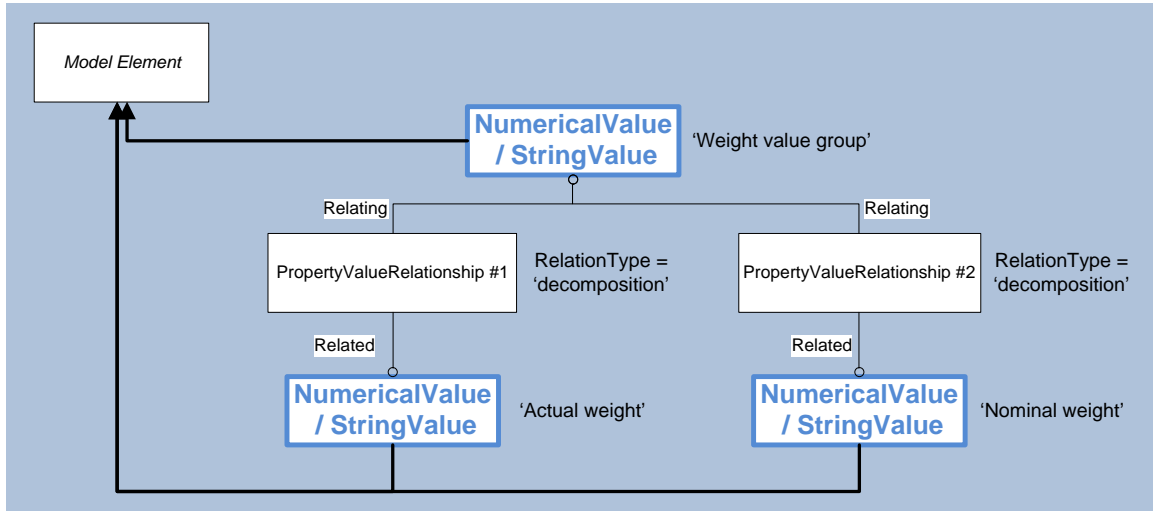


Figure 51: Defining a group of attribute values

**Preprocessor Recommendations:**

- As defined in the EXPRESS data model via a WHERE rule, the related Property Value and the relating Property Value shall be different objects.

**12.4 Specifying the target for the attribute**

User defined attributes can be attached to the geometry in a STEP file at different levels of granularity, i.e. individual solids or surfaces, or entire parts. While all CAD systems support the definition of attributes at the part level, only some systems can handle attributes at the level of individual shape elements.

**12.4.1 Attributes at the part level**

To assign a user defined attribute to both individual parts and assemblies, it is recommended to use “PropertyAssignment” template defined in chapter 6.2.

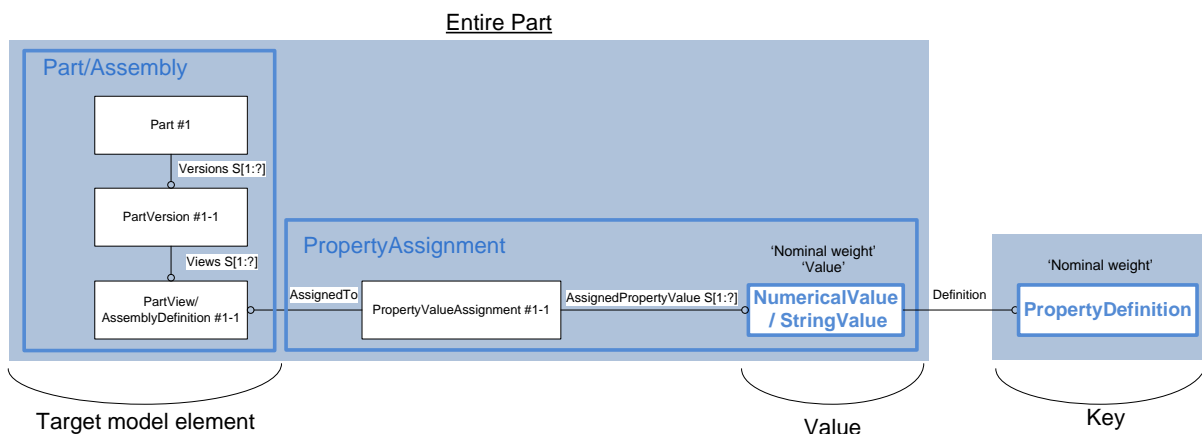


Figure 52: User defined attribute at the part/assembly level

**12.4.2 Attributes at component instances in an assembly**

To assign a user defined attribute to a specific instance of a component within an assembly, the property needs to be attached to the assembly definition. If the instance in question is an immediate child of the assembly node, the attribute will be attached to the NextAssem-

blyOccurrenceUsage, it is recommended to use “PropertyAssignment” template defined in chapter 6.2 directly in the NextAssemblyOccurrenceUsage entity.

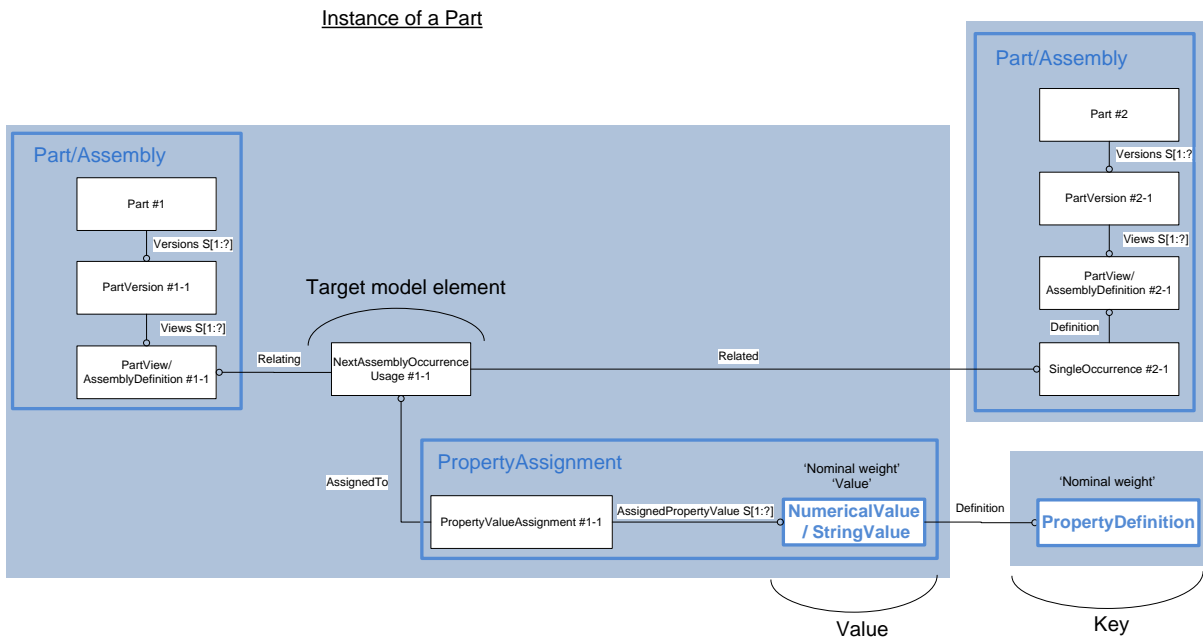


Figure 53: User defined attribute for a simple component instance in an assembly

### 12.4.3 Attributes at the shape level

To assign a user defined attribute to a shape or a portion of a part shape, it is recommended to use “ShapeDependentProperty” template defined in chapter 6.3.

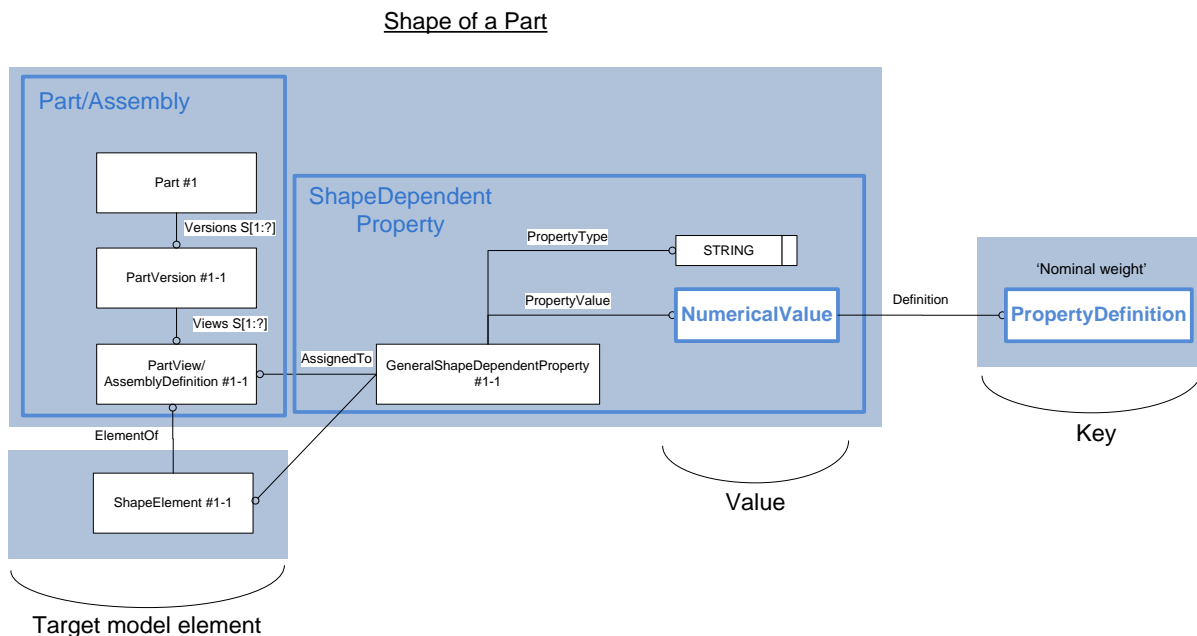


Figure 54: User defined attribute for a shape or a portion of shape

### 12.5 Definition of attribute value

In the same way that in section 7 of the CAx-IF Recommended Practices for User Defined Attributes V1.2, the aim of this section is set up a property value for a property.

If the Preprocessor does not support grouping of attributes (see section 12.3) nor metadata to the use defined attributes (see section 12.6), it is recommended to set `PropertyValue.Definition` with `PropertyDefinitionString` and set it with the `PropertyType`. Otherwise, it is recommended to use the same value for `PropertyDefinition.Id` and `PropertyValue.Name` like it is shown in 12.2 with the “Attribute A” example.

There are two types of values:

- Values with unit (e.g., measure values)
- Values without unit (String, Integer, Real, Boolean)

### 12.5.1 Values with Unit

A value attribute transports a general value with an associated unit. To define a value attribute, the “NumericalValue” template defined in chapter 4.6.9. is used.

**Note** that `NumericalValue` is a subtype of `ValueWithUnit`, hence the definition of a unit is mandatory. For transfer of values without applicable unit, see section below.

Example of use of the “NumericalValue” template:

```
<PropertyValue uid="id1" xsi:type="ap:NumericalValue">
  <Definition>
    <PropertyDefinition uidRef="pd--000000320"/>
  </Definition>
  <Name>
    <CharacterString>angle</CharacterString>
  </Name>
  <Unit uidRef="u--000000002"/>
  <ValueComponent>45</ValueComponent>
</PropertyValue>

<PropertyDefinition uid="pd--000000320">
  <Id id="quality property"/>
  <PropertyType>
    <ClassString>user defined attribute</ClassString>
  </PropertyType>
</PropertyDefinition>
```

### 12.5.2 Values without Unit

Values without applicable unit convey all kinds of information, in particular in the context of PDM Properties. To define such an attribute, the “StringValue” template defined in chapter 4.6.10. shall be used.

Four main types can be distinguished, which shall be handled as recommended below.

#### **Preprocessor Recommendations:**

- A *descriptive attribute* stores an arbitrary text string in the `ValueComponent` attribute. As usual in STEP, any special characters in the name or description need to be encoded in Unicode.
- An *integer attribute* stores a whole number (and nothing else) as text string in the `ValueComponent` attribute. Examples for this include counts and sequences.
- A *real attribute* stores a decimal number (and nothing else) as text string in the `ValueComponent` attribute. Examples for this include ratios and percentages.
- A *Boolean* or *Logical attribute* stores a pre-defined value as text string in the `ValueComponent` attribute. The value shall be either ‘TRUE’, ‘FALSE’, or ‘UNDEFINED’.

#### **Postprocessor Recommendations:**



- The values shall be mapped to the internal attribute types of the importing system according to its default typecasting rules. In case an unexpected value is encountered – e.g. an arbitrary string when an integer value is expected – an error message shall be given.

Note: For implementations based on AP242 MIM / STEP Part 21, the EXPRESS data model defines four dedicated entity types for the transfer of values without unit:

- `descriptive_representation_item`
- `integer_representation_item`
- `real_representation_item`
- `boolean_representation_item`

Having such explicit counterparts in the AP242 BO Model would improve the stability of PDM Property exchange. BugZilla [#6290](#) has been created to add these elements in a future revision of the standard.

### **12.6 Transfer of Meta-Data for the User Defined Attributes**

In the same way as in section 7.4 of the CAX-IF Recommended Practices for User Defined Attributes V1.2, the aim of this section is to give as an option the possibility to add additional information about an attribute, an attribute value, or group thereof.

Note that this is not supported for user defined attributes defined on shape elements (see section 13.4.3 above), since the `PropertyValueAssignment` cannot point to a `ShapeDependentProperty`.

Using the `PropertyValueAssignment` template it is possible to add even more information about an attribute, an attribute value, or group thereof. This may include CAD-system specific data, such as whether the attribute is relevant for a data management system or not. The identifier of the additional information is carried in the `PropertyValue.Name` attribute, and the value is transferred in the `PropertyValue.ValueComponent`.

The “meta data” will be defined as a “property of a property”, and they can be distinguished easily from the actual user defined attributes by two means:

- its `PropertyValueAssignment` will point to a `PropertyValueAssignment`, and not one of the model elements identified in section 6.2.
- its `PropertyValueAssignment` will have no associated `PropertyDefinition`

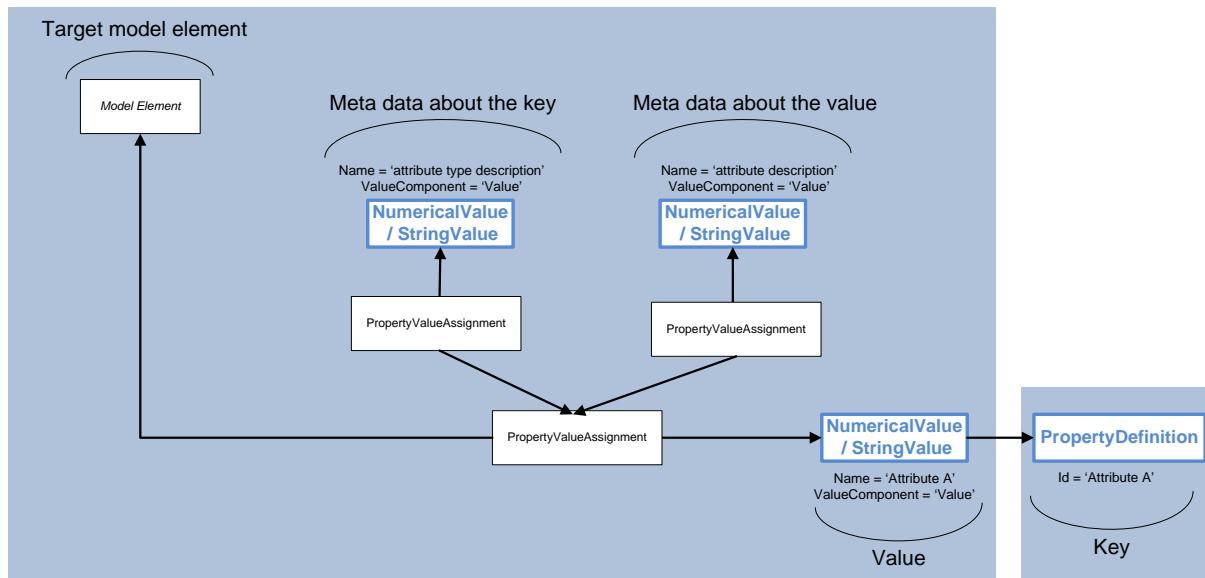


Figure 55: Specification of meta data for an attribute and its value

### 12.6.1 Designation of the Attribute Type

In order to transfer the name of the type for the user defined attribute as given in the originating system, add an additional property with the PropertyValueAssignment template with the following attribute characteristics:

- PropertyValue.name: 'attribute type designation'
- PropertyValue.ValueComponent: The designation of the attribute type as given in the native system

This shall be linked to the attribute value definition (PropertyValueAssignment) as shown in the figure above.

#### 12.6.1.1 Attribute Value / Group Description

In order to transfer a description for the attribute (**Note** that this is a description *about* the attribute, in contrast to a descriptive attribute as defined in section 12.5), add an additional property with the PropertyValueAssignment template with the following attribute characteristics:

- PropertyValue.name: 'attribute description'
- PropertyValue.ValueComponent: Textual information about the attribute

This shall be linked to the PropertyValueAssignment of the UDA or a group of values as shown in the figure above.

## 13 Validation Properties

This chapter describes how to confirm the correctness of exchanged geometry information and assembly information compared to its source. The following exchange process is suggested to enable validation of exchanged information. It is optional to apply validation properties in exchange files.

Product data are created in a source system and shall be sent to a target system using the exchange format described in this document.

The source system derives – from its own representation of the product data - validation properties that reflect the main semantics of the product data. For the purpose of this document the following two types of validation properties are distinguished:

- *Assembly Validation Properties (AVP)*: They provide a verification capability for product structure data where geometry is not present. Two properties are recommended: one to ensure that the number of instances found at each node is correct and another one to ensure that the position and orientation information for each instance is correct. See section 7 of the Recommended Practices for Geometric and Assembly Validation Properties (see reference in Annex C) and section 13.1, below for details.
- *Geometric Validation Properties (GVP)*: They describe characteristics of a solid or surface model or of a collection of them and are assigned to parts and assemblies. See section 4 of the Recommended Practices for Geometric and Assembly Validation Properties (see reference in Annex C) and section 13.2, below for details.

These validation properties enable the verification of geometry and assembly information of received data sets. Values for these properties are added to the exchanged data set of the product structure, that is, to the representation that is sent to the target system. The target system reads the received data set including the source validation properties. It converts the data set, but not the validation properties, to the target representation. The target system derives the validation properties from this local representation after conversion using the same algorithms that are described here and that were applied at the source system.

The validation property values of the source and the target representations are then compared, manually or by a dedicated application. If the values are identical within an agreed tolerance, the semantics of the source product data were exchanged correctly to the target representation. With this, the validation of the exchange is completed successfully.

**Preprocessor Recommendations:** It is recommended that all the validation properties of one AssemblyDefinition use the same PropertyValueAssignment.

### **13.1 Assembly Validation Properties (Notional Solid, Number of Children)**

Section 7 of the Recommended Practices for Geometric and Assembly Validation Properties (see reference in Annex C) specifies the semantics of two Assembly Validation Properties:

- *Number of Children*: For each node the number of instances or branches is recorded.
- *Notional Solids Centroid Position*: The positional information for each instance in the product structure is recorded, i.e. position and orientation of the coordinate systems for each child node relative to its parent. Note that this condition is not mathematically guaranteed by this methodology, but the chance of an incorrect position and orientation combining to give the correct result is extremely small.

These two validation properties allow verifying that the number of instances found at each node is correct and that the position and orientation information for each instance is correct.

The following sub-sections describe the representations of values of these properties in an AP242 BO XML exchange structure.

#### **13.1.1 Number of Children**

Each Part node which is a parent part of at least one other Part node will have a property attached to enumerate the actual number of child instances of that parent node.

This property shall be assigned to an AssemblyDefinition as property, using the “PropertyAssignment” template; see chapter 6.2. The property value counts the number of NextAssemblyOccurrenceUsage instances that reference this AssemblyDefinition by their relating-attribute; see Figure 56, below.

The property value shall be instantiated according to the following description, which is depicted in the instance diagram in Figure 56 below.

All instances of type AssemblyDefinition that are used as relating AssemblyDefinition instances by one or more NextAssemblyOccurrenceUsage instances will have a single PropertyValueAssignment of the PropertyAssignment template assigned to it to represent the number of children count. The classification of this PropertyValueAssignment instance shall follow the recommendation for attribute classifiedAs in section 6.2.

Only one NumericalValue shall be referenced by this PropertyValueAssignment, that is, there shall be only one element in the set of assignedPropertyValues. The name of this NumericalValue shall be 'number of children'. The PropertyDefinitionString shall be 'assembly validation property'. The Unit of the NumericalValue shall be 'each'.

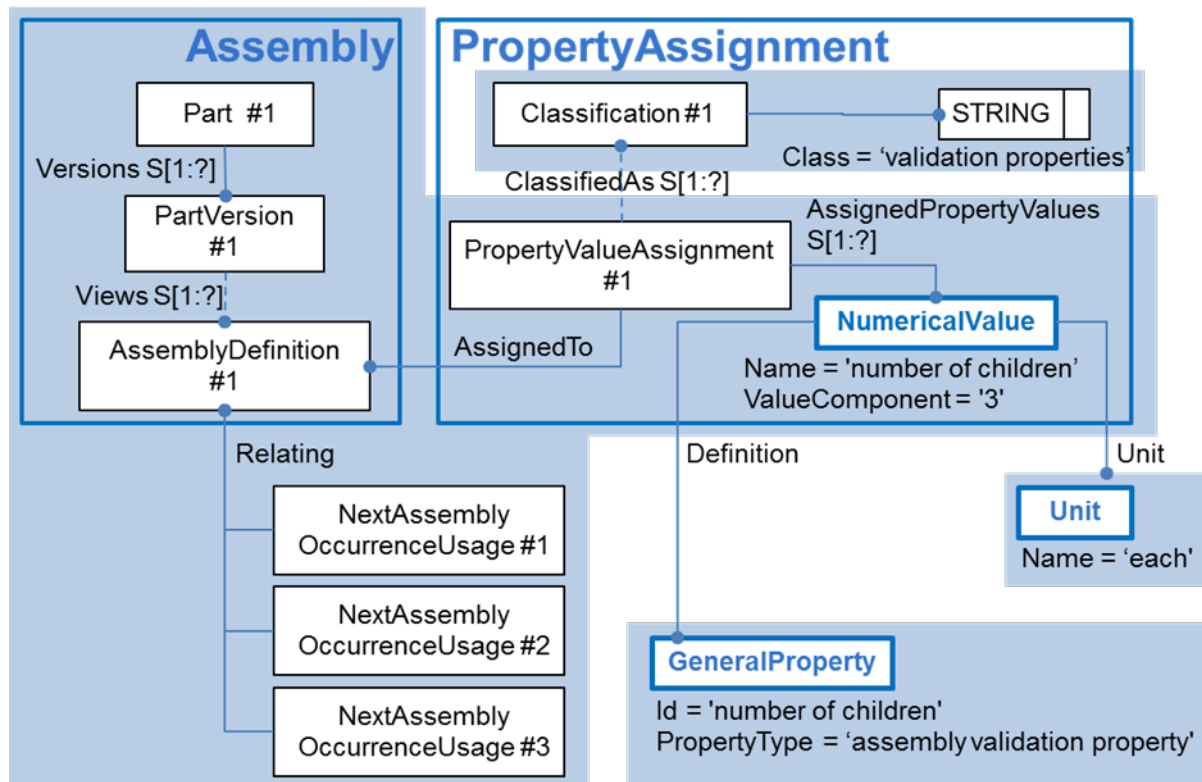


Figure 56: Instantiation of AVP 'number of children' for 3 children

### 13.1.2 Notional Solids Centroid Position

This property is similar to the geometric validation property "centroid" (see section 13.2): here, as well, a location property is defined for each sub-assembly. However, in this case the property is not calculated based on the real geometry of the product.

The details of this property are specified in section 7.2 of the Recommended Practices for Geometric and Assembly Validation Properties and repeated here.

For the top node and each intermediate node of a product structure, a notional solid is assumed within the child node of each child instance of that node. Using the positional and orientation relationship for each child instance, a centroid position can be calculated for the combined set of notional solids within the set of child instances.

The notional solid will be a cube of size 10 x 10 x 10. The notional solid will be positioned with its centroid at (10.0, 10.0, 10.0) of the coordinate system of the child node. Note that the actual size and shape of the notional solid will not, in fact, affect the overall result. The key data is the centroid position and the assumption that the volume of the notional solid in each

child node is the same. Mathematically, the calculated point is the mean of the set of points at (10.0, 10.0, 10.0) within the child nodes.

Note that in contrast to an actual solid centroid, the notional solid itself is not in the STEP file – it is just a convention. Thus, it has to be ensured that the correct geometrical context is used for the notional solids centroid position, in order to guarantee that the units are applied correctly. In addition, the notional solid does not have any material properties such as density and weight, thus centroid, center of mass and center of geometry are synonymous.

The child node may be a leaf node of the overall assembly or another intermediate node within the sub-assembly. Each case is treated in the same way. Even though the child node might be an intermediate node with no actual geometry defined, a notional solid will be assumed for the purpose of this calculation.

The notional centroid for each sub-assembly is influenced only by the notional solids in each of its direct child nodes. The CentreOfMass represents the notional solid centroid position. The child nodes that this centroid position is valid for are those Occurrences that are referenced by the NextAssemblyOccurrenceUsage.related attributes.

The CentreOfMass property value shall be instantiated according to the instance diagram in Figure 57, below, and the subsequent attribute recommendations.

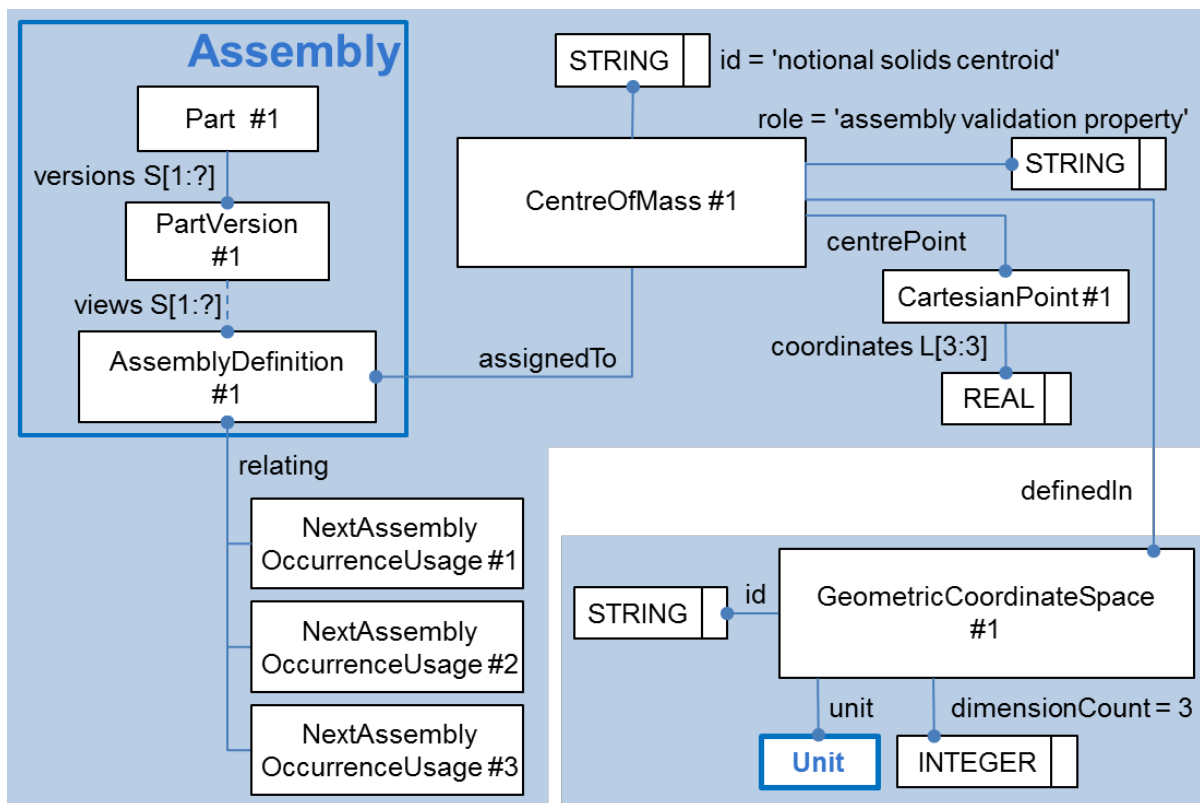


Figure 57: Instantiation of AVP 'notional solids centroid'

ENTITY CentreOfMass	Attribute Type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
DefinedIn	GeometricCoordinateSpace
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL Id

ENTITY CentreOfMass	Attribute Type
Role	OPTIONAL ClassSelect
ValueDetermination	OPTIONAL ClassSelect
CentrePoint	CartesianPoint

Table 54: "CentreOfMass" Attributes

**Attribute recommendations:**

- **DefinedIn:** the GeometricCoordinateSpace in which the property is applicable. To denote the coordinate space of the CartesianPoint, a GeometricCoordinateSpace is instantiated with dimensionCount equal three, as this is a centroid in three dimensional space. The unit that the coordinate values are measured in shall be provided as string-value in the attribute CentreOfMass.definedIn.unit . In case of relative positioning, it is up to the pre-processor to provide a distinct instance of GeometricCoordinateSpace for each CentreOfMass.
- **Id:** the identifier for the CentreOfMass. The id shall be provided as type IdentifierString; the string value shall be 'notional solids centroid'. An AssemblyDefinition object shall be assigned at maximum one object of CentreOfMass with this id.
- **Role:** the meaning of the assignment. The role of the CentreOfMass shall be provided as type ClassString; the string value shall be 'assembly validation property'.
- **CentrePoint:** a point in three-dimensional space that defines the location of a CentreOfMass in a GeometricCoordinateSpace. The CentreOfMass.centrePoint is a CartesianPoint with exactly three coordinate values of type REAL. The CartesianPoint defines the calculated centroid for the notional solids assumed for each child node.
- Other attributes than these are not covered by these Recommended Practices; their use is discouraged as it would depend on mutual agreements between data exchange partners.

**Preprocessor Recommendations:**

- If the Part/Assembly for the CentreOfMass has a GeometricModel associated to it, both (CentreOfMass and GeometricModel) shall be defined in the same GeometricCoordinateSpace.

**Postprocessor Recommendations:** None specified.

**Related Entities:**

- See Table 26 in section 6.1 for the definition of CartesianPoint.
- See Table 24 in section 6.1 for the definition of GeometricCoordinateSpace.

**13.2 Geometric Validation Properties (Repeated from referenced Parts)**

Geometric Validation Properties (GVP) describe characteristics of a solid or surface model or of a collection of them. The original specification of GVPs is in section 4 of the "Recommended Practices for Geometric and Assembly Validation Properties" (see reference in Annex C) for details. The most important information is repeated here.

The validation properties defined in this document represent different types of measures: for volume, for area, and for length. Each of these requires a correct definition of the applied unit of measure in the XML file.

Geometric Validation Properties in an AP242 BO Model XMLfile can only be attached to parts or assemblies, not to geometry, as the representation of detailed geometry is out of scope of the AP242 BO Model.

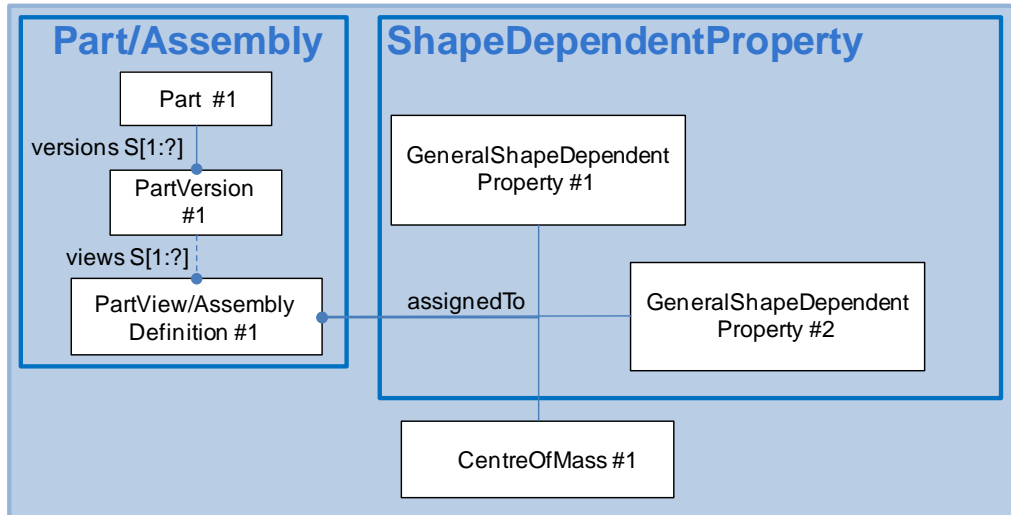


Figure 58: Geometric Validation Properties at the Part and Assembly level

Geometric Validation Properties will be defined dependent on the class of geometry:

- Solids: volume, surface area, and centroid;
- Surfaces: surface area, and centroid;
- Curves: curve length, and centroid.

This means that if a model contains a solid, independent surfaces, and independent curves, there will be three different centroids given in the validation properties: one for each class.

**Note:** Geometric Validation Properties shall be computed solely off the part geometry. They should not take any Supplemental Geometry into account, since not all target applications support Supplemental Geometry. In addition, Supplemental Geometry may contain unbounded elements.

### 13.2.1 Validation Properties for Solid Geometry

#### 13.2.1.1 Volume

Volume specifies the amount of space occupied by the solid model as measured in cubic units. During an exchange this GVP can be used to validate the success of creating an equivalent solid via the translation.

Figure 59 illustrates the instances required to specify in the XML file the volume property value of the original part, as calculated in the native system.

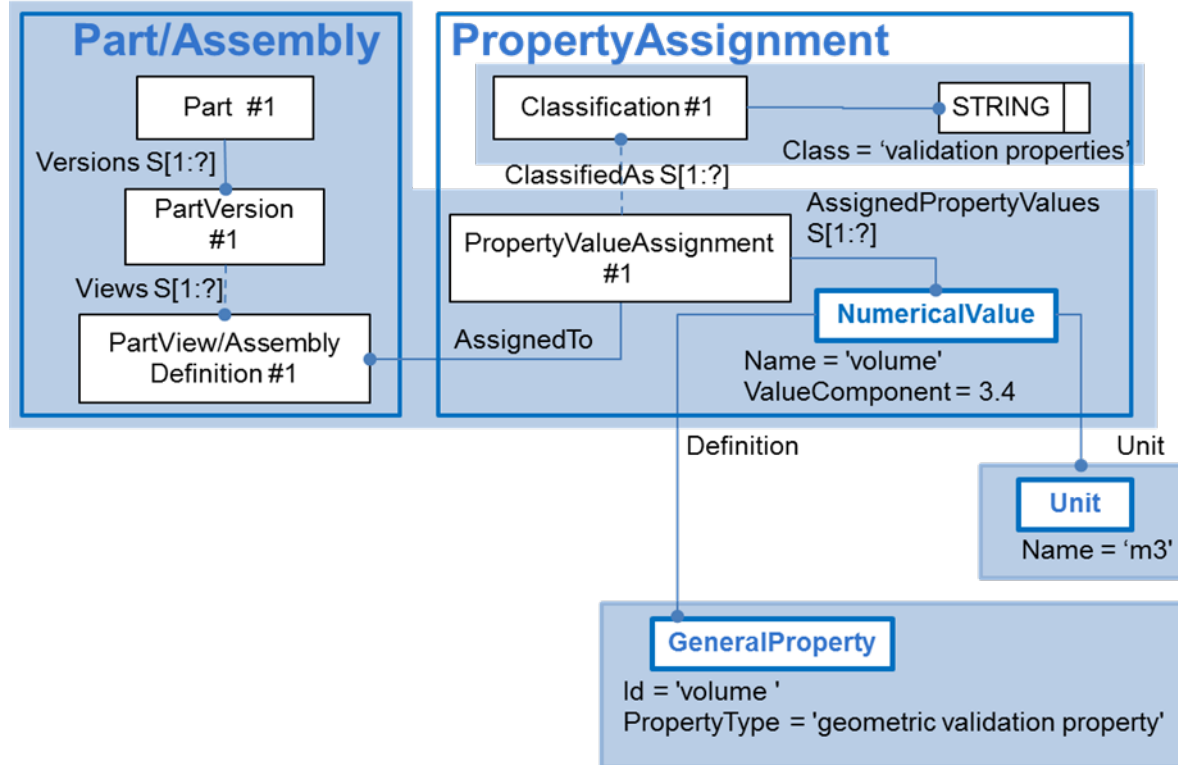


Figure 59: GVP 'volume' of 3.4 cubicmetre assigned to a Part or Assembly

### 13.2.1.2 SurfaceArea

Surface area specifies the area measurement of the surface of an entire solid. By default, this will include any voids in the model. Figure 60 below illustrates the relevant entities and their mandatory attributes used in the assignment of the surface area validation property.

**Note:** Since CATIA calculates the 'wetted area' (i.e. voids will not be taken into account) instead of the total surface area, the validation mechanism will report a 'false error' when exchanging a model with voids in it between a CATIA-based and a non-CATIA-based system. Therefore, when exporting validation properties from a CATIA-based system, the name of the NumericalValue (see Figure 60) shall be 'wetted surface area' instead of 'surface area'.



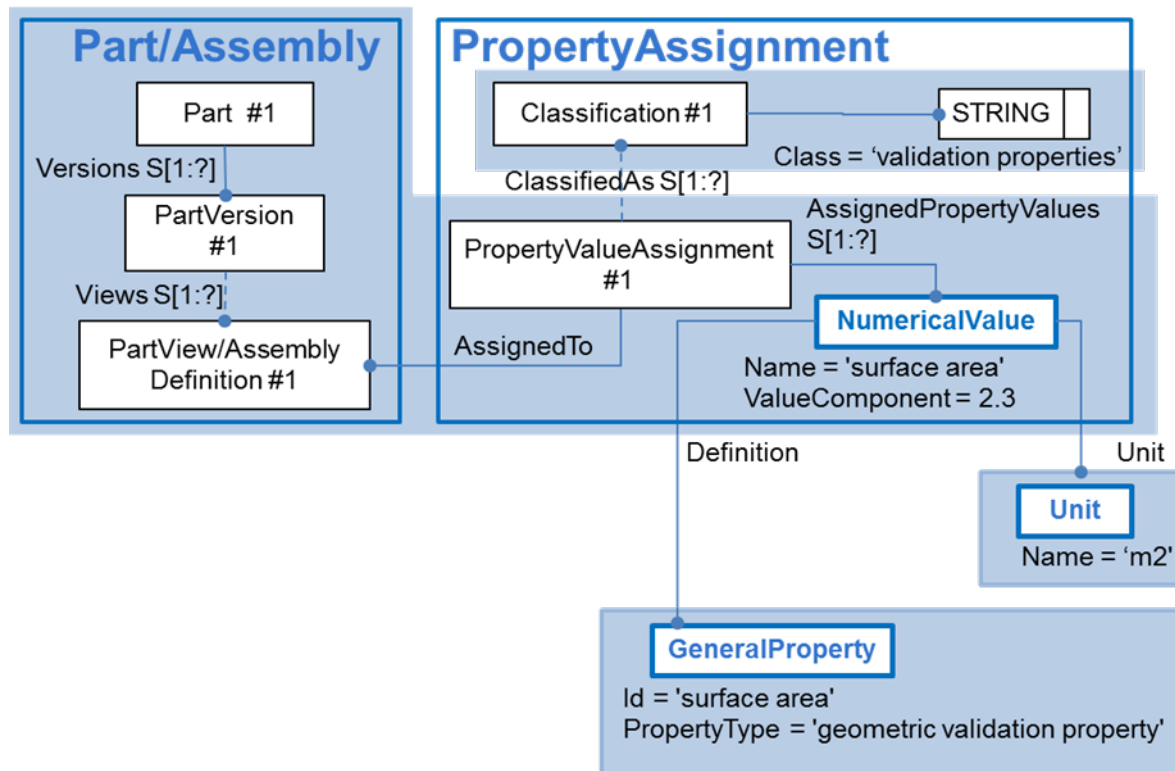


Figure 60: GVP 'surface area' of 2.3 squaremetre assigned to a Part or Assembly

### 13.2.1.3 Solid Centroid

A centroid is the center of volume of a geometric solid model; in the AP242 BO Model this corresponds to the entity CentreOfMass. The position of the centroid is an invariant datum relative to the model origin, thus during an exchange, this can be used to validate the positional integrity of any geometric translations.

If the Part/Assembly for the CentreOfMass has a GeometricModel associated to it, both (CentreOfMass and GeometricModel) shall be defined in the same GeometricCoordinateSpace.

Note that material properties, such as density and weight, are currently not in scope of this document. Hence, the terms “center of mass”, “center of geometry” and “centroid” are used synonymously.

Figure 61 illustrates the relevant entities and their mandatory attributes used in the assignment of a solid centroid for validation. Instantiation follows the same principles as for the AVP 'notional solids centroid'; see 13.1.2.

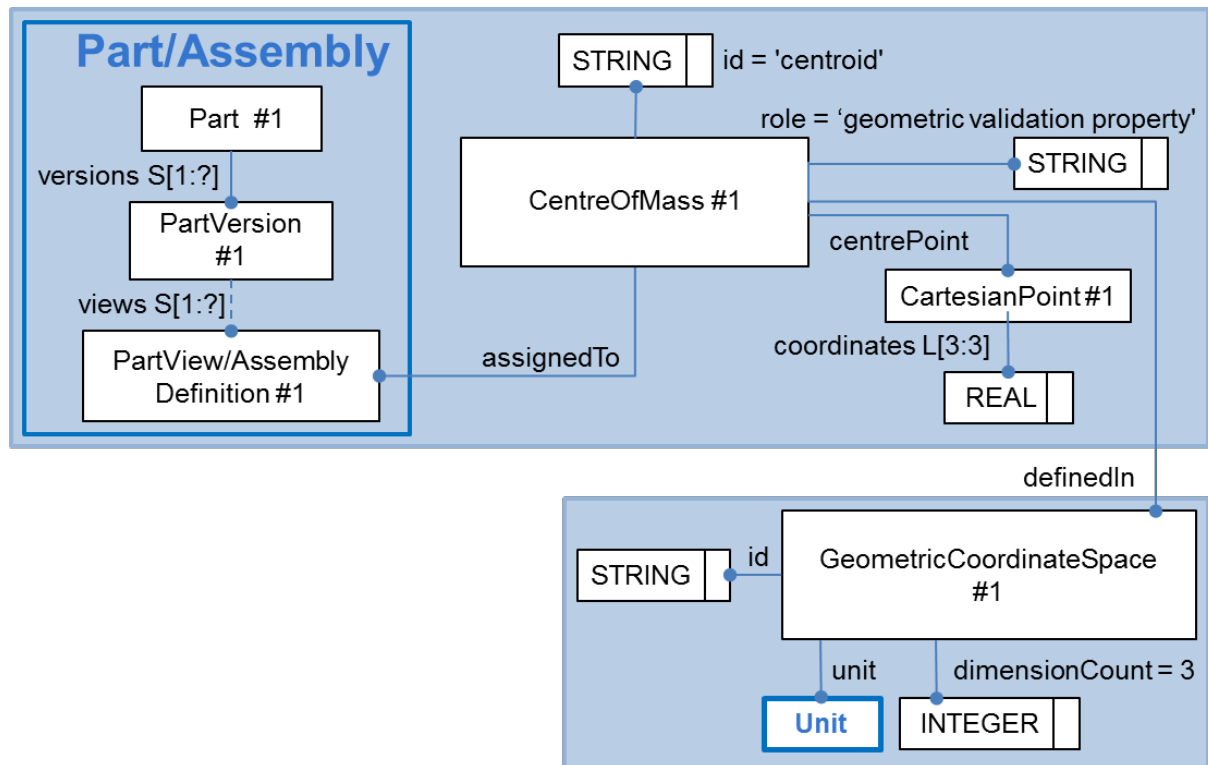


Figure 61: GVP 'centroid' or CentreOfMass assigned to a Part or Assembly

## 13.2.2 Validation Properties for Surface Geometry

### 13.2.2.1 Independent Surface Area

The designation “independent” for a surface means that it is not a face of a solid. Such surfaces can occur as constituents of a surface model (open or closed shell), or as additional elements in a solid model. The total area of these surfaces in a model can be validated to ensure completeness of the exchanged data.

The instantiation follows the exact same pattern as defined in section 13.2.1.2, Figure 60, using the following magic strings instead:

- NumericalValue.name = “independent surface area”

### 13.2.2.2 Independent Surface Centroid

In addition to the total area of independent surfaces (see section above), their positioning is of interest as well. This can be validated using the combined centroid of all independent surfaces in the model.

The instantiation follows the exact same pattern as defined in section 13.2.1.3, Figure 61, using the following magic strings instead:

- CentreOfMass.id = “independent surface centroid”.

## 13.2.3 Validation Properties for Curve / Wireframe Geometry

### 13.2.3.1 Independent Curve Length

The designation “independent” for a curve means that it is not the edge curve of a surface or solid. Such curves can occur as constituents of a wireframe model, or as additional elements in a surface or solid model. The total length of these curves in a model can be validated to make sure no information was lost during transfer. Use cases for this are electric harnesses and piping installations, where independent curves are used as center curves of wires or pipes.

The instantiation follows the exact same pattern as defined in section 13.2.1.2, Figure 60, using the following magic strings instead:

- NumericalValue.name = “independent curve length”

The values of NumericalValue.valueComponent and Unit.name in Figure 60 will need to be changed according to the use case at hand, that is, to the length of a curve and a length unit instead of the area of a surface and an area unit.

### 13.2.3.2 Independent Curve Centroid

In addition to the total length of independent curves in a model (see previous section), their position is of interest as well. The independent curve centroid shall store the combined centroid of all independent curves at the part level. Use cases for this are electric harnesses and piping installations, where an independent curve is used as the center curve of the wire or pipe.

The instantiation follows the exact same pattern as defined in section 13.2.1.3, Figure 61, using the following magic strings instead:

- CentreOfMass.id = “independent curve centroid”.

### 13.2.4 Bounding Box

The bounding box is a means of providing information about the model extent and location. It can be used as a further way of validating the position of the model by providing the space it fits into, in addition to the centroid. As there are many different ways to define a bounding box, the CAx-IF has agreed on a common definition, which uses two three-dimensional points (minimum X, minimum Y, minimum Z) and (maximum X, maximum Y, maximum Z).

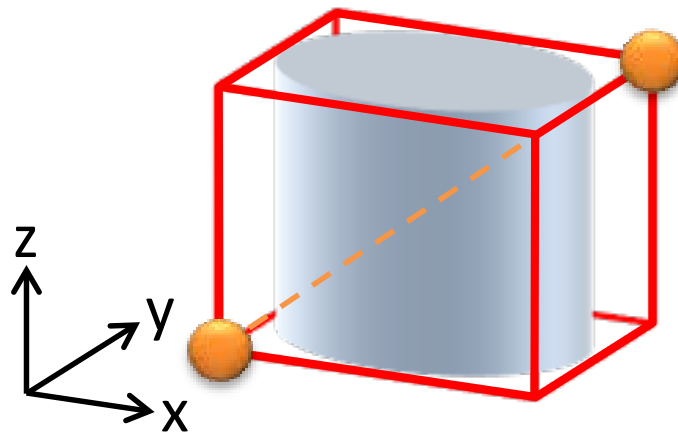


Figure 62: Bounding Box defined by two opposing corner points

The detailed definition is given in section 4.9 of the Recommended Practices for Geometric and Assembly Validation Properties (see Annex C).

The instantiation follows the same pattern as defined in section 13.2.1.3, Figure 61; however there will be two instances of CentreOfMass to the same instance of Part or Assembly, both using the same magic string:

- CentreOfMass.id = “bounding box corner point”.

## 14 Outlook

As written in the introductory sections of this document, the Recommended Practices for AP242 BO Model XML Product and Assembly Structure have been written to guide implementations of the new BO Model XML format for the well-established exchange of product

and assembly information. This document focuses on the core scope in the area of CAD-PDM interaction.

Since meta data and product structures are key data for almost all businesses and life cycle processes, it is clear that the agreements documented in this document have to be discussed with and accepted by all involved communities; users and implementors alike. User groups and implementor forums as described in the introduction in section 1.2 provide the platform for this. Also, in order to guarantee process stability, the documented agreements have to remain stable.

STEP Business Object Models and the XML representations are still a rather new concept, which is only now being implemented on a broad basis. This means that the data model will change over time, in order to address issues discovered during implementation, or to support new requirements. Annex B below gives an overview on such issues that are currently being worked by the AP242 project. In cases where future versions of the AP242 BO Model require changes to existing implementations, the recommended practices will clearly point out these differences and how to support them in pre- and postprocessors.

Building on this core scope, the data scope of AP242 BO Model XML implementations will increase in the future. Functionalities listed as out of scope of this document, for instance Composites or advanced PDM capabilities such as Configuration Management, will be addressed by the respective communities as soon as a stable basis has been established. These specific capabilities will then be defined in separate documents referencing this one where necessary. This will ensure the manageability of this document, and also allow the documentation of in-development capabilities to be updated more frequently without excessive harmonization overhead.

#### **14.1 “Model-based” approach for future versions of this document**

A model based approach for capturing and interrelating recommended practices should be used, in order to allow:

- Easy publication through dynamic web sites;
- Easy change management and consistency with automated regeneration of documents from a valid model;
- An implementable document;
- Allow computer aided verification and validation

Principles introduced in the STEP new architecture:

- Definition of use cases called DEXs, describing the business need with templates vocabulary
- Definition of a building block mechanism called “Core Technical Capabilities”, describing a recommendation of use on a subset of the STEP AP242 BO Model entities

In order to prepare the future versions of this document, the concept of templates has been used (see a list of available templates in chapter 15). The next stage will be the creation of DEXs for automatic verification and validation.

## **15 List of Templates**

This chapter gives a summary of the templates described in this document:

<b>Name</b>	<b>Paragraph</b>
Template “ExchangeContext”	4.6.1

Name	Paragraph
Template "Organization"	4.6.2
Template "Unit"	4.6.3
Template "Class"	4.6.4
Template "Classification"	4.6.5
Template "Identifier"	4.6.6
Template "Description"	4.6.7
Template "ViewContext"	4.6.8
Template "NumericalValue"	4.6.9
Template "StringValue"	4.6.10
Template "DateTime"	4.6.11
Template "Approval"	4.6.12
Template "Person"	4.6.13
Template "PersonInOrganization"	4.6.14
Template "Part"	5.1
Template "Assembly"	5.1
Template "GeometricModel"	6.1
Template "PropertyAssignment"	6.2
Template "ShapeDependentProperty"	6.3
Template "SingleOccurrence"	7.1
Template "SpecifiedOccurrence"	7.2
Template "Simplified Positioning Representation"	7.3.1
Template "Document"	8.1
Template "DocumentDefinitionRelationship"	8.2
Template "DocumentVersionRelationship"	8.3
Template "DigitalFile"	9.1
Template "FileRelationship"	9.2
Template "FormatProperty"	10.1
Template "ContentProperty"	10.2
Template "CreationProperty"	10.3
Template "SizeProperty"	10.4
Template "DocumentFileProperty"	10.5
Template "CAx Representation for DocumentAssignment"	11.1
Template "PDM Representation for DocumentAssignment"	11.2
Template "PropertyDefinition"	12.2

## Annex A XML Schema derivation from BO Model EXPRESS Schema

This annex is derived from the Annex B of **ISO/TS 10303-3001:2014 'Business object model: Managed model based 3D engineering business object model'**, and has been corrected (crossed text) and completed (*text in italic*) to cover all aspects. The crossed and italic sentences are issued to Bugzilla under #5079 through #5087 + #5110.

### A.1 General concepts

This section describes the general concepts for the derivation of the XML Schema from the corresponding BO Model EXPRESS Schema. These concepts were used to create the XML Schema Definition from the EXPRESS Schema using the configuration directives of ISO 10303-28.

#### A.1.1 General concepts

*In general*, the XML name derived from a BO MODEL EXPRESS identifier is the BO MODEL EXPRESS identifier modified with following rules:

- Names of XML ~~tags elements and attributes~~ shall be written using upper camel case.
- ~~For identifiers that will be represented by XML element attributes lower camel case shall be used.~~
- This convention requires removing underscore characters " \_" from EXPRESS names.

*The structure of the EXPRESS model is preserved in XML:*

- *No changes in cardinality*
- *One EXPRESS instance is represented by one XML instance*
  - *No aggregation of several EXPRESS entities into one XML element*
  - *No splitting of one EXPRESS entity into several XML elements*

*Reverting the direction of associations is allowed:*

- *Does not violate the principle of structure preservation*

#### A.1.2 Mapping of EXPRESS entity data types

For each EXPRESS entity data type declaration the XML Schema contain the definition of a new complex type corresponding to that EXPRESS entity data type.

```
<xsd:complexType name="PartVersion">
  <xsd:complexContent>
    <xsd:extension base="cmn:BaseObject">
      <xsd:sequence>
        <!-- the attributes of the entity -->
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

~~For each entity data type that does not inherit from another entity data type a new ComplexType will be declared. For each EXPRESS attribute appearing in the entity declaration, the ComplexType shall contain one corresponding element.~~

*By default, each complexType is based on cmn:BaseObject (defined in common.xsd).*

*For each entity data type that does not inherit from another entity data type, the complexType is based on cmn:BaseRootObject (defined in defined in common.xsd). The BaseRootObjects occur as top level elements in the DataContainer and cannot be contained by any other element.*

```
<xsd:complexType name="Part">
  <xsd:complexContent>
    <xsd:extension base="cmn:BaseRootObject">
      <xsd:sequence>
        ...
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

In addition to the declaration of simple entity data types EXPRESS allows the specification of entity data types as subtypes of other entity data types. This establishes an inheritance relationship (subtype/supertype) and through successive subtype/supertype relationships an inheritance graph in which every instance of a subtype is also an instance of its supertype(s). An entity declared by using inheritance relationships with supertypes is said to be a complex entity data type.

```
<xsd:complexType name="ActualActivity">
  <xsd:complexContent>
    <xsd:extension base="Activity">
      <xsd:sequence>
        <!-- the attributes of the entity -->
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

A complex entity data type inherits not only the EXPRESS attributes and rules appearing in the EXPRESS entity declarations of all of its supertypes, but also all the EXPRESS attributes and rules they inherit. So subtype entities are specialisations of any of their supertypes, where a specialisation means a more constrained form of the original declaration. The mapping of complex entity data types uses the technique of derivation by extension for those entity data types that do not inherit from multiple supertypes. When a complex type is derived by extension, its effective content model is the content model of the base type plus the content model specified in the type derivation.

EXPRESS allows the declaration of entities that are not intended to be directly instantiated. For each EXPRESS entity data type declared to be ABSTRACT, the Schema shall contain an XML element declaration corresponding to the EXPRESS entity data type. The XML element shall be declared to be abstract so it cannot be used in a XML instance document.

There are some exceptions to the mapping described above that do not require one-to-one mapping between EXPRESS entity and XML Schema ComplexType:

- EXPRESS entities that have equivalents that are already defined in XML Schema, e.g., dateTime, duration, language
- Utilizing XML Schema constructs rather than one-to-one mapping e.g., Multilanguage support

### A.1.3 Mapping of named data types

For each defined EXPRESS data type with a final underlying type of STRING, INTEGER, REAL, NUMBER or BOOLEAN the XML schema contains a new element using the corresponding built-in types of XML schema.

### A.1.4 Mapping of SELECT data types

A SELECT data type has a select list where each item shall be an entity data type or a defined data type. SELECT data types that are used in EXPRESS are *mostly* created as *cmn:Reference* IDREF(s) XML data types. *In some cases (like in AxisPlacementOrTransformationSelect for AxisPlacement and CartesianTransformation, or in ClassSelect for ClassString) some elements of the SELECT type are mapped by containment. Therefore it is not needed to create SELECT type definitions.* In order to make resolving of IDREF(s) *cmn:Reference* and containments for SELECT types possible for each SELECT type, an XML Schema Group definition shall be created. It shall contain *the* list of items that belong to *the* given SELECT type. Groups might be used for validation purposes.

```
<xsd:group name="TransformationSelect">
  <xsd:choice>
    <xsd:element name="CartesianTransformation"
type="CartesianTransformation"/>
    <xsd:element name="GeometricRepresentationRelationship"
type="cmn:Reference"/>
  </xsd:choice>
</xsd:group>
```

### A.1.5 Mapping of EXPRESS attributes

For each EXPRESS explicit attribute of an EXPRESS entity data type declaration the corresponding ComplexType in the XML Schema Definition shall contain an element definition. In case of EXPRESS attributes that have simple semantics, XML *elements* shall be used. (e.g., name, description, role, relationType, versionId etc.).

*There are four main cases for EXPRESS attribute mapping:*

- *Single attribute*
  - *by containment*
  - *by reference*
- *Aggregation attribute (SET, BAG, LIST, ARRAY)*
  - *by containment*
  - *by reference*

The order of elements is fixed - this shall be realized with XML Schema sequence grouping.

*The order of the elements in the XSD is defined as following:*

- a) *alphabetic, with some rare exceptions*
- b) *the embedded objects come after the local attributes, also in alphabetic order, with some rare exceptions*
- c) *the inherited attributes come first (top-down from the supertype hierarchy), the local attributes at the very end*

Example:

Entity A



*Entity B, subtype of A*

*Entity C, subtype of B*

*Attributes of C:*

*Inherited attributes from A in alphabetical order*

*Embedded objects from A in alphabetical order*

*Inherited attributes from B in alphabetical order*

*Embedded objects from B in alphabetical order*

*Local attributes of C in alphabetical order*

*Local embedded objects of C in alphabetical order*

Except in very rare exceptions like Identifier.role (mapped to idRoleRef) and Identifier.identificationContext (mapped to idContextRef), the name of the XML element shall be the name of the EXPRESS entity written in upper camel casing style. ~~EXPRESS attributes that are mapped to XML attributes shall be written in lower camel casing style.~~

*Comment: here an issue in Bugzilla has been created under #5094 in order to add these exceptions to the Configuration Specification File referenced by the annex B.2 and contained in the Annex D of the BO Model specification, or at least to describe these renamings in the Annex B*

If the EXPRESS attribute is declared to be OPTIONAL, then the minOccurs pattern of the XML element shall be declared to be "0". ~~The type of the XML element shall be declared according to the data type of the EXPRESS attribute.~~

```
<xsd:complexType name="Activity">
  <xsd:complexContent>
    <xsd:extension base="cmn:BaseRootObject">
      <xsd:sequence>
        <xsd:element name="Requestor" type="DateAndPersonOrganization"
minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

*An INVERSE attribute is constrained to a SET or a BAG [1:1] of exactly one element (like ApprovingPersonOrganization. authorizedApproval of type Approval), as all INVERSE attributes, shall not be mapped to the XSD, but shall be used as an indicator to apply the XML containment.rules (i.e. to define ApprovingPersonOrganization as being contained into Approval). For more details, see below.*

The re-declaration of EXPRESS attributes shall have no effect on XML schema declaration.

#### **A.1.5.1 EXPRESS attribute types corresponding to XML complex type**

The XML element corresponding to an EXPRESS attribute type that is an EXPRESS entity data type shall be mapped in one of following ways:

- XML element type shall be the name of a complexType defined in XML Schema. Element in XML document shall be instantiated inside a parent element i.a. attribute shall be represented by containment.
- XML element type shall be defined as *type="cmn:Reference" IDREF(s)*. The IDREF attribute shall reference the uid attribute of an XML complexType corresponding to the EXPRESS entity data type of the attribute.

The XML *attribute element* corresponding to an EXPRESS attribute whose data type is a SELECT data type shall reference the XML group defined for the SELECT be declared to have type IDREF in the XML schema declaration.

```
<xsd:complexType name="AssemblyOccurrenceRelationship">
  <xsd:complexContent>
    <xsd:extension base="ViewOccurrenceRelationship">
      <xsd:sequence>
        <xsd:element name="Placement" minOccurs="0">
          <xsd:group ref="TransformationSelect" minOccurs="0"/>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

*cmn:Reference in XML is untyped. To map the type of the referenced object from the EXPRESS schema, xsd:ref and xsd:keyref definitions are defined in the XSD. Based on XPATH, each xsd:keyref definition lists all the places where the referenced object may occur together with the xsd:ref of the referenced Entity. There is one xsd:ref per Entity and one xsd:keyref per attribute of type cmn:Reference plus one xsd:keyref per Entity. This allows automatic consistency check for XML file during XML Schema validation*

### A.1.5.2 EXPRESS attribute types corresponding to XML simple type

The XML element corresponding to an EXPRESS attribute whose data type is a defined data type with a final underlying type of STRING, INTEGER, REAL, NUMBER, or BOOLEAN shall be declared to have the XML type corresponding to the underlying type of the defined data type in the EXPRESS type declaration, if not otherwise specified in a configuration directive:

EXPRESS attribute type	ISO XML element type
NUMBER, REAL	xs:double
INTEGER	xs:integer
STRING	xs:string
BOOLEAN	xs:boolean

*In the TC version of AP242, LOGICAL is mapped to a simpleType having a <xsd:restriction base="xsd:string">. Each value element is defined within it as <xsd:enumeration value="..."/>.*

```
<xsd:simpleType name="logical">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="false"/>
    <xsd:enumeration value="true"/>
    <xsd:enumeration value="unknown"/>
  </xsd:restriction>
</xsd:simpleType>
```

*EXPRESS ENUMERATION types are mapped to a simpleType having a <xsd:restriction base="xsd:string">. Each value element is defined within it as <xsd:enumeration value="..."/>.*

```
<xsd:simpleType name="ActuatedDirectionEnum">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="bidirectional"/>
    <xsd:enumeration value="positive_only"/>
    <xsd:enumeration value="negative_only"/>
    <xsd:enumeration value="not_actuated"/>
  </xsd:restriction>
</xsd:simpleType>
```

If an EXPRESS attribute type REAL is mapped to an XML element type STRING the format of this content string shall be according IEEE 754-1985.

### A.1.5.3 Attributes with aggregate data types

*Aggregations (by containment) are mapped as a sequence of elements with the EXPRESS type as name and type:*

```
<Part uid="p--000000001E720B30">
  ...
  <Versions>
    <PartVersion uid="pv--000000001E720B30-id7">
      ...
      <Views>
        <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--
000000001E720B30-id7">
          ...
          </PartView>
        <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--
000000001E720B30-id8">
          ...
          </PartView>
        ...
      </Views>
    </PartVersion>
  </Versions>
  ...
</Part>
```

*Aggregations (by reference) are represented as a sequence of elements with the EXPRESS type as name and "cmn:Reference" as type:*

```
<xsd:complexType name="Activity">
  <xsd:complexContent>
    <xsd:extension base="cmn:BaseRootObject">
      <xsd:sequence>
        <xsd:element name="PossibleMethods" minOccurs="0">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="ActivityMethod" type="cmn:Reference" max-
Occurs="unbounded"/>
            </xsd:sequence>
          ...
        ...
      </xsd:sequence>
    ...
  </xsd:complexContent>
</xsd:complexType>
```

EXPRESS provides four kinds of aggregation data types: ARRAY, LIST, BAG and SET. These data types have as their domains collections of values of a given base data type where the base data type can be a simple type, a named type or another aggregation type.

The XML element corresponding to an aggregate valued EXPRESS attribute shall be declared to be of type "IDREFS" in case of reference or for containment usage. In both cases, they shall make use of maxOccurs="unbounded".

Caution: "IDREFS" or maxOccurs="unbounded" most match the EXPRESS Type BAG, since redundant values are allowed. The uniqueness of the elements (like in a SET), the indexing of the elements (like in a LIST or an ARRAY) and OPTIONAL ARRAY values are not supported.

*Multi-Dimensional Aggregates are mapped as One-Dimensional, for example: MomentsOfInertia.InertiaValue of kind ARRAY[1:3] OF ARRAY[1:3] OF NumericalValue; is mapped to:*

```
<xsd:element name="NumericalValue" type="NumericalValue" maxOccurs="9"/>
```

*An exception is made for CartesianTransformation.RotationMatrix defined as LIST[2:3] OF LIST[2:3] OF LengthMeasure, which is mapped to xsd:string for reasons of compactness.*

*In both cases, the separator is a blank et the ordering of the elements is as following: xx xy xz yx yy yz zx zy zz*

*Some One-Dimensional Aggregates have been also mapped to xsd:string for reasons of compactness:*

- *Direction.DirectionRatios (LIST[2:3] OF REAL)*
- *CartesianTransformation.TranslationVector (LIST[2:3] OF LengthMeasure)*
- *AxisPlacement.Axis/Position/RefDirection (LIST[2:3] OF LengthMeasure)*
- *CartesianPoint.Coordinates (LIST[2:3] OF LengthMeasure)*

*Here the order is obvious: x y z*

### **A.1.6 Not mapped EXPRESS Constructs**

*The UNIQUE, WHERE and global rules are not mapped to XML*

*Also the uniqueness of the elements within a SET, UNIQUE LIST or UNIQUE ARRAY is not mapped to XML.*

### **A.1.7 Containment and referencing rules**

*An EXPRESS attribute whose data type is an EXPRESS entity data type shall be mapped in one of the following ways:*

- *By containment*
- *By reference*

*Containment is the preferred approach*

- *It is recommended to use containment wherever possible*
- *Reference should only be used for elements that are commonly reused*

*Motivation:*

- *To place as much information as possible about an object within it highly increases human readability*
- *Less complexity with script based analysis (e.g. XSLT).*
- *Increases the XSD validation quality*

Reference mapping rules:

- Master data - for elements defined directly under XML root element e.g., Organization, Person
- Structure elements - for elements that are reused and referenced as structure elements e.g., Documents, Parts

Here the following criteria apply:

1. In case the referenced type is not an aggregate and is not a SELECT type: only the name of the attribute is mapped to the XSD, not the name of the referenced entity (since it is implicitly clear). In the case where a subtype is referenced, the subtype has to be given in a xsi:type-clause.

```
<xsd:complexType name="ActivityAssignment">  
  <xsd:complexContent>  
    <xsd:extension base="cmn:BaseObject">  
      <xsd:sequence>  
        <xsd:element name="AssignedActivity" type="cmn:Reference"/>  
      </xsd:sequence>  
    </xsd:extension>  
  </xsd:complexContent>  
</xsd:complexType>
```

Instantiated as:

```
<ActivityAssignment uid="ID_747">  
  <AssignedActivity uidRef="ID_400"/>  
</ActivityAssignment>
```

Remark: within the TC changes to the XSD, aggregates like:

```
<xsd:complexType name="Occurrence" abstract="true">  
  <xsd:complexContent>  
    <xsd:extension base="cmn:BaseObject">  
      <xsd:sequence>
```

...

```
        <xsd:element name="ShapeElement" type="cmn:Reference" minOccurs="0" maxOccurs="unbounded"/>  
      </xsd:sequence>  
    </xsd:extension>  
  </xsd:complexContent>  
</xsd:complexType>
```

and instantiated as:

```
<Occurrence xsi:type="n0:SingleOccurrence" uid="pi--  
000000001EB04CF0--10">  
  <Id id="plate.1"/>  
  <ShapeElement uidRef="se-123"/>  
  <ShapeElement uidRef="se-124"/>  
</Occurrence>
```

...

have been either changed to containments or change to case 2:

2. In case the referenced type is an aggregate or a SELECT type: the name of the attribute and the name of the referenced are given.

```
<xsd:element name="Description" minOccurs="0">  
  <xsd:complexType>  
    <xsd:group ref="DescriptorSelect" minOccurs="0"/>  
  </xsd:complexType>  
</xsd:element>
```

Instantiated as:

```
<ExchangeContext uid="ec--1">
```

...

```
<Description>  
  <Descriptor uid="id123">  
    <Text>  
      <CharacterString>abcd</CharacterString>  
    </Text>  
  </Descriptor>  
</Description>
```

</Descriptor>

Or in the case of an aggregate:

```
<xsd:element name="SameAs" minOccurs="0">
  <xsd:complexType>
    <xsd:group ref="ProxySelect" minOccurs="0" max-
Occurs="unbounded" />
  </xsd:complexType>
</xsd:element>
```

Instantiated as:

```
<SameAs>
  <ExternalOwlObject uidRef="ID_382" />
  <ProxyString>test</ProxyString>
  <ExternalItem uidRef="ID401" />
  <Proxy uidRef="ID402" />
</SameAs>
```

Containment mapping rules:

- Simple attributes e.g., String values
- Elements that cannot exist *standalone* but depend from another object and cannot be reused e.g., DateTime, TranslatedString, PropertyValue
- Grouping of elements in master-revision pattern, like Part -> PartVersion, Document -> DocumentVersion (*usually the EXPRESS schema defines a mandatory INVERSE attribute for the contained element, for example PartVersion.versionOf*)
- For relationships containment shall be performed along the relating attribute (*see below*)
- ~~For EXPRESS SET attributes that shall be represented as containment plural form shall be changed to singular form, e.g., ConcernedOrganizations -> ConcernedOrganization. It is needed to keep semantics correctness (maxOccurs="unbounded")~~
- A contained element cannot be defined as BaseRootObject
- ~~If a containment is made along an attribute of kind SELECT type, the above rule shall apply to all members of the SELECT type~~

Here the following criteria apply:

1. In case the contained type is not an aggregate and is not a SELECT type: only the name of the attribute is mapped to the XSD, not the name of the contained entity (since it is implicitly clear). In the case where a subtype is contained, the subtype has to be given in a xsi:type-clause.

```
<xsd:complexType name="AssemblyJoint">
  <xsd:complexContent>
    <xsd:extension base="cmn:BaseObject">
      <xsd:sequence>
        <xsd:element name="AssemblyShape" type="AssemblyDefinition"/>
        ...
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

2. In case the contained type is an aggregate or a SELECT type: the name of the attribute and the name of the contained entity are given.

```
<xsd:complexType name="Part">
  <xsd:complexContent>
    <xsd:extension base="cmn:BaseRootObject">
      <xsd:sequence>
        ...
        <xsd:element name="Versions">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="PartVersion" type="PartVersion" max-
                Occurs="unbounded"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Instantiated as:

```
<Part uid="p--0000000017D374A0">
  ...
  <Versions>
    <PartVersion uid="pv--0000000017D374A0--id1">
      ...
    </PartVersion>
    <PartVersion uid="pv--0000000017D374A0--id2">
      ...
    </PartVersion>
  </Versions>
</Part>
```

3. In case of change of direction for associations, see the next paragraph

## A.1.8 Change of Direction for Associations

*For various associations in EXPRESS, the direction has been inverted:*

- *The original attribute is omitted*
- *A new attribute is added to the originally reference entity*

```
ENTITY MeasuredCharacteristic
...
measureActivity : OPTIONAL MeasureActivitySelect;
...
TYPE MeasureActivitySelect = SELECT
(Activity,...

<xsd:complexType name="Activity" abstract="true">
  <xsd:complexContent>
    <xsd:extension base="cmn:BaseRootObject">
      <xsd:sequence>
```

```
<xsd:element name="MeasuredCharacteristic"
type="MeasuredCharacteristic" minOccurs="0" maxOccurs="unbounded"/>
```

The advantage of this 'reverse' mapping is to see all the characteristics (like dates, approvals, projects, persons and organizations, properties, ...) of one object at one single place: within the object.

### A.1.8.1 Entities of kind Relationship

Entities of kind ...Relationship, which are instantiated once in EXPRESS and reference one or many relating instance, are mapped (and instantiated separately) in the relating object. The relating attribute is omitted.

The related attribute is mapped by reference.

Example:

```
ENTITY ActivityRelationship;
...
relating : Activity;
related : Activity;
...
END_ENTITY;

<xsd:complexType name="Activity" abstract="true">
  <xsd:complexContent>
    <xsd:extension base="cmn:BaseRootObject">
      <xsd:sequence>
        <xsd:element name="ActivityRelationship"
type="ActivityRelationship" minOccurs="0" maxOccurs="unbounded"/>
        ...
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
  <xsd:complexType name="ActivityRelationship">
    <xsd:complexContent>
      <xsd:extension base="cmn:BaseObject">
        <xsd:sequence>
          <xsd:element name="Related" type="cmn:Reference"/>
          ...
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:complexType>
```

### A.1.8.2 Entities of kind Assignment

Entities of kind ...Assignment, which are instantiated once in EXPRESS and reference one or many 'assignedTo' instances, are mapped (and instantiated separately) in each assignedTo object.

The assignedTo attribute is omitted. The assignedXxx attribute is mapped by reference.

Example:

```
ENTITY DateTimeAssignment;
  id : OPTIONAL IdentifierSelect;
  description : OPTIONAL DescriptorSelect;
  classifiedAs : OPTIONAL SET[1:?] OF Classification;
  role : ClassSelect;
  assignedDate : DateTimeString;
  assignedTo : SET[1:?] OF DateTimeAssignmentSelect;
END_ENTITY;
```

is mapped to XML in each member element of DateTimeAssignmentSelect (they are about 130!), for example in Activity:

```
<xsd:complexType name="Activity" abstract="true">
```



```
<xsd:complexContent>
  <xsd:extension base="cmn:BaseRootObject">
    <xsd:sequence>
      ...
      <xsd:element name="DateTimeAssignment" type="DateTimeAssignment"
minOccurs="0" maxOccurs="unbounded"/>
      ...
    </xsd:sequence>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

The semantic is nearly the same (at least within the scope of a single XML file): the SET attribute `DateTimeAssignment.AssignedTo` can be computed out of all `DateTimeAssignment`s where `DateTimeAssignedDate` point to the same `DateTimeString`.

Remark: in the TC release, all `assignedTo` attributes defined as `SET[1:?]` will be changed to single values (no SETs anymore).

### A.1.9 Representation of Id Attribute

Two representations are supported:

- Compact representation of the most used variant „simple string“
- Optimized representation for the other variants
  - Associated context and/or role
  - Multiple identifiers

```
<xsd:complexType name="Id">
  <xsd:sequence>
    <xsd:element name="Identifier" type="Identifier" minOccurs="0" max-
Occurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string" use="optional"/>
</xsd:complexType>
<xsd:complexType name="Identifier">
  <xsd:complexContent>
    <xsd:extension base="cmn:BaseObject">
      ...
      <xsd:attribute name="id" type="xsd:string" use="optional"/>
      <xsd:attribute name="idRoleRef" type="xsd:string" use="optional"/>
      <xsd:attribute name="idContextRef" type="xsd:string" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

#### Notes

- `idRoleRef` references the `uid` of a `Class`, `ExternalClass` or `ExternalOwlClass`
- `idContextRef` references the `uid` of an `Identifier` or an `Organization`
- if `Id.Identifier` is set, `Id.id` shall not be set

#### Examples:

```
<Organization uid="o-1">
```

```
<Id id="mercedes-benz.com"/>
</Organization>
<Part uid="p-1">
  <Id>
    <Identifier uid="pid-1" id="as1" idContextRef="o-1"/>
  </Id>
  ...
</Part>
<Part uid="p-1">
  <Id>
    <Identifier uid="pid-1" id="as1" idContextRef="o-1"/>
    <Identifier uid="pid-2" id="assy1" idContextRef="o-2"/>
  </Id>
  ...
</Part>
```

### A.1.10 Multilanguage Support

Two representations are supported:

- Compact text strings with optional language indication
- `xsd:language` is used for the language indication
  - Country and language code conforming to RFC 3066

```
<xsd:group name="MultiLingualStringSelect">
  <xsd:choice>
    <xsd:element name="CharacterString" type="xsd:string"/>
    <xsd:element name="LocalizedString" type="LocalizedString" max-
Occurs="unbounded"/>
  </xsd:choice>
</xsd:group>
<xsd:complexType name="LocalizedString" mixed="true">
  <xsd:complexContent mixed="true">
    <xsd:extension base="cmn:BaseObject">
      <xsd:attribute name="lang" type="xsd:language" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Example:

```
<ActivityMethod uid="am-2">
  <Consequence>
    <LocalizedString>no consequence</LocalizedString>
    <LocalizedString lang="en-GB">no consequence</LocalizedString>
    <LocalizedString lang="de-DE">keine Konsequenz</LocalizedString>
    <LocalizedString lang="fr-FR">aucun conséquence</LocalizedString>
  </Consequence>
</ActivityMethod>
```

The AP242 specification refers to ISO 639-2 for the language code and to ISO 3166-1 for the country code. They enable the specification of a language code optionally followed by a country code, for example 'en' or 'en-US'. The EXPRESS TYPE Language defined as LIST[1:2] OF STRING is mapped to XML as `xsd:language`, where the language code and the country code are concatenated (the country code is optional).

The W3C definition of `xsd:language` englobes these two ISO standards plus further ones in RFC 3066 (IANA and unofficial languages) => these shall not be used.

### A.1.11 Representation of Date and Time

`xsd:dateTime` is used instead of `String`.

As in EXPRESS Time is not optional, to avoid conversion problems it shall be provided as "T00:00:00".

## A.2 Unit of Serialization

~~The EXPRESS Schema is mapped to AP242\_BusinessObjectModel.xsd into a DataContainer. To be a valid XML, a DataContainer has to be included into an UoS object (Unit of Serialization) defined in common.xsd. The UoS contains a mandatory header element that contains administrative information that characterizes the content of the data package.~~

The header elements are described in ISO 10303-28:2007, section 5.2, as follows:

- *Name*: human readable identifier for the XML resource
- *TimeStamp*: date and time when the XML resource was created
- *Author*: identifies the person or group of persons who created the XML resource
- *Organization*: identifies the organization that created, or is responsible for, the XML resource
- *PreprocessorVersion*: identifies the software system that created the XML resource itself, including platform and version identifiers.

*NOTE*: The `preprocessor_version` will identify the system that was used to produce the XML resource. It may well be distinct from the software system that created or captured the original information.

- *OriginatingSystem*: identifies the software system that created or captured the information contained in the XML resource, including platform and version identifiers
- *Authorization*: specifies the release authorization for the XML resource and the signatory, where appropriate.

*NOTE*: This may be distinct from the authorizations for various information units contained within the document.

- *Documentation*: free text field for information

`common.xsd` is defined under <http://standards.iso.org/iso/ts/10303/-3000/-ed-1/tech/xml-schema/common>.

## A.3 XML configuration specification

This section contains the configuration specification.

```
<iso_10303_28 version="" xmlns="urn:iso:std:iso:10303:-28:ed-2:tech:XMLschema:document" xmlns:cnf="urn:iso:std:iso:10303:-28:ed-2:tech:XMLschema:configuration_language" xmlns:exp="urn:iso:std:iso:10303:-28:ed-2:tech:XMLschema:common" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:iso:std:iso:10303:-28:ed-2:tech:XMLschema:document
../../../../implementation_resources/iso10303_28_document_schema/doc.xsd
urn:iso:std:iso:10303:-28:ed-2:tech:XMLschema:configuration_language
../../../../implementation_resources/iso10303_28_configuration_language_schema/cn
```



---

```
f.xsd urn:iso:std:iso:10303:-28:ed-2:tech:XMLschema:common  
../../../../implementation_resources/iso10303_28_base_xml_schema/exp.xsd ">
```

## Annex B Known Issues

This section lists known issues with the AP242 Business Object Model. These include errors in the XSD, mismatches between the EXPRESS and XML schemas, deficiencies in the documentation and other issues that have already been communicated to the AP242 maintenance / development team for resolution.

Issue# (Link)	Short Description	Target Milestone	Status Februar 2017
<a href="#">#5078</a>	Instantiation of SpecifiedOccurence in XSD is not possible	AP242e1 TC	Resolved in TC1
<a href="#">#5079</a>	Definition and support of UNIQUE rules	AP242e1 TC	Confirmed - De-ferred
<a href="#">#5080</a>	Definition and support of WHERE rules	AP242e2 CD	Confirmed - De-ferred
<a href="#">#5081</a>	Cardinality Constraints	---	Rejected
<a href="#">#5082</a>	Multidimensional Aggregates	AP242e1 TC	Resolved in TC1
<a href="#">#5083</a>	Mapping of the INVERSE attributes	AP242e1 TC	Resolved in TC1
<a href="#">#5084</a>	Mappping of attributes of type LOGICAL	AP242e1 TC	Resolved in TC1
<a href="#">#5085</a>	Mapping of attributes of type ENUMERATION	AP242e1 TC	Resolved in TC1
<a href="#">#5086</a>	Multiple inheritance for ConditionalEffectivities and the other sub-types of Effectivity	AP242e2 CD	Confirmed - De-ferred
<a href="#">#5087</a>	key and keyref	AP242e1 TC	Resolved in TC1
<a href="#">#5088</a>	URL of XML Schema	AP242e1 TC	Resolved in TC1
<a href="#">#5089</a>	idRoleRef	AP242e1 TC	Resolved in TC1
<a href="#">#5090</a>	Two identifier values are possible in XML	AP242e1 TC	Resolved in TC1
<a href="#">#5091</a>	CartesianTransformation.TranslatonVector	AP242e1 TC	Resolved in TC1
<a href="#">#5092</a>	AxisPlacement.Position	AP242e1 TC	Resolved in TC1
<a href="#">#5093</a>	CreationProperty.CreatingSystem	AP242e1 TC	Resolved in TC1
<a href="#">#5094</a>	EXPRESS attributes renamed in the XSD	AP242e1 TC	Resolved in TC1
<a href="#">#5095</a>	ViewOccurrenceRelationship.relationType	AP242e1 TC	Resolved in TC1
<a href="#">#5110</a>	common.xsd	---	Rejected
<a href="#">#5111</a>	File.FileType	AP242e1 TC	Resolved in TC1
<a href="#">#5153</a>	semantical type checking in bom.xsd for PartView.ClassifiedAs	AP242e1 TC	Resolved in TC1

Issue# (Link)	Short Description	Target Milestone	Status Februar 2017
<a href="#">#5155</a>	ShapeElement defined as RootObject, but misses the attribute 'ElementOf'	AP242e1 TC	Resolved in TC1
<a href="#">#5558</a>	Header.Documentation as LIST	AP242e1 TC	Resolved in TC1
<a href="#">#5619</a>	introduce 'GeneralGeometricRepresentationRelationship' as subtype of 'GeometricRepresentationRelationship' with attribute 'RelationshipType'	AP242e1 TC	Resolved in TC1
<a href="#">#5620</a>	change PartView.DefiningGeometry to a	AP242e1 TC	Resolved in TC1
<a href="#">#5664</a>	RepresentationItem shall be referenced rather than being embedded	AP242e1 TC	Resolved in TC1
<a href="#">#5665</a>	RepresentationItem.External should be a reference and not embedded	---	Rejected
<a href="#">#5666</a>	Clarifications needed concerning Kinematics	AP242e1 TC	Resolved in TC1
<a href="#">#5667</a>	Incorporate Part105 into the BO Model documentation	AP242e1 TC	Resolved in Rec. Practices
<a href="#">#5668</a>	Embed KinematicLinkToOccurrenceAssociation into KinematicLink	AP242e1 TC	Resolved in TC1
<a href="#">#5669</a>	embed KinematicMechanismAssociation into Mechanism	AP242e1 TC	Resolved in TC1
<a href="#">#5884</a>	Two representation of Id and concept of IdentifierSet, DescriptorSet and TranslatedStringSet in the XSD	AP242e1 TC2	Confirmed
<a href="#">#6020</a>	Enable PropertyValueAssignments on Document	AP242e1 TC2	Confirmed
<a href="#">#6021</a>	Clarify added-value of GeneralShapeDependentProperty.PropertyType	AP242e1 TC2	Confirmed
<a href="#">#6022</a>	Combined use of PartViewRelationship and ViewOccurrenceRelationship	AP242e1 TC2	Confirmed
#6273	Definition of the order of the elements in the XSD	AP242e1 TC2	Confirmed
<a href="#">#6290</a>	Dedicated XML elements for the transfer of units without value	AP242e1 TC2	Confirmed
#6291	Add 'Role' to ApprovalAssignment	AP242e1 TC2	Confirmed

## Annex C Reference Documents

This recommended practices document is based on and derived from various other documents, schemas, and technical presentations. Those resources are listed below:

- AP242 IS BO Model XML / EXPRESS Schema
  - Dated May 5, 2014
- Model Usage Guidance for STEP AP242 BO Model CAD Exchange (DOC)
  - Release 0.2, July 16, 2012
  - Authors: F. Darré, A. Fournier
- STEP AP242 XML Nested Assembly Approach (PPT)
  - February 24, 2014
  - Authors: M. Ungerer, G. Hirel
- AP242 BO Model IS - Simplified Shape Association and Transformation (PPT)
  - February 24, 2014
  - Author: M. Ungerer
- CAX-IF Recommended Practices (<http://www.cax-if.de/> - <http://www.cax-if.org/>)
  - Geometric and Assembly Validation Properties: Release 4.4; Aug. 17, 2016
  - External References: Release 3.1; Jan. 20, 2014
  - User Defined Attributes: Release 1.5; Aug. 15, 2016
  - STEP File Compression: Release 1.2; Aug. 15, 2016
  - PDM Schema Usage Guide: Release 4.3; Jan., 2002
- LOTAR Part 115 “Explicit CAD Assembly Structure”
- VDA Recommendation 4956-1 “Assembly Data Exchange” ([http://www.prostep.org/fileadmin/freie\\_downloads/Empfehlungen-Standards/VDA/VDA\\_4956-1\\_Product-Data-Management\\_1.0.pdf](http://www.prostep.org/fileadmin/freie_downloads/Empfehlungen-Standards/VDA/VDA_4956-1_Product-Data-Management_1.0.pdf)).
- JT Application Benchmark 2013-2016 Experiences
- MIL-STD-31000A (<http://model-based-enterprise.org/mil-standards.html>)

## Annex D Conversion from implicitly to explicitly defined transformation

The first step is now to extract the two matrices implicitly given by each of the two placements. The AxisPlacement has a name, a location and two axes as attributes. The axes are the axis and ref\_direction attribute, where axis is the placement Z axis direction and the ref\_direction is an approximate to the placement X axis direction. From this information, a right-handed coordinate system is computed:

Let  $z$  be the placement Z axis direction and  $a$  be the approximate placement X axis direction. Approximate here means that  $a$  and  $z$  are not required to be orthogonal. Then the exact placement X axis direction is given as  $x = \langle a - (a \cdot z)z \rangle$  and the placement Y axis direction calculates to  $y = \langle z \times x \rangle$ .

For the first representation item, the following calculations would result:

$$\text{axis: } z = \begin{pmatrix} 0.0 \\ -0.8660254 \\ 0.5 \end{pmatrix}, \quad \text{ref\_direction: } a = \begin{pmatrix} 1.0 \\ 0.0 \\ 0.0 \end{pmatrix},$$

therefore  $x = \langle a - (a \cdot z)z \rangle = \langle a - 0z \rangle = a$  because  $z$  and  $a$  are already orthogonal in this example.

Next step is calculating  $y$  using the vector product:  $y = \langle z \times x \rangle = \begin{pmatrix} 0.0 \\ 0.5 \\ 0.8660254 \end{pmatrix}$ .

So the geometric function which leads from the coordinates of the global coordinate system to those of the first axis placement is represented by the rotation matrix  $A$  given by the three vectors  $x$ ,  $y$  and  $z$  plus the translation vector  $t$  given by the AxisPlacement's location attribute:

$$A = \begin{pmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.5 & -0.8660254 \\ 0.0 & 0.8660254 & 0.5 \end{pmatrix}, \quad t = \begin{pmatrix} 1.0 \\ 1.0 \\ 3.0 \end{pmatrix}$$

In the same way, the matrix  $B$  and the vector  $u$  are computed from the second axis placement:



$$B = \begin{pmatrix} 0.7071068 & -0.7071068 & 0.0 \\ 0.7071068 & 0.7071068 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{pmatrix}, \quad u = \begin{pmatrix} 2.0 \\ 2.0 \\ 1.0 \end{pmatrix}$$

Geometrically, the matrix  $A$  defines a  $60^\circ$  rotation around the global X axis and the matrix  $B$  gives a  $45^\circ$  rotation around the global Z axis.

To get the explicit transformation from the information gained so far, the matrices have to be combined. The idea is as follows: To move a point from a location within the first placement (called 'source') into a location within the second one (called 'target'), three steps have to be made:

First, the point has to be multiplied with the inverted matrix  $A^{-1}$  to undo the rotation, which occurs when going from the global coordinate system into the first placement system.

Next, multiply it with the second matrix  $B$  to get it into the right position for the target placement.

Finally, a translation vector is needed to put the point into its correct location within the second axis placement. Calculation of this vector can be seen below.

As  $A$  is a rotation matrix, the inverted matrix  $A^{-1} = A^T$ , the transposed matrix. Steps 1 and 2 can be combined:

$$C = BA^{-1} = \begin{pmatrix} 0.7071068 & -0.3535534 & -0.6123724 \\ 0.7071068 & 0.3535534 & 0.6123724 \\ 0.0 & -0.8660254 & 0.5 \end{pmatrix}$$

The translation vector needed is

$$v = u - Ct = \begin{pmatrix} 3.4835639 \\ -0.8977775 \\ 0.3660254 \end{pmatrix}$$

This means moving any point  $P$  from a location within the first placement into the second one follows the calculation

$$P' = C \cdot P + v$$

## Annex E Recommendation for the Definition of Units<sup>2</sup>

This clause provides recommendations for instance population for the definition of units in the data set. Once the definition is created, other data instances reference the units as required.

For the use of this recommended practices we take the assumption that the partners have agreed beforehand which units to use in a project.

**Note:** The definitions given in this Annex are valid for AP242 edition 1 BO Model schema.

### E.1 SI Base Unit Definitions

The following is the recommendation for exchange of SI base unit definitions<sup>3</sup>:

Millimetre:

```
<Unit uid="u--100000001">
  <Kind><ClassString>SI system</ClassString></Kind>
  <Name><ClassString>metre</ClassString></Name>
  <Prefix><ClassString>milli</ClassString></Prefix>
  <Quantity><ClassString>length</ClassString></Quantity>
</Unit>
```

Kilogram:

```
<Unit uid="u--100000002">4
  <Kind><ClassString>SI system</ClassString></Kind>
  <Name><ClassString>gram</ClassString></Name>
  <Prefix><ClassString>kilo</ClassString></Prefix>
  <Quantity><ClassString>mass</ClassString></Quantity>
</Unit>
```

Seconds:

```
<Unit uid="u--100000003">
  <Kind><ClassString>SI system</ClassString></Kind>
  <Name><ClassString>second</ClassString></Name>
  <Quantity><ClassString>time</ClassString></Quantity>
</Unit>
```

Ampère:

```
<Unit uid="u--100000004">
  <Kind><ClassString>SI system</ClassString></Kind>
  <Name><ClassString>ampere</ClassString></Name>
  <Quantity><ClassString>electric current</ClassString></Quantity>
</Unit>
```

Kelvin:

```
<Unit uid="u--100000005">
  <Kind><ClassString>SI system</ClassString></Kind>
  <Name><ClassString>kelvin</ClassString></Name>
  <Quantity><ClassString>thermodynamic temperature</ClassString></Quantity>
</Unit>
```

Mole:

```
<Unit uid="u--100000006">
  <Kind><ClassString>SI system</ClassString></Kind>
  <Name><ClassString>mole</ClassString></Name>
  <Quantity><ClassString>amount of substance</ClassString></Quantity>
</Unit>
```

---

<sup>2</sup> <http://www.nist.gov/pml/wmd/metric/si-units.cfm>

<sup>3</sup> [https://en.wikipedia.org/wiki/SI\\_base\\_unit](https://en.wikipedia.org/wiki/SI_base_unit)

<sup>4</sup> This instance is created to support definition of SI derived units and is the formal definition that the kilo-gram is the SI unit of mass.

Candela:

```
<Unit uid="u--100000007">  
  <Kind><ClassString>SI system</ClassString></Kind>  
  <Name><ClassString>candela</ClassString></Name>  
  <Quantity><ClassString>luminous intensity</ClassString></Quantity>  
</Unit>
```

## E.2 SI Derived Units

List of SI derived units from SI derived unit on Wikipedia. <sup>5</sup>

The AP242 edition 1 BO Model takes a more pragmatic approach to units than Part 21 does: it is based on the assumption that a current target system simply knows what the used units are. In consequence it is not necessary to define a unit with conversion and relation to other predefined units.

### E.2.1 Named units derived from SI base units

SI derived unit exchange should use the `Unit` element with the `Unit.kind` attribute set to 'SI derived unit'.

List of named units derived from SI base units:

- hertz
- radian
- steradian
- newton
- pascal
- joule
- watt
- coulomb
- volt
- farad
- ohm
- siemens
- weber
- tesla
- henry
- degree Celsius
- lumen
- lux
- becquerel
- gray
- sievert
- katal

### E.2.2 Derived quantities and units

To exchange these covered units, use the `Unit` element with the `Unit.kind` attribute set to 'Unspecified SI derived unit'.

---

<sup>5</sup> [https://en.wikipedia.org/wiki/SI\\_derived\\_unit](https://en.wikipedia.org/wiki/SI_derived_unit)

Examples of quantity:

- acceleration
- area
- velocity
- volume

Examples of SI derived unit:

- square metre
- cubic metre
- metre per second
- cubic metre per second
- metre per second squared
- metre per second cubed
- metre per quartic second
- radian per second
- newton second
- newton metre second
- newton metre
- newton per second
- reciprocal metre
- kilogram per square metre
- kilogram per cubic metre
- cubic metre per kilogram
- mole per cubic metre
- cubic metre per mole
- joule second
- joule per kelvin
- joule per kelvin mole
- joule per kilogram kelvin
- joule per mole
- joule per kilogram
- joule per cubic metre
- newton per metre
- watt per square metre
- watt per metre kelvin
- square metre per second
- pascal second
- coulomb per square metre
- coulomb per cubic metre
- ampere per square metre
- siemens per metre
- siemens square metre per mole
- farad per metre
- henry per metre
- volt per metre
- ampere per metre
- candela per square metre
- lumen second
- lux second
- coulomb per kilogram
- gray per second
- ohm metre

mm2:

```
<Unit uid="u--100000007">
  <Kind><ClassString>Unspecified SI derived unit</ClassString></Kind>
  <Name><ClassString>square metre</ClassString></Name>
  <Prefix><ClassString>milli</ClassString></Prefix>
  <Quantity><ClassString>area</ClassString></Quantity>
</Unit>
```

### E.3 Unspecified Units

To exchange units which are neither SI units nor derived from SI units, the Unit element shall be used with the Unit.kind attributes set to 'Unspecified'.

#### E.3.1 Byte

Byte:

```
<Unit uid="u--100000009">
  <Kind><ClassString>Unspecified</ClassString></Kind>
  <Name><ClassString>byte</ClassString></Name>
  <Quantity><ClassString>data size</ClassString></Quantity>
</Unit>
```

Note that for bytes, two classes of prefixes exist; the SI prefixes (base 10) and the IEC prefixes (base 2). So 1 Kilobyte = 1000 Byte, while 1 Kibibyte = 1024 Byte. The following table gives an overview:

Decimal Prefixes			Difference (rounded)	Binary Prefixes		
SI Name	SI Symbol	Factor		IEC Name	IEC Symbol	Factor
Kilobyte	kB	10 <sup>3</sup>	2.40%	Kibibyte	KiB	2 <sup>10</sup>
Megabyte	MB	10 <sup>6</sup>	4.86%	Mebibyte	MiB	2 <sup>20</sup>
Gigabyte	GB	10 <sup>9</sup>	7.37%	Gibibyte	GiB	2 <sup>30</sup>
Terabyte	TB	10 <sup>12</sup>	9.95%	Tebibyte	TiB	2 <sup>40</sup>
Petabyte	PB	10 <sup>15</sup>	12.6%	Pebibyte	PiB	2 <sup>50</sup>
Exabyte	EB	10 <sup>18</sup>	15.3%	Exbibyte	EiB	2 <sup>60</sup>
Zettabyte	ZB	10 <sup>21</sup>	18.1%	Zebibyte	ZiB	2 <sup>70</sup>
Yottabyte	YB	10 <sup>24</sup>	20.9%	Yobibyte	YiB	2 <sup>80</sup>

The names of IEC prefixes are derived from the SI prefixes, where “kibi” means “kilo-binary”, “mebi” means “mega-binary” and so on. As shown in the table above, the difference in data size between using base 2 or base 10 for the prefixes is significant for higher factors.

See <http://physics.nist.gov/cuu/Units/binary.html> for background information.

#### E.3.2 Each

To represent a quantity of instances with QuantifiedOccurrence, the unit ‘each’ is recommended. It represents a total number of objects.

```
<Unit uid="u--100000010">
  <Kind><ClassString>Unspecified</ClassString></Kind>
  <Name><ClassString>each</ClassString></Name>
  <Quantity><ClassString>number of elements</ClassString></Quantity>
</Unit>
```

## E.4 Imperial Units

The following is the recommendation for exchange of Imperial unit definitions<sup>6</sup>, the `Unit` element shall be used with the `Unit.kind` attributes set to 'Imperial':

### Length:

- thou
- inch
- foot
- yard
- chain
- furlong
- mile
- league
- fathom
- cable
- nautical mile
- link
- rod
- chain

### Area:

- perch
- rood
- acre

### Volume:

- fluid ounce
- gill
- pint
- quart
- gallon

### Inch:

```
<Unit uid="u--100000009">  
  <Kind><ClassString>Imperial</ClassString></Kind>  
  <Name><ClassString>inch</ClassString></Name>  
  <Quantity><ClassString>length</ClassString></Quantity>  
</Unit>
```

---

<sup>6</sup> [https://en.wikipedia.org/wiki/Imperial\\_units](https://en.wikipedia.org/wiki/Imperial_units)

## Annex F AP214 AIM to AP242 BOM Mapping

STEP AP214 / STEP AP203 AIM			Mapping to STEP AP242 BO Model	
Entity	Attribute	Comments	Entity	Attribute
<b>Part Identification</b>				
product	id		Part	Id
	name		Part	Name
	description		Part	Description
	frame_of_reference	ref. to product_context		no mapping needed
product_context	name	no recommendation on the value		no mapping needed
	frame_of_reference	ref. to application_context		no mapping needed
	discipline_type	no recommendation on the value		no mapping needed
application_context	application			xsi.schemaLocation
application_protocol_definition	application_interpreted_model_schema_name			xsi.schemaLocation
	application_protocol_year			xsi.schemaLocation
	status			xsi.schemaLocation
	applications	ref. to application_context		no mapping needed
product_related_product_category	name		Part	PartTypes
	description	no recommendation on the value		no mapping needed
	products	ref. to product		no mapping needed
product_category_relationship	name	no recommendation on the value		no mapping needed
	description	optional		no mapping needed
	category		SubsetMember	Superset
	sub_category		SubsetMember	Subset
product_definition_formation	id		PartVersion	Id
	description		PartVersion	Description
	of_product	ref. to product	Part	Versions
product_definition	id		PartView / AssemblyDefinition	Id
	description		PartView / AssemblyDefinition	Description
	formation	ref. to product_definition_formation	PartVersion	Views
	frame_of_reference	ref. to product_definition_context	PartView / AssemblyDefinition	InitialContext
product_definition_context	name		ViewContext	Description
	frame_of_reference	ref. to application_context		no mapping needed
	life_cycle_stage		ViewContext	LifeCycleStage
application_context	application		ViewContext	ApplicationDomain
<b>Part Properties</b>				
product_definition_shape	name	recommendation as empty string		no mapping needed
	description	optional		no mapping needed
	definition	ref. to product_definition		no mapping needed
shape_definition_representation	definition	ref. to product_definition_shape		no mapping needed
	used_representation	ref. to shape_representation	PartView / AssemblyDefinition	DefiningGeometry
shape_representation	name		ExternalGeometricModel	Id
	items		ExternalGeometricModel	Items
	context_of_items		ExternalGeometricModel	ContextOfItems
geometric_representation_context	context_identifier		GeometricCoordinateSpace	Id
	context_type	Unique value recommended : "External"		no mapping needed
	coordinate_space_dimension		GeometricCoordinateSpace	DimensionCount

geometric_representation_item	name		RepresentationItem	Name
axis2_placement_3d	name		AxisPlacement	Name
	location		AxisPlacement	Position
	axis		AxisPlacement	Axis
	ref_direction		AxisPlacement	RefDirection
shape_aspect	name		ShapeElement	Name
	description	optional	ShapeElement	Description
	of_shape		ShapeElement	ElementOf
	product_definitional	logical		
property_definition	name	Unique value recommended : "external definition"		no mapping needed
	description	optional		no mapping needed
	definition	ref. to document file	ExternalGeometricModel	ExternalFile
property_definition_representation	definition	ref. to property_definition		no mapping needed
	used_representation	ref. to shape_representation		no mapping needed
representation_relationship_with_transformation	name			no mapping needed
	description	optional		no mapping needed
	rep_1	ref. to shape_representation		no mapping needed
	rep_2	ref. to shape_representation		no mapping needed
	transformation_operator	ref. to item_defined_transformation	NextAssemblyOccurrenceUsage	Placement
item_defined_transformation	name		Use of the appropriate entity regarding the type of transformation: - GeometricRepresentationRelationshipWith-PlacementTransformation - GeometricRepresentationRelationship-WithCartesianTransformation - GeometricRepresentationRelationshipWith-SameCoordinateSpace	
	description	optional		
	transform_item_1			
	transform_item_2			
cartesian_transformation_operator	name		CartesianTransformation	Name
	description	optional		
	axis1	optional		
	axis2	optional		
	local_origin			
	scale	optional	CartesianTransformation	Scale
	axis3	optional		
<b>Part Structure and Relationships</b>				
next_assembly_usage_occurrence	id		NextAssemblyOccurrenceUsage	Id
	name		NextAssemblyOccurrenceUsage	RelationType
	description		NextAssemblyOccurrenceUsage	Description
	relating_product_definition		NextAssemblyOccurrenceUsage	Relating
	related_product_definition		SingleOccurrence	Definition
	reference_designator	optional		no mapping needed
product_definition_shape	name	recommendation as empty string		no mapping needed
	description	optional		no mapping needed
	definition	ref. to next_assembly_usage_occurrence		no mapping needed
	representation_relation	ref. to representation_relationship		no mapping needed
context_dependent_shape_representation	represented_product_relation	ref. to product_definition_shape		no mapping needed





Document Identification				
product	id		Document	Id
	name		Document	Name
	description		Document	Description
	frame_of_reference	ref. to product_context		no mapping needed
product_context	name	no recommendation on the value		no mapping needed
	frame_of_reference	ref. to application_context		no mapping needed
	discipline_type	no recommendation on the value		no mapping needed
application_context	application			xsi.schemaLocation
application_protocol_definition	application_interpreted_model_schema_name			xsi.schemaLocation
	application_protocol_year			xsi.schemaLocation
	status			xsi.schemaLocation
	applications	ref. to application_context		no mapping needed
product_related_product_category	name		Document	DocumentTypes
	description	no recommendation on the value		no mapping needed
	products	ref. to product		no mapping needed
product_category_relationship	name	no recommendation on the value		no mapping needed
	description	optional		no mapping needed
	category		SubsetMember	Superset
	sub_category		SubsetMember	Subset
product_definition_formation	id		DocumentVersion	Id
	description		DocumentVersion	Description
	of_product	ref. to product	Document	Versions
product_definition	id		DigitalDocumentDefinition	Id
	description		DigitalDocumentDefinition	Description
	formation	ref. to product_definition_formation	DocumentVersion	Views
	frame_of_reference	ref. to product_definition_context		no mapping needed
product_definition_context	name		managed by DocumentDefinition subtypes	
	frame_of_reference	ref. to application_context		no mapping needed
	life_cycle_stage			no mapping needed
application_context	application			no mapping needed
External Files				
document_file	id		DigitalFile	Id
	name			
	description	optional	DigitalFile	Description
	kind	ref. to document_type		no mapping needed
document_representation_type	name		managed by File subtypes	
	represented_document	ref. to document_file		no mapping needed
document_type	product_data_type		DigitalFile	FileType
applied_identification_assignment	assigned_id		DigitalFile	VersionId
	role	ref. to identification_role		no mapping needed
	items	ref. to document_file		no mapping needed
identification_role	name	recommendation as "version" string		managed by DigitalFile attribute name
	description	optional		no mapping needed
Relationship Between Documents and Constituent Files				
product_definition_with_associated_documents	id		DigitalDocumentDefinition	Id
	description		DigitalDocumentDefinition	Description
	formation	ref. to product_definition_formation	DocumentVersion	Views
	frame_of_reference	ref. to product_definition_context		no mapping needed
	documentation_ids	ref. to document_file	DigitalDocumentDefinition	Files

Document and File Properties				
property_definition	name	recommendation as "document property" string		no mapping needed
	description	optional		no mapping needed
	definition	ref. to document_file		no mapping needed
property_definition_representation	definition	ref. to property_definition		no mapping needed
	used_representation	ref. to representation		no mapping needed
representation	name		managed by the name of the appropriate property entity	
	items	ref. to descriptive/measure_representation_item		no mapping needed
	context_of_items	ref. to representation_context		no mapping needed
	id	should not be instantiated		no mapping needed
	description	should not be instantiated		no mapping needed
representation_context	context_identifier	no value specified		no mapping needed
	context_type	recommendation as "document parameters" string		no mapping needed
descriptive_representation_item	name		Use of the appropriate property : - ContentProperty for content information - CreationProperty for creation information - FormatProperty for format information - SizeProperty for size information - DocumentFileProperty template for additional document properties	
	description			
measure_representation_item	name			
	value_component			
	unit_component			
identification_role	name		ExternalItem	Description
	description	optional		no mapping needed
applied_external_identification_assignment	assigned_id		ExternalItem	Id
	role	ref. to identification_role		no mapping needed
	source	ref. to external_source		no mapping needed
	items	ref. to document_file	DigitalFile	Locations
external_source	source_id		ExternalItem	Source
	description	should not be instantiated		no mapping needed
Document and File Association with Product Data				
document_product-equivalence	name	recommendation as "equivalence" string	Out of scope	
	description	optional		
	related_product	ref. to product_definition_formation		
	related_document	ref. to document		
document	id	should not be instantiated	Out of scope	
	name			
	description	optional		
	kind	ref. to document_type		
document_type	product_data_type		Out of scope	
applied_document_reference	source	no value specified		no mapping needed
	assigned_document	ref. to document or document_file	DocumentAssignment	AssignedDocument
	items	ref. to product_definition	DocumentAssignment	AssignedTo
role_association	item_with_role	ref. to applied_document_reference		no mapping needed
	role	ref. to object_role		no mapping needed
object_role	name		DocumentAssignment	Role
	description	optional	DocumentAssignment	Description