Recommended Practices

for

STEP AP242 Edition 3
Domain Model XML
Product & Assembly Structure

Release 3.2

11 January 2024

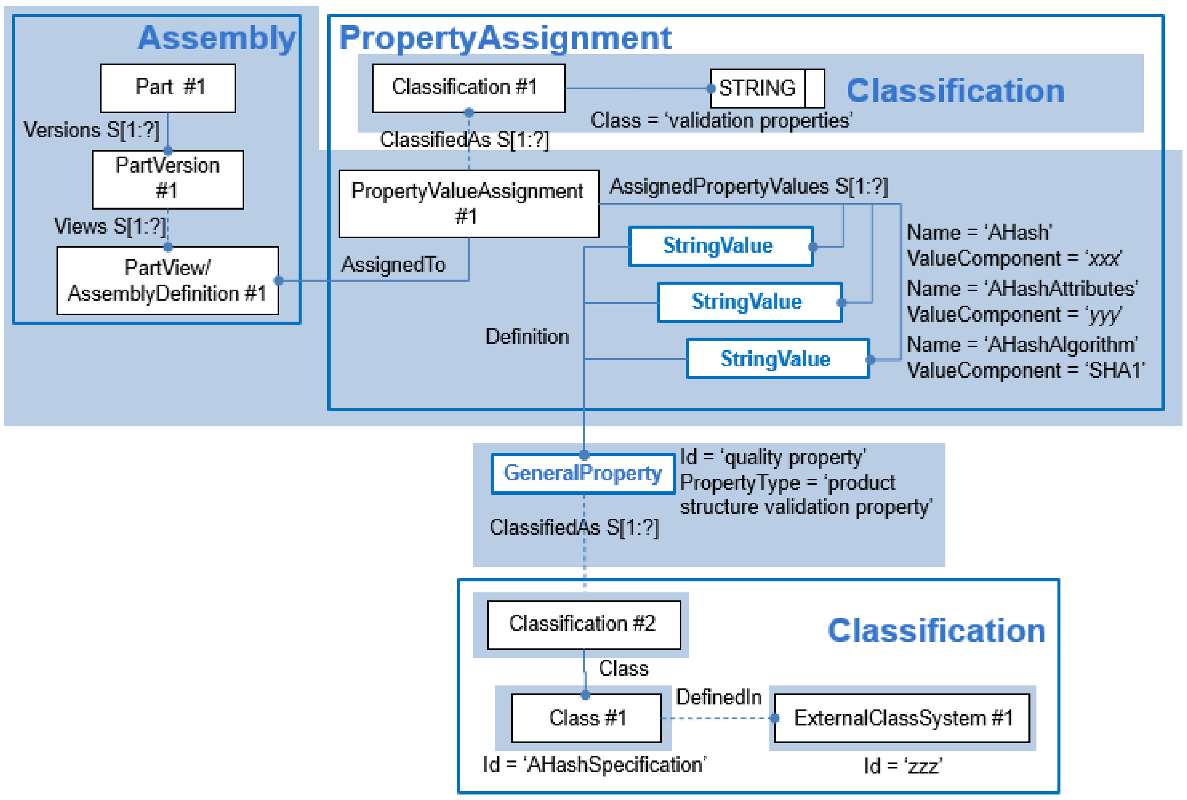
Contacts:

Organizational		
Jochen Boy PROSTEP AG jochen.boy@prostep.com	**Phil Rosché** ACCR, LLC. phil.rosche@accr-llc.com	**Frédéric Darré** Airbus Operations frederic.f.darre@airbus.com

Technical	
Guillaume Hirel T-Systems guillaume.hirel@t-systems.com	**Jochen Haenisch** Jotne EPM jochen.haenisch@jotne.com

Table of Contents

List of Figures



List of Tables

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 Over-loading”; improved several descriptions Section 8: Improved definition of DocumentType; new section Docu-mentDefinitionRelationship Section 9: Improved descriptions; new section FileRelationship Section 10: Updated recommendations for CreationProperty Section 11: Updated descriptions to include Model Splitting / Shar-ing and Alternate Models, for CAx as well as PDM map-ping Section 12: Improved description for attribute types Section 13: Updates description for classification of attributes
1.2	2017-06-30	Overall: updated instantiation diagrams for Class / String; improved illustration of XML embedding Section 2: Refined Scope statements Section 4.1: Case handling in Attributes Section 4.2: Updated attributes for Version ID Section 4.6: Updated XML examples and attribute recommendations Section 5.1: Added PartViewRelationship; updated Property Value Assignment; added Multiple Part Versions and Relation-ships; added support for Supplier Part Number Section 6: Clarification for use of ShapeElement Section 6.1: Refined descriptions for CartesianPoint and Geometric-CoordinateSpaces Section 6.2: Added Definition of Property Types; renamed PartProp-erty to PropertyAssignment Section 6.3: Added clarifications for Centroid and MomentsOfInertia Section 7: Added clarification for creation of occurrences; added PartViewRelationship Section 7.1: Updated recommendations for SingleOccurrence Section 7.3: Revised RotationMatrix attribute recommendations Section 7.4: Updated NextAssemblyOccurrenceUsage Section 8.1: Mulitple DocumentVersions and -Definitions Section 8.3: Added DocumentVersionRelationship Section 9: File & Document Structure updates Section 9.3: Added clarifications to the use of nested files and re-moval of PDM representation Section 10.1: Extended list of values for non-CAD documents; up-date FileFormat Property recommendations; added clari-fications to the use of compression Section 10.4: Updated Property Assignment Section 12: Revision based on PDM-IF Discussions; Clarification for use of ShapeElement

Revision	Date	Change
		Section 12:2: Added 'general property' to PropertyDefinition.Id Section 12.5: Re-wrote attribute definition for values without unit Section 13: Added attribute table for CentreOfMass; clarifications for Centroid Annex A.1.5: Added explanation for order of elements Annex B: Updated list of BugZilla Issues Annex C: Updated Reference Document Versions Annex E.3: Unit 'each' for QuantifiedOccurrence
2.0	2018-10-30	**Update for AP242-TC** **See section 2.3 for Implementation differences to AP242-IS** Section 4.2.1: added recommendation for unique attributes Section 4.6.3: Updated recommendations for Unit prefixes Section 4.6.6: Updated 'identifier' information for exchange context Section 4.6.9, 4.6.10, 4.6.14: Updated Attribute Recommendations Section 5.1.2: added EffectivityAssignment to PartVersion Section 5.1.6: Replaced textual Definitions; moved example for multiple ids from section 4.6.6; added second example Section 7.1: Updated Pre-Processor Recommendations for Single-Occurrence and added EffectivityAssignment Section 8.1.1: Updated Post-Processor Recommendations Section 9.3: Updated Post-Processor Recommendations Section 9.4: Updated Pre- and Post-Processor Recommendations Section 12.2, 12.5: Updated Attribute Recommendations Section 13.1: Updated Definitions and Figures 59-64 New Section 14 for Exchange of Customized PDM Information New Section 15 for CAx-PDM Interoperability
2.1	2019-12-20	**Public Release – Update for AP242 TC** Sections 1.1.2 and 1.2: Introduction of the MBx Interoperability Forum (MBx-IF) Section 2.2: Updated scope statement Section 4.2.1: Update for the handling of empty valued multi-language strings Section 4.6.1: Updated recommendations for ExchangeContext Section 4.6.6: Updated recommendations for Identifier Section 4.6.7: Replaced previous section with support for multiple language; updated recommendations Section 5.1.1: Added support for alternate part relationships Section 5.1.2 and 7.1: Added support of ActivityAssignment Section 5.1.5: Added support 'alternative' PartVersionRelationship Section 6.1.: Clarified required unit definitions Section 6.2: Added support of Activity properties Section 7.1.: Added support for alternate part relationships; Updated recommendations for uniqueness of occurrence IDs Section 7.4: Added support of AssemblyViewRelationshipSubstitutions Section 7.5: New Template "AlternatePartRelationship"

Revision	Date	Change
		Section 7.6: New Template "AssemblyOccurrenceRelationship-Substitution" Section 7.7: New Template "AssemblyViewRelationshipSubstitution" Section 9.1: Added support of unset DigitalFile.Locations Section 11.1, 11.2: updated figures with use of AuxiliaryGeometry Section 12.2, 12.4: updated figures for PropertyDefinition.Id Section 12.5: Updated the recommendations for transferring values without units Section 12.6: figure 65 updated PropertyDefinition.Id Section 13.1.1: Update format of validation property 'number of chil-dren'
2.2	2020-09-30	**Public Release – Update for AP242 TC** Section 1.1.2: Added support for TC patch schema Section 4.2.1: updated base identifiers Section 4.3: updated recommendations for uniqueness of IDs Section 4.6.1: refined postprocessor recommendations Section 4.6.3: updated unit definitions Section 4.6.4, 4.6.5: Updated to support 'make or buy' Section 4.6.6: updated identifier attributes and recommendations Section 4.6.8: updated preprocessor recommendation Section 5.1.2: Updated to support 'make or buy' Section 5.1.3: added PartView.ShapeElement Section 5.1.5: Added support 'alternative' PartVersionRelationship and updated recommendations Section 6.1, 6.2: updated attributes recommendations Section 7.2: added SpecifiedOccurrence.PropertyValueAssignment Section 7.4, 7.5, 7.6, 7.7: Updates based on testing feedback Section 7.5: restructured for Alternate and Substitute parts Section 8.2, 8.3: updated diagrams Section 9.3: adapted definitions for multiple IDs Section 11.2: extended definitions for alternate models Section 12.2: figure 58 updated PropertyDefinition.Id Section 12.2, 12.4, 12.6: updated diagrams Section 12.4.1: figure 62 updated PropertyDefinition.Id Section 12.4.2: figure 63 updated PropertyDefinition.Id Section 12.4.3: figure 64 updated PropertyDefinition.Id Section 12.6: figure 65 updated PropertyDefinition.Id
3.0	2021-11-23	**Public Release – Update for AP242 Edition 2 Domain Model** **See sections 2.4 for Implementation changes vs. AP242 Ed.1 TC** Throughout: Updated references from AP242 Ed.1 TC BO Model to AP242 Ed.2 Domain Model Some 0 chapter references fixed Sections 1.1.2, 4.1.6: URL of Domain Model corrected Section 1.1.2: remark how to find the Ed. 2 MR Domain Model XSD if not published yet by the ISO Section 2.4: new section

Revision	Date	Change
		outlining requiremed implementation changes compared to AP242 Ed.1 TC Section 3.6.14: added Preprocessor Recommendation for OrganizationOrPersonInOrganizationAssignment.Role Section 4.1.6, 4.1.7: Updated schema references Section 4.1.12: Added 'Role' to attribute recommendations Section 4.2.3: added handling of unset optional attributes during import Section 4.3: added uniqueness rule applying to objects having Id and VersionId. Section 4.6.1: added recommendation to handle explicit configurations Section 4.6.5: enhancement and harmonization of the usage of Classification.Role + XML examples updated throughout the document; 'VDA reference mechanism' added as Classification.Role Sections 4.6.6 + 5.1.6: added recommendation for setting Occurrence.Id for multiple identifiers Section 4.6.8: added recommendations for the use of PredefinedApplicationDomainEnum Section 4.6.9: attribute DeterminationMethod added Section 4.6.9, 4.6.10: added recommendations for the use of PropertyDefinitionEnum Sections 4.6.9, 7.4 and 10.4: rework of the handling of ValueWithUnit.Definition and .Name for DigitalFile.FileSize and NextAssemblyViewUsage.Quantity Section 4.6.11: added Postprocessor Recommendation for DateTime;additional recommendation regarding date/time format New sections 4.6.11, 4.6.12, 4.6.13 and 4.6.15 for the handling of ValueRange, ValueWithTolerances, ValueList and ValueSet Section 4.6.20: added object DateAndPersonOrganization; Added File.VersionId, PropertyDefinition.VersionId, Class.Version.Id; Added recommendation for the use of Role='sequence' for PartVersionRelationship, DocumentVersionRelationship and FileRelationship. Section 5.1.1: added recommendations for the use of PartCategoryEnum Section 5.1.5: PartVersionRelationship: update in mapping table; update in postprocessor recommendation; update of the uniqueness rules Section 6.1.3: Added 'ShapeDependentProperty' to recommendations Sections 6.2 and 10.5: usage of ValueRange, ValueWithTolerances, ValueList and ValueSet added Section 6.2: additional recommendation regarding properties with value 0, empty string or empty ValueList/ValueSet Section 7.1: Updated recommendations on 'Placement' new picture to illustrate the cross-reference between multiple geometries and multiple placements Section 7.3.2.2: updated figure 28 & XML example

Revision	Date	Change
		Section 7.4: Added recommendations for 'AssemblyViewRelationshipSubstitution'; mapping of FindNumber/PositionNumber added; updated uniqueness rule of NextAssemblyViewUsage.Id Section 7.5.2: AssemblyOccurrenceRelationshipSubstitution: added alternative proposal as comment; AssemblyOccurrenceRelationshipSubstitution: update in postprocessor recommendation + computing of the AVPs; update of the uniqueness rules Section 7.5.3: NextAssemblyViewUsages PartVersionRelationship update in postprocessor recommendation; NextAssemblyViewUsages: computing of the AVPs + non-obligation to define NextAssemblyOccurrenceUsages for each NextAssemblyViewUsage; new rule regarding the Id of the Relating/Related NextAssemblyViewUsages of a AssemblyViewRelationshipSubstitution; update of the uniqueness rules Section 8.1.1: Added 'Document.PropertyValueAssignment' to recommendations Definition of some recommended values for DocumentTypes edited Section 9.1: Added 'FileLocations' to recommendations added recommendations for the use of DigitalFile.FileLocations and DigitalFile.Locations.Source Section 9.3: updated recommendation regarding the consistency of ExternalGeometricModel.ExternalFile new comment regarding the external reference of ShapeElements; new comment regarding the naming of the nested files; added handling of NextAssemblyViewUsage for 'nested' files; handling of special characters for naming nested files added Sections 9.3 and 9.4: corrected figures regarding 'ClassifiedAs'; handling of AlternatePartRelationship for nested and for incremental exchange; update of the file naming rules for nested files Section 9.4: changes regarding incremental exchange New section 9.6 with planned activities for the support of External Element References (EER) Section 10.1: FormatProperty.DataFormat recommended mandatory Ed. 2 TC replaced by Ed. 3 Section 12.2: CartesianPoint added for ExternalShapeReference.PropertyValue; added PropertyDefinition.DefinedIn (EWIS-IF) Sections 12.2, 12.5: Usage of PropertyDefinitionString updated Section 12.3: added PropertyDefinitionRelationship Section **Fehler! Verweisquelle konnte nicht gefunden werden.**: grouping of attributes removed since redundant with section 0 Sections **Fehler! Verweisquelle konnte nicht gefunden werden.**, 12.4: ValueRange, ValueWithTolerances, ValueList and ValueSet added in pictures Section 12.7 removed

Revision	Date	Change
		Section 13.1.2: Update regarding the units of Notional Solids Centroid Position, regarding the precision expected by validating the data and typo in the diagram. Section 14.2: added mapping of customized extensions/forms as property groups + new value for PropertyDefinition.PropertyType in section 12.2Annex A: Updated based on ISO 10303-4442:2019 Section 0: added postprocessor handling of non unique property value names of different property groups Section 14.3: changes regarding the handling of empty customized properties Section 15.4: overwork of consistency rules for tracking relationships. Section 16.2: typo corrected /ANY replaced by /NULL Section A.1.5: update of the description of the alphabetical ordering of the XML elements Section A.1.5.3: support of ValueList and multi-dimensional aggregate PropertyValues update Section E: incorporation of the values for Unit.Quantity Section E.2.1: unit ‘degree’ added Section E.3.3: unit ‘percent’ added Section E.3.4: unit ‘ratio’ added Figures 58, 61, 67 to 69 updated
3.1	2022-11-18	**Public Release – Update for AP242 Edition 2 Domain Model** **See sections 2.4 for Implementation changes vs. AP242 Ed.1 TC** Over all the document: correct the XML examples using Unit.Quantity Section 1.1.2 and 4.1.6: update Ed3 Domain Model URL to official value from the ISO Section 2.3: Change of CentreOfMass.CentrePoint from IS to TC added Sections 4.6.5, 7.5.2 and 7.5.3: postprocessor recommendation added not to interprete alternative NAVUs/NAOUs as being cumulated in case the target system does not support substitutions. Section 4.6.13 and 4.6.15: add recommendation that the ValueList/Set element shall have the same Unit (in case of NumericalValue) Section 5.1.5: proposal to change the usage of relating and Related for nesting reasons => to be discussed… Section 5.1.5, 7.7: relating/related inversed in PartVersionRelationship and GeometricalRelationship for mirrored parts Sections 5.1.6 and 13.3.1.6: IdentifierSet replaced by ‘multiple Identifiers’, since IdentifierSet is used only for non-Id attributes => nowhere recommended Sections 7.1, 7.4, 7.5.2 and 7.5.3: ClassifiedAs=’alternative’ added to better distinguish between the main view/occurrence usage and the alternative ones Section 7.1 and 7.2: handling of `xsi:type="n0:SpecifiedOccurrence"`.

Revision	Date	Change
		Section 7.2: updated diagram for SpecifiedOccurrence Section 7.4: edit recommendation to allow multiple PartViewRelationships between the same parent and child; add recommendation to concatenate NextAssemblyViewUsage.Id into NextAssemblyOccurrenceUsage.Id in case multiple PartViewRelationships exist between the same parent and child; new JIRA ticket created to relate the NAVUs and the NAOUs + several editorial change Section 7.5: Update of the overview tables New Section 7.7 for mirrored parts Section 7.7.2: RelationType 'mirrored' added Section 9.3: use of PartVersion.Id for the file name not longer recommended; example diagrams for the nesting of alternate and substitute parts; example diagram for the nesting in case of SpecifiedOccurrences; recommendation to handle non-processed nested files + use of relative paths added Sections 9.3 and 9.4: mapping of Part/Document.Name, PartTypes and DocumentTypes for nested files and incremental exchange Section 10.2: update of recommended values for DetailLevel and GeometryTypes; update of recommended value 'exact geometry' and in general explanation for GeometryTypes; new comment to remove LOD Section 12: CAD-related limitations of UDAs introduced Section 12.2: add handling of customized PDM properties to the scope of CAx data exchange (tested by the CAx-IF) Section 12.2: add recommendation to leave PropertyValue.Definition unset for the value elements of a ValueList/ValueSet => XML examples edited in section 4.6.13 and 4.6.15 Section 12.4 removed (Group of attribute values) Section **Fehler! Verweisquelle konnte nicht gefunden werden.**: review comment added to clarify if UDAs on geometry level should be kept out of this recommendation practices, since mapped to Part 21 within CAx Data Exchange… Dito for Geometry Validation Properties (Section 13.2) Section 13.2: GVPs removed New Section 13.3: LOTAR Product Structure Validation Properties; update of the XPath examples + overworking of the specification Section 0 : update in mapping of extensions due to issue with uniqueness of PropertyDefinition.Id Section 14.3: training mode removed Section A.1.9: added mapping difference between Id/VersionId attributes and other attributes of kind IdentifierSelect Annex B: new issue numbers after switch from PDES Inc. Bugzilla to ISO Jira

3.2.01	2023-12-22	**Public Release – Update for AP242 Edition 3 Domain Model** **See sections 2.5 for Implementation changes vs. AP242 Ed.2** Sections 2.3, 2.4, 4.1.7: update of ISO release number Section 4.6.6: update handling of idContextRef for embedded objects Section 4.6.7: LocalizedString.uid now recommended as mandatory (prerequisite for AP242 Edition 4 EER mechanism) New Section 4.6.7.1: added handling of string attributes that have a different value depending on the IdentificationContext (like Part.Name) Adding of ValueLimit to section 4.6.9 Sections 4.6.9 to 4.6.15: adding of attribute “Qualifiers”; for upward compatibility reasons, recommend to support also PropertyDefinitionString during import; add recommendation how to interprete PropertyDefinitionString New section 4.6.13: LimitsAndFits Sections 4.6.16, 4.6.18 and 4.6.19: reference to section 14.3 added in case of customized PDM properties of kind Date, Approval or PersonInOrganization New Section 4.6.21: ValueFormatTypeQualifier for the handling of property qualifiers (Edition 4); added use of ValueFormatTypeQualifier for customized properties of kind DateTimeAssignment, ApprovalAssignment or OrganizationOrPersonInOrganizationAssignment Section 5.1.5: Special handling of additional ‘Related’ Parts after archiving/releasing the ‘Relating’ Part Sections 5.1.5 + 7.7.2: harmonization of the definition of RelationType in PartVersionRelationship and GeometricalRelationship (now matches the sample test data) Section 6.1: enhanced recommendation for the consistency of RepresentationContext.Items with RepresentationContext.Representations.Items; (GeometricModel): recommendation added to use only the subtypes of GeometricModel; recommendation added regarding the consistency between PartView.GeometricModel, GeometricModel.Items and the ShapeElement.RepresentedGeometry defined on a part (relevant for the PMIs); recommendation added for the reuse of RepresentationItems in different GeometricModels; adding of ‘External’ in the context of PMIs Section 6.1.1: new recommendation for ExternalGeometricModel.Items (for Edition 4) Section 6.1.2: update of the recommendation for the use of the subtypes of ExternalGeometricModel (relevant for the PMI use case in AP242 Edition 4) Removed section 6.3: GeneralShapeDependentProperty Section 7.1: added handling of NextAssemblyOccurrenceUsage.Id Section 7.3.3 and 7.8.2: GeneralGeometricRepresentationRelationship added for the mapping of Flexible Parts. Section 7.7: replacement of left/right by original/mirrored New Section 7.8: Flexible Parts Section 8.1.3: added handling of document without file

		Section 9.1: handling of empty Locations/FileLocations added; added handling of multiple versions of the same Digital-File Section 9.3: new preprocessor recommendation about consistency of ExchangeContext.IdentificationContext over all nested files. New section 9.3.1: nested handling of Relationships Section 9.4.1: handling of incremental exchange on other objects than Part and Document New section 9.5 Delta Ex-change (updated on the 7.6.23) Section 9.4.1 and 9.5: add handling of multiple identifiers in case of incremental exchange and delta exchange (analog nested files) New Section 9.6: External Element References (EER) from Part 15 Edition 2 (for AP242 Edition 4) Sections 11.1 and 11.2: recommendation added to non-referenced DigitalFiles Section 12.2: added handling of multiple versions of the same Prop-ertyDefinition; PropertyType='semantic text' added for PMIs Section 12.4.3: removed (attributes at shape level) Section 13.3: updates and additional precisions about the AHash computation; removal of unit in the computation of the AHash and details added about handling of significant digits Section 14.2.1: adding of the uniqueness of the Proper-tyValue.Name over all PropertyValues assigned by a PropertyValueAssignment Section 0: update regarding the mapping of date/approval/per-son/organization properties within a property group Section 14.3: handling of customized properties updated to unset (using DeltaChange mechanism) Annex B: new issue numbers after switch from PDES Inc. Bugzilla to ISO Jira

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” Domain 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 Domain 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 third edition of STEP AP242, which was published in 2022. It contains the Domain Model (ISO 10303-4442:2022) and the corresponding XML schemas, which can be found at `http://standards.iso.org/iso/ts/10303/-4442/ed-3/tech/xml-schema/domain_model`

Therefore, the the XML header of AP242 Domain Model XML files which shall comply to Ed.3 shall look like the following:

```
xmlns:n0="http://standards.iso.org/iso/ts/10303/-4442/ed-3/tech/xml-
schema/domain_model" xsi:schemaLocation="http://stand-
ards.iso.org/iso/ts/10303/-4442/ed-3/tech/xml-schema/domain_model
http://standards.iso.org/iso/ts/10303/-4442/ed-3/tech/xml-schema/do-
main_model/DomainModel.xsd"
```

Note: due to some delays in the publication of the standard, this URL may not be available yet. In this case, please use a local copy of the Domain Model XSD as distributed within the interoperability forums.

AP242 is the first STEP Domain Model that has been implemented. During the implementation of the first edition, the writing of the Recommended Practices, the corresponding testing activities in the involved Interoperability Forums (see section1.2), as well as the definition of extended capabilities to support new user requirements, various shortcomings in the original schema have been identified. These have been documented in Bugzilla as official maintenance issues for AP242 and were dealt with as part of the ISO maintenance procedures. The agreed upon changes to the schema were gathered in the second edition of the BO Model (now called the Domain Model) and published as Ed.3. Some of these changes require implementation changes compared to the first edition of the BO Model published with AP242 as International Standard (IS) in 2014.

- Version 1.x of this document supports AP242 Ed.1 IS (2014).
- Version 2.x of this document supports AP242 Ed.1 Technical Corrigendum (2016).
- Version 3.x of this document supports AP242 Ed.3 (2022).

The development of the third edition of AP242 will commence in 2022. This will add new capabilities to the standard and eventually lead to a new version 4.x of this document. The development of these extended capabilities, as well as changes resulting from implementation feedback and harmonization efforts, follows the standard development procedures defined by ISO. Technical issues, as far as they affect the scope of this document, 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 Domain model, and guidelines for the consistent interpretation by a postprocessor of the STEP AP242 Domain 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 Domain 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 XML schema specification of ISO 10303-4442. The table includes not only the attributes of the EXPRESS schema of the AP242 Domain 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 Domain 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 Domain 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.

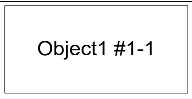
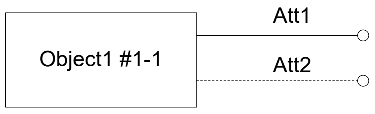
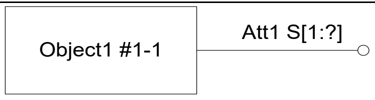
1.1.7 Instantiation Diagrams

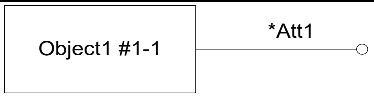
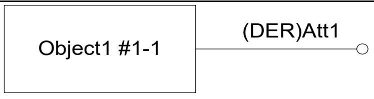
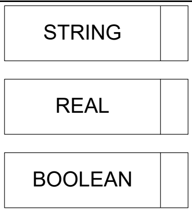
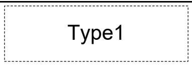
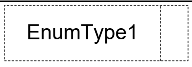
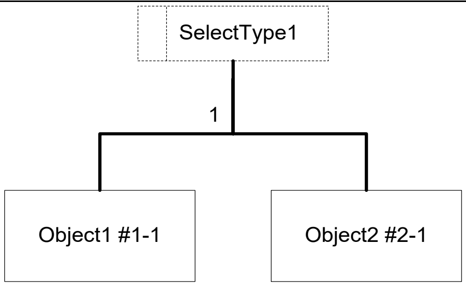
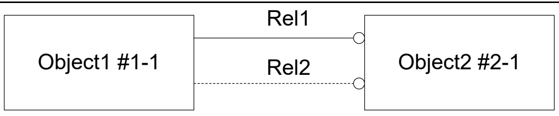
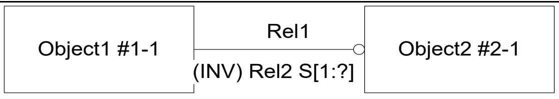
The diagrams are presented using a graphical notation intended to illustrate the instance model.

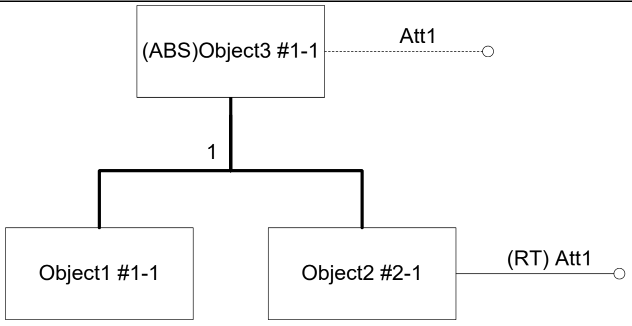
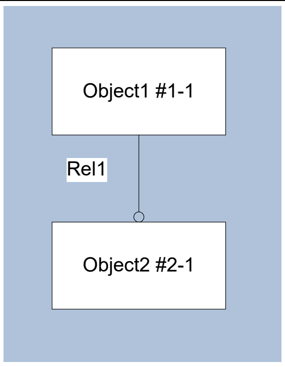
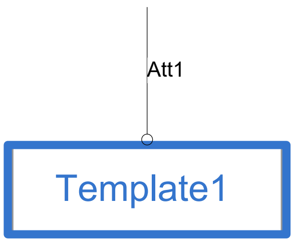
This notation is not EXPRESS-G and does not illustrate the XML schema; rather it is a graphical illustration of a specific population of a particular instance model of the schema. This notation supports:

- 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:

	Object (instance of an EXPRESS ENTITY) After the #, an instance number is given
	Att1: mandatory attribute Att2: optional attribute
	Aggregate type for the definition of the cardinality constraint: B: Bag (non-ordered and my 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
	Additional constraint on the object: the attribute(s) depicted with '*' have to contain unique values. *Currently not used in this document*
	Derived Information from another object or attribute *Currently not used in this document*
	Simple data types
	User defined data type *Currently not used in this document*
	Enumeration Type (consists of a limited list of possible values defined for this type)
	Select Type (is used if a relationship from an object may apply either to Object1 or Object2) 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. *Currently not used in this document*
	Attribute as relationship between two Objects (mandatory or optional), The circle at the end of the line gives the direction. Rel1: mandatory relationship Rel2: optional relationship
	Rel2: inverse attribute (i.e. in the converse direction to Rel1) with cardinality constraint

	Currently not used in this document
 (ABS)Object3 #1-1 Att1 1 Object1 #1-1 Object2 #2-1 (RT) Att1	Inheritance from a Supertype to its Subtypes An Abstract Supertype (ABS) cannot be instanced without one of its non-abstract Subtypes 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) RT: an inherited attribute is retyped, for example through restriction of its type, length, value range, cardinality, mandatory instead of optional or DERIVED *Currently not used in this document*
	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
	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’

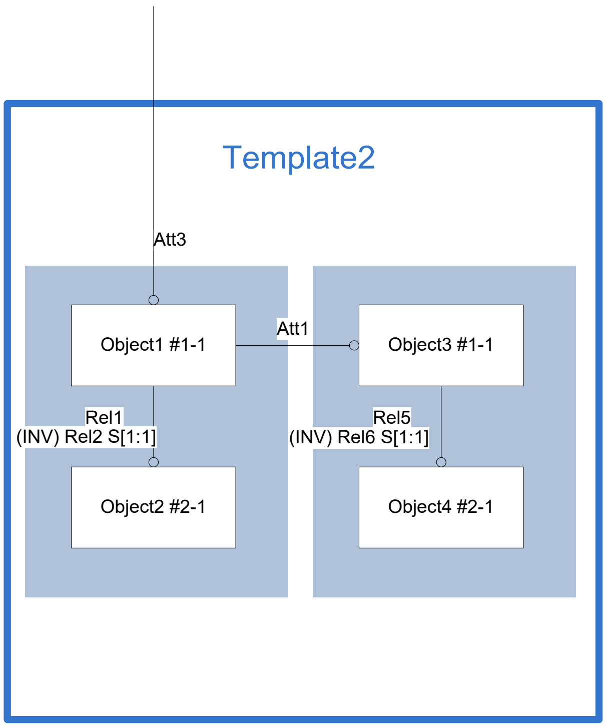
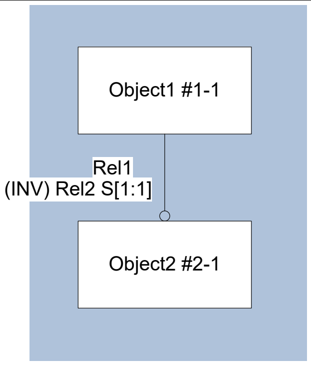
 Template2 Att3 Object1 #1-1 Att1 Object3 #1-1 Rel1 (INV) Rel2 S[1:1] Rel5 (INV) Rel6 S[1:1] Object2 #2-1 Object4 #2-1	If the template is more complex and the object referenced within the template is shown explicitly, portions of the reused template are displayed within a blue frame.
	Alternatively, a template may be re-used through adding new XML containments to it

Table 1: Instance Diagram Notation

1.2 Organizational Framework

These Recommended Practices for AP242 Domain 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 Domain Model. This section describes those communities’ roles and responsibilities.

1.2.1 Vendor Communities

The **MBx Interoperability Forum (MBx-IF) Implementor Groups (IG)** will be responsible for the overall organization and development of this document. The MBx-IF IGs will:

- Coordinate the creation of the document
- Verify the approach of the recommended practices in MBx-IF Test Rounds
- Publish result summaries of testing AP242 Domain Model product and assembly structure
- Ensure the consistency of other “AP242 Domain Model XML Recommended Practices”
- Ensure the consistency with existing recommended practices from all domains, namely CAx, CAE and EWIS.

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 **MBx Interoperability Forum (MBx-IF) User Groups (UG)** are forums of domain experts from the Aerospace and Defense as well as the automotive industries. The MBx-IF UGs are 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

The **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 Domain 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 2.1. Domain-specific extensions, such as Configuration Management or Change 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 roadblocks 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.

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 AP242 Edition 3 (2022) Domain Model (ISO 10303-4442) 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
- Identifiers at OEM and supplier
- Customized PDM Information
- Multi-View Product Structure Representation
- CAx-PDM Compatibility
- Geometric and Assembly Validation Properties
- AP242 IS Domain 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
- 7.2: SpecifiedOccurrence
- 8.2: DocumentDefinitionRelationship
- 8.3: DocumentVersionRelationship
- 9.2: FileRelationship
- 10: DocumentFileProperty

- 11.1: Model splitting and alternative geometries (CAx mapping)
- 11.2: Model splitting and alternative geometries (PDM mapping)
- 12.3: PropertyDefinitionRelationship
- 14.2.2: Groups of customized properties

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, Wire Harness, etc.
- Advanced PDM Capabilities:
  - Configuration, Effectivity, Change Management, …
- External Element References (into Part 21 files, into JT or between Domain Model XML files): planned for section 9.5
- Implementations based on any edition of the AP242 Domain Model (ISO 10303-4442) other than Edition 3 (2022).

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
- Certification
- Contract
- ConstituentPart (including measured, calculated, and estimated / asserted weight for single parts and assemblies)
- FrozenAssignment
- InformationUsageRightAssignment
- MaterialPropertyAssignment
- ModelPropertyAssignment
- Subtypes of OccurrenceRelationship (ReplacedUsageRelationship and SameTimeMachiningRelationship)
- OrganizationRelationship
- MakeFromRelationship
- QuantifiedOccurrence
- ReplacedPartViewRelationship
- ToolPartRelationship
- PhysicalDocumentDefinition
- Project
- ProjectAssignment
- ProjectRelationship
- PropertyDefinitionAssignment
- SecurityClassificationAssignment
- ShapeElementRelationship
- TimeIntervalAssignment
- Validation Properties for Product Structures and External References
- Document structures

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

2.3 Implementation Changes between AP242 Ed.1 IS and AP242 Ed. 1 TC

For a number of constructs, the implementation changed between the original version of AP242 and its Technical Corrigendum, i.e., ISO 10303-3001 Ed.1 and Ed.2 respectively. These changes became necessary due to deficiencies found in the original definitions of these elements, or to lay the foundation for future extensions of the data model.

The changes include structural changes such as embedding and referencing of XML elements as well as technical changes such as new element types or changed definition of attributes. Several of these changes are incompatible ones, but have been agreed to by all stakeholders, nonetheless.

The list below provides an overview for the necessary implementation changes to support AP242-TC in scope of this document:

- Change of the URL of the AP242 XSD (from IS to TC) (see sections 1.1.2 and 4.1.6)
- Introduction of RepresentationContext (as supertype of GeometricCoordinateSpace) and embedding of the Representations and referencing RepresentationItems embedded in the Representations
    - This affects the Full Positioning Representations (see chapter 7.3.2), the CentreOfMass (see sections 13.1.2) as well as the Kinematics features (see Recommended Practices for AP242 Domain Model XML Kinematics for details)
    - Since the EXPRESS schema misses RepresentationContext.Items, it is not possible to export CentreOfMass as XML out of an EXPRESS based preprocessor.
- Introduction of PartView.auxiliaryGeometry for the mapping of alternate Models (see sections 11.1 and 11.2)
- Introduction of GeneralGeometricRepresentationRelationship as subtype of GeometricRepresentationRelationship for the mapping of model splitting (see sections 11.1 and 11.2)
- idRoleRef now mandatory in the XSD (as already mandatory in the EXPRESS Schema)
- enhanced semantical type checking in the XSD (fix of some key and keyref definitions)
- Header.Documentation now as LIST (see section 4.1.5)
- ShapeElement is no root object anymore
- Fix of the modeling of SpecifiedOccurrence (see section 7.2)
- Improvements of the specification of XSD derivation from EXPRESS (see Annex A)
- CentreOfMass.CentrePoint (of type CartesianPoint) not anymore embedded but referenced (now embedded into RepresentationContext.Items)

2.4 Implementation Changes between AP242 Ed.1 TC and AP242 Ed.2

For a number of constructs, the implementation changed between the Ed.1 and Ed.2, i.e., ISO 10303-3001 Ed.2 and ISO 10303-4442 Ed.1. These changes became necessary due to deficiencies found in the original definitions of these elements, or to lay the foundation for future extensions of the data model.

The changes include structural changes such as embedding and referencing of XML elements as well as technical changes such as new element types or changed definition of attributes. Except very few of them, all these changes are upward compatible.

The list below provides an overview for the necessary implementation changes to support AP242 Ed.2 in scope of this document:

- Change of the URL of the AP242 XSD (see sections 1.1.2 and 4.1.6)
- New attribute ApprovalAssignment.Role (#6291) (see sections 4.6.17 and14.3)
- Attributes PropertyValue.Definition (#6451), PropertyDefinition.Id and PropertyDefinition.PropertyType (#6021) now optional (no changes in the Rec. Pracs.)
- New attribute FileLocations in File, parallel to Locations (#6204) (no changes in the Rec. Pracs.)
- Type of Part.PartTypes changed from ClassSelect to PartClassSelect => new ENUMERATION type PartCategoryEnum (see section 5.1.1)
- Type of ViewContext.ApplicationDomain: new ENUMERATION type PredefinedApplicationDomainEnum added to ApplicationDomainSelect (see section 4.6.8).
- Type of PropertyValue.Definition: new ENUMERATION type PropertyDefinitionEnum added to PropertyDefinitionSelect (see section 4.6.9 and 4.6.10).
- PropertyValueAssignment added to Document (#6020) and to DocumentVersion (see section 8.1.1)
- Optional attributes ClassLibrary and ContractAssignment added to ExchangeContext (no changes in the Rec. Pracs.)
- CartesianTransformation not anymore subtype of DetailedGeometricModelItem => no embedment in RepresentationContext (#7537) (see section 7.3.2.2)
- Type of PartView/ProductConfiguration.Occurrence changed from DefinitionBasedOccurrence (**Single** and QuantifiedOccurrence) to Occurrence (embraces also SpecifiedOccurrence) (see section 5.1.3) (rollbacked in Ed3)
- NextAssemblyOccurrenceUsage.Placement is now a ***SET OF*** TransformationSelect (#8080) (see section 7.1)
- New EXPRESS attributes Items and Representations in RepresentationContext (to be compliant to the (unchanged) XSD) (no changes in the Rec. Pracs.)
- Alphabetical order of AssemblyViewRelationshipSubstitution in AssemblyViewRelationship changed (#6273, #6904) (see section 7.4)
- Embedment of SpecifiedOccurrence into Occurrence renamed and retyped from SpecifiedOccurrence to Occurrence (see section 7.1) (rollbacked in Ed3)
- Changes in the specification of XSD derivation from SysML (instead of EXPRESS) => see Annex A

2.5 Implementation Changes between AP242 Ed.2 and AP242 Ed.3

Some smaller changes which could not be submitted in time for Ed.2 have been added within a Minor Revision published as Ed.3.

All these changes are upward compatible.

The list below provides an overview for the necessary implementation changes to support AP242 Ed.3 in scope of this document:

- In NextAssemblyOccurrenceUsage.Placement, *SET OF* TransformationSelect is renamed to *SET OF* TransformationAndAssociationSelect (no change of content) (see section 7.1)
- PropertyValueAssignment added to AssemblyOccurrenceRelationshipSubstitution (#8183) (see section 7.5.2)
- Rollback of the Ed2 change “Embedment of SpecifiedOccurrence into Occurrence renamed and retyped from SpecifiedOccurrence to Occurrence” (see section 7.1)
- Rollback of the Ed2 change “Type of PartView/ProductConfiguration.Occurrence changed from DefinitionBasedOccurrence (Single and QuantifiedOccurrence) to Occurrence (embraces also SpecifiedOccurrence)” (see section 5.1.3)

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>MBx-IF Rec.Pracs.---AP242 Domain Model XML
Assembly Structure---3.2---2024-01-11</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 ¼ 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" idRoleRef="rl—ii"
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 Domain 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 Domain 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 `Representation`:

```
<Representation 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[1]: version of the Recommended Practices used to implement the preprocessor
- (optionally) Documentation[2-n]: some free texts explaining the contents of the exchange file

Here is an example:

```
<Header xmlns="">
  <Name>as1.stpx</Name>
  <TimeStamp>2023-12-19T09:54:06Z</TimeStamp>
  <Organization>
    <Name>mercedes-benz.com</Name>
  </Organization>
  <PreprocessorVersion>T-Systems International GmbH COM/FOX V6.1.4</Prepro-
cessorVersion>
  <OriginatingSystem>CATIA V5 B25 SP0 HF0</OriginatingSystem>
  <Documentation>MBx-IF Rec.Pracs.---AP242 Domain Model XML
Assembly Structure---3.2---2024-01-11</Documentation>
  <Documentation>free text 1</Documentation>
  <Documentation>free text 2</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 a 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 Domain 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).
- "Documentation" is a set. The following pattern is recommended:
  - 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 Domain Model XML file:

```
<n0:Uos
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:n0="http://standards.iso.org/iso/ts/10303/-4442/ed-3/tech/xml-
schema/domain_model"
xmlns:cmn="http://standards.iso.org/iso/ts/10303/-3000/-ed-2/tech/xml-
schema/common"
xsi:schemaLocation="http://standards.iso.org/iso/ts/10303/-4442/ed-
3/tech/xml-schema/domain_model http://standards.iso.org/iso/ts/10303/-
4442/ed-3/tech/xml-schema/domain_model/DomainModel.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 Domain Model XML file shall be compliant to shall be listed at the beginning of the uos element.

Example: To indicate AP242 Domain Model XML Ed.3, the following URL is used:

```
xsi:schemaLocation="http://standards.iso.org/iso/ts/10303/-4442/ed-
3/tech/xml-schema/domain_model http://standards.iso.org/iso/ts/10303/-
4442/ed-3/tech/xml-schema/domain_model/DomainModel.xsd">
```

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: 2010-08-19T00:00:00Z

4.1.9 STEP Domain 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 Domain 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,…) Domain Model XML files
- “***.stpxZ**” – for compressed STEP Domain Model XML files
- “***.stpA**” – for compressed archive containing one/multiple root files and possibly files of different types (for example stpx + Part 21, PDF, …). An stpa.ini file contains the list of root file(s).

The compression of STEP files – Part 21 as well as Domain 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 and adds recommendations for derived attributes and for mandatory numerical attribtues.

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 an empty value. So, the receiver may interprete it in a different way than if the tag was not mentioned at all (unset value).
- For string-value attributes: string <tag>/NULL</tag> indicates user data in a mandatory attribute that is not managed by the sending system, or currently not known.

  Mandatory base identifiers like Organization.Id, Class.Id, Part.Id, Document.Id, File.Id, … shall not be mapped to ‘/NULL’, ‘/ANY nor ‘/UNSET’, except for Part.Name and Document.Name in case of nested files or incremental exchange.

  If an attribute is required or recommended to be unique (for example Occurrence.Id), it is not recommended to set it to ‘/NULL’, ‘/ANY nor ‘/UNSET’. since the sender should be aware that what he sends may not necessarily be interpreted correctly. If set to /NULL, ‘/ANY or ‘/UNSET’., the receiving system may have to construct an Id in some way to achieve uniqueness.

  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>/UNSET</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 /UNSET. This may apply for example to DateAndPersonOrganization.PersonOrOrganization.

Accordingly, it is not recommended to use the empty string or the default strings '/ANY', '/UNSET' 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 Domain 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 Domain Model postprocessor will generally provide user access to also these values.

Since a target PDM system will define many optional attributes not provided by the partner, it is not recommended to overwrite their value with 'unset' or an empty value during import. If this is wished, an empty value could be provided by the partner, or a specific agreement between exchange partners has to be defined.

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 (for each single object type) the EXPRESS schema of the AP242 Domain 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.

   For those objects having an attribute VersionId (optional), if the VersionId is set, then the uniqueness applies to the combination of Id and VersionId. Like for objects having a dedicated Version entity, the idContextRef of the VersionId shall be the Id(Identifier).uid.

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 Domain 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. See dedicated recommendation for the exchange of a single explicit Configuration in [242-CFG] section 4.1.2

The Instance Model: AP242 Domain Model XML entities and attributes

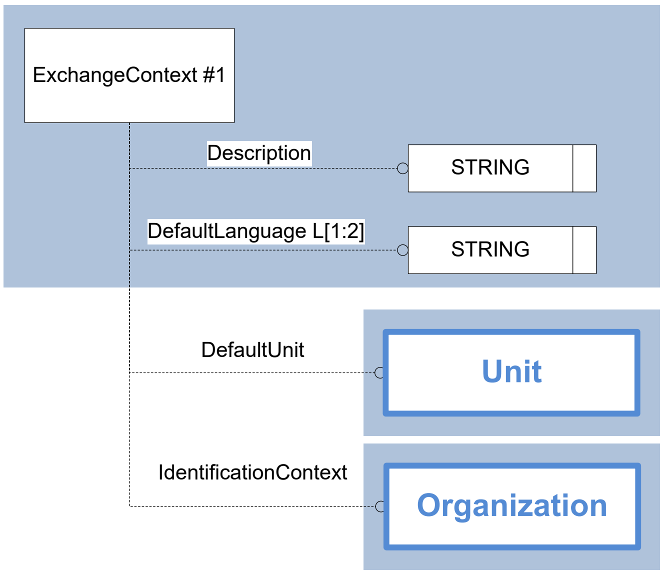


Figure 1: Template “ExchangeContext”

ENTITY ExchangeContext	**Attribute Type**
ClassLibrary	OPTIONAL IdentifierString
DefaultLanguage	OPTIONAL Language
DefaultUnit	OPTIONAL Unit
Description	OPTIONAL DescriptorSelect
IdentificationContext	OPTIONAL IdentificationContextSelect
ContractAssignment	OPTIONAL SET[1:?] of ContractAssignment

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
- The organization referenced by ExchangeContext.IdentificationContext is not necessarily redundant with the one referenced by Identifier.idContextRef, even if the Identifiers have only one context: for example, ExchangeContext.IdentificationContext could largely be representing a sight, but the data could be assigned to several organization and need not be always the sight.

Postprocessor Recommendations:

- 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, descriptions or StringValues, the DefaultLanguage should be set.

Related Entities: There are no specific related entities.

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

```
<ExchangeContext uid="ec--000000001">
  <DefaultLanguage>en-US</DefaultLanguage>
  <DefaultUnit uidRef="u--000000002"/>
  <Description>
    <CharacterString>AP242 Domain Model XML Assembly Structure ex-
change</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>
  <Quantity>
    <ClassString>length</ClassString>
  </Quantity>
</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 Domain Model XML entities and attributes

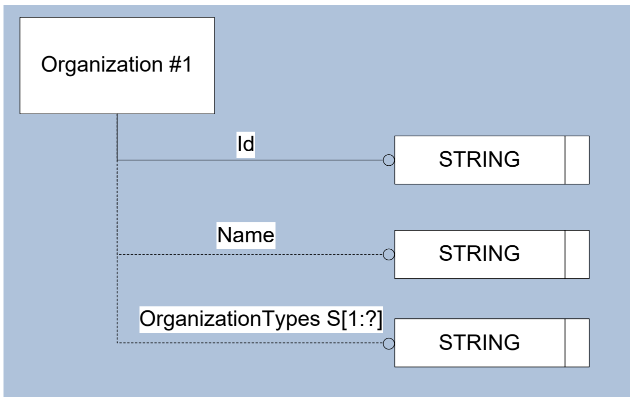


Figure 2: Template "Organization"

ENTITY Organization	**Attribute Type**
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
Name	OPTIONAL DescriptorSelect

ENTITY Organization	**Attribute Type**
OrganizationTypes	OPTIONAL SET[1:?] of ClassSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
AddressAssignment	OPTIONAL SET[1:?] of AddressAssignment
AnalysisAssignment	OPTIONAL SET[1:?] of AnalysisAssignment
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
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
OrganizationRelationship	OPTIONAL SET[1:?] of OrganizationRelationship
StateAssignment	OPTIONAL SET[1:?] of StateAssignment
StateDefinitionAssignment	OPTIONAL SET[1:?] of StateDefinitionAssignment
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 Domain 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 Domain Model XML entities and attributes

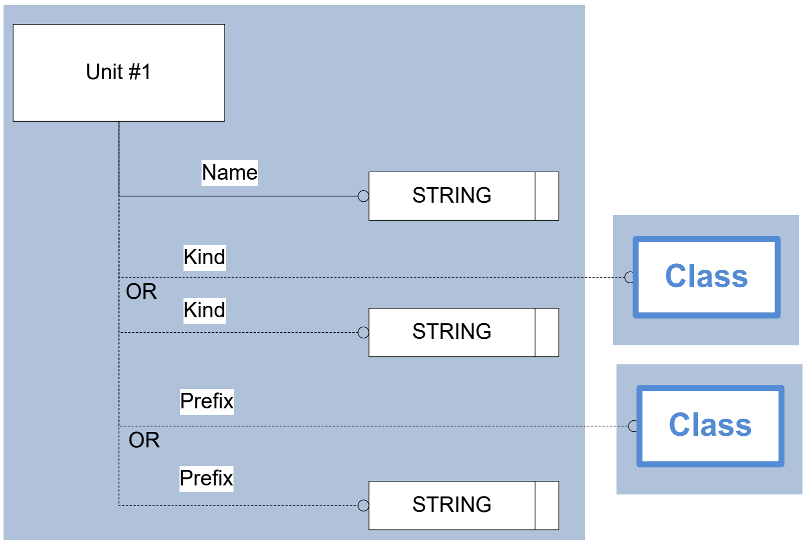


Figure 3: Template "Unit"

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. Refer to the chapter Annex E for the recommended values.
- The ***Kind*** attribute is the type of system used. 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, and according to the chapter E.2 the following values are also applicable: ‘derived SI unit’, ‘unspecified SI derived unit’, ‘unspecified’, ‘imperial’.
- 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’.
- The ***Quantity*** attribute is the type of property that the **Unit** shall be used for. Use ClassString. Refer to the chapter Annex E for the recommended values.
- 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 recommendation of the Units definition in Annex E lets the preprocessor free to use any prefix value recommended above.

Postprocessor Recommendations: The prefix shall be evaluated, and if necessary, the property value shall be converted into the appropriate prefix for the target system.

Related Entities: There are no specific related entities.

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

```
<Unit uid="u--000000002">
  <Kind>
    <ClassString>SI system</ClassString>
  </Kind>
  <Name>
    <ClassString>metre</ClassString>
  </Name>
  <Prefix>
    <ClassString>milli</ClassString>
  </Prefix>
  <Quantity>
    <ClassString>length</ClassString>
  </Quantity>
</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 Domain Model XML entities and attributes

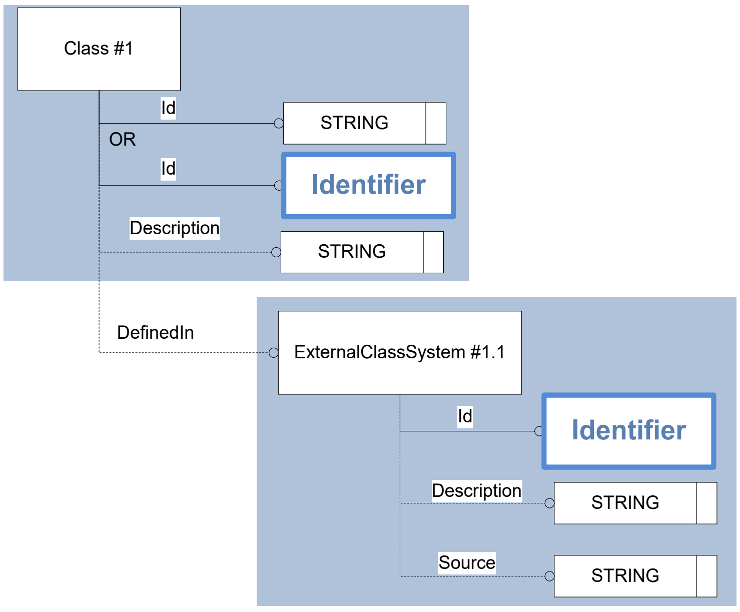


Figure 4: Template "Class"

ENTITY Class	**Attribute Type**
ClassAttribute	OPTIONAL SET[1:?] of ClassAttribute
DefinedIn	OPTIONAL ExternalClassSystem
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
Source	OPTIONAL ExternalSourceSelect
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 or in case the Class is used to map one of the predefined values for idRoleRef within the "Identifier" template (see 4.6.6), 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.
- The ***VersionId*** attribute specified the identification or set of identifications of a specific version of the Class. The value of this attribute need not be specified. Use "Class" template (see 4.6.4).
  The ISO Jira Task TCSC410303-1306 has been created to introduce the entity ClassRelationship in order to relate two Classes (like FileRelationship for example).

- 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
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
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. Use “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.

When the Classification.Role is used with the value ‘make or buy’, the following values shall be used for the attribute Id (according to value defined in 10303-41: product_definition_schema):

- ‘made’: characterize a product is to be manufactured within an organization;
- ‘bought’: characterize a product is to be purchased;

Postprocessor Recommendations: None specified.

Related Entities: There are no specific related entities.

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

```
<ExternalClassSystem uid="ecs--fdf">
  <Id>
    <Identifier uid="fdf--filedataformat-id1" id="file data format" idRol-
eRef="rl--ii" 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 Domain Model XML entities and attributes

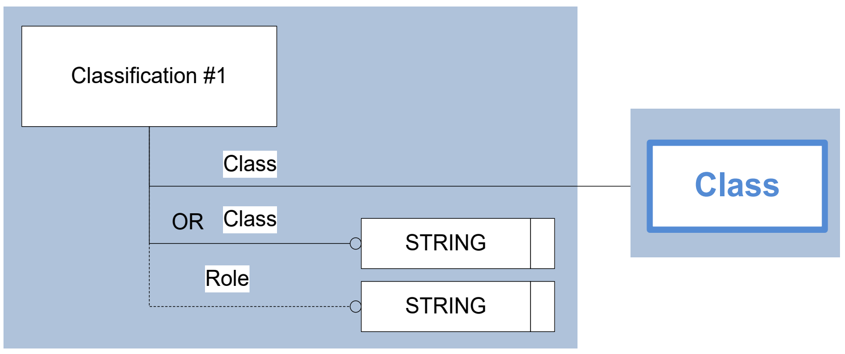


Figure 5: Template "Classification"

ENTITY Classification	**Attribute Type**
Class	ClassSelect
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:

- ‘customized type’: The classification categorizes the classified element as being a customized subtype of it (see section 14.2)
- ‘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
- ‘geometry type’: The classification categorizes the GeometryTypes of a ContentPropertiy (see section 10.2)
- ‘kind of properties’: The classification categorizes the PropertyValueAssigment (see section 6.2)
- ‘make or buy’: The classification categorizes the classified element with respect to its provenance, the company made the element or purchased it (see section 15.4)

- ‘specified reference’: The classification categorizes the classified element as being an incomplete/unchanged object according to the nested files exchange recommendations of section 9.3
- ‘VDA reference mechanism’: The classification categorizes the classified element as being a incomplete/unchanged object according to the incremental data exchange recommendations of section 9.4
- ‘alternative’: The classification categorizes the classified element as being an alternative to a part usage/occurrence and shall not be considered cumulated to the original part usage/occurrence. See sections 7.5.2 and 7.5.3.
- ‘delta change’: The classification categorizes the classified element as being a incomplete object according to the delta exchange recommendations of section 9.5

Postprocessor Recommendations: None specified.

Related Entities: There are no specific related entities.

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

```
<Classification uid="—gtc--1">
  <Class>
    <ClassString>specified reference</ClassString>
  </Class>
  <Role>specified reference</Role>
</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 one of the identifiers of the root object (in case of an embedded object).

The Instance Model: AP242 Domain Model XML entities and attributes

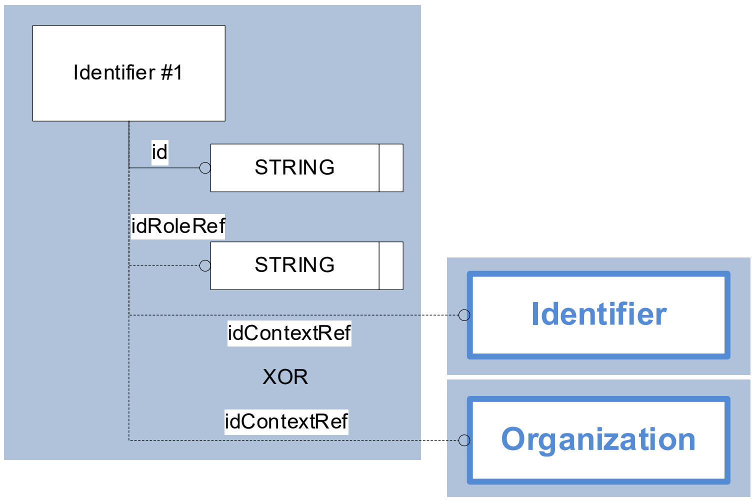


Figure 6: Template "Identifier"

ENTITY Identifier	**Attribute Type**
ClassifiedAs	OPTIONAL SET[1:?] of Classification
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
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
OrganizationOrPersonInOrganiza-tionAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganiza-tionAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
Id	OPTIONAL String
idRoleRef	ClassSelect
idContextRef	OPTIONAL IdentificationContextSelect

Table 8: "Identifier" Attributes

Attribute recommendations

- 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. The value of this attribute should be specified. Use “Class” template (see 4.6.4) and the following value:
  - ‘identification information’: the id identifies the object.
  - ‘exchange identification information’: the id identifies the object along the exchange process.

  For each idContextRef value, one any only one of these two roles is mandatory:
    - idRoleRef ‘identification information’: for example with an internal uid for the object in the PDM system used at the company idContextRef.
    - idRoleRef ‘exchange identification information’: for example with a readable part number in the context of idContextRef

  If an object has multiple contexts, one and only one identifier should have the role ‘exchange identification information’
- 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) for root objects or for embedded objects if they have a unique identifier independtly from the identifier of the root object, like ProductConfiguration.Id or Specification.Id (see [242-CFG]) or “Identifier” template (this section) for embedded objects like PartVersion.Id, DocumentVersion.Id, … (see complex example in section 5.1.6).
- Other attributes than these are not covered by this document; 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 fre-quently 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 Identifi-er/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. For a given context, only one id shall have the role ‘identification information’ or ‘exchange identification information’ (both roles shall not occur in parallel)

To map further ids for the same context, dedicated roles shall be defined and agreed between the exchange partners. According to the “Class” Template (see 4.6.4), such dedicated roles shall be mapped as a Class with an ExternalClassSystem.

In the case where the root object and its embedded objects have all the same context and only one role, embedded objects are already identified through their root object => IdentifierString can be used alternatively to Identifier.

If Identifier is used for an embedded object, its idContextRef shall reference one of the Identifiers of its root object, and the idRoleRef value (‘identification information’ or ‘exchange identification information’) shall be the same in the embedded object and in its idContextRef.

Examples:

- PartVersion.Id.idContextRef shall be Part.Id.id
- PartView.Id.idContextRef (if set) shall be PartVersion.Id.id
- Occurrence.Id.idContextRef shall be PartView.Id.id or (if unset) PartVersion.Id.id.
- DocumentVersion.Id.idContextRef shall be Document.Id.id
- DocumentDefinition.Id.idContextRef (if set) shall be DocumentVersion.Id.id

This recommendation does not apply necessarily to dedicated roles (for example the system internal object Id of PartVersion (idRoleRef='unique object identification information') could be defined having an Organization as ifContextRef).

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 Domain Model. All preprocessors must avoid it. The Id.id shall only be used in special cases like Organization.Id

This has been documented in the ISO Jira Task BS10303-3817.

Postprocessor Recommendations: If a target system does not support multiple identifiers, at least the one having the role 'exchange identification information' shall be imported. If none of them have this role, the one having the context of the receiver organization should be chosen. This is especially important if the data exchange is bidirectional and the data loops between the partners.

If an embedded object has less Identifier contexts that its root object, the missing Identifiers can be interpreted as '/NULL'.

Related Entities: There are no specific related entities.

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain 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>
```

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 (see Annex A.1.9 "Representation of Id Attribute"):

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

4.6.7 Template "Description"

If only one language is considered, the description shall be set with the `CharacterString` type. The applying language is implicitly the one defined in the ExchangeContext.

If multiple languages are to be exchanged for one description, LocalizedStrings shall be defined a described in A.1.10.

If values for multiple identification contexts are to be exchanged for one description, Descriptors shall be defined.

Remark: for a better readability, the instantiation figures included in the recommended practices only consider the simple case 'CharactString'.

Preprocessor Recommendation:

- If LocalizedStrings are used (i.e. if multiple languages are supported), at least the default language defined in the ExchangeContext should be provided, while further languages may be provided or omitted for each Description.
- If LocalizedStrings are used and there is no value set for the default language in the ExchangeContext, at least one common language shall occur in all Descriptions.

Postprocessor Recommendations:

- If a target system does not support multiple languages for some attributes of type Description, one language (one of the LocalizedStrings) has to be picked out (if provided, accordingly to the default language defined in the ExchangeContext, otherwise freely chosen among all available languages).

4.6.7.1 ENTITY Descriptor

If the same item can have several descriptions associated with it, depending on the identification context (for example Part is called 'part A' within the Organization O1 and is called 'part B' within the Organization O2), the use of Descriptor is recommended.

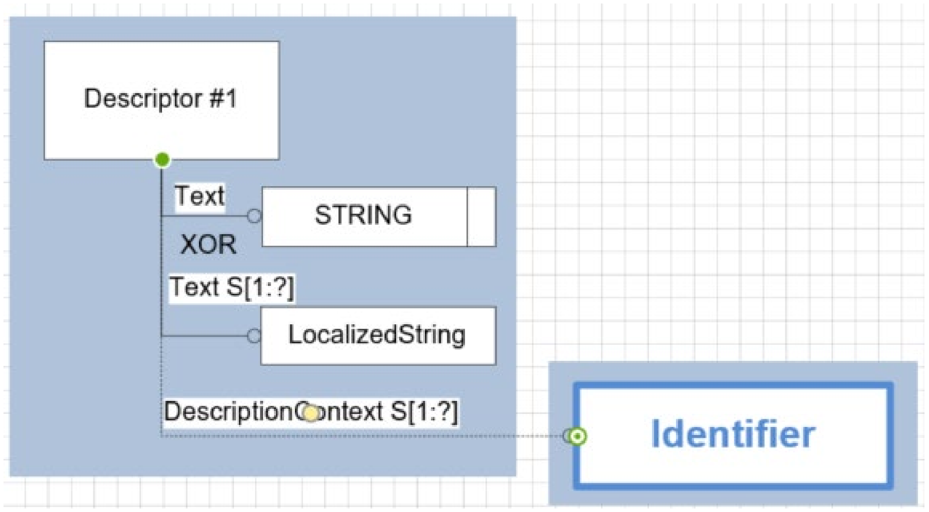


Figure 7: Template "Descriptor"

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

```
<Part uid="ID_14">
…
  <Id>
    <Identifier id="as1-MB" idRoleRef="ID_4" idContextRef="ID_22"
uid="ID_23"/>
    <Identifier id="as1-BO" idRoleRef="ID_3" idContextRef="ID_24"
uid="ID_25"/>
    <Identifier id="as1-TSI" idRoleRef="ID_3" idContextRef="ID_2"
uid="ID_26"/>
  </Id>
  <Name>
    <Descriptor uid="ds--1">
      <DescriptionContext>
        <Identifier uidRef="ID_23"/>
      </DescriptionContext>
      <Text>
        <LocalizedString lang="de-DE">MB as1 deutsch</LocalizedString>
        <LocalizedString lang="en-US">MB as1 english</LocalizedString>
        <LocalizedString lang="fr-FR">MB as1 francais</LocalizedString>
      </Text>
    </Descriptor>
    <Descriptor uid="ds--2">
      <DescriptionContext>
        <Identifier uidRef="ID_25"/>
        <Identifier uidRef="ID_26"/>
      </DescriptionContext>
      <Text>
        <LocalizedString lang="de-DE">as1 deutsch</LocalizedString>
        <LocalizedString lang="en-US">as1 english</LocalizedString>
        <LocalizedString lang="fr-FR">as1 francais</LocalizedString>
      </Text>
    </Descriptor>
  </Name>
…
</Part>
```

Entity Descriptor	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
DescriptionContext	OPTIONAL SET[1:?] OF IdentificationContextSelect
Id	OPTIONAL IdentifierSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
Text	MultiLingualStringSelect
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
ContractAssignment	OPTIONAL SET[1:?] of ContractAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssignment
DatetimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DescriptorRelationship	OPTIONAL SET[1:?] of DescriptorRelationship
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUsageRightAssign-ment
OrganizationOrPersonInOrganiza-tionAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganiza-tionAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 9: "Descriptor" Attributes

Attribute recommendations

- ***DescriptionContext***: the context(s) (specified by IdentificationContextSelect) to which the Descriptor applies. See idContextRef in "Identifier" template (section 4.6.6).
- ***Text***: the string providing the textual content of the description (as CharacterString or as one/many LocalizedStrings a described in A.1.10).
- 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 part-ners.

Preprocessor Recommendation:

- If LocalizedStrings are used (i.e. if multiple languages are supported), at least the default language defined in the ExchangeContext should be provided, while further languages may be provided or omitted for each Description.
- If LocalizedStrings are used and there is no value set for the default language in the ExchangeContext, at least one common language shall occur in all Descriptions.

- The idContextRefs present in Descriptor shall be the same as the one present in the Identifier of the object that contains the Descriptor.
- If the attribute of type Descriptor is mandatory (like Part.Name), a value shall be given for each idContextRef present in the Identifier of the object that contains the Descriptor.
- Like for Identifiers used for embedded objects in section 4.6.6, DescriptorContext shall reference one of the Identifiers of its root object.

Postprocessor Recommendations:

- If a target system does not support multiple languages for some attributes of type Description, one language (one of the LocalizedStrings) has to be picked out (if provided, accordingly to the default language defined in the ExchangeContext, otherwise freely chosen among all available languages).
- If an object contains idContextRefs in a Descriptor and the object Identifier doesn't have such an idContextRef, the Descriptor shall be ignored or a warning shall be recorded by the application.

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 Domain Model XML entities and attributes

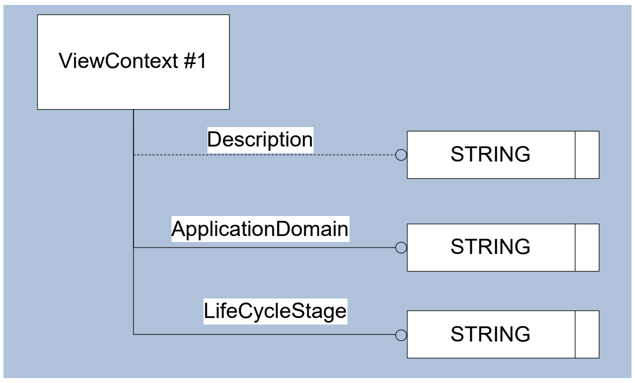


Figure 8: Template "ViewContext"

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

Table 10: "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. As defined in the ISO AP242 specification, use ProxyString if the value of this element is one or several of the following:
  'assembly study', 'digital mock-up', 'electrical design', 'mechanical design', 'preliminary design', 'process planning', 'product delivery', 'product support' and 'not specific', otherwise use PredefinedApplicationDomainEnum as far as possible, otherwise use ApplicationDomain.
- The ***LifeCycleStage*** attribute is the text representing the life cycle stage. As defined in the ISO AP242 specification, use ProxyString if the value of this element is one or several of the following:
  'design', 'manufacturing planning', 'manufacturing realization', 'documentation'/'certification', 'built'/'manufactured', 'delivery', 'support', 'recycling' and 'not specific'.
- 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.
- Although some values already recommended for ViewContext are defined in PredefinedApplicationDomainEnum, for upward compatibility reasons, their mapping as ProxyString is recommended.For the recommended combinations of values for ApplicationDomain and LifeCycleStage, refer to the management of complex structure with different views (15.1).
- All preprocessors should ensure that the LifeCycleStage is unique over the Initial/AdditionalContexts of a PartView.

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.
- For upward compatibility reasons: PredefinedApplicationDomainEnum should be also supported.

Related Entities: There are no specific related entities.

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain 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.

A `ValueLimit` is a subtype of `NumericalValue` with a qualified numerical value representing either the lower limit or the upper limit of a specific physical characteristic.

EXAMPLE '30.5 max' and '5 min' are examples for a ValueLimit.

The Instance Model: AP242 Domain Model XML entities and attributes

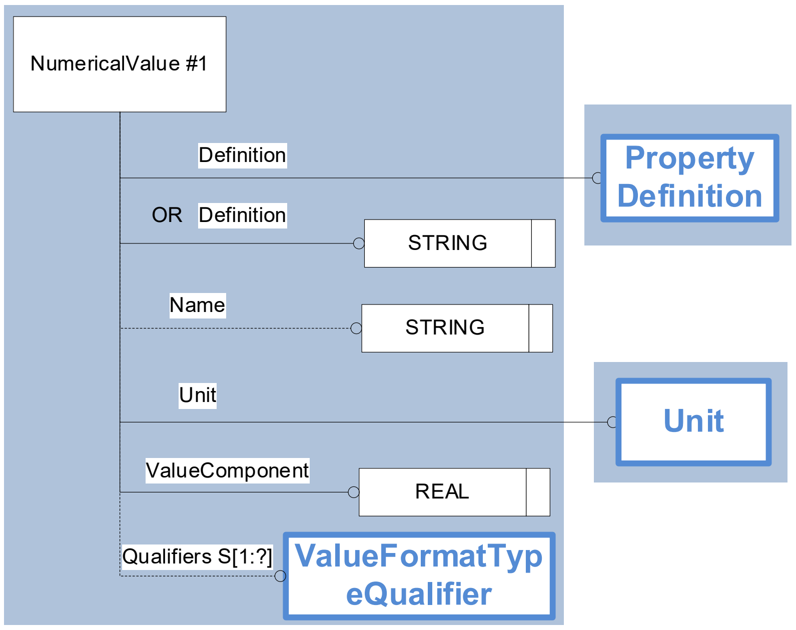


Figure 9: Template "NumericalValue

ENTITY NumericalValue	**Attribute Type**
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Definition	OPTIONAL PropertyDefinitionSelect
DeterminationMethod	OPTIONAL ClassSelect
Name	OPTIONAL DescriptorSelect
Qualifiers	OPTIONAL ValueQualifierSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
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

ENTITY NumericalValue	**Attribute Type**
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
(only for ValueLimit) Qualifier	TypeQualifierSelect

Table 11: "NumericalValue" Attributes

Attribute recommendations

- The ***Definition*** attribute is the description of the property. In the case of system property, PDM property, User Defined attributes (see chapter 12) or validation properties (see chapter 13), use “PropertyDefinition” template (see 12.2), otherwise (for example FileSize or Quantity) use PropertyDefinitionString type.
- The ***DeterminationMethod*** attribute informs on how the property value shall be interpreted. 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).

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

Determination-Method	**Explanation**
‘calculated’	The value has been calculated
‘designed’	The value represents a value intended by the design
‘estimated’	The value has been estimated
‘measured’	The value has been measured
‘required’	The value represents a requirement
‘set point’	The value is used as the initialization value

- The ***Name*** attribute is the text by which the property is known. The value of this attribute shall be specified except in special cases like NextAssemblyViewUsage.Quantity and DigitalFile.FileSize (here the Name shall be set the attribute name (‘quantity’ or ‘filesize’) or unset). Use “Description” template (see 4.6.7).
- The ***Qualifiers*** attribute allows for the designation of the data type ValueFormatTypeQualifier (see 4.6.21). The value of this attribute need not be specified.
- 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.
- The ***Qualifier*** attribute specifies the kind of the limit (via PredefinedTypeQualifier.Name).
- 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:

- Even if PropertyValue,Definition becomes optional with Ed2, for upward compatibility reasons, it is still recommended to set it as if it was mandatory.
- Although some PropertyValue.Definition values are now defined defined in PropertyDefinitionEnum, for upward compatibility reasons, their mapping as PropertyDefinition is recommended.

Postprocessor Recommendations:

- For upward compatibility reasons: PropertyDefinitionString and PropertyDefinitionEnum should be also supported.
- In case of PropertyDefinitionString, it is assumed that the value corresponds to PropertyDefinition.PropertyType while PropertyDefinition.Id is ‘general property’.

Related Entities: There are no specific related entities.

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

```
<Unit uid="u--000000003">
  <Kind>
    <ClassString>unspecified</ClassString>
  </Kind>
  <Name>
    <ClassString>byte</ClassString>
  </Name>
  <Prefix>
    <ClassString>kilo</ClassString>
  </Prefix>
  <Quantity>
    <ClassString>data size</ClassString>
  </Quantity>
</Unit>
<File xsi:type="n1:DigitalFile" uid="df--0000000088">
  <FileSize uid="fsp--1">
    <FileSize xsi:type="n0:NumericalValue">
      <Definition>
        <PropertyDefinitionString>file size property</PropertyDefini-
tionString>
      </Definition>
      <Name>
        <CharacterString>file size</CharacterString>
      </Name>
      <Unit uidRef="u--000000003"></Unit>
      <ValueComponent>46</ValueComponent>
    </FileSize>
  </FileSize>
</File>
```

Example of ValueLimit:

```
<PreDefinedTypeQualifier uid="ptq--1">
  <Name>
```

```
    <ClassString>max</ClassString>
  </Name>
</PreDefinedTypeQualifier>

…
<Part uid="ID_14">
…
  <Versions>
    <PartVersion uid="ID_27">
…
      <Views>
        <PartView xsi:type="ns2:AssemblyDefinition" uid="ID_31">
…
          <PropertyValueAssignment uid="ID_52">
            </AssignedPropertyValues>
              <PropertyValue xsi:type="ns2:ValueLimit" uid="ID_59">
                <Definition>
                  <PropertyDefinition uidRef="ID_60"/>
                </Definition>
                <Name>
                  <CharacterString>PDMIFPart_Real</CharacterString>
                </Name>
                <Unit uidRef="ID_61"/>
                <ValueComponent>5.678</ValueComponent>
                <Qualifier uidRef="ptq--1"/>
              </PropertyValue>
            </AssignedPropertyValues>
…
          </PropertyValueAssignment>
…
        </PartView>
      </Views>
…
    </PartVersion>
  </Versions>
…
</Part>
```

4.6.10 Template “StringValue”

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

The Instance Model: AP242 Domain Model XML entities and attributes

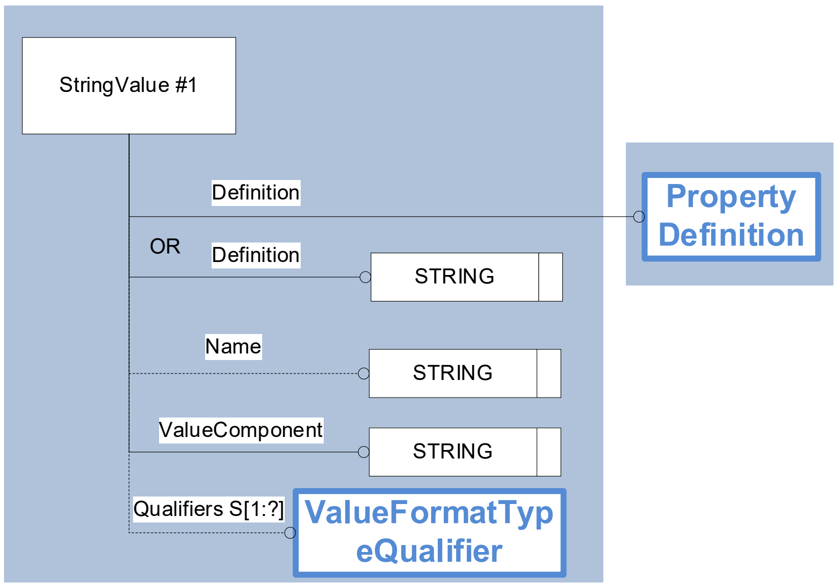


Figure 10: Template "StringValue"

ENTITY StringValue	**Attribute Type**
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Definition	OPTIONAL PropertyDefinitionSelect
DeterminationMethod	OPTIONAL ClassSelect
Name	OPTIONAL DescriptorSelect
Qualifiers	OPTIONAL ValueQualifierSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
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
ProjectAssignment	OPTIONAL SET[1:?] of ProjectAssignment
PropertyValueRelationship	OPTIONAL SET[1:?] of PropertyValueRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
ValueComponent	MultiLingualStringSelect

Table 12: "StringValue" Attributes

Attribute recommendations

- The ***Definition*** attribute is the definition of the property. In the case of e of system property, PDM property, User Defined attributes (see chapter 12) or validation properties (see chapter0), use “PropertyDefinition” template (see 12.2) .
- The ***Name*** attribute is the text by which the property is known. The value of this attribute shall be specified. Use of “Description” template (see 4.6.7).
- The ***Qualifiers*** attribute allows for the designation of the data type ValueFormatTypeQualifier (see 4.6.21). The value of this attribute need not be specified.
- 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:

- Even if PropertyValue,Definition becomes optional with Ed2, for upward compatibility reasons, it is still recommended to set it as if it was mandatory.
- Although some PropertyValue.Definition values are now defined defined in PropertyDefinitionEnum, for upward compatibility reasons, their mapping as PropertyDefinition is recommended.

Postprocessor Recommendations:

- For upward compatibility reasons: PropertyDefinitionString and PropertyDefinitionEnum should be also supported.
- In case of PropertyDefinitionString, it is assumed that the value corresponds to PropertyDefinition.PropertyType while PropertyDefinition.Id is ‘general property’.

Related Entities: There are no specific related entities.

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain 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>

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

4.6.11 Template "ValueRange"

The `ValueRange` is a subtype of `ValueWithUnit` with a pair of numerical values (sharing the same Unit) representing the range in which the value shall lie.

The Instance Model: AP242 Domain Model XML entities and attributes

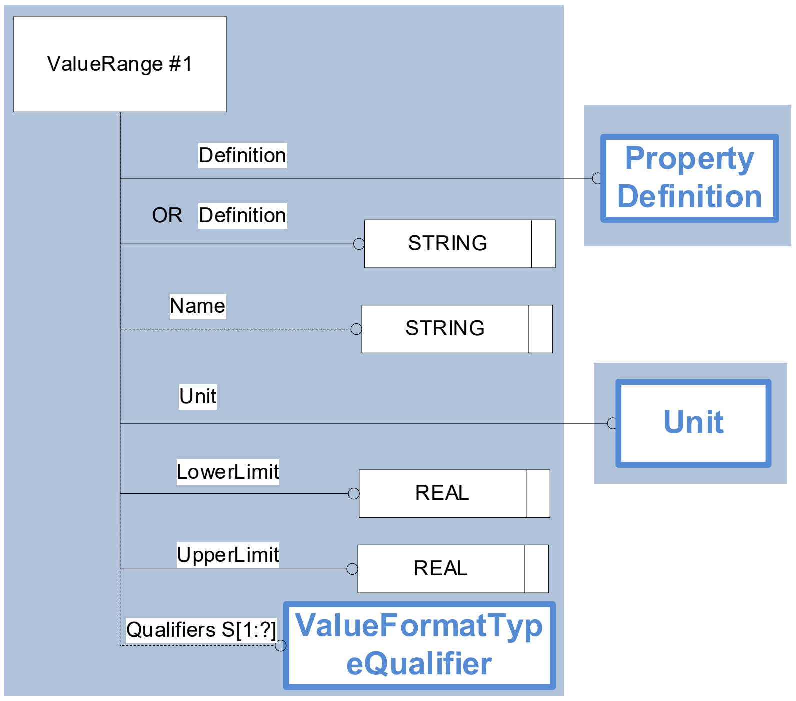


Figure 11: Template "ValueRange"

ENTITY ValueRange	**Attribute Type**
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Definition	OPTIONAL PropertyDefinitionSelect
DeterminationMethod	OPTIONAL ClassSelect
Name	OPTIONAL DescriptorSelect
Qualifiers	OPTIONAL ValueQualifierSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
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

ENTITY ValueRange	**Attribute Type**
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
LowerLimit	Double
UpperLimit	Double

Table 13: "ValueRange" Attributes

Attribute recommendations

- The ***Definition*** attribute is the description of the property. In the case of system property, PDM property, User Defined attributes (see chapter 12) or validation properties (see chapter 13), use “PropertyDefinition” template (see 12.2).
- The ***Name*** attribute is the text by which the property is known. The value of this attribute shall be specified. Use “Description” template (see 4.6.7).
- The ***Qualifiers*** attribute allows for the designation of the data type ValueFormatTypeQualifier (see 4.6.21). The value of this attribute need not be specified.
- The ***Unit*** attribute is the Unit of the expressed value. Use “Unit” template (see 4.6.3).
- The ***LowerLimit*** attribute specifies the minimum acceptable value that is constrained by the ValueRange.
- The ***UpperLimit*** attribute specifies the maximum acceptable value that is constrained by the ValueRange.
- 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:

- Even if PropertyValue,Definition becomes optional with Ed2, for upward compatibility reasons, it is still recommended to set it as if it was mandatory.
- Although some PropertyValue.Definition values are now defined defined in PropertyDefinitionEnum, for upward compatibility reasons, their mapping as PropertyDefinition is recommended.

Postprocessor Recommendations:

- For upward compatibility reasons: PropertyDefinitionString and PropertyDefinitionEnum should be also supported.
- In case of PropertyDefinitionString, it is assumed that the value corresponds to PropertyDefinition.PropertyType while PropertyDefinition.Id is ‘general property’.

Related Entities: There are no specific related entities.

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

```
<Unit uid="ID_161">
      <Kind>
            <ClassString>unspecified SI derived unit</ClassString>
      </Kind>
      <Name>
            <ClassString>litre</ClassString>
      </Name>
      <Prefix>
            <ClassString>centi</ClassString>
      </Prefix>
      <Quantity>
            <ClassString>volume</ClassString>
      </Quantity>
</Unit>
<PropertyDefinition uid="ID_57">
      <Id id="general property"/>
      <PropertyType>
            <ClassString>customized PDM property</ClassString>
      </PropertyType>
</PropertyDefinition>
<PropertyValue uid="ID_162" xsi:type="n0:ValueRange">
      <Definition>
            <PropertyDefinition uidRef="ID_57"/>
      </Definition>
      <Name>
            <CharacterString>PDMIFPartVers_Range</CharacterString>
      </Name>
      <Unit uidRef="ID_161"/>
      <LowerLimit>6.789000000000000E+00</LowerLimit>
      <UpperLimit>7.890000000000000E+00</UpperLimit>
</PropertyValue>
```

4.6.12 Template “ValueWithTolerances”

The `ValueWithTolerances` is a subtype of `ValueWithUnit` that specifies a range of values by specifying a single nominal value and two tolerances that are offsets from the single value. The range is defined to be the closed interval [tolerancedValue + lowerLimit, tolerancedValue + upperLimit].

The Instance Model: AP242 Domain Model XML entities and attributes

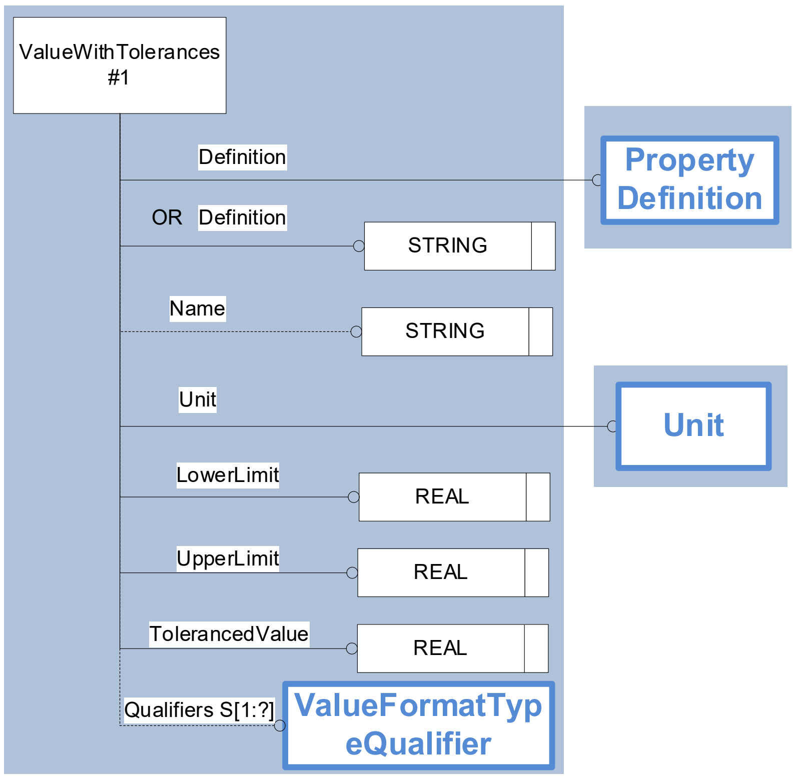


Figure 12: Template "ValueWithTolerances"

ENTITY ValueWithTolerances	**Attribute Type**
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Definition	OPTIONAL PropertyDefinitionSelect
DeterminationMethod	OPTIONAL ClassSelect
Name	OPTIONAL DescriptorSelect
Qualifiers	OPTIONAL ValueQualifierSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
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
ProjectAssignment	OPTIONAL SET[1:?] of ProjectAssignment

ENTITY ValueWithTolerances	**Attribute Type**
PropertyValueRelationship	OPTIONAL SET[1:?] of PropertyValueRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
SignificantDigits	Optional Integer
Unit	UnitSelect
LowerLimit	Double
ToleranceValue	Double
UpperLimit	Double

Table 14: "ValueWithTolerances" Attributes

Attribute recommendations

- The ***Definition*** attribute is the description of the property. In the case of system property, PDM property, User Defined attributes (see chapter 12) or validation properties (see chapter 13), use “PropertyDefinition” template (see 12.2).
- The ***Name*** attribute is the text by which the property is known. The value of this attribute shall be specified. Use “Description” template (see 4.6.7).
- The ***Qualifiers*** attribute allows for the designation of the data type ValueFormatTypeQualifier (see 4.6.21). The value of this attribute need not be specified.
- The ***Unit*** attribute is the Unit of the expressed value. Use “Unit” template (see 4.6.3).
- The ***LowerLimit*** attribute specifies the value used to build the minimum acceptable value as [toleranceValue + lowerLimit].
- The ***UpperLimit*** attribute specifies the value used to build the minimum acceptable value as [toleranceValue + upperLimit].
- 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:

- Even if PropertyValue,Definition becomes optional with Ed2, for upward compatibility reasons, it is still recommended to set it as if it was mandatory.
- Although some PropertyValue.Definition values are now defined defined in PropertyDef-initionEnum, for upward compatibility reasons, their mapping as PropertyDefinition is recommended.

Postprocessor Recommendations:

- For upward compatibility reasons: PropertyDefinitionString and PropertyDefinitionEnum should be also supported.
- In case of PropertyDefinitionString, it is assumed that the value corresponds to Proper-tyDefinition.PropertyType while PropertyDefinition.Id is ‘general property’.

Related Entities: There are no specific related entities.

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

```
<Unit uid="U__1">
      <Kind>
            <ClassString>SI system</ClassString>
      </Kind>
      <Name>
            <ClassString>metre</ClassString>
      </Name>
      <Prefix>
            <ClassString>milli</ClassString>
      </Prefix>
      <Quantity>
            <ClassString>length</ClassString>
      </Quantity>
</Unit>
<PropertyDefinition uid="ID_57">
      <Id id="general property"/>
      <PropertyType>
            <ClassString>customized PDM property</ClassString>
      </PropertyType>
</PropertyDefinition>
<PropertyValue uid="ID_159_3" xsi:type="n0:ValueWithTolerances">
      <Definition>
            <PropertyDefinition uidRef="ID_57"/>
      </Definition>
      <Name>
            <CharacterString>PDMIFPvr_Tol</CharacterString>
      </Name>
      <Unit uidRef="U__1"/>
      <LowerLimit>-3.000000000000000E-03</LowerLimit>
      <TolerancedValue>8.901000000000000E+00</TolerancedValue>
      <UpperLimit>4.000000000000000E-03</UpperLimit>
</PropertyValue>
```

4.6.13 Template “LimitsAndFits”

In the future PMI area of Edition 4, `LimitsAndFits` is a tolerance specified by reference to the limits-and-fits system standardized by ISO 286. The application of `LimitsAndFits` shall conform to the requirements of ISO 286. Limits and fits is a system for assigning tolerances associated with the assembly of mating product features.

EXAMPLE 1 A hole and a shaft that are intended to mate in an assembly.

In limits-and-fits system, a tolerance is assigned to a feature of size by classification rather than by giving an explicit value. A tolerance classification is composed of a fundamental deviation or "position letter" and a tolerance grade. These are used to calculate the size of the tolerance zone and its location relative to the basic or nominal size of the feature according to the tables and formulas found in ISO 286.

The Instance Model: AP242 Domain Model XML entities and attributes

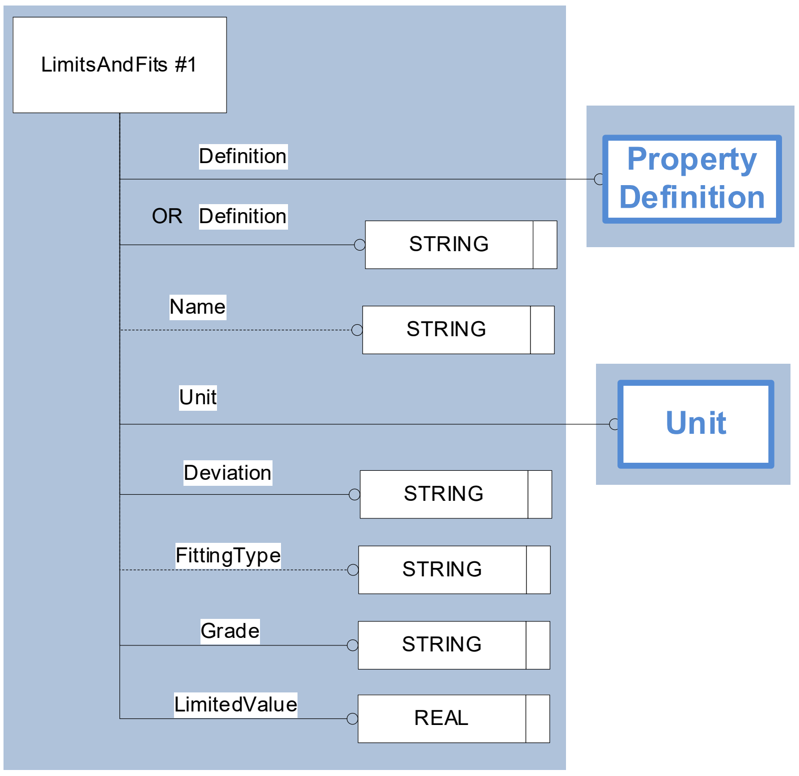


Figure 13: Template "LimitsAndFits"

ENTITY LimitsAndFits	**Attribute Type**
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Definition	OPTIONAL PropertyDefinitionSelect
DeterminationMethod	OPTIONAL ClassSelect
Name	OPTIONAL DescriptorSelect
Qualifiers	OPTIONAL ValueQualifierSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
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
ProjectAssignment	OPTIONAL SET[1:?] of ProjectAssignment

ENTITY LimitsAndFits	**Attribute Type**
PropertyValueRelationship	OPTIONAL SET[1:?] of PropertyValueRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
SignificantDigits	Optional Integer
Unit	UnitSelect
Deviation	String
FittingType	OPTIONAL String
Grade	String
LimitedValue	Double

Table 15: "LimitsAndFits" Attributes

Attribute recommendations

- The ***Definition*** attribute is the description of the property. In the case of system property, PDM property, User Defined attributes (see chapter 12) or validation properties (see chapter 13), use “PropertyDefinition” template (see 12.2).
- The ***Name*** attribute is the text by which the property is known. The value of this attribute shall be specified. Use “Description” template (see 4.6.7).
- The ***Qualifiers*** attribute allows for the designation of the data type ValueFormatTypeQualifier (see 4.6.21). The value of this attribute need not be specified.
- The ***Unit*** attribute is the Unit of the expressed value. Use “Unit” template (see 4.6.3).
- ***Deviation***: a representation of the fundamental deviation or "position letter" of the ISO 286 limits-and-fits tolerance classification. Only symbols defined in ISO 286 shall be provided. NOTE 1 The symbols 'A' to 'ZC' for holes or 'a' to 'zc' for shafts may be used to classify deviation.
- ***FittingType***: the kind of fitting to which the tolerance applies. The value of this attribute need not be specified. Normally the tolerance applies to an external or internal feature of the work-piece. Where applicable the following values shall be used:

FittingType	
‘hole’	the tolerance applies to an internal feature of the work-piece
‘shaft’	the tolerance applies to an external feature of the work-piece

  NOTE 2 Whether the tolerance applies to an interior or exterior feature can be determined from the position letter of the fundamental deviation, therefore fitting_type need not be specified.
- ***Grade***: a representation of the quality or the accuracy grade of a tolerance. Only symbols defined in ISO 286 shall be provided.
  NOTE 3 The grade is one of the 18 international standard tolerance grades defined in ISO 286.
  EXAMPLE 2 'IT07' is a standard tolerance grade.
- ***LimitedValue***: the value that is limited.

- 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:

- …

Postprocessor Recommendations:

- …

Related Entities: There are no specific related entities.

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

…

4.6.14 Template "ValueList"

The `ValueList` is a subtype of `PropertyValue` with an ordered collection of PropertyValue objects.

The Instance Model: AP242 Domain Model XML entities and attributes

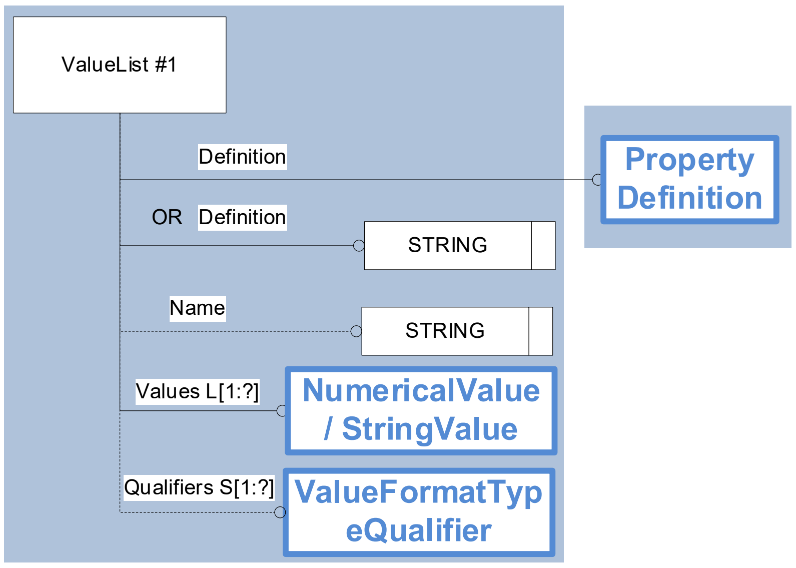


Figure 14: Template "ValueList"

ENTITY ValueList	**Attribute Type**
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Definition	OPTIONAL PropertyDefinitionSelect
DeterminationMethod	OPTIONAL ClassSelect
Name	OPTIONAL DescriptorSelect
Qualifiers	OPTIONAL ValueQualifierSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment

ENTITY ValueList	**Attribute Type**
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
ProjectAssignment	OPTIONAL SET[1:?] of ProjectAssignment
PropertyValueRelationship	OPTIONAL SET[1:?] of PropertyValueRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
Values	LIST[1:?] OF PropertyValue

Table 16: "ValueList" Attributes

Attribute recommendations

- The ***Definition*** attribute is the description of the property. Use “PropertyDefinition” template (see 12.2).
- The ***Name*** attribute is the text by which the property is known. The value of this attribute shall be specified. Use “Description” template (see 4.6.7).
- The ***Qualifiers*** attribute allows for the designation of the data type ValueFormatTypeQualifier (see 4.6.21). The value of this attribute need not be specified.
- The ***Values*** attribute specifies the ordered collection of PropertyValue objects that together are provided as a PropertyValue. Use one of “StringValue” template (see 4.6.10), “NumericalValue” template (see 4.6.9). The use of ValueList for element having further types (ValueList, ValueSet, ValueRange, ValueWithTolerances) is not recommended, since probably not supported by the target system.
- 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:

- Even if PropertyValue,Definition becomes optional with Ed2, for upward compatibility reasons, it is still recommended to set it as if it was mandatory.
- Although some PropertyValue.Definition values are now defined defined in PropertyDefinitionEnum, for upward compatibility reasons, their mapping as PropertyDefinition is recommended.
- It is recommended that all the value elements of a ValueList have the same type (StringValue or NumericalValue), the same PropertyDefinition (can be omitted), the same Name (can be omitted) and the same Unit (in case of NumericalValue).
- If the list of values is not an ordered collection, rather use ValueSet (see 4.6.15).

Postprocessor Recommendations:

- For upward compatibility reasons: PropertyDefinitionString and PropertyDefinitionEnum should be also supported.
- In case of PropertyDefinitionString, it is assumed that the value corresponds to PropertyDefinition.PropertyType while PropertyDefinition.Id is 'general property'.
- If ValueList is not supported by the target system, either concatenate the values to a large string or map each value to a distinct attribute.

Related Entities: There are no specific related entities.

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

```
<PropertyDefinition uid="ID_57">
      <Id id="general property"/>
      <PropertyType>
            <ClassString>customized PDM property</ClassString>
      </PropertyType>
</PropertyDefinition>
<PropertyValue uid="ID_170_3" xsi:type="n0:ValueList">
  <Definition>
    <PropertyDefinition uidRef="ID_57"/>
  </Definition>
  <Name>
    <CharacterString>PDMIFPvr_ValueList</CharacterString>
  </Name>
  <Values>
    <PropertyValue xsi:type="n0:StringValue" uid="ID_170_3_1">
      <ValueComponent>
        <CharacterString>PDMIFPvr_ValueList#1</CharacterString>
      </ValueComponent>
    </PropertyValue>
    <PropertyValue xsi:type="n0:StringValue" uid="ID_170_3_2">
      <ValueComponent>
        <CharacterString>PDMIFPvr_ValueList#2</CharacterString>
      </ValueComponent>
    </PropertyValue>
    <PropertyValue xsi:type="n0:StringValue" uid="ID_170_3_3">
      <ValueComponent>
        <CharacterString>PDMIFPvr_ValueList#3</CharacterString>
      </ValueComponent>
    </PropertyValue>
  </Values>
</PropertyValue>
```

4.6.15 Template "ValueSet"

The `ValueSet` is a subtype of `PropertyValue` with an <u>unordered collection</u> of PropertyValue objects.

The Instance Model: AP242 Domain Model XML entities and attributes

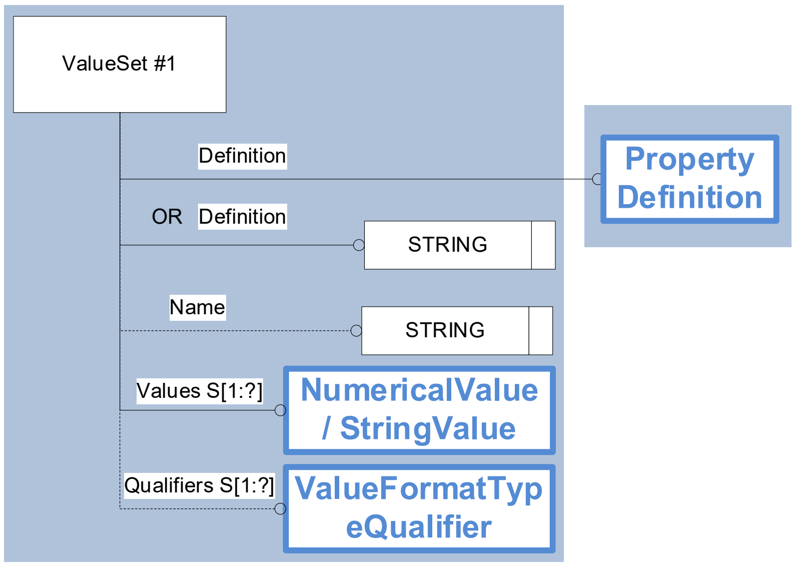


Figure 15: Template "ValueSet"

ENTITY ValueSet	**Attribute Type**
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Definition	OPTIONAL PropertyDefinitionSelect
DeterminationMethod	OPTIONAL ClassSelect
Name	OPTIONAL DescriptorSelect
Qualifiers	OPTIONAL ValueQualifierSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
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
ProjectAssignment	OPTIONAL SET[1:?] of ProjectAssignment
PropertyValueRelationship	OPTIONAL SET[1:?] of PropertyValueRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
Values	SET[1:?] OF PropertyValue

Table 17: "ValueSet" Attributes

Attribute recommendations

- The ***Definition*** attribute is the description of the property. Use “PropertyDefinition” template (see 12.2).
- The ***Name*** attribute is the text by which the property is known. The value of this attribute shall be specified. Use “Description” template (see 4.6.7).
- The ***Qualifiers*** attribute allows for the designation of the data type ValueFormatTypeQualifier (see 4.6.21). The value of this attribute need not be specified.
- The ***Values*** attribute specifies the ordered collection of PropertyValue objects that together are provided as a PropertyValue. Use one of “StringValue” template (see 4.6.10), “NumericalValue” template (see 4.6.9). The use of ValueSet for elements having further types (ValueList, ValueSet, ValueRange, ValueWithTolerances) is not recommended, since probably not supported by the target system.
- 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:

- Even if PropertyValue,Definition becomes optional with Ed2, for upward compatibility reasons, it is still recommended to set it as if it was mandatory.
- Although some PropertyValue.Definition values are now defined defined in PropertyDefinitionEnum, for upward compatibility reasons, their mapping as PropertyDefinition is recommended.
- It is recommended that all the value elements of a ValueSet have the same type (StringValue or NumericalValue), the same PropertyDefinition (can be omitted), the same Name (can be omitted) and the same Unit (in case of NumericalValue).
- If the set of values is an ordered collection, rather use ValueList (see 4.6.13).
- If the values are not unique within the ValueSet, some EXPRESS-based preprocessor might purge duplicate values, since a SET may not contain duplicates. In this case, rather use ValueList (see 4.6.13) even if the ordering in not relevant.

Postprocessor Recommendations:

- For upward compatibility reasons: PropertyDefinitionString and PropertyDefinitionEnum should be also supported.
- In case of PropertyDefinitionString, it is assumed that the value corresponds to PropertyDefinition.PropertyType while PropertyDefinition.Id is ‘general property’.
- If ValueSet is not supported by the target system, either concatenate the values to a large string or map each value to a distinct attribute.

Related Entities: There are no specific related entities.

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

```
<PropertyDefinition uid="ID_57">
      <Id id="general property"/>
      <PropertyType>
            <ClassString>customized PDM property</ClassString>
      </PropertyType>
</PropertyDefinition>
<PropertyValue uid="ID_170_3" xsi:type="n0:ValueSet">
  <Definition>
    <PropertyDefinition uidRef="ID_57"/>
  </Definition>
  <Name>
    <CharacterString>PDMIFPvr_ValueSet</CharacterString>
  </Name>
  <Values>
    <PropertyValue xsi:type="n0:StringValue" uid="ID_170_3_1">
      <ValueComponent>
        <CharacterString>PDMIFPvr_ValueSet#1</CharacterString>
      </ValueComponent>
    </PropertyValue>
    <PropertyValue xsi:type="n0:StringValue" uid="ID_170_3_2">
      <ValueComponent>
        <CharacterString>PDMIFPvr_ValueSet#2</CharacterString>
      </ValueComponent>
    </PropertyValue>
    <PropertyValue xsi:type="n0:StringValue" uid="ID_170_3_3">
      <ValueComponent>
        <CharacterString>PDMIFPvr_ValueSet#3</CharacterString>
      </ValueComponent>
    </PropertyValue>
  </Values>
</PropertyValue>
```

4.6.16 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 Domain Model XML entities and attributes

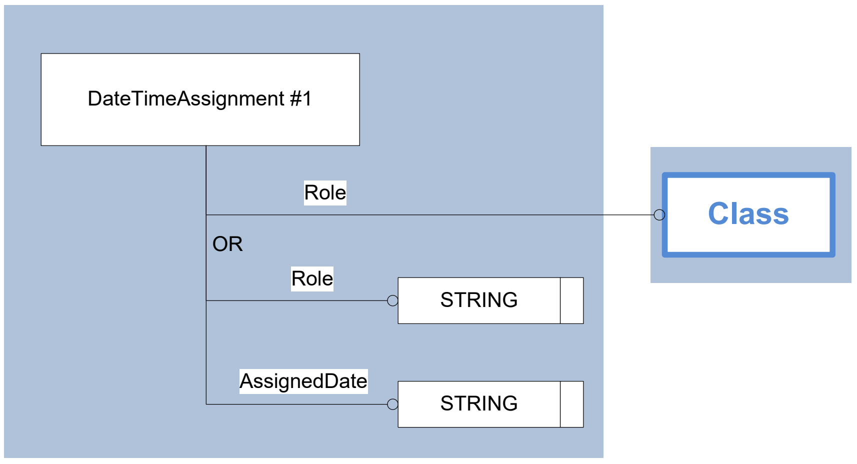


Figure 16: Template “DateTime”

ENTITY DateTimeAssignment	**Attribute Type**
AssignedDate	Xsd:dateTime
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
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
PropertyValueRelationship	OPTIONAL SET[1:?] of PropertyValueRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 18: "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 (YYYY-MM-DDThh:mm:ss[.mmm][TZD]). In the Domain 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. The hour component shall be expressed out of 24 hours. If no time is available, it is recommended to set it with the following default value: YYYY-MM-DDT00:00:00.

Postprocessor Recommendations:

If no time is given, then assume 00:00:00. Except in case of customized PDM properties (see section 14.3), 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 Domain Model XML syntax)

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

4.6.17 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 Domain Model XML entities and attributes

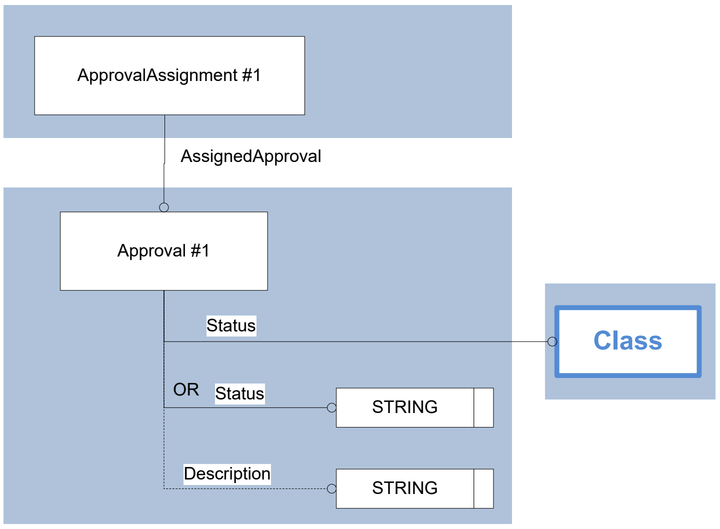


Figure 17: Template "Approval"

ENTITY ApprovalAssignment	**Attribute Type**
AssignedApproval	Approval
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
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
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganiza-tionAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganiza-tionAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueRelationship	OPTIONAL SET[1:?] of PropertyValueRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 19: "ApprovalAssignment" Attributes

Attribute recommendations

- The ***AssignedApproval*** attribute is the reference to the Approval entity.
- **Role:** the meaning of the assignment. The value of this attribute need not be specified. In case of customized PDM properties (see section 14.3), use ClassString, 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: 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 Proxy
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
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
StateAssignment	OPTIONAL SET[1:?] of StateAssignment
StateDefinitionAssignment	OPTIONAL SET[1:?] of StateDefinitionAssignment
PropertyValueRelationship	OPTIONAL SET[1:?] of PropertyValueRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 20: "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 Domain 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>
```

4.6.18 Template "Person"

The entity `Person` represents an individual human being.

The `Id` attriute 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 Domain Model XML entities and attributes

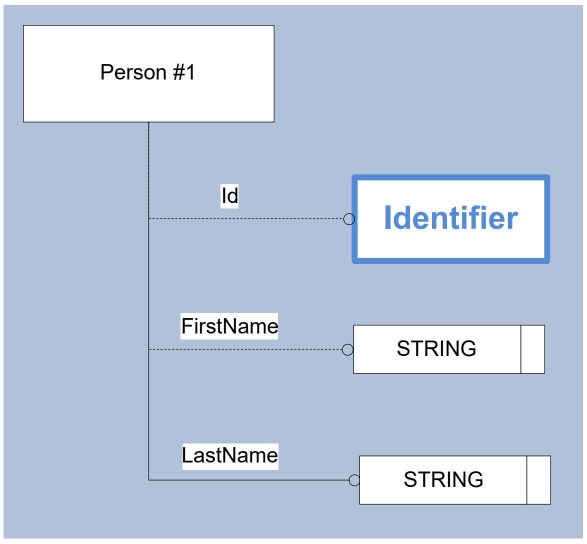


Figure 18: Template "Person"

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
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
SameAs	OPTIONAL SET[1:?] of Proxy
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
OrganizationOrPersonInOrganiza-tionAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganiza-tionAssignment
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 21: "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. Use String type.
- 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. Use String 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: None specified.

Postprocessor Recommendations: None specified.

Related Entities: There are no specific related entities.

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

```
<Person uid="pers--Mustermann">
  <FirstName>Max</FirstName>
  <Id>
    <Identifier uid="pers--Mustermann--1" id="4711" idRoleRef="rl--ii" idCon-
textRef="o--000000178"/>
  </Id>
  <LastName>Mustermann</LastName>
</Person>
```

4.6.19 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 whith 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 Domain Model XML entities and attributes

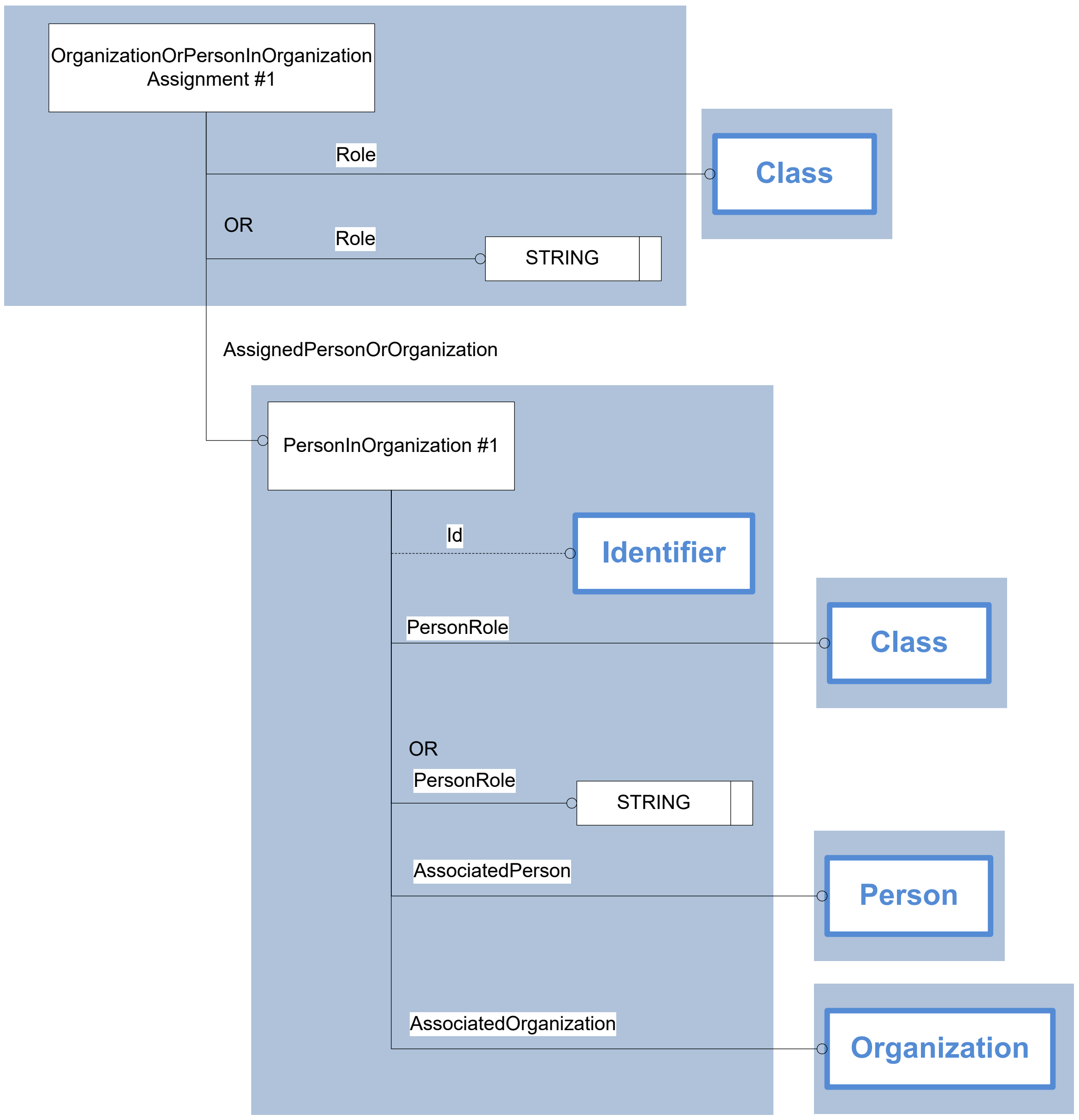


Figure 19: Template “PersonInOrganization”

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

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

Table 22: "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: Except in case of customized PDM properties (see section 14.3), when applicable, the following values shall be used for the attribute Role:

Role	**Explanation**
‘assignee’	The assigned person or organization to complete, review, comment or update the referenced object
'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

Role	Explanation
'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 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

Table 23: Recommended Values of "OrganizationOrPersonInOrganizationAssignment.role"

Preprocessor Recommendations:

- The roles ‘owner’ and ‘id owner’ shall not be used redundantly to the `idContextRef` of the Identifier of the current object (“Identifier” template (see 4.6.6)), except if the Organization-OrPersonInOrganizationAssignment refers to a given Person or Department.

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	OPTIONAL IdentifierSelect

ENTITY PersonInOrganization	**Attribute Type**
PersonRole	ClassSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
SameAs	OPTIONAL SET[1:?] of Proxy
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
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
StateAssignment	OPTIONAL SET[1:?] of StateAssignment
StateDefinitionAssignment	OPTIONAL SET[1:?] of StateDefinitionAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssignment

Table 24: "PersonInOrganization" Attributes

Attribute recommendations

- The ***AssociatedPerson*** attribute defines the person. Use “Person” template (see 4.6.18).
- The ***AssociatedOrganization*** attribute defines the organization. Use “Organization” template (see 4.6.2).
- The ***Id*** attribute is the identification of the entity. The value of this attribute need not be specified. 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 Recommendation: At least one of Person.Id or PersonInOrganization.Id shall be set.

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 Domain 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"
idRoleRef="rl--ii" idContextRef="o--000000178"/>
  </Id>
  <PersonRole>
    <ClassString>employee</ClassString>
  </PersonRole>
</PersonInOrganization>
```

4.6.20 Template "DateAndPersonOrganization"

A DateAndPersonOrganization is a PersonInOrganization or an Organization associated with a DateTimeString.

The Instance Model: AP242 Domain Model XML entities and attributes

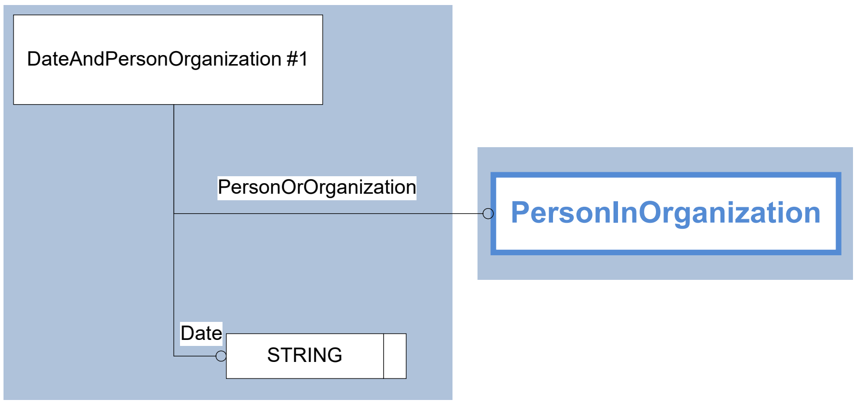


Figure 20: Template "DateAndPersonOrganization"

ENTITY DateAndPersonOrganization	Attribute Type
Date	DateTimeString
PersonOrOrganization	OrganizationOrPersonInOrganizationSelect

Table 25: "DateAndPersonOrganization" Attributes

Attribute recommendations

- **Date**: specifies the date and an optional time of day. Use “DateTime” template (see 4.6.11).
- **PersonOrOrganization**: specifies the Organization or the PersonInOrganization. Use template “PersonInOrganization” (see 4.6.19).

Preprocessor Recommendations:

Except if the data model requires it (like for WorkRequest.Requestor), rather use separated DateTime and PersonInOrganization or Organization (e.g., in DateTimeAssignment and OrganizationOrPersonInOrganizationAssignment rather than DateAndPersonAssignment), since most PDM systems do not associate DateTime and PersonInOrganization in one attribute/object.

Postprocessor Recommendations: None specified.

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

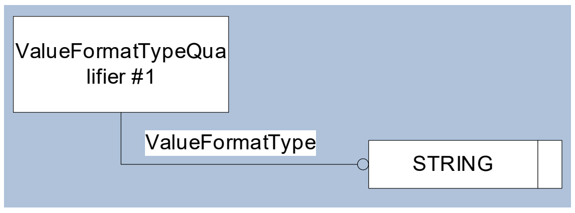
```
<WorkRequest uid="wr--1">
…
      <Requestor uid="dpo--1" xsi:type="n0:DateAndPersonOrganization">
            <Date>2019-10-16T09:08:07</Date>
            <PersonOrOrganization uidRef="pio--005-TPEVD-Mustermann"/>
      </Requestor>
…
</WorkRequest>
…

<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" idRoleRef="rl--ii" idContextRef="o--000000178"/>
      </Id>
      <PersonRole>
            <ClassString>employee</ClassString>
      </PersonRole>
</PersonInOrganization>
…
<Organization uid="org--005-TPEVD">
      <Id id="005-TP/EVD"/>
      <Name>
            <CharacterString>005-TP/EVD</CharacterString>
      </Name>
      <OrganizationTypes>
            <ClassString>department</ClassString>
      </OrganizationTypes>
</Organization>
…
```

```
<Person uid="pers--Mustermann">
      <LastName>Mustermann</LastName>
</Person>
```

4.6.21 Template "ValueFormatTypeQualifier"

Introduced with AP242 Domain Model Edition 4, a ValueFormatTypeQualifier is the specification of the allowed format of a property value). Also the presentation aspects of the format may be mapped (for example In the future PMI area of Edition 4).

The Instance Model: AP242 Domain Model XML entities and attributes



.

Figure 21: Template "ValueFormatTypeQualifier"

ENTITY ValueFormatTypeQualifier	Attribute Type
ValueFormatType	ProxyItemSelect

Table 26: "ValueFormatTypeQualifier" Attributes

Attribute recommendations

- **ValueFormatType**: specifies the format and the type of the string or numerical value. According to ISO 13584-42 section 7.2.17:
  - the following format type definitions are recommended:

Format type	Explanation
A	alphabethic characters, letters only
M	mixed, all characters allowed
N	(non-quantitative) numeric characters, digits only
X	(non-quantitative) alphanumeric characters, letters or digits only
B	boolean
NR1	integer
NR2	real number with decimal-mark
NR3	real number with decimal-mark and floating point

  - the following format definitions are recommended:
    For NR1, NR2:
    - in case presentation aspects are relevant: <Format type> [S] [..]|[ ] [x.y]
    - otherwise: <Format type> x.y
    For NR3:
    - in case presentation aspects are relevant: <Format type> [S] [..]|[ ] [x.y] [E [S] [z]

- otherwise: <Format type> x.y [E]
For A, M, N, X:
- in case presentation aspects are relevant: <Format type> [..] a
- otherwise: <Format type> a
For B, none of the format parameters apply

All format parameters given in [] are optional. | means 'or'.

Format parameter	Explanation	Format example	Value example
S	Signed (only in case of NR2 or NR3)	NR2 NR2S	0.5625 -0.5625 +0.5625 -0.5625
x.y	number of digits before.after the decimal point (only in case of NR2 or NR3)	NR2 1.3	0.562
a	string field length (longer string values might be truncated) (only for A, M, N, X)		
..	variable field length	A..35 NR2.. NR2..3.3	string example 0.5625 .562
E	exponent mark, base 10 (only in case of NR3)	NR3 1.3E2	5.625E-01 (in case of 0,5625)
z	Number of digits for the exponent (only in case of NR3)	NR3 1.3E2 NR3 1.3ES2	5.625E01 (in case of 56.25) 5.625E+01 (in case of 56.25)

For example in case of PDM properties, optionally, some more parameters are recommended (beyond the scope of ISO 13584-42). All of them may be freely combined, blank separated:
<Format type> <Format Parameters> [O]|[M] [U] [P{p}] [D{d}]

Additional parameter	Explanation	Format example	Value example
O/M	Property is Optional or Mandatory	A 35 M	may not be unset
U	Property value shall be unique over all the objects having the same idContextRef to which the property is assigned to	A35 U	
P{p}	Property value shall comply the pattern p in order to restrict a string to a particular regular expression (accord-	P{^(\\([0-9]{3}\\))?[0-9]{3}-[0-9]{4}$}	555-1212 (888)555-1212

	ing to the syntax of xs:pattern, see for example under http://www.java2s.com/Tutorial/XML/0060__XML-Schema/Patternsyntax.htm)	P{male|female} (value domain)	male female
D{d}	Default property value if the property if not set	D{+49} (german phone number country code)	+49 (if the property value is not set)

Preprocessor Recommendations:

ValueFormatTypeQualifier is only supported on PropertyValues, not on customized PDM properties of kind DateTimeAssignment, ApprovalAssignment or OrganizationOrPersonInOrganizationAssignment.

A Workaround could be to map a PropertyValueAssignment to DateTimeAssignment, ApprovalAssignment or OrganizationOrPersonInOrganizationAssignment with a dummy StringProperty having:

- Definition(PropertyDefinitionString)='dummy property value for ValueFormatTypeQualifier of the Date/Approval/Person/Organization'
- Name='/NULL'
- ValueComponent='/NULL'

Postprocessor Recommendations:

T.B.S.

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

```
<ValueFormatTypeQualifier uid="_2165">
  <ValueFormatType>
    <ProxyString>NR2S 3.2</ProxyString>
  </ValueFormatType>
</ValueFormatTypeQualifier>

<Part uid="ID535">
…
  <Versions>
    <PartVersion uid="ID730">
…
      <Views>
        <PartView xsi:type="n0:AssemblyDefinition" uid="ID731">
…
          <PropertyValueAssignment uid="ID733">
            <AssignedPropertyValues>
              <PropertyValue uid="ID734" xsi:type="n0:NumericalValue">
                <Definition>
                  <PropertyDefinition uidRef="ID9"/>
                </Definition>
                <Name>
                  <CharacterString>calculated density</CharacterString>
                </Name>
                <Qualifiers>
                  <ValueFormatTypeQualifier uidRef="_2165"/>
                </Qualifiers>
                <Unit uidRef="ID5"/>
                <ValueComponent></ValueComponent>
              </PropertyValue>
            </AssignedPropertyValues>
…
          </PropertyValueAssignment>
…
        </PartView>
      </Views>
…
    </PartVersion>
  </Versions>
…
</Part>
```

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

Identification of Parts in the AP242 Domain 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 Part classification the AP242 Domain 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 section 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 section 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 Domain Model of the following structurally distinct data types as also shown in Figure 22 and Figure 23:

- 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, that is, below PartVersion, as shown in Figure 22 and Figure 23 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 22 and Figure 23 are root elements and are, thus, outside of the containment blocks.

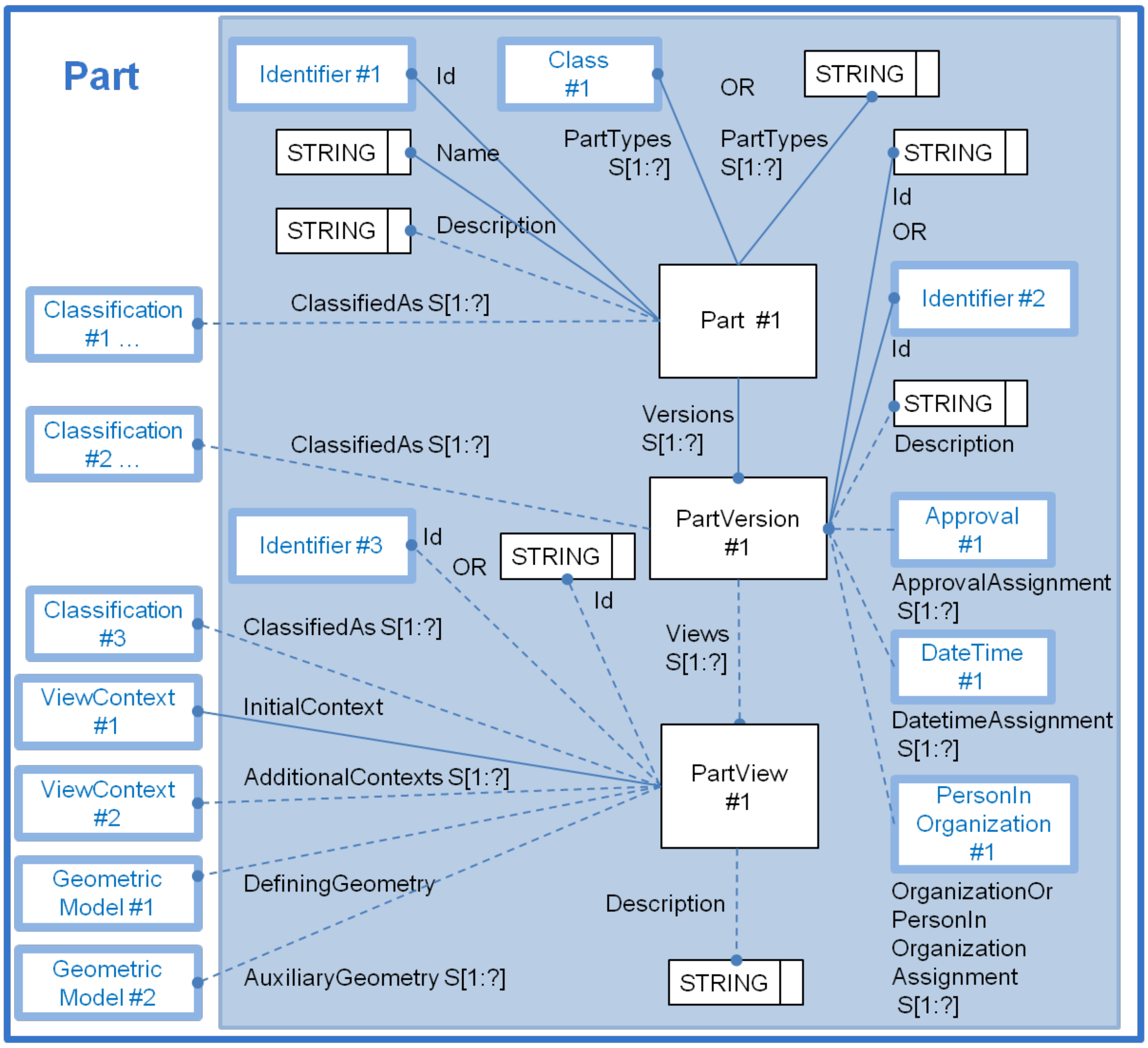


Figure 22: Template "Part"

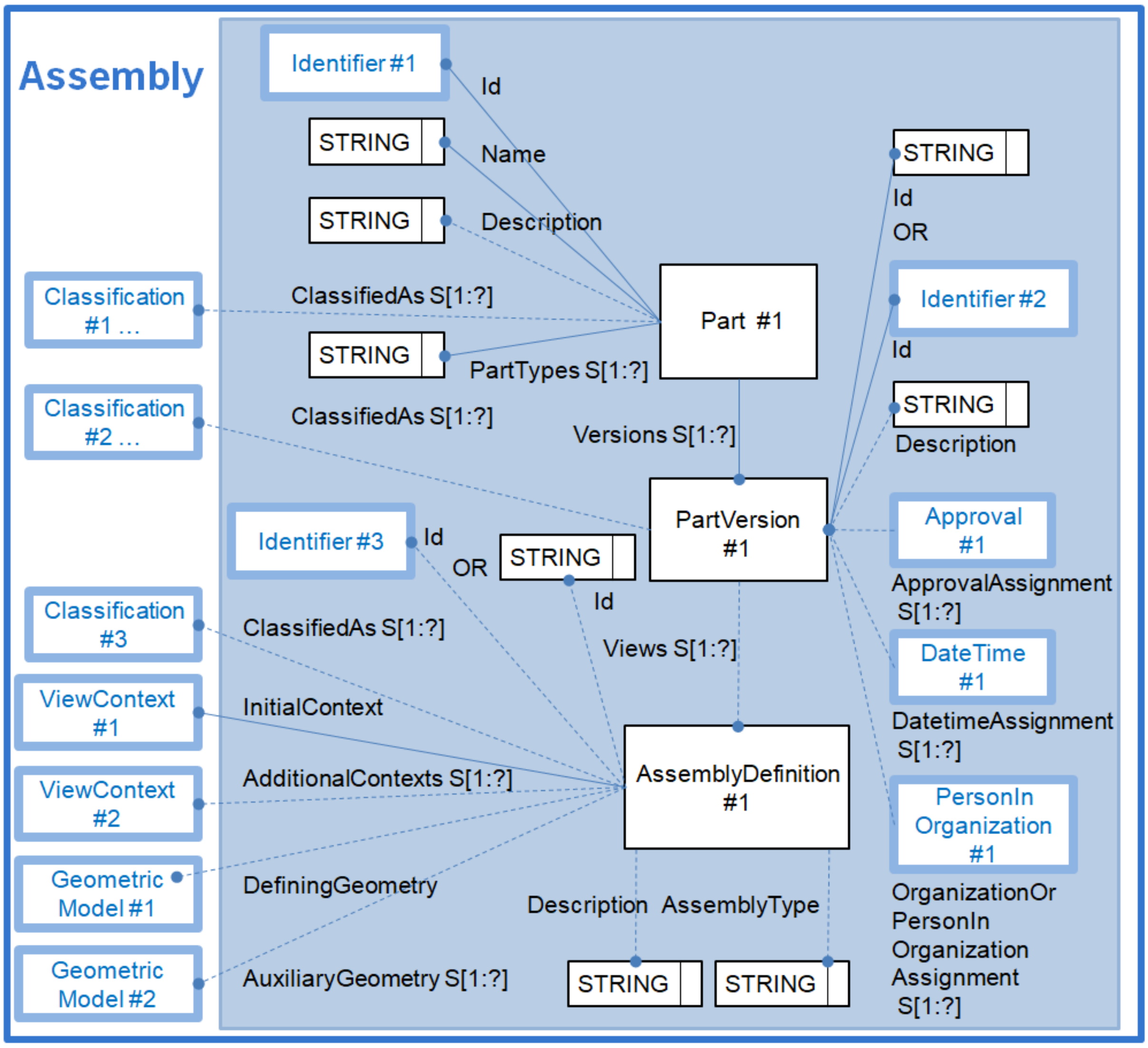


Figure 23: Template “Assembly”

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 22.

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

```
<Part uid="p--0000000017086CB0">
  <Id>
    <Identifier uid="pid--0000000017086CB0--id1" id="as1" idRoleRef="rl--ii"
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" idRoleRef="rl--
ii" 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 PartClassSelect

Entity Part attributes	Attribute type
SameAs	OPTIONAL SET[1:?] of Proxy
Versions	SET[1:?] of PartVersion
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
AnalysisAssignment	OPTIONAL SET[1:?] of AnalysisAssignment
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
StateAssignment	OPTIONAL SET[1:?] of StateAssignment
StateDefinitionAssignment	OPTIONAL SET[1:?] of StateDefinitionAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssignment

Table 27: "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. PartTypes is a set of one or many strings. As defined in the ISO AP242 specification, use ClassString if the value of this element is 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

  otherwise use PartCategoryEnum as far as possible, otherwise use Class
- **Versions**: the related variants of the Part; a Part shall have at least one PartVersion.
- **AlternatePartRelationship:** to assign (optionally) an alternate part to this part. For more details, refer to 7.5.1.
- **PropertyValueAssignment**: to assign a PropertyValue to the Part. 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.

***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).

- Although some values already recommended for PartTypes are defined in PartCategoryEnum, for upward compatibility reasons, their mapping as ClassString is recommended.

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.
- For upward compatibility reasons: PartCategoryEnum should be also supported.

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.

In order to represent the 'make or buy' information as in STEP AP214 AIM, the usage of ClassifiedAs attribute is recommended (see ISO Jira Task TCSC410303-909).

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 PartVersionandonly 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.

Examples of PartVersion instantiations are in the XML-snippet in section 4.1.2.

Entity PartVersion attributes	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
SameAs	OPTIONAL SET[1:?] of Proxy
Views	OPTIONAL SET[1:?] of PartView
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
AnalysisAssignment	OPTIONAL SET[1:?] of AnalysisAssignment
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

Entity PartVersion attributes	Attribute type
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 InformationUs-ageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssign-ment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPer-sonInOrganizationAssignment
PartVersionRelationship	OPTIONAL SET[1:?] of PartVersionRelationship
ProjectAssignment	OPTIONAL SET[1:?] of ProjectAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefini-tionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssign-ment
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassifica-tionAssignment
StateAssignment	OPTIONAL SET[1:?] of StateAssignment
StateDefinitionAssignment	OPTIONAL SET[1:?] of StateDefinitionAssign-ment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelation-ship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssign-ment

Table 28: "PartVersion" Attributes

Attribute recommendations

- **ActivityAssignment**: the Activities associated to the PartVersion. The value of this attribute need not be specified. Use "Activity" template (see Recommended Practices for AP242 Domain Model XML Change Management for details).
- **ApprovalAssignment**: the level of acceptance of the PartVersion. The value of this attribute need not be specified. Use "Approval" template (see 4.6.17).
- **ClassifiedAs**: the classifications of the PartVersion. The value of this attribute need not be specified except for the ‘make or buy’ information. 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).

- ***EffectivityAssignment***: to assign (optionally) one or multiple Effectivities to the usage of the PartVersion. Use the “EffectivityAssignment” template (see Recommended Practices for AP242 Domain Model Configuration Management for details).
- **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 4.6.19).
- **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 lifecycle. 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 Domain 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 4.1.2

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

Entity PartView attributes	Attribute type
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssign-ment
AnalysisAssignment	OPTIONAL SET[1:?] of AnalysisAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
BreakdownVersionAssignment	OPTIONAL SET[1:?] of BreakdownVer-sionAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssign-ment
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 InformationUs-ageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssign-ment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPer-sonInOrganizationAssignment
PartViewRelationship	OPTIONAL SET[1:?] of PartViewRelationship
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefini-tionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssign-ment
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassifica-tionAssignment
StateAssignment	OPTIONAL SET[1:?] of StateAssignment
StateDefinitionAssignment	OPTIONAL SET[1:?] of StateDefinitionAssign-ment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelation-ship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
ViewOccurrenceRelationship	OPTIONAL SET[1:?] of ViewOccurrenceRe-lationship
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssign-ment

Table 29: "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 except for nested files (see 9.3) and incremental exchange (see 9.4). Use “Classification” template (see 4.6.5).
- **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.
- **AuxiliaryGeometry:** the set of GeometricModels that provide additional shapes for the PartView. These additional shapes do not define the primary geometry of 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 Domain 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 the subtype “SingleOccurrence” (see template in 7.1) shall be used.
- **ShapeElement:** to assign one or many ShapeElement to the PartView. See [AP242-PMI] for details.
- **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

An example of an AssemblyDefinition instantiation is in the XML-snippet in section 7.1.

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
AuxiliaryGeometry	OPTIONAL SET[1:?] of 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 DefinitionBasedOccurrence
SameAs	OPTIONAL SET[1:?] of Proxy
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

Entity AssemblyDefinition attributes	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 InformationUs-ageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssign-ment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPer-sonInOrganizationAssignment
PartViewRelationship	OPTIONAL SET[1:?] of PartViewRelationship
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefini-tionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssign-ment
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassifica-tionAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelation-ship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
ViewOccurrenceRelationship	OPTIONAL SET[1:?] of ViewOccurrenceRe-lationship
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssign-ment
AssemblyType	OPTIONAL ClassSelect
KinematicMechanismAssociation	OPTIONAL SET OF KinematicMechanismAsso-ciation
MotionModelAssociation	OPTIONAL SET OF MotionModelAssociation

Table 30: "AssemblyDefinition" Attributes, including attributes inherited from “PartView”

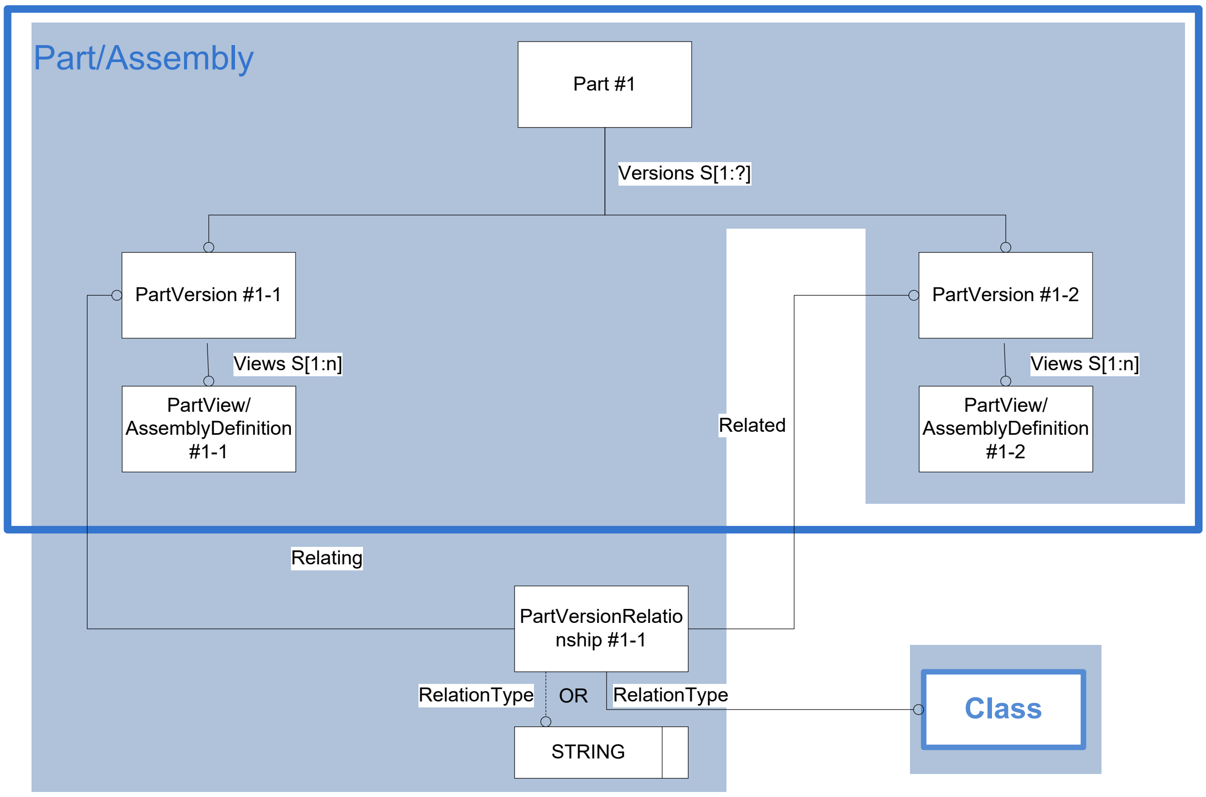
Attribute recommendations

- **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 “Single-Occurrence” 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 Domain Model XML Kinematics for details).
- **MotionModelAssociation**: to assign (optionally) one or multiple Kinematic Motions to the AssemblyDefinition. Use the "LinkMotionAlongPath" template (see Recommended Practices for AP242 Domain Model XML Kinematics for details).

5.1.5 PartVersionRelationship

Used to relate several versions of the same part:



or of different parts.

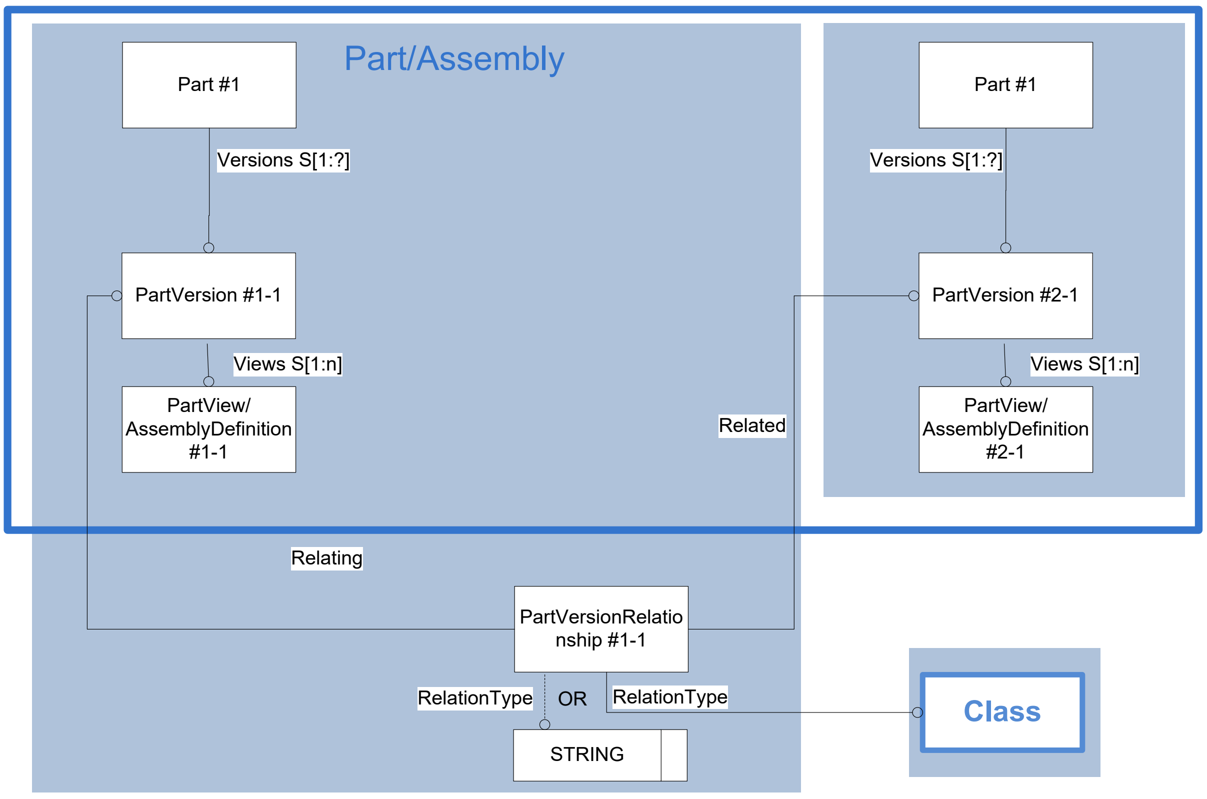


Figure 24: Template "PartVersionRelationship"

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

```
<Part uid="p--0000000017D374A0">
  <Id>
    <Identifier uid="pid--0000000017D374A0--id1" id="as1" idRoleRef="rl--ii"
idContextRef="o--000000178"/>
  </Id>
  <Name>
    <CharacterString>as1</CharacterString>
  </Name>
…
  <Versions>
    <PartVersion uid="pv--0000000017D374A0--id1">
…
      <Id id="A.1"/>
…
      <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	OPTIONAL IdentifierSelect
Related	PartVersion
RelationType	ClassSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssign-ment
AnalysisAssignment	OPTIONAL SET[1:?] of AnalysisAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
CertificationAssignment	OPTIONAL SET[1:?] of CertificationAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssign-ment
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 InformationUs-ageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssign-ment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPer-sonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefini-tionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssign-ment
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 31: "PartVersionRelationship" Attributes

Attribute recommendations

- **Id**: the identifier or set of identifiers for the PartVersionRelationship. The value of this attribute need not be specified. Use IdentifierString type or “Identifier” template (see 4.6.6).

- ***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'. The Part associated to the Relating/Related PartVersions shall be the same.
'supplied item'	The business object defines a relationship between two PartVersion objects 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
'alternative'	The business object defines a relationship where the related PartVersion is an alternative to the relating PartVersion (not symmetric) ISO Jira Task TCSC410303-1308 has been created to add a new subtype of PartVersionRelationship called AlternatePartVersionRelationship (similar to AlternatePartRelationship) or to define 'alternative part version relationship' as RelationType
'mirrored'	The business object defines a relationship where the relating PartVersion is mirrored from the related (original) PartVersion (see section 7.7)
'alternate shape'	The business object defines a relationship where the relating PartVersion defines an Alternative Instance Shape of the related PartVersion (for example in case of Flexible Part, see section 7.8)

- ***Related***: the other object of PartVersion that is part of the relationship
- **PropertyValueAssignment**: to assign a PropertyValue to the PartVersionRelationship. 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.

Preprocessor Recommendations:

- The relating and the related PartVersions shall be different objects
- In case of RelationType ‘alternative’, ‘mirrored’ or ‘alternate shape’, the relating and the related PartVersions shall belong to different Parts, otherwise they shall belong to the same Part.
- In case of RelationType ‘alternative’, if the relation is symmetric in the source
- ~~In case of RelationType ‘alternative’, Either the Id of all the PartVersionRelationships having the same Relating PartVersion shall have all the same value (or all being unset), or they shall have a unique value.~~
- PDM system, then a reverse-relationship shall state that it applies in both directions.
- If there are more than one alternate part involved, a ‘star’ structure from the base PartVersion to each alternate PartVersion shall be mapped (this star maps the semantic, that there is a base PartVersion and a number of alternate PartVersions. All the alternate PartVersions are on the same semantical level, but not on the same semantical level than the base PartVersion.
- Alternate PartVersions shall have no revision effectivities and their usages shall have no occurrence effectivities, since the ones defined on the base part also apply to the alternate PartVersions.
- The combination of the base PartVersion and the alternate PartVersion shall be unique.

Postprocessor Recommendations:

- If the target system does not support alternate PartVersionRelationship, but only AlternatePartRelationship (see section 7.5.1), it shall check if the same alternate rule applies to all versions of the alternate part (mentioned in the XML file) for all versions of the base part (mentioned in the XML file) and map them as one instance of AlternatePartRelationship. Otherwise, an error shall be returned and the PartVersionRelationships shall be ignored.

5.1.6 Distinction between Identifiers given by OEM and Supplier

The same Parts and Assemblies may be assigned different identifiers by different and collaborating organizations, for example, by OEM and by suppliers. To exchange more than one identifier many Identifiers shall be used. In the context of Part and Assembly this applies specifically to Part.Id, PartVersion.Id, PartView.Id (respectively AssemblyDefinition.id if set) and optionally Occurrence.Id.

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

```
<Class uid="rl--ii">
  <Id id="identification information"/>
</Class>
<Class uid="rl--eii">
  <Id id="exchange identification information"/>
</Class>
<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" idRoleRef="rl--
eii" idContextRef="o--000000178"/>
    <Identifier uid="pid--000000001E60C660--id7" id="BO-bolt" idRoleRef="rl--
ii" idContextRef="o--000000179"/>
  </Id>
…
  <Versions>
    <PartVersion uid="pv--000000001E60C660">
...
      <Id>
            <Identifier uid="pid--000000001EAA8110--id6" id="A.1" idRol-
eRef="rl--eii" idContextRef="pid--000000001E60C660--id6"/>
            <Identifier uid="pid--000000001EAA8110--id7" id="001,3" idRol-
eRef="rl--ii" idContextRef="pid--000000001E60C660--id7"/>
      </Id>
...
    </PartVersion>
...
  </Versions>
...
</Part>
```

Collaborating partners may mutually agree how many and which identifiers they want to exchange. The case that one organization assigns several identifiers to the same instance is covered by recommendations for template “Identifier” in chapter 4.6.6.

To avoid being obliged to exchange the full id list for all idContextRefs between the OEM, the tier-1 and the tier-n suppliers, the ids having the role ‘exchange identification information’ shall be exchanged (the others become optional) => it is sufficient that each exchange partner stores the mapping between the exchange id(s) and his own internal id. For a given Identifier, there shall be one or many ids with this value in idRoleRef (over all idContextRef values).

If a Part or Assembly is assigned identifiers by different organizations, ownership of the Part or Assembly shall be documented using template “OrganizationOrPersonInOrganizationAssignment”; see chapter 4.6.19. OrganizationOrPersonInOrganizationAssignment.role shall be given the value “owner” from Table 23. Thus, by matching the Organization in Identifier.idContextRef with the OrganizationOrPersonInOrganizationAssignment.assignedPersonOrOrganization.associatedOrganization that plays the role of "owner", the original identifier of a Part or Assembly can be uniquely distinct from other identifiers.

The same applies to Document/DocumentVersion/DocumentDefinition and similarly identified objects.

```
<Class uid="rl--ii">
  <Id id="identification information"/>
</Class>
<Class uid="rl--eii">
  <Id id="exchange identification information"/>
</Class>

<Organization uid="o--000000178">
  <Id id="mercedes-benz.com"/>
  <Name>
    <CharacterString>Mercedes-Benz</CharacterString>
  </Name>
  <OrganizationTypes>
    <ClassString>company</ClassString>
  </OrganizationTypes>
</Organization>
<PersonInOrganization uid="pio--005-TPEVD-Mustermann">
  <AssociatedOrganization uidRef="o--000000178"/>
  <AssociatedPerson uidRef="pers--Mustermann"/>
  <Id>
    <Identifier uid="pio--005-TPEVD-Mustermann--1" id="005-TP/EVD-Mustermann"
idRoleRef="rl--ii" idContextRef="o--000000178"/>
  </Id>
  <PersonRole>
    <ClassString>employee</ClassString>
  </PersonRole>
</PersonInOrganization>

<Part uid="p--000000001E60C660">
  <Id>
    <Identifier uid="pid--000000001E60C660--id6" id="bolt" idRoleRef="rl--
eii" idContextRef="o--000000178"/>
    <Identifier uid="pid--000000001E60C660--id7" id="BO-bolt" idRoleRef="rl--
ii" idContextRef="o--000000179"/>
  </Id>
…
  <Versions>
    <PartVersion uid="pv--000000001E60C660">
...
      <Id>
            <Identifier uid="pid--000000001EAA8110--id6" id="A.1" idRol-
eRef="rl--eii" idContextRef="pid--000000001E60C660--id6"/>
            <Identifier uid="pid--000000001EAA8110--id7" id="001,3" idRol-
eRef="rl--ii" idContextRef="pid--000000001E60C660--id7"/>
      </Id>
...
      <OrganizationOrPersonInOrganizationAssignment uid="poa--
0000000017D374A0--id1">
        <AssignedPersonOrOrganization uidRef="pio--005-TPEVD-Mustermann"/>
        <Role>
          <ClassString>owner</ClassString>
        </Role>
      </OrganizationOrPersonInOrganizationAssignment>
...
    </PartVersion>
...
  </Versions>
...
</Part>
```

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.

Even if GeometricModel is not an ABSTRACT supertype, it is recommended to always use one of its subtypes. The different usage of the subtypes are detailed in the chapter 6.1.1.

The Instance Model: AP242 Domain Model XML entities and attributes

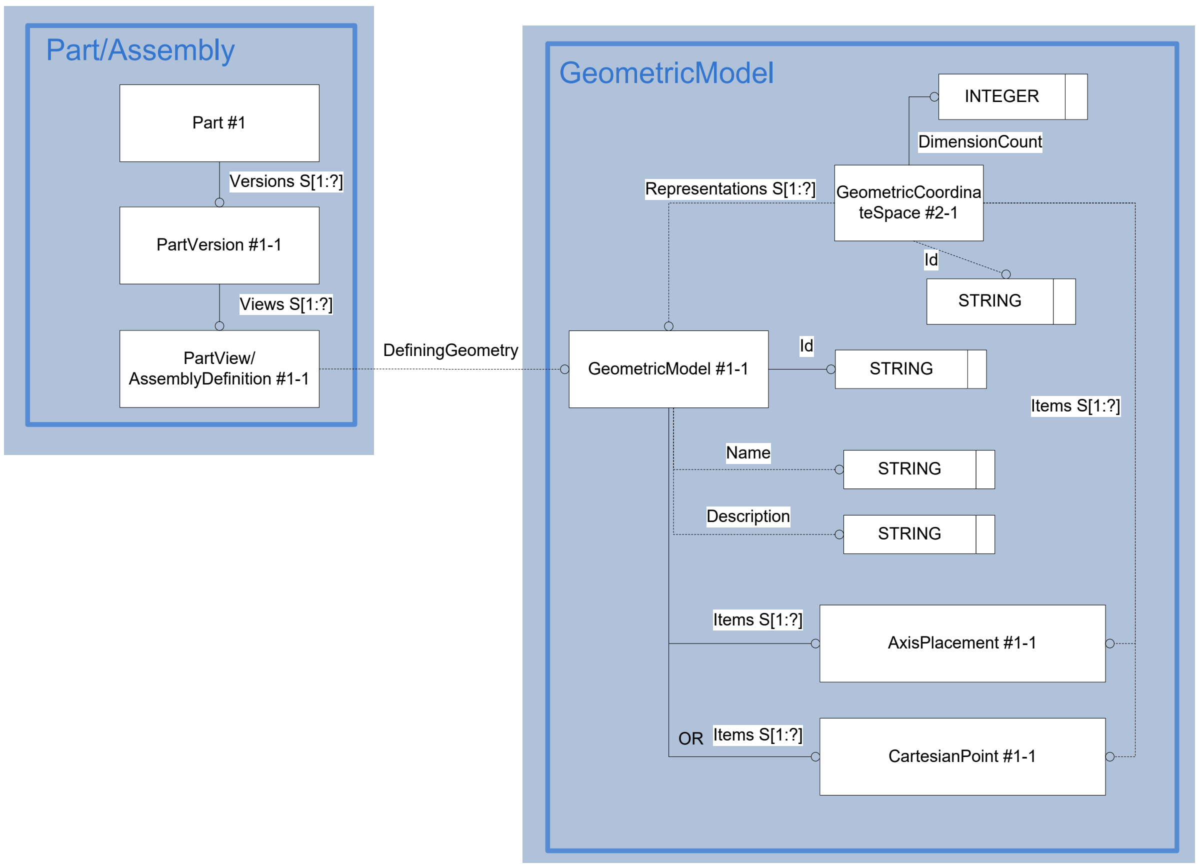


Figure 25: Template "GeometricModel"

ENTITY GeometricModel	**Attribute Type**
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
Items	SET[1:?] of RepresentationItem
Name	OPTIONAL DescriptorSelect

ENTITY GeometricModel	**Attribute Type**
RepresentationTypes	OPTIONAL SET[1:?] of ClassSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
VersionId	OPTIONAL IdentifierSelect
RepresentationRelationship	OPTIONAL SET[1:?] of RepresentationRelationship
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
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
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
ModelExtent	OPTIONAL REAL
ModelProperty	OPTIONAL SET[1:?] of ModelProperty

Table 32: "GeometricModel" Attributes

Attribute recommendations

- ***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.
- The ***Items*** attribute is the SET of elements representing the different kind of representation item attached to a shape. Reference to AxisPlacement (see below), CartesianPoint (see section 13.1.2 for examples), KinematicPair element or subtypes of DetailedGeometricModelItem (AdvancedFace, Edge, Curve, Point, …) involved in PMIs.
- ***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.

Preprocessor Recommendations:

- In the future PMI area of Edition 4: for a given part, all the RepresentationItems referenced by ShapeElement.RepresentedGeometry shall be included into GeometricModel.Items
  and this GeometricModel shall be directly or indirectly (via GeometricRepresentationRelationship) be referenced by PartView.DefiningGeometry (see section 5.1.3).

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

Table 33: "GeometricCoordinateSpace" Attributes

Attribute recommendations

- 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.
- The ***Representations*** attribute is the SET of elements representing the different kind of representations defined in the Coordinate Space. Reference to GeometricModel (see 6.1), KinematicLink or Mechanism element.
- The ***Items*** attribute is the SET of elements representing the different kind of representation items in the Coordinate Space. Reference to AxisPlacement (see below), CartesianPoint (see section 13.1.2 for examples), KinematicPair or any further RepresentationItems.
- The ***DimensionCount*** attribute specifies the dimensionality.
- 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” does not apply in general (see below).

Preprocessor Recommendations:

- In case of relative positioning, each Geometry shall be associated to its own GeometricCoordinateSpace.
- The value of GeometricCoordinateSpace.Items shall be the sum of:

- all the Items referenced directly by GeometricCoordinateSpace.Representations (like GeometricModels).Items
- and those referenced indirectly. For example:
  - in the Kinematic area via KinematicPathDefinedByNodes.Segments, …
  - in the Composite area via Polar.Placement, Curve11.GuideCurves, GeometricSet.Elements, …
  - or in the future PMI area of Edition 4 via CameraModel3d.ViewWindow, Callout.Contents, …)

- The RepresentationItems referenced by ShapeElement.RepresentedGeometry for one ExternalGeometricModel may be reused in other ExternalGeometricModels within the same GeometricCoordinateSpace, but not within another GeometricCoordinateSpace. One use case is for GeometricModels that are defined as alternatives to describe one part (DefiningGeometry, AuxiliaryGeometry), for example Brep and tessellated, in the case the RepresentationItem has the same identifier in both GeometricModels.
- The length unit used in the positioning (implicit/explicit transformation) and in the ExternalGeometricModel shall be the same.
- The use of GeometricCoordinateSpace.Units is necessary only in either of the following cases:
  - no default length unit is given in the ExchangeContext
  - the length unit of a particular assembly node deviates from the default length unit (so-called ‘mixed-unit assembly’)
  - angles are defined (for example for Lower/UpperLimits of KinematicPairs).
- If elements with different length or angle units are required, they each have to have their own GeometricCoordinateSpace
- The GeometricCoordinateSpace.DimensionCount must be greater than 0.

Postprocessor Recommendations:

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

ENTITY AxisPlacement	**Attribute Type**
External	OPTIONAL ExternalItem
Name	OPTIONAL DescriptorSelect
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
Axis	OPTIONAL String
Position	OPTIONAL String
RefDirection	OPTIONAL String

Table 34: "AxisPlacement" Attributes

Attribute recommendations

- **External**: the reference to an ExternalItem, i.e. indirectly to a DigitalFile where the RepresentationItem can be found). It's recommended to set this attribute only if the AxisPlacement is defined in an external geometric model via PartView.ShapeElement (see 5.1.3) but for example not if only used within a GeometricRepresentationRelationship to position an ExternalGeometricModel within an assembly. Use template "DigitalFile" (see 9.1).
- ***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. An issue has been created in ISO Jira Task TCSC410303-1307 in order to add this statement (coming from AP214): If Axis is not provided, then Axis shall be assumed to have the value (0, 0, 1).
- ***Position:*** the absolute x, y and z value (in the GeometricCoordinateSpace they are defined in) specifying the origin position of the AxisPlacement.

  Although changed to optional in Edition 4, this attribute shall be specified..

  Remark: As stated in the ISO 10303-4442 documentation, the coordinates in each AxisPlacement are absolute in the GeometricCoordinateSpace there are defined in. The combination of two AxisPlacements in a GeometricRepresen-tationrelationshipWithPlacementTransformation provides a relative placement.

  An issue has been created in ISO Jira Task TCSC410303-1307 in order to add this statement (coming from AP214):
  Remark: If both Axis and RefDirection are not provided, the AxisPlacement describes a pure translation without rotation.
- ***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. An issue has been created in ISO Jira Task TCSC410303-1307 in order to add this statement (coming from AP214): If RefDirection is not provided then RefDirection shall be assumed to have the value (1, 0, 0)
- 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 a RefDirection is given, it shall be specified so that it is orthogonal to the Axis.

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 35: "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 Domain Model XML syntax)

```
…
  <PartView uid="pvv--000000001AA415B0--id6">
    <DefiningGeometry uidRef="egm--000000001AA415B0"/>
    …
  </PartView>
…
  <RepresentationContext uid="ccs--origin-bolt" xsi:type="n0:GeometricCoordi-
nateSpace">
    <Id id="/NULL"/>
    <Representations>
      <Representation uid="egm--000000001AA415B0" xsi:type="n0:ExternalGe-
ometricModel">
        <Id id="bolt.stp"/>
        <Items>
          <RepresentationItem uidRef="repi--000000001AA415B0--18"/>
        </Items>
        <ExternalFile uidRef="df--000000001AA415B0"/>
      </Representation>
    </Representations>
    <Items>
      <RepresentationItem xsi:type="n0:AxisPlacement" uid="repi--
000000001AA415B0--18">
        <Position>0.0 0.0 0.0</Position>
      </RepresentationItem>
    </Items>
    <DimensionCount>3</DimensionCount>
  </RepresentationContext>
```

6.1.1 ExternalGeometricModel / ComposedGeometricModel

The ExternalGeometricModel / ComposedGeometricModel entities are subtype of the GeometricModel.

Entity ComposedGeometricModel / ExternalGeometricModel (additionally to GeometricModel)	Attribute type
Items (in case of ComposedGeometricModel)	SET[1:?] of AxisPlacementOrTransformationSelect
Items (in case of ExternalGeometricModel)	SET[1:?] of DetailedGeometricModelItem
RepresentationRelationship	OPTIONAL SET[1:?] of RepresentationRelationship
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ExternalFile (only for ExternalGeometricModel)	DigitalFile
ParameterValues (only for ExternalGeometricModel)	OPTIONAL SET[1:?] OF PropertyValue

Table 36: "ComposedGeometricModel" / "ExternalGeometricModel" Attributes

Attribute recommendations

- ***RepresentationRelationship:*** add for each element stored in 'Items' an instance of RepresentationRelationship to the ComposedGeometricModel This attribute shall not be used in the case of an ExternalGeometricModel, but shall be used in the case of a ComposedGeometricModel
- ***ActivityAssignment***: the Activities associated to the GeometricModel. The value of this attribute need not be specified. Use "Activity" template (see Recommended Practices for AP242 Domain Model XML Change Management for details)).
- ***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 use of the attribute ExternalGeometricModel.ExternalFile is described in the chapter 9.3.

Preprocessor Recommendations:.All DetailedGeometricModelItem in .Items shall have a value in .External, except AxisPlacements that are defined within the stpx file, for example to position the geometry within an assembly.

Postprocessor Recommendations: None specified.

Related Entities: There are no specific related entities.

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

```
…
  <RepresentationContext uid="ccs--origin-bolt" xsi:type="n0:GeometricCoordi-
nateSpace">
    <Id id="/NULL"/>
    <Representations>
      <Representation uid="egm--000000001AA415B0"
xsi:type="n0:ExternalGeometricModel">
      <Id id="bolt.stp"/>
      <Items>
        <RepresentationItem uidRef="repi--000000001AA415B0--18"/>
      </Items>
…
    </Representation>
…
  </Representations>
…
</RepresentationContext>
```

6.1.2 The subtypes of ExternalGeometricModel

The Domain 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 de-fined. 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:

- Since these subtypes are all defined in the DomainModel capability CompositeStructuralShapeAndStructure, they shall be only used within this context. For non-composite parts, it is recommended to use only the supertype ExternalGeometricModel.

Postprocessor Recommendations:

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

6.2 Template "PropertyAssignnment"

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), "StringValue" templates (see 4.6.10), ValueRange (see 4.6.11), ValueWithTolerances (see 4.6.12), ValueList (see 4.6.13) or ValueSet (see 4.6.15).

Preprocessor Recommendations:

- It is recommended that all the part properties use the same PropertyValueAssignment.

The Instance Model: AP242 Domain Model XML entities and attributes

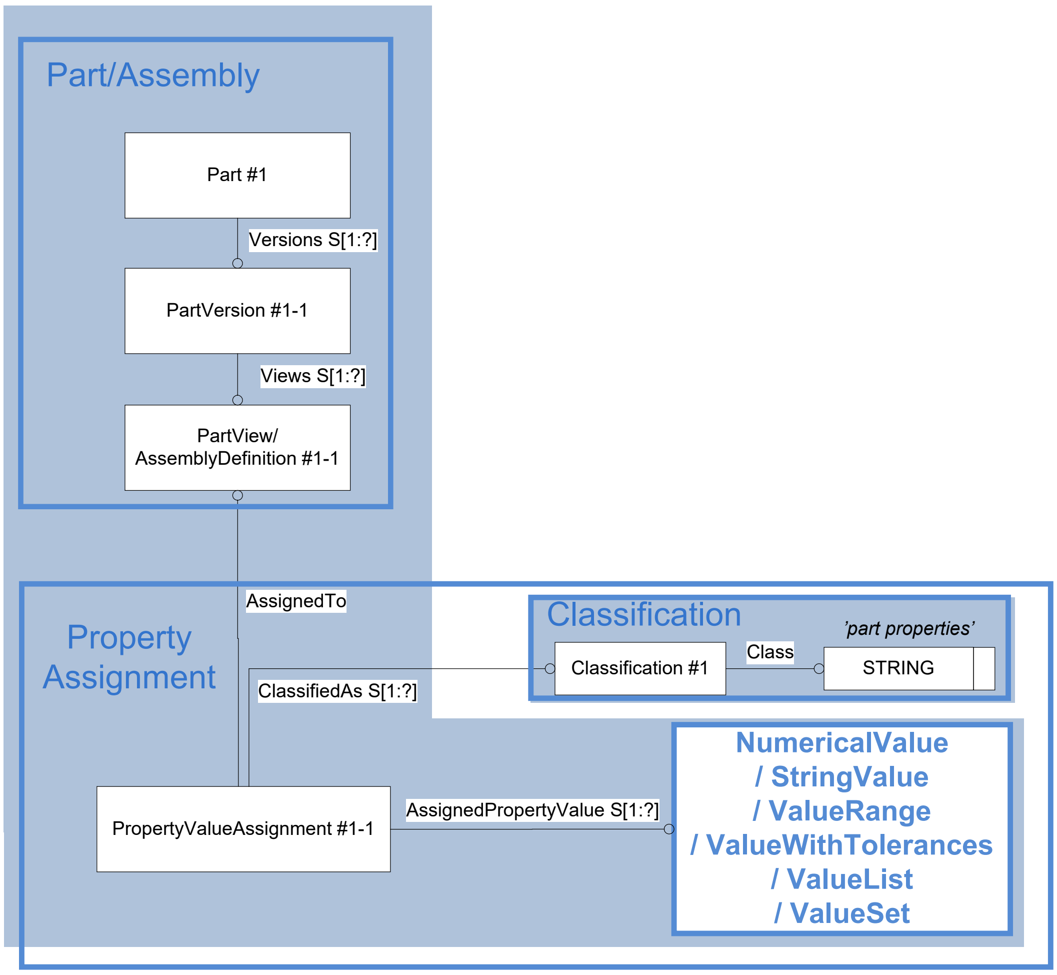

Part/Assembly
Part #1
Versions S[1:?]
PartVersion #1-1
Views S[1:?]
PartView/ AssemblyDefinition #1-1
AssignedTo
Property Assignment
Classification
'part properties'
Classification #1
Class
STRING
ClassifiedAs S[1:?]
NumericalValue / StringValue / ValueRange / ValueWithTolerances / ValueList / ValueSet
PropertyValueAssignment #1-1
AssignedPropertyValue S[1:?]

Figure 26: Template "PropertyAssignment"

ENTITY PropertyValueAssignment	**Attribute Type**
AssignedPropertyValues	SET[1:?] of PropertyValue
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
Role	OPTIONAL ClassSelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
AnalysisAssignment	OPTIONAL SET[1:?] of AnalysisAssignment
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 37: "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), “StringValue” templates (see 4.6.10), ValueRange (see 4.6.11), ValueWithTolerances (see 4.6.12), ValueList (see 4.6.13) or ValueSet (see 4.6.15).

- The **ClassifiedAs** attribute shall be represented by the template "Classification" (see 4.6.5). Use the following values for Classification.Class:

ClassString	Explanation
'validation properties'	the AssignedPropertyValues are ot type validation properties associated to a PartView according to section 13
'part master properties'	the AssignedPropertyValues are ot type part properties associated to a Part according to section 5.1.1
'part properties'	the AssignedPropertyValues are ot type part properties associated to a PartView, PartViewRelationship, NextAssemblyOccurrenceUsage, SpecifiedOccurrence, AlternatePartRelationship, PartVersionRelationship, AssemblyOccurrenceRelationshipSubstitution or AssemblyViewRelationshipSubstitution according to the current section or 5.1.3 (PartView), 7.4 (PartViewRelationship), 7.1 (NAOU), 7.2 (SpecifiedOccurrence), 7.5.1 (AlternatePartRelationship), 5.1.5 (PartVersionRelationship), 7.5.2 (AssemblyOccurrenceRelationshipSubstitution) or 7.5.3 (AssemblyViewRelationshipSubstitution)
'document master properties'	the AssignedPropertyValues are ot type document properties associated to a Document according to section 8.1.1.
'document properties'	the AssignedPropertyValues are ot type document properties associated to a DocumentDefinition or DigitalFile according to section 8.1.3 and 9.1
'work request properties'	the AssignedPropertyValues are ot type activity properties associated to a WorkRequest (see Recommended Practices for AP242 Domain Model XML Change Management for details)
'work order properties'	the AssignedPropertyValues are ot type activity properties associated to a WorkOrder according to section (see Recommended Practices for AP242 Domain Model XML Change Management for details)
'activity properties'	the AssignedPropertyValues are ot type activity properties associated to an Activity or ActivityAssignment according to section (see Recommended Practices for AP242 Domain Model XML Change Management 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.

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.
- Properties with a '' string, value 0 or List/Set having no elements shall be omitted if these values don't have a specific semantic. An example is the validation properties of kind 'number of …': the value 0 has a specific semantic, telling there are no instances of the counted elements.

Postprocessor Recommendations: None specified.

Related Entities: There are no specific related entities.

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

```
<Classification uid="ID_63">
  <Class>
    <ClassString>part properties</ClassString>
  </Class>
  <Role>kind of properties</Role>
</Classification>
<Classification uid="ID_217">
  <Class>
    <ClassString>validation properties</ClassString>
  </Class>
  <Role>kind of properties</Role>
</Classification>

<Part uid="p--0000000017D374A0">
…
  <Versions>
    <PartVersion uid="pv--0000000017D374A0--id1">
…
      <Views>
        <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--
0000000017D374A0--id1">
…
          <PropertyValueAssignment uid="ID_170">
            <AssignedPropertyValues>
              <PropertyValue uid="ID_156" xsi:type="n0:NumericalValue">
                <Definition>
                  <PropertyDefinition uidRef="ID_154"/>
                </Definition>
                <Name>
                  <CharacterString>PDMIFPartReal_xxx</CharacterString>
                </Name>
                <Unit uidRef="ID_155"/>
                <ValueComponent>5.678000000000000E+00</ValueComponent>
              </PropertyValue>
              <PropertyValue uid="ID_164" xsi:type="n0:StringValue">
                <Definition>
                  <PropertyDefinition uidRef="ID_163"/>
                </Definition>
                <Name>
                  <CharacterString>PDMIFPartInt_xxx</CharacterString>
                </Name>
                <ValueComponent>
                  <CharacterString>2</CharacterString>
                </ValueComponent>
              </PropertyValue>
…
            </AssignedPropertyValues>
            <ClassifiedAs>
              <Classification uidRef="ID_63"/>
            </ClassifiedAs>
          </PropertyValueAssignment>
          <PropertyValueAssignment uid="ID_225">
            <AssignedPropertyValues>
```

```
              <PropertyValue uid="ID_224" xsi:type="n0:NumericalValue">
                <Definition>
                  <PropertyDefinition uidRef="ID_216"/>
                </Definition>
                <Name>
                  <CharacterString>number of children</CharacterString>
                </Name>
                <Unit uidRef="ID_x543008816"/>
                <ValueComponent>4</ValueComponent>
              </PropertyValue>
            </AssignedPropertyValues>
            <ClassifiedAs>
              <Classification uidRef="ID_217"/>
            </ClassifiedAs>
          </PropertyValueAssignment>
…
        </PartView>
      </Views>
…
    </PartVersion>
  </Versions>
</Part>

<PropertyDefinition uid="ID_154">
  <Id id="general property"/>
  <PropertyType>
    <ClassString>customized PDM property</ClassString>
  </PropertyType>
</PropertyDefinition>
<PropertyDefinition uid="ID_163">
  <Id id="general property"/>
  <PropertyType>
    <ClassString>customized PDM property</ClassString>
  </PropertyType>
</PropertyDefinition>
<PropertyDefinition uid="ID_216">
  <Id id="quality property"/>
  <PropertyType>
    <ClassString>assembly validation property</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 (except in special cases described in section 7.4), 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 exists only in the context of an assembly structure, whereas in the part occurrences based assembly structure concept, part occurrences can exists 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 three possible kinds of occurrences in AP242 Domain 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 subtype of Occurrence (QuantifiedOccurrence) is used in the area of BoM systems (not 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.

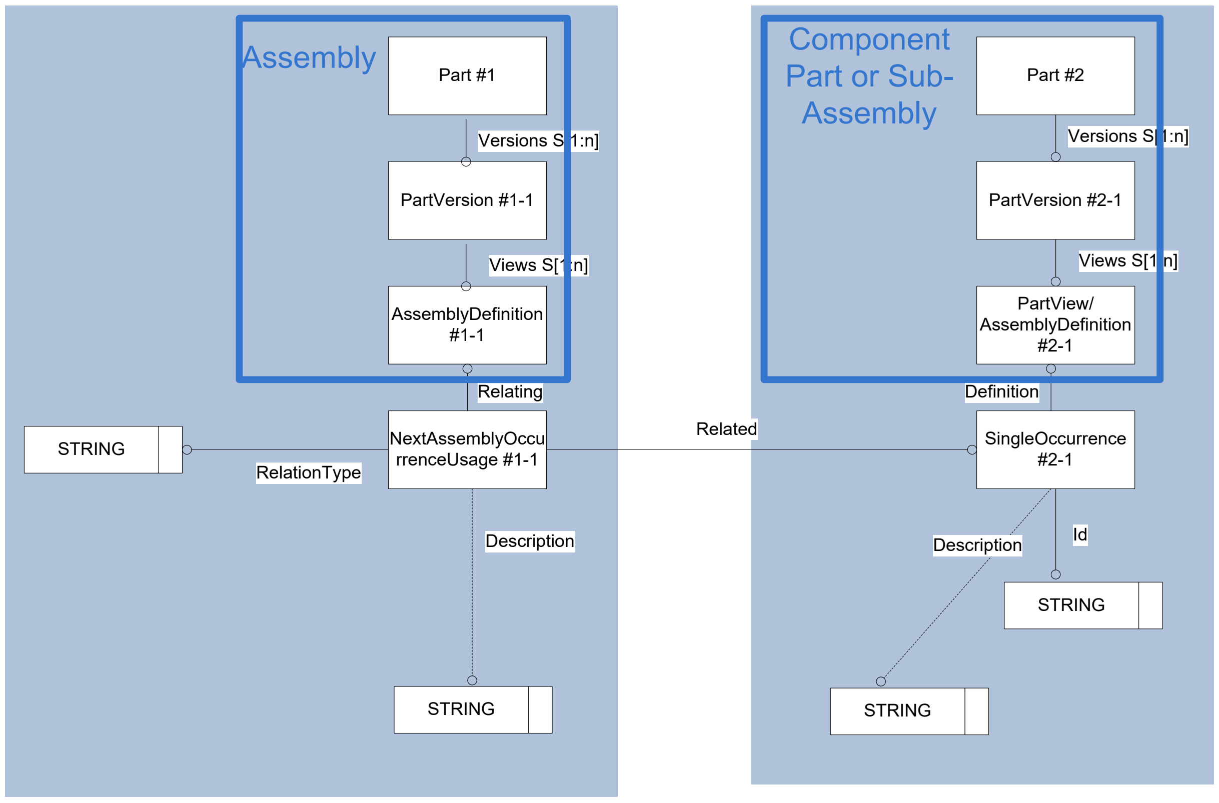


Figure 27: Template "SingleOccurrence"

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

```
<Part uid="p--000000017086CB0">
  <Id>
    <Identifier uid="pid--000000017086CB0--id1" id="as1" idRoleRef="rl--ii"
idContextRef="o--000000178"/>
  </Id>
…
  <Versions>
    <PartVersion uid="pv--000000017086CB0--id1">
...
      <Views>
        <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--
000000017086CB0--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" idRoleRef="rl--
ii" idContextRef="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.

If a single level or multiple level assembly structure contains multiple usages of the same component part, a distinct instance of SingleOccurrence shall be created for each usage of this component. In some PDM systems, each instance of SingleOccurrence referenced via NextAssemblyOccurrenceUsage by an assembly part shall have a unique Id. All of them shall be defined under the PartView/AssemblyDefinition of the component part and they shall be referenced only once via NextAssemblyOccurrenceUsage.Related. Doing so, each usage can be described via its own properties.

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 circular/cyclic link 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 Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
MaterialIdentification	OPTIONAL SET[1:?] of MaterialIdentification
Related	Occurrence
RelationType	ClassSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
SurfaceCondition	OPTIONAL SET[1:?] of SurfaceCondition
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssign-ment
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 InformationUs-ageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssign-ment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPer-sonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefini-tionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssign-ment
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
Placement	OPTIONAL SET[1:?] OF TransformationAn-dAssociationSelect
AssemblyOccurrenceRelationshipSubstitution	OPTIONAL SET[1:?] of AssemblyOccur-renceRelationshipSubstitution
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassifica-tionAssignment
ProductStructureKinematicPathAssociation	OPTIONAL SET[1:?] OF ProductStructureKine-maticPathAssociation
ProductStructureLinkMotionAssociation	OPTIONAL SET[1:?] OF ProductStructureLink-MotionAssociation

Table 38: "NextAssemblyOccurrenceUsage" Attributes

Attribute recommendations

- **ClassifiedAs:** the classifications of the NextAssemblyOccurrenceUsage. The value of this attribute need not be specified. Use “Classification” template (see 4.6.5) with value ‘alternative’ if referenced by AssemblyOccurrenceRelationshipSubstitution.Related (see 7.5.2).
- ***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***: the identifier or set of identifiers for the NextAssemblyOccurrenceUsage. The value of this attribute need not be specified. One possible use case is to map the internal system identifier in case this relationship is a business object in the PDM system.
- ***Related***: Reference to a SingleOccurence of the component part built into the assembly part.
- ***RelationType***: the meaning of the relationship. Use ClassString type. Mandatory value: 'next assembly occurrence'.
- ***ActivityAssignment***: the Activities associated to the SingleOccurrence. The value of this attribute need not be specified. Use "Activity" template (see Recommended Practices for AP242 Domain Model XML Change Management for details)).
- ***EffectivityAssignment***: to assign (optionally) one or multiple Effectivities to the usage of the Part referenced by NextAssemblyOccurrenceUsage.Related. Use the “EffectivityAssignment” template (see Recommended Practices for AP242 Domain Model Configuration Management for details).
- ***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 constituents (PartView.DefiningGeometry and optionally also the PartView.AuxiliaryGeometries) in the coordinate space of the AssemblyDefinition. Placement shall be embedded CartesianTransformations (when using the recommended simplified positioning representation, see section 7.3.1) or references to GeometricRepresentationRelationshipWithPlacementTransformations, GeometricRepresentationRelationshipWithCartesianTransformations or GeometricRepresentationRelationshipWithSameCoordinateSpaces (when using the non-recommended full positioning representations, see section 7.3.2).
- **AssemblyOccurrenceRelationshipSubstitution:** to assign (optionally) that this NextAssemblyOccurrenceUsage may be substituted for another NextAssemblyOccurrenceUsage. For more details, refer to section 7.5.2.
- ***ProductStructureKinematicPathAssociation***: to assign (optionally) one or multiple Kinematic Motions to the Part referenced by. NextAssemblyOccurrenceUsage.Related. Use the “ProductStructureKinematicPathAssociation” template (see Recommended Practices for AP242 Domain Model XML Kinematics for details).
- ***ProductStructureLinkMotionAssociation***: to assign (optionally) one or multiple Kinematic Motions to the Part referenced by. NextAssemblyOccurrenceUsage.Related. Use the “ProductStructureLinkMotionAssociation” template (see Recommended Practices for AP242 Domain Model XML Kinematics 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 SingleOccurrence	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
DefiningGeometry	OPTIONAL GeometricModel
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
KinematicLinkToOccurrenceAssociation	OPTIONAL SET[1:?] of KinematicLinkToOccurrenceAssociation
MaterialIdentification	OPTIONAL SET[1:?] of MaterialIdentification
Occurrence	OPTIONAL SET[1:?] of SpecifiedOccurrence
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
SameAs	OPTIONAL SET[1:?] of Proxy
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
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OccurrenceRelationship	OPTIONAL SET[1:?] of OccurrenceRelationship
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment
ProjectAssignment	OPTIONAL SET[1:?] of ProjectAssignment

Entity SingleOccurrence	Attribute type
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelation-ship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 39: "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 (in some PDM systems, it shall be unique over all Occurrences directly referenced via NextAssemblyOccurrenceUsage by an assembly part). Use IdentifierString type.
- **KinematicLinkToOccurrenceAssociation**: the KinematicLinks associated to this occurrence in mechanisms (see Recommended Practices for AP242 Domain Model XML Kinematics for details)
- **Occurrence**: the specific occurrences of this occurrence in a product structure. Use SpecifiedOccurrence template in 7.2. Remark: this attribute is of type SpecifiedOccurrence so it is not necessary (but allowed) to mention `xsi:type="n0:SpecifiedOccurrence"`.
- ***PropertyValueAssignment:*** adds the value of a property to the SingleOccurrence. 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.

Preprocessor Recommendations:

- When using one of the full positioning representations, each SET element within ‘Placement’ shall be cross-referenced by PartView.DefiningGeometry and PartView.AuxiliaryGeometry of the child part, so it is clear which Placement applies to which Geometry.

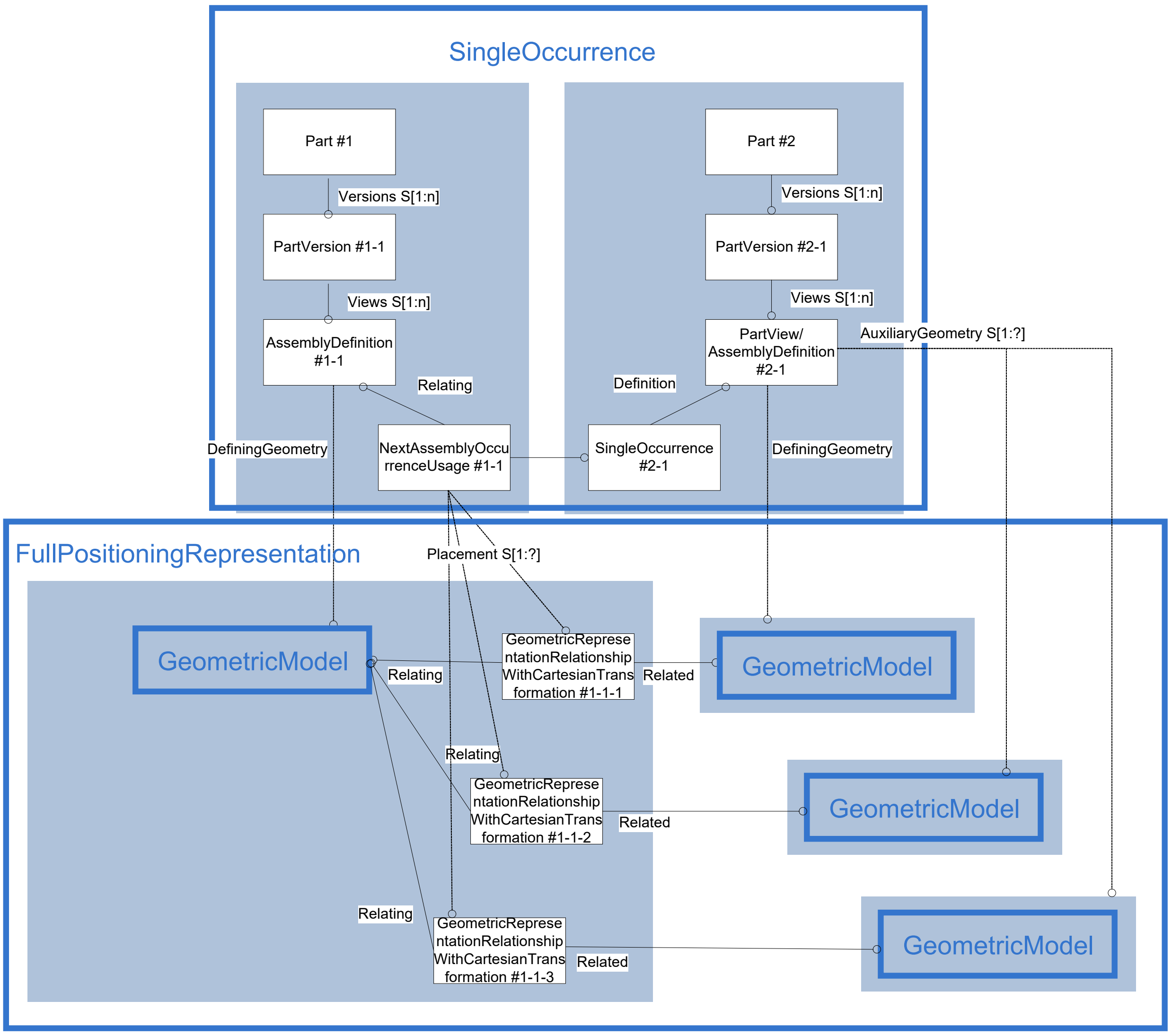


Figure 28: Cross-reference of Defining/AuxiliaryGeometry and NextAssemblyOccurrenceUsage.Placement

- When using the simplified positioning representation, there is no way to associate each Geometry to each Placement => there shall be only one Placement of kind CartesianTransformation and this Placement shall apply to all Geometries (defining and auxiliary).

Postprocessor Recommendations:

- In case the target PDM requires that the Occurrence.Id is unique over all Occurrences directly referenced via NextAssemblyOccurrenceUsage by an assembly part, the Occurrence.Id shall be made unique, for example by adding a unique suffix like _1, _2, _3, …
- Although it is recommended to attach the properties via PropertyValueAssignment to NextAssemblyOccurrenceUsage and not to SingleOccurrence, during import, since Properties may be attached to SpecifiedOccurrences, it is recommended to evaluate both NextAssemblyOccurrenceUsage.PropertyValueAssignment and SingleOccurrence.PropertyValueAssignment.

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 Train assembly contains many wagons. The Wagon again contains a sub-assembly Bogie, which again contains an Axle, etc... The requirement is to individually identify the front axle of the second bogie of the fifth wagon of the train, for example.

This is achived by having a hierarchy of SpecifiedOccurrences.

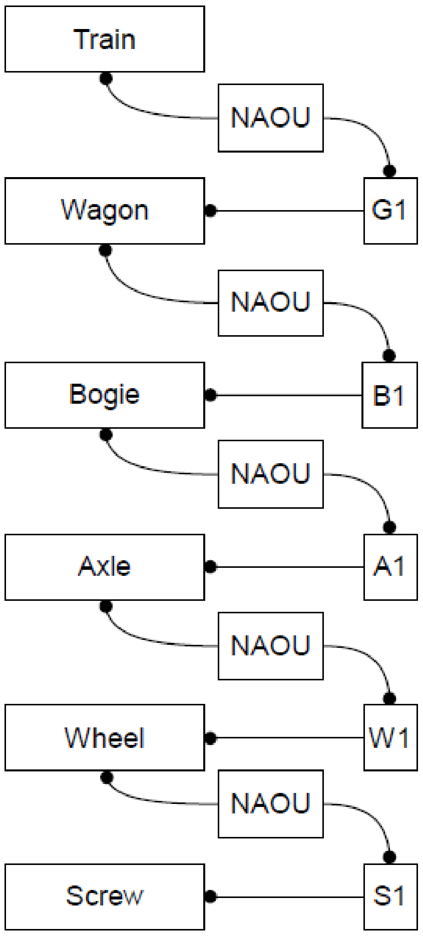


Figure 29: Example structure for multi-level assembly

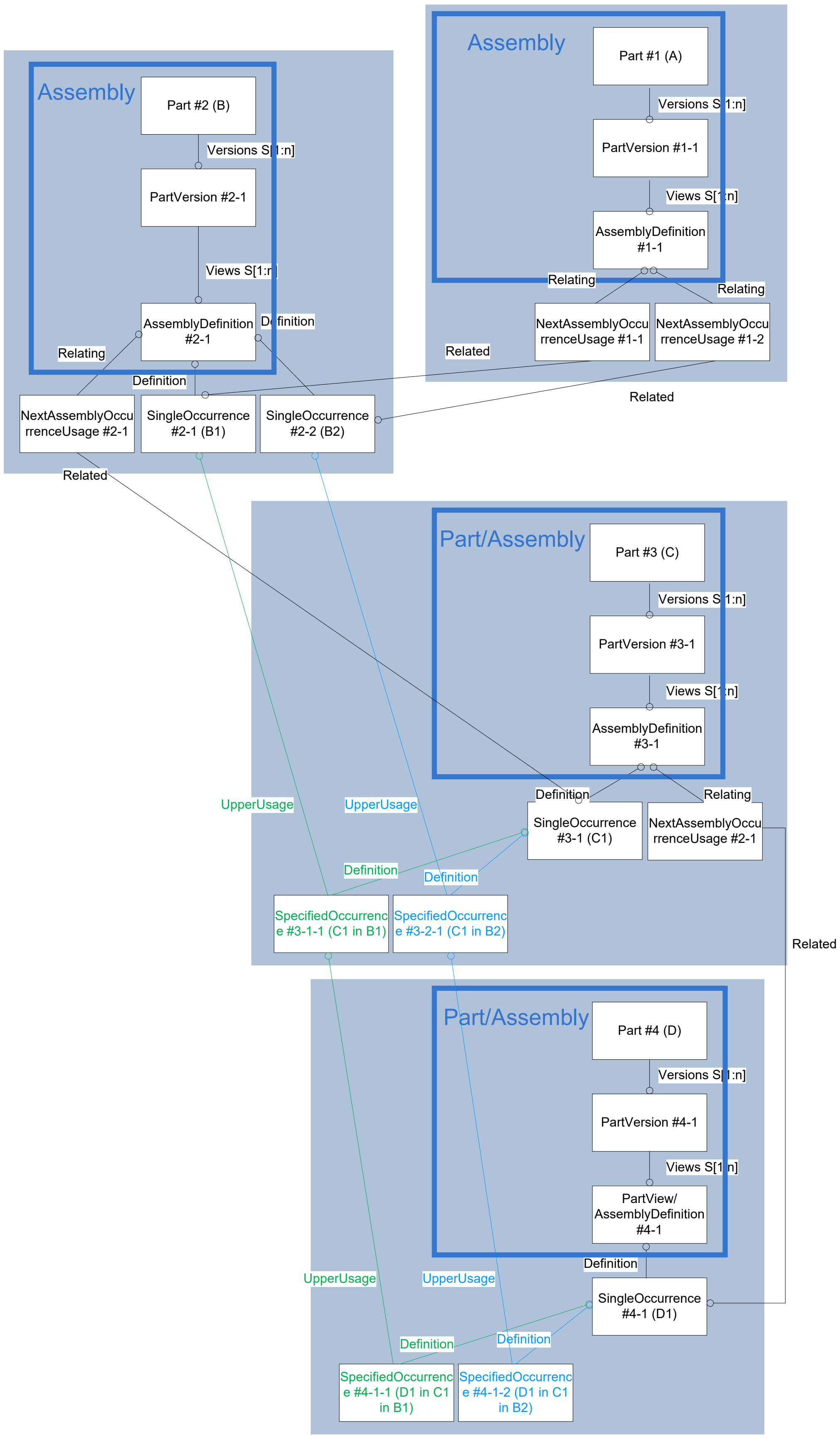


Figure 30: Template "SpecifiedOccurrence"

Note: due to the complexity of the picture, the attributes of SpecifiedOccurrence Description (reference the "Description" template) and "Id" (as IdentifierString) are not mentioned here. The same applies for the attributes of NextAssemblyOccurrenceUsage: Description ("Description template") and RelationType (ClassString).

Through the chain of SpecifiedOccurrences, it is possible to compute the 3D position of each occurrence.

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

Entity SpecifiedOccurrence	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
DefiningGeometry	OPTIONAL GeometricModel
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
KinematicLinkToOccurrenceAssociation	OPTIONAL SET[1:?] of KinematicLinkToOccurrenceAssociation
MaterialIdentification	OPTIONAL SET[1:?] of MaterialIdentification
ShapeDependentProperty	OPTIONAL SET[1:?] of ShapeDependentProperty
ShapeElement	OPTIONAL SET[1:?] of ShapeElement
Occurrence	OPTIONAL SET[1:?] of SpecifiedOccurrence
SurfaceCondition	OPTIONAL SET[1:?] of SurfaceCondition
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
BreakdownVersionAssignment	OPTIONAL SET[1:?] of BreakdownVersionAssign-ment
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 InformationUs-ageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OccurrenceRelationship	OPTIONAL SET[1:?] of OccurrenceRelationship
OrganizationOrPersonInOrganization-As-signment	OPTIONAL SET[1:?] of OrganizationOrPersonInOr-ganizationAssignment
ProjectAssignment	OPTIONAL SET[1:?] of ProjectAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of

Entity SpecifiedOccurrence	Attribute type
	PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelationship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
UpperUsage	Occurrence

Table 40: "SpecifiedOccurrence" Attributes

Attribute recommendations

- ***DefiningGeometry:*** the GeometricModel that contains the shape information. See 6.1 for details of instantiating a GeometricModel and linking it to a PartView. The value of this attribute need not be specified, except if the geometry of the related part is overloaded (see below).
- ***Description***: the text or the set of texts that provides further information about the SpecifiedOccurrence. The value of this attribute need not be specified. Use “Description” template.
- ***Id:*** stores the Identifier for the SpecifiedOccurrence (in some PDM systems, shall be unique over all Occurrences defined under this PartView). Use IdentifierString type.
- **Occurrence**: the specific occurrences of this occurrence in a product structure. Only necessary in case the intermediate SpecificOccurrences are mapped (like on Figure 32: Instantiation with all the intermediate SpecifiedOccurrences). Remark: this attribute is of type SpecifiedOccurrence so it is not necessary (but allowed) to mention `xsi:type="n0:SpecifiedOccurrence"`.
- **KinematicLinkToOccurrenceAssociation**: the KinematicLinks associated to this occurrence in mechanisms (see Recommended Practices for AP242 Domain Model XML Kinematics for details)
- ***PropertyValueAssignment:*** adds the value of a property to the SpecifiedOccurrence. Use “PropertyAssignment” template (see 6.2)
- ***UpperUsage:*** the Occurrence in which the related instance is used. This Occurrence shall be the immediate upper level instance or another SpecifiedOccurrence
- 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:

- To overload the position of the part (for example if the position depends on the load to be supported by the product), a dedicated instance of NextAssemblyOccurrenceUsage shall contain the deviating positioning (Placement) and reference via ‘Related’ an instance of SpecifiedOccurrence (instead of SingleOccurrence).
- To overload the geometry of the part, depending on its SpecifiedOccurrence, the deviating geometry shall be referenced via SpecifiedOccurrence.DefiningGeometry.
- It is not mandatory to instantiate all intermediate SpecifiedOccurrences if not needed:

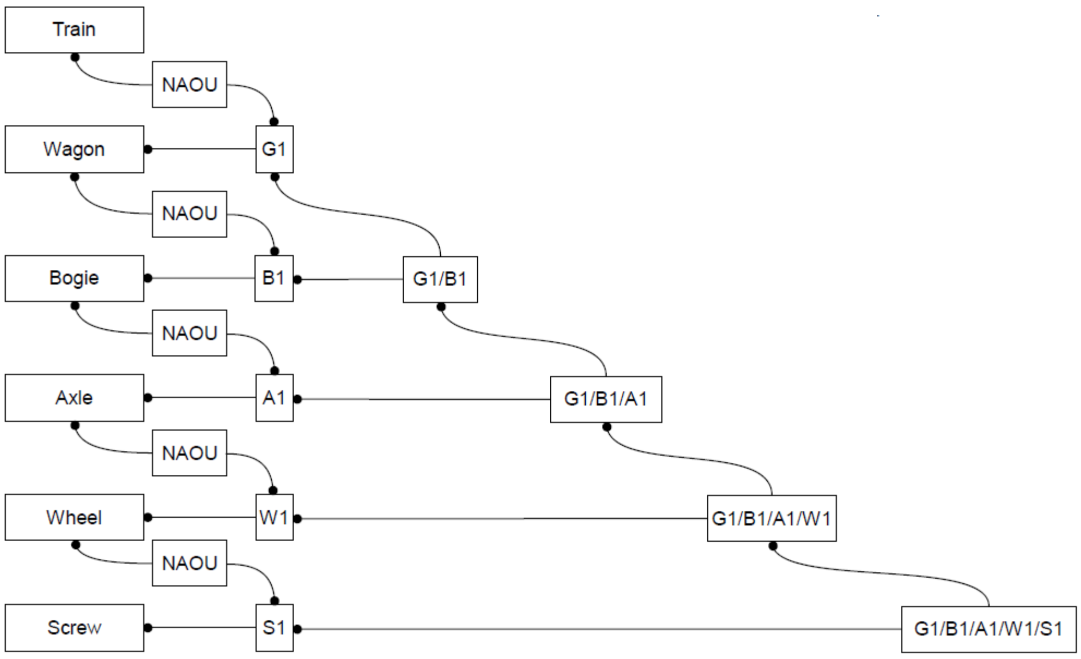


Figure 31: Instantiation without the intermediate SpecifiedOccurrences

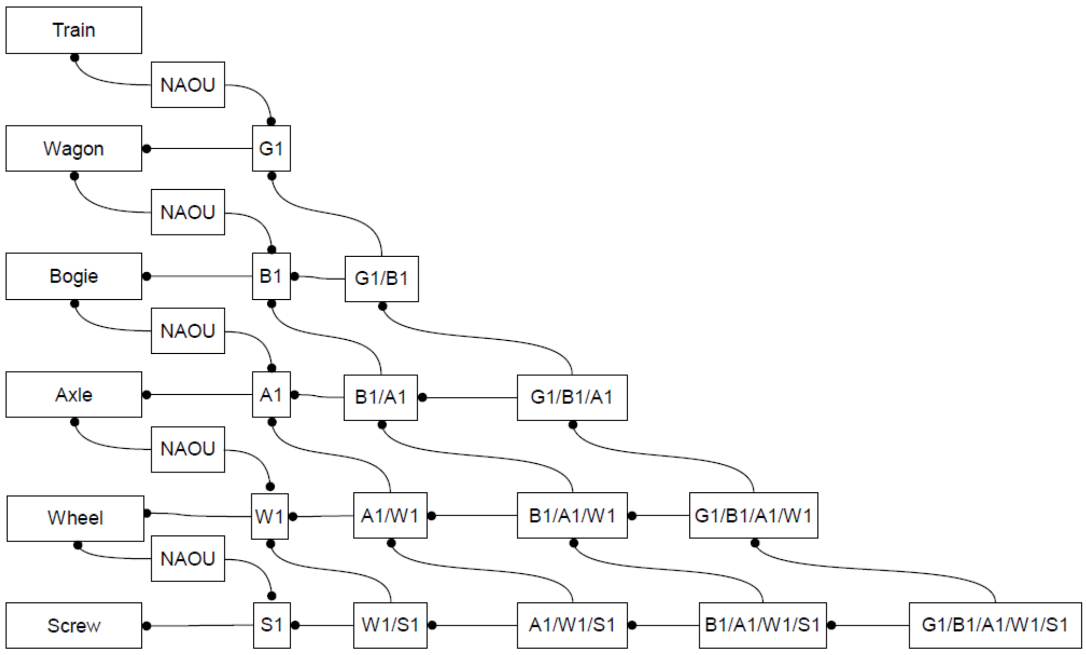


Figure 32: Instantiation with all the intermediate SpecifiedOccurrences

Instantiating all intermediate SpecifiedOccurrences provides a clear containment along the “Definition” attribute:

- S1 inherits all attributes from Screw

- W1/S1 inherits from S1

- A1/W1/S1 inherits from W1/S1

- B1/A1/W1/S1 inherits from A1/W1/S1

- G1/B1/A1/W1/S1 inherits from B1/A1/W1/S1

Postprocessor Recommendations:

- Instances of NextAssemblyOccurrenceUsage that reference a SpecifiedOccurrence via 'Related' shall only be interpreted if the target system supports the overloading of the position of parts
- SpecifiedOccurrence.DefiningGeometry shall only be interpreted if the target system supports the overloading of the geometry.

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. See more about mirroring in section 7.7.

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 Domain 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 Domain 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 instantiating GeometricModel if there is none. If there is a GeometricModel, it has to be referenced by the RepresentationRelationship 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 GeometricCoordinateSpace.

This recommendation applies independently from the fact that nested or monolithic mapping is used (see section 9.3), 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.

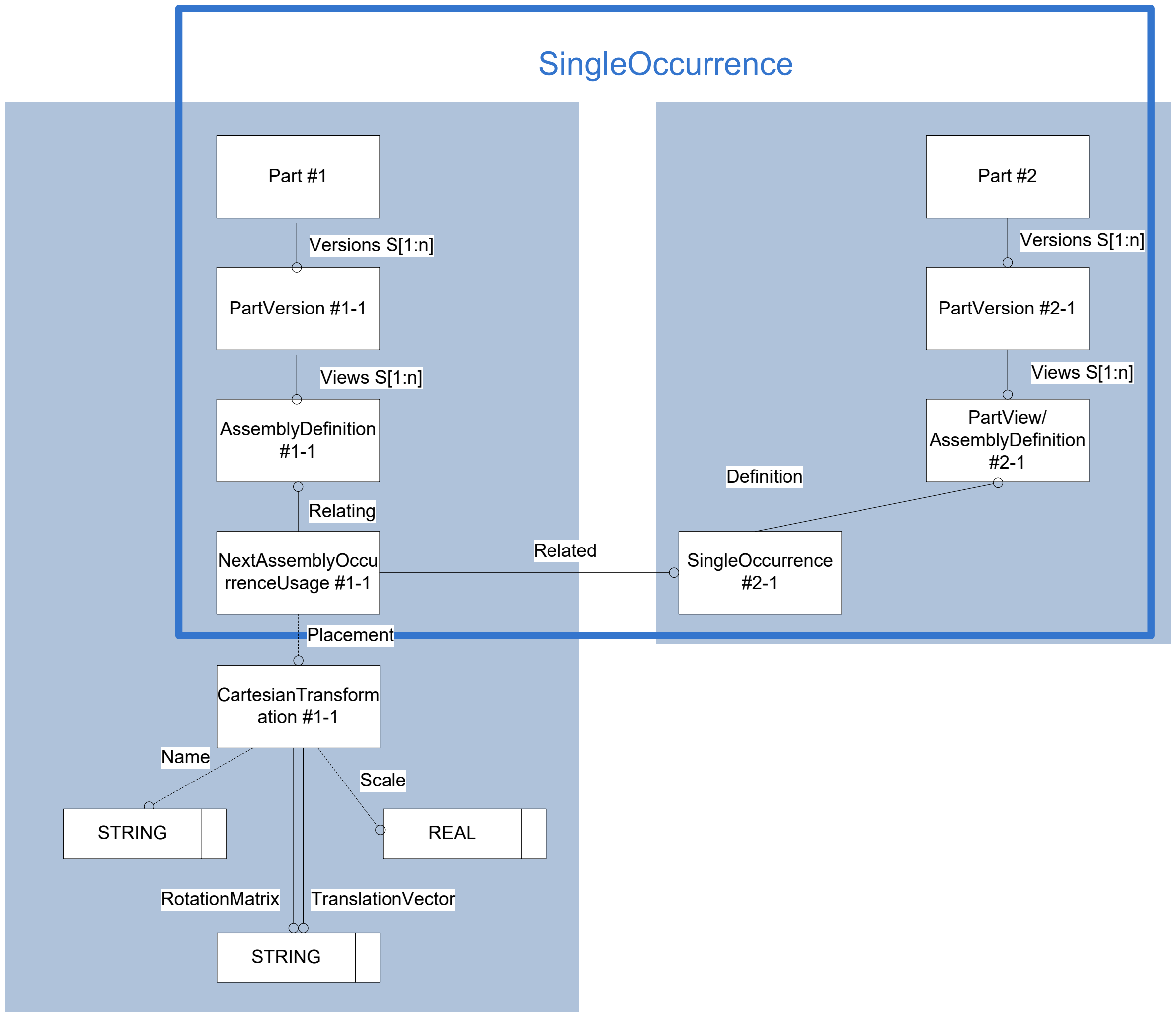
Preprocessor Recommendations:

- Unlike for the full positioning representations, the use of a GeometricCoordinateSpace is not necessary here.
- 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.

Postprocessor Recommendations:

- None.

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



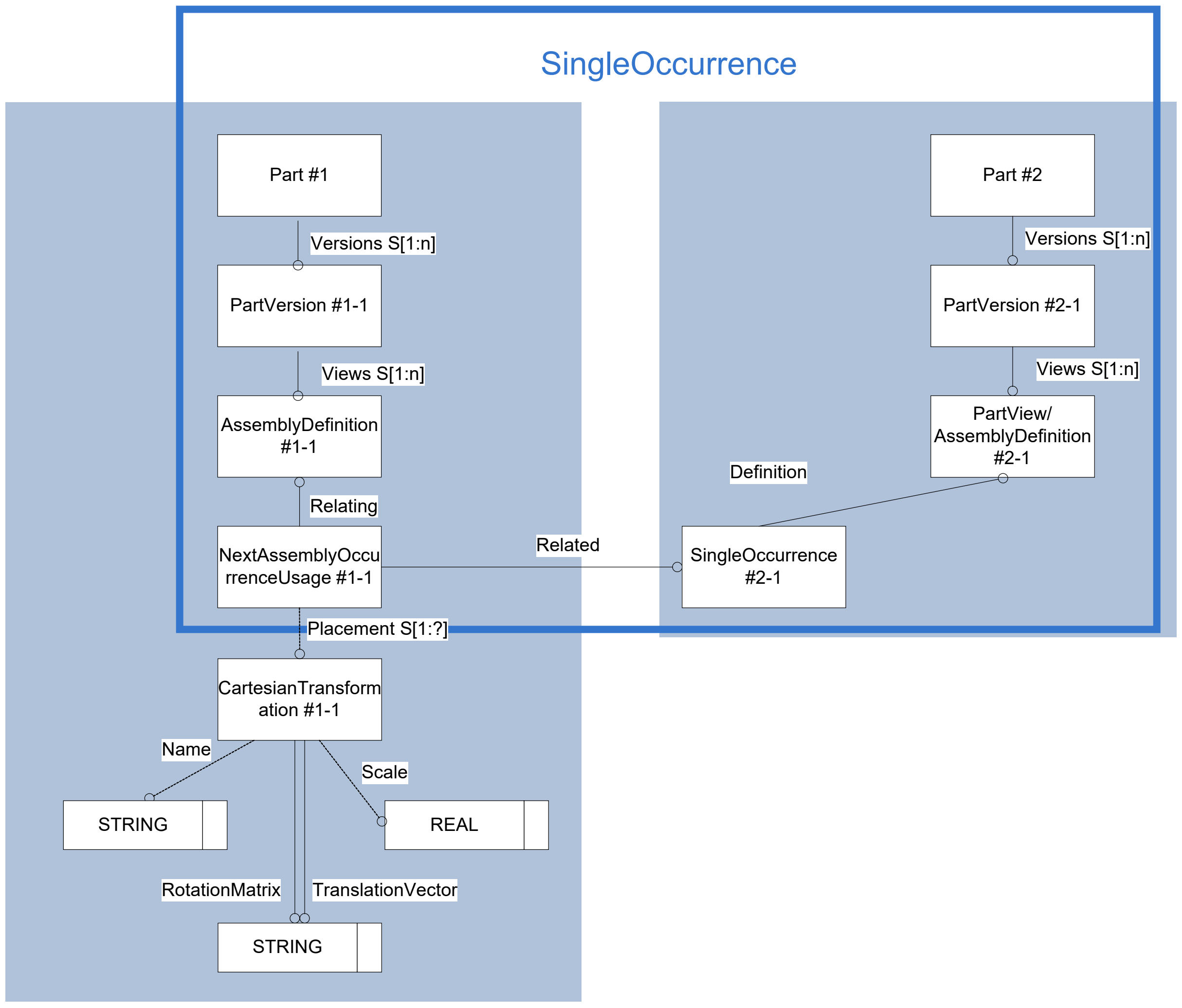


Figure 33: Template "Simplified Positioning Representation"

```
<Part uid="p--0000000017085F00">
  <Id>
    <Identifier uid="pid--0000000017085F00--id3" id="bracket_asm" idRol-
eRef="rl--ii" 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</TranslationVec-
tor>
              </CartesianTransformation>
            </Placement>
          </ViewOccurrenceRelationship>
…
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
```

Entity CartesianTransformation	Attribute type
RotationMatrix	String
Scale	Real
TranslationVector	String

Table 41: "CartesianTransformation" Attributes

Attribute recommendations

- ***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 ISO specification, the scale factor shall be omitted or set to 1.0.
- ***TranslationVector:*** 3-dimensional vector with the following values: `x y z`
- 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.

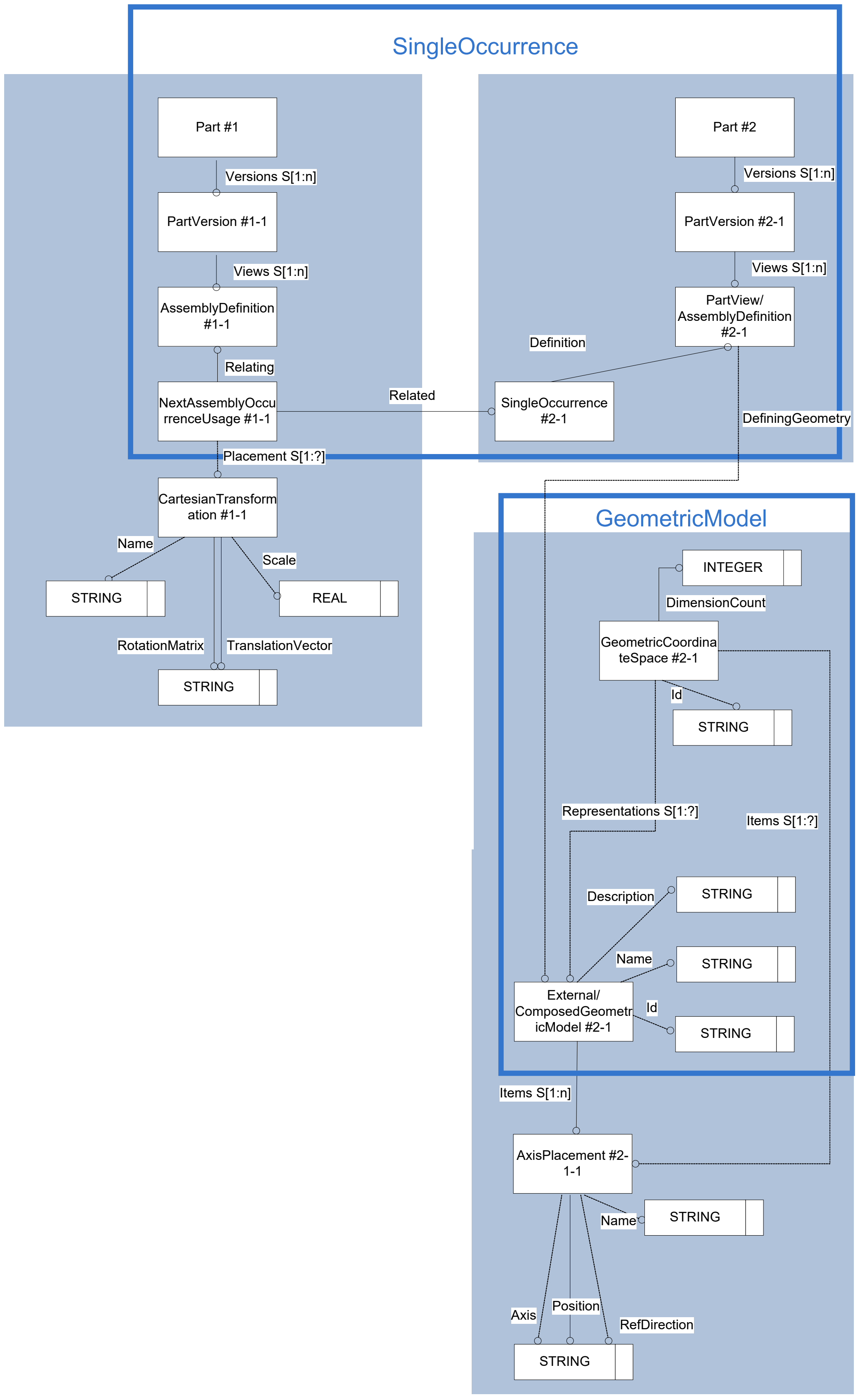


Figure 34: Instantiation example if Part #2 has geometry

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 RepresentationRelationship 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 interpreteted purely as 'for information purpose only' by the postprocessor. There are several easons 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 External-CurveSweptSolidShapeRepresentation).
- 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 GeometricRepresentationRelationshipWithCartesianTransfor-mation 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 Ge-ometricModelRelationship 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 RepresentationRelationship is of kind GeometricRepresentationRelationship-WithPlacementTransformation.

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.

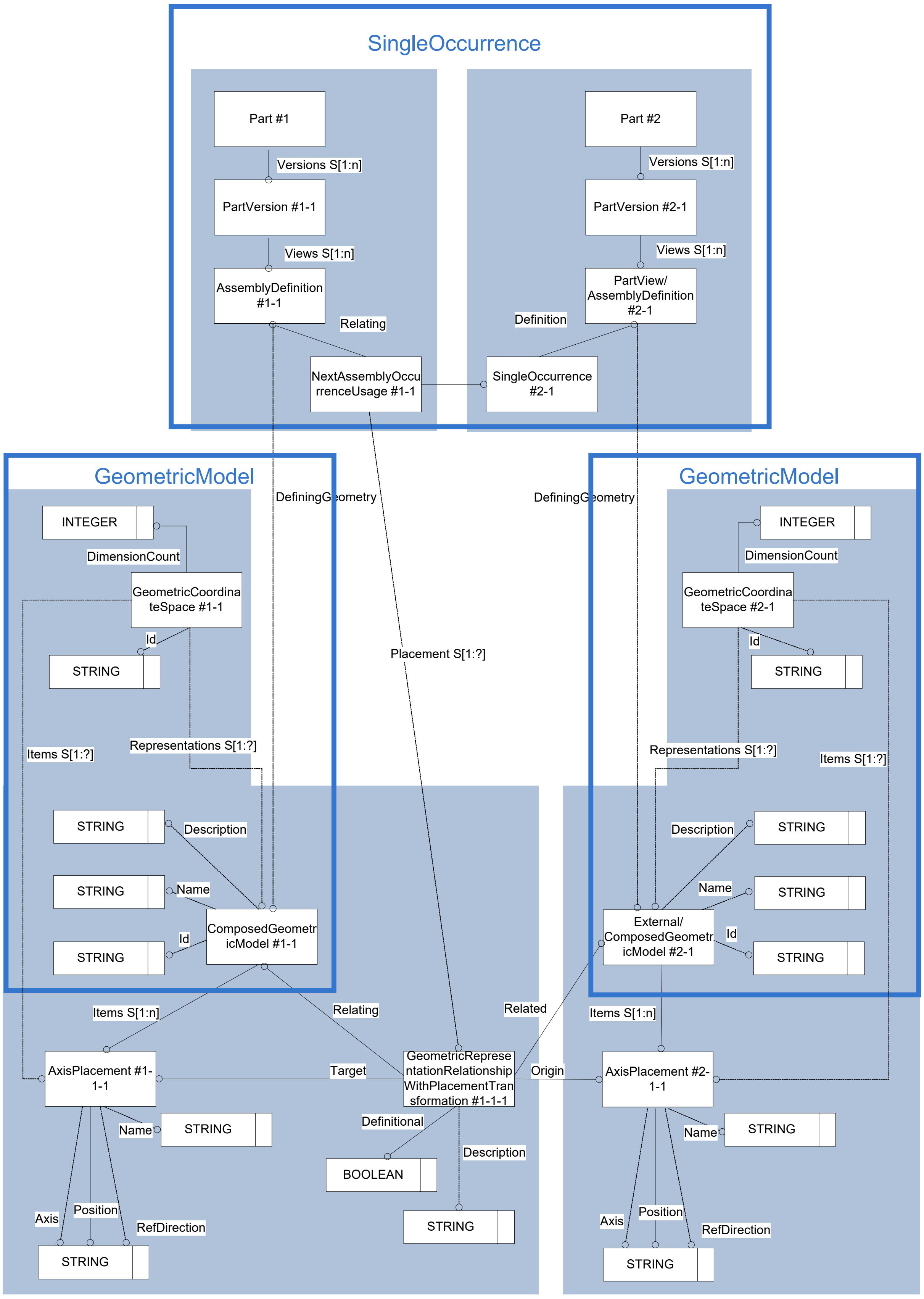


Figure 35: Full Positioning Representation with Implicit Transformation

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

```
<Part uid="p--0000000020AB6290">
  <Id>
    <Identifier uid="pid--0000000020AB6290--id5" id="nut and bolt"
idRoleRef="rl--ii" 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>
              <RepresentationRelationship 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>
              <RepresentationRelationship uidRef="ctrafo--0000000013DF75A0--
19"/>
            </Placement>
          </ViewOccurrenceRelationship>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
<RepresentationContext uid="ccs--origin-nut-and-bolt" xsi:type="n0:Geometric-
CoordinateSpace">
  <Id id="/NULL"/>
  <Representations>
    <Representation uid="egm--0000000020AB6290"
xsi:type="n0:ComposedGeometricModel">
      <Id id="nut and bolt"/>
      <Items>
        <RepresentationItem uidRef="repi--000000001AA41A00--18--2"/>
        <RepresentationItem uidRef="repi--000000001AA41A00--19--2"/>
      </Items>
      <RepresentationRelationship 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"/>
      </RepresentationRelationship>
      <RepresentationRelationship 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"/>
      </RepresentationRelationship>
    </Representation>
  </Representations>
  <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>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>

<Part uid="p--000000001AA415B0">
  <Id>
    <Identifier uid="pid--000000001AA415B0--id6" id="bolt" idRoleRef="rl--ii"
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>

<RepresentationContext uid="ccs--origin-bolt" xsi:type="n0:GeometricCoordi-
nateSpace">
  <Id id="/NULL"/>
  <Representations>
    <Representation uid="egm--000000001AA415B0"
xsi:type="n0:ExternalGeometricModel">
      <Id id="bolt.stp"/>
      <Items>
        <RepresentationItem uidRef="repi--000000001AA415B0--18"/>
      </Items>
      <ExternalFile uidRef="df--000000001AA415B0"/>
    </Representation>
  </Representations>
  <Items>
    <RepresentationItem xsi:type="n0:AxisPlacement" uid="repi--
000000001AA415B0--18">
      <Position>0.0 0.0 0.0</Position>
    </RepresentationItem>
  </Items>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>

<Part uid="p--000000001AA41A00">
  <Id>
    <Identifier uid="pid--000000001AA41A00--id7" id="nut" idRoleRef="rl--ii"
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>
```

```
<RepresentationContext uid="ccs--origin-nut" xsi:type="n0:GeometricCoordi-
nateSpace">
  <Id id="/NULL"/>
  <Representations>
    <Representation uid="egm--000000001AA41A00"
xsi:type="n0:ExternalGeometricModel">
      <Id id="nut.stp"/>
      <Items>
        <RepresentationItem uidRef="repi--000000001AA41A00--19"/>
      </Items>
      <ExternalFile uidRef="df--000000001AA41A00"/>
    </Representation>
  </Representations>
  <Items>
    <RepresentationItem xsi:type="n0:AxisPlacement" uid="repi--
000000001AA41A00--19">
      <Position>0.0 0.0 0.0</Position>
    </RepresentationItem>
  </Items>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>
```

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

Table 42: "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 RepresentationRelationship. 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 RepresentationRelationship 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.

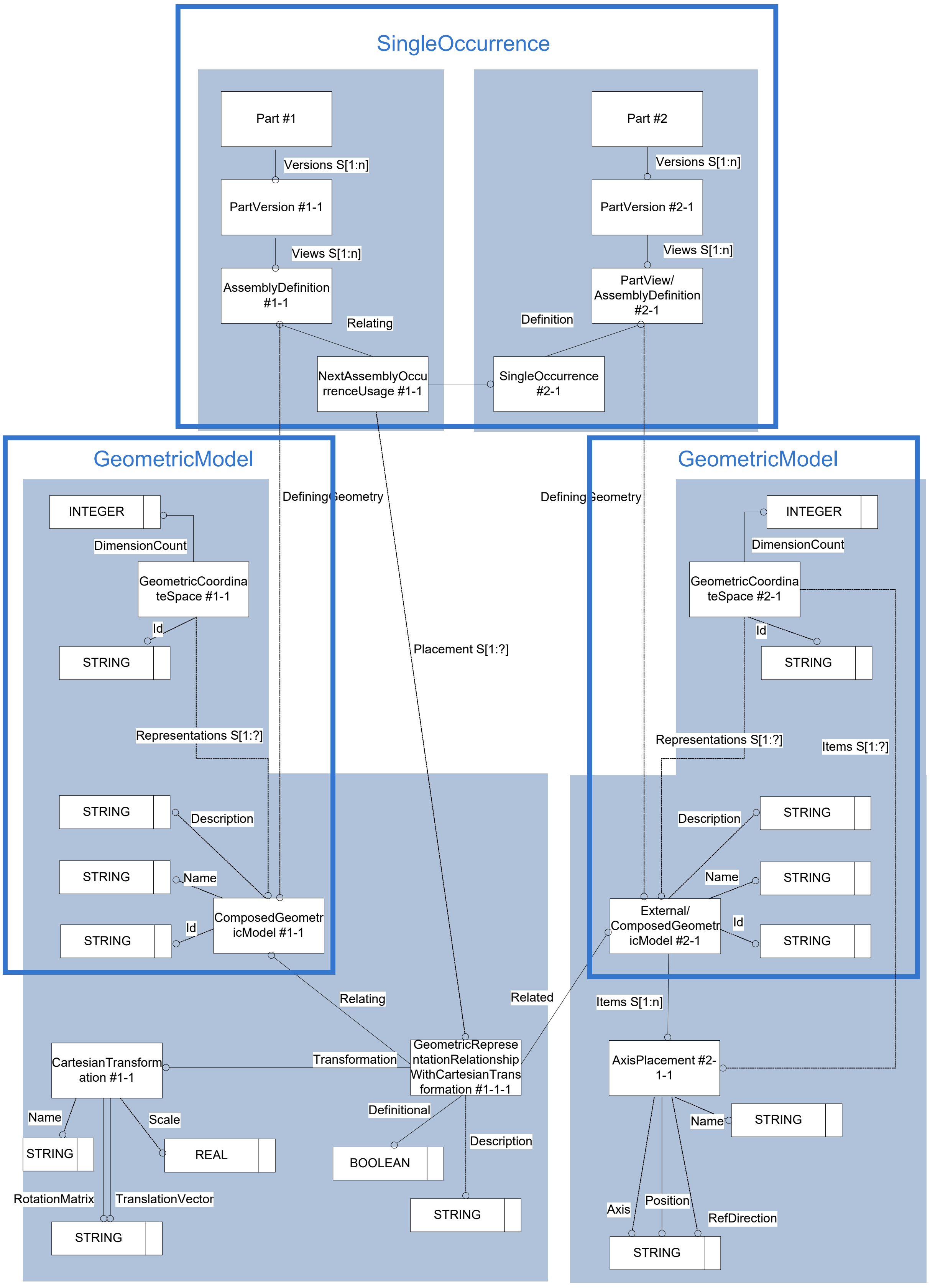


Figure 36: Full Positioning Representation with Explicit Transformation

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

```
<Part uid="p--0000000020AB6290">
  <Id>
    <Identifier uid="pid--0000000020AB6290--id5" id="nut and bolt" 
idRoleRef="rl--ii" 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>
              <RepresentationRelationship 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>
              <RepresentationRelationship uidRef="ctrafo--0000000013DF75A0--
19"/>
            </Placement>
          </ViewOccurrenceRelationship>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
<RepresentationContext uid="ccs--origin-nut-and-bolt" xsi:type="n0:Geometric-
CoordinateSpace">
  <Id id="/NULL"/>
  <Representations>
    <Representation uid="egm--0000000020AB6290" 
xsi:type="n0:ComposedGeometricModel">
      <Id id="nut and bolt"/>
      <RepresentationRelationship uid="ctrafo--0000000013DF75A0--18"
xsi:type="n0:GeometricRepresentationRelationshipWithCartesianTransformation">
        <Definitional>true</Definitional>
        <Related uidRef="egm--000000001AA415B0"/>
        <Transformation    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>
        </Transformation>
      </RepresentationRelationship>
      <RepresentationRelationship uid="ctrafo--0000000013DF75A0--19"
xsi:type="n0:GeometricRepresentationRelationshipWithCartesianTransformation">
        <Definitional>true</Definitional>
        <Related uidRef="egm--000000001AA41A00"/>
        <Transformation ~~uidRef="repi--000000001AA41A00--19--2"/>~~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>
        </Transformation>
      </RepresentationRelationship>
    </Representation>
  </Representations>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>

<Part uid="p--000000001AA415B0">
  <Id>
    <Identifier uid="pid--000000001AA415B0--id6" id="bolt" idRoleRef="rl--ii"
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>
```

```
<RepresentationContext uid="ccs--origin-bolt" xsi:type="n0:GeometricCoordi-
nateSpace">
  <Id id="/NULL"/>
  <Representations>
    <Representation uid="egm--000000001AA415B0"
xsi:type="n0:ExternalGeometricModel">
      <Id id="bolt.stp"/>
      <Items>
        <RepresentationItem uidRef="repi--000000001AA415B0--18"/>
      </Items>
      <ExternalFile uidRef="df--000000001AA415B0"/>
    </Representation>
  </Representations>
  <Items>
    <RepresentationItem xsi:type="n0:AxisPlacement" uid="repi--
000000001AA415B0--18">
      <Position>0.0 0.0 0.0</Position>
    </RepresentationItem>
  </Items>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>

<Part uid="p--000000001AA41A00">
  <Id>
    <Identifier uid="pid--000000001AA41A00--id7" id="nut" idRoleRef="rl--ii"
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>

<RepresentationContext uid="ccs--origin-nut" xsi:type="n0:GeometricCoordi-
nateSpace">
  <Id id="/NULL"/>
  <Representations>
    <Representation uid="egm--000000001AA41A00"
xsi:type="n0:ExternalGeometricModel">
      <Id id="nut.stp"/>
      <Items>
        <RepresentationItem uidRef="repi--000000001AA415B0--18"/>
      </Items>
      <ExternalFile uidRef="df--000000001AA415B0"/>
    </Representation>
  </Representations>
  <Items>
    <RepresentationItem xsi:type="n0:AxisPlacement" uid="repi--
000000001AA41A00--19">
      <Position>0.0 0.0 0.0</Position>
    </RepresentationItem>
  </Items>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>
```

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

Table 43: "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 RepresentationRelationship. 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 CartesianTransforamtion associated to the relating ComposedGeometricModel of the assembly part
- Other attributes than these are not covered by this document; 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 RepresentationRelationship is of kind GeometricRepresentationRelationship-WithSameCoordinateSpace.

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.

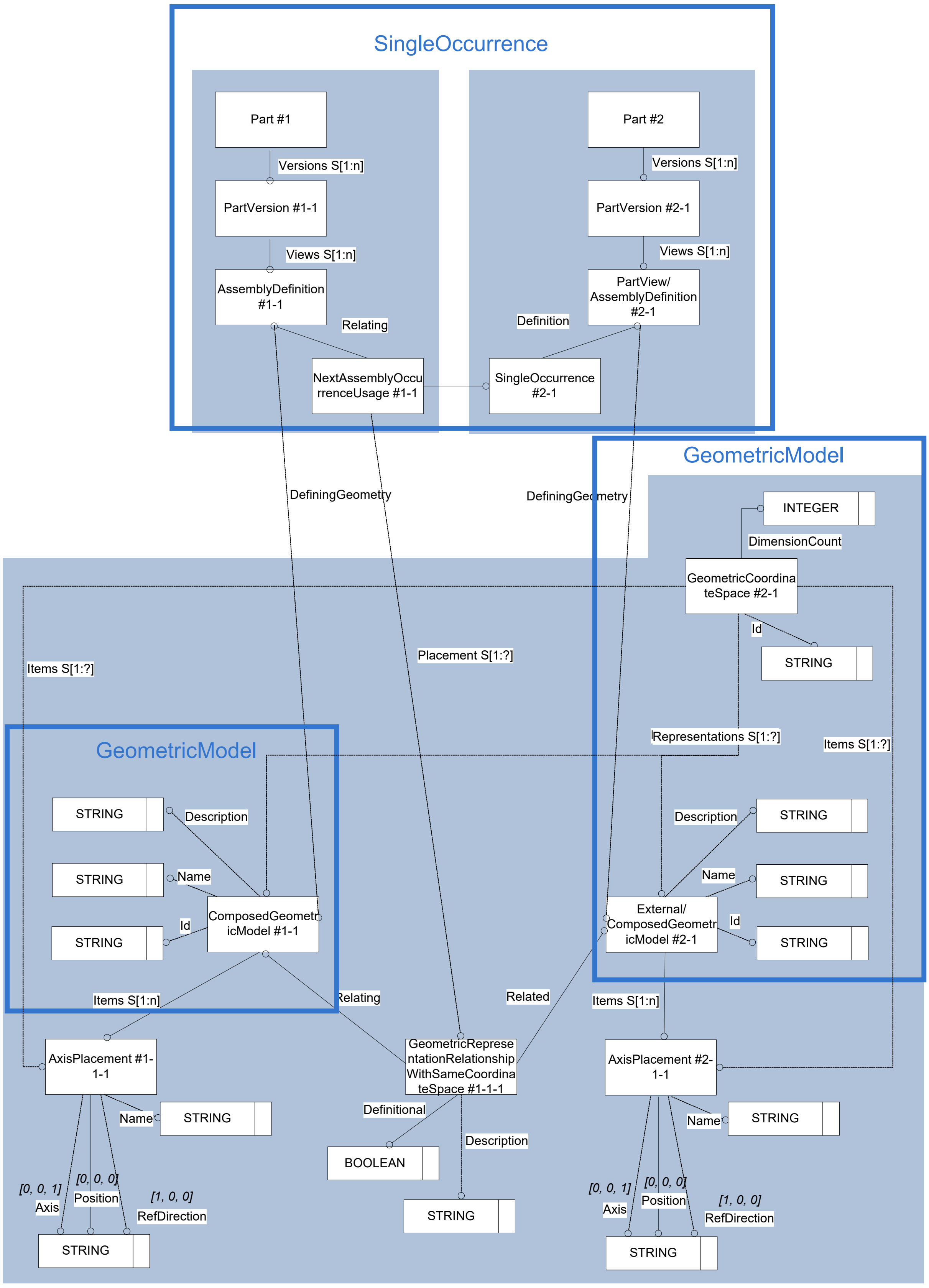


Figure 37: Full Positioning Representation with Same Coordinate Space

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

```
<Part uid="p--0000000020AB6290">
  <Id>
    <Identifier uid="pid--0000000020AB6290--id5" id="nut and bolt"
idRoleRef="rl--ii" 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>
              <RepresentationRelationship 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>
              <RepresentationRelationship uidRef="ctrafo--0000000013DF75A0--
19"/>
            </Placement>
          </ViewOccurrenceRelationship>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>

<RepresentationContext uid="ccs--origin-nut-and-bolt" xsi:type="n0:Geometric-
CoordinateSpace">
  <Id id="/NULL"/>
  <Representations>
    <Representation uid="egm--0000000020AB6290"
xsi:type="n0:ComposedGeometricModel">
```

```
      <Id id="nut and bolt"/>
      <Items>
        <RepresentationItem uidRef="repi--000000001AA41A00--18--2"/>
        <RepresentationItem uidRef="repi--000000001AA41A00--19--2"/>
      </Items>
      <RepresentationRelationship uid="ctrafo--0000000013DF75A0--18"
xsi:type="n0:GeometricRepresentationRelationshipWithSameCoordinateSpace">
        <Definitional>true</Definitional>
        <Related uidRef="egm--000000001AA415B0"/>
      </RepresentationRelationship>
      <RepresentationRelationship uid="ctrafo--0000000013DF75A0--19"
xsi:type="n0:GeometricRepresentationRelationshipWithSameCoordinateSpace">
        <Definitional>true</Definitional>
        <Related uidRef="egm--000000001AA41A00"/>
      </RepresentationRelationship>
    </Representation>
    <Representation uid="egm--000000001AA415B0" xsi:type="n0:ExternalGe-
ometricModel">
      <Id id="bolt.jt"/>
      <Items>
        <RepresentationItem uidRef="repi--000000001AA415B0--18"/>
      </Items>
      <ExternalFile uidRef="df--000000001AA415B0"/>
    </Representation>
    <Representation uid="egm--000000001AA41A00" xsi:type="n0:ExternalGe-
ometricModel">
      <Id id="nut.jt"/>
      <Items>
        <RepresentationItem uidRef="repi--000000001AA41A00--19"/>
      </Items>
      <ExternalFile uidRef="df--000000001AA41A00"/>
    </Representation>
  </Representations>
  <Items>
    <RepresentationItem uid="repi--000000001AA41A00--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>
    <RepresentationItem xsi:type="n0:AxisPlacement" uid="repi--
000000001AA415B0--18">
      <Position>0.0 0.0 0.0</Position>
    </RepresentationItem>
    <RepresentationItem xsi:type="n0:AxisPlacement" uid="repi--
000000001AA41A00--19">
      <Position>0.0 0.0 0.0</Position>
    </RepresentationItem>
  </Items>
  <DimensionCount>3</DimensionCount>
</RepresentationContext>

<Part uid="p--000000001AA415B0">
  <Id>
```

```
    <Identifier uid="pid--000000001AA415B0--id6" id="bolt" idRoleRef="rl--ii"
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>

<Part uid="p--000000001AA41A00">
  <Id>
    <Identifier uid="pid--000000001AA41A00--id7" id="nut" idRoleRef="rl--ii"
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>
```

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

Table 44: "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 RepresentationRelationship. 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

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

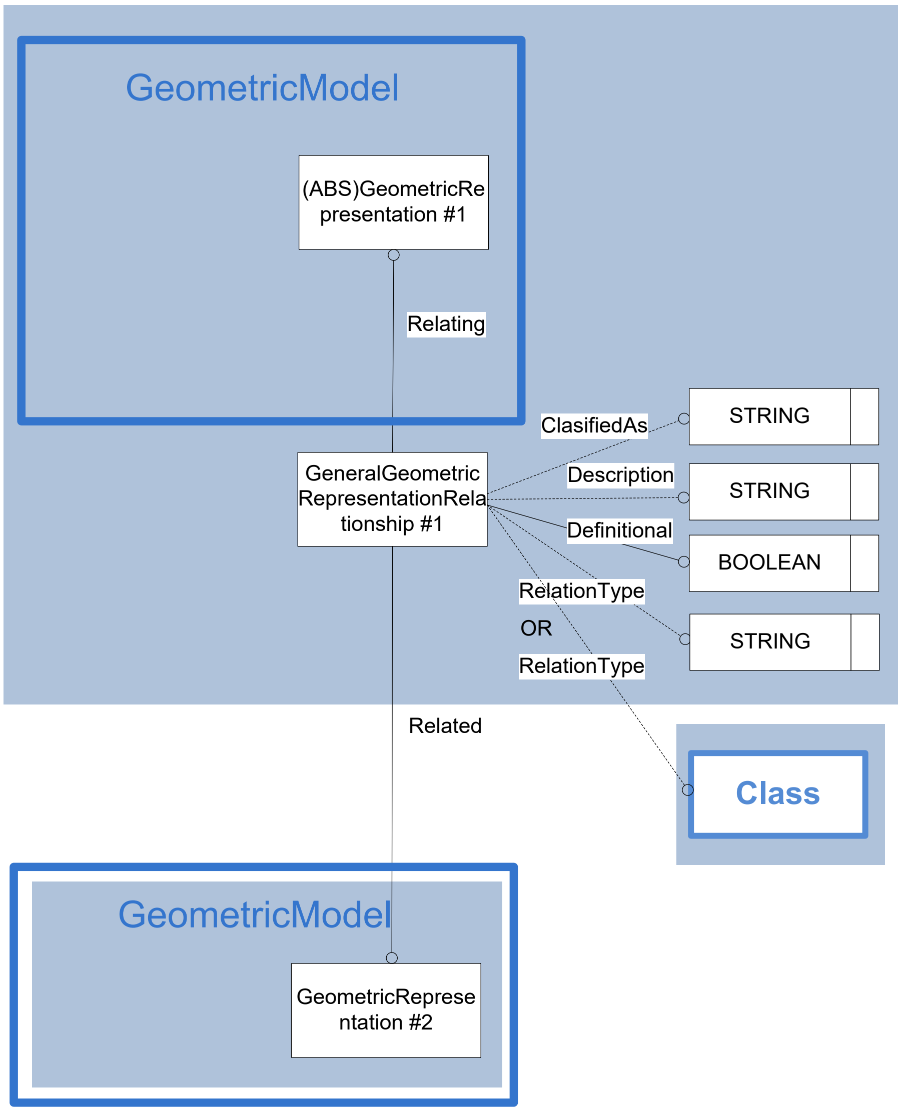


Figure 38: General Geometric Representation Relationship

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

```
<RepresentationContext uid="ccs--1" xsi:type="n0:GeometricCoordinateSpace">
  <Id id="/NULL"/>
  <Representations>
    <Representation xsi:type="n0:Mechanism" uid="kin--0000000006039F00--m">
…
      <Id id="kin--0000000006039F00--m"/>
```

```
…
      <RepresentationRelationship uid="kin--0000000006039F00--mrel"
xsi:type="n0:GeneralGeometricRepresentationRelationship">
        <Definitional>false</Definitional>
        <Related uidRef="kin--0000000006039F00--m2"/>
        <RelationType>import</RelationType>
      </RepresentationRelationship>
    </Representation>
…
  </Representations>
…
</RepresentationContext>
```

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

Table 45: "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 RepresentationRelationship. 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
‘alternate shape’	The business object defines a relationship where the relating GeometricModel defines an Alternative Instance Shape of the related GeometricModel (for example in case of Flexible Part, see section 7.8)

- 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”

These are the special cases, where the use of PartViewRelationship and its subtypes is recommended:

- to relates different views of a complex product structure (see section 15.2)
- to map a mirroring 3D placement of a part that reuses the geometry of another part (see section 7.7)
- where some properties need to be exchanged on PartViewRelationship or its subtypes
- where the quantity to be stored in NextAssemblyViewUsage deviates from the number of Occurrences given by NextAssemblyOccurrenceUsage
- where part usage substitution rules shall be mapped between NextAssemblyViewUsages (so-called AssemblyViewRelationshipSubstitutions, see chapter 7.5.3)
- where NextAssemblyViewUsage.Id (so-called FindNumber or Position number) shall be exchanged to bind multiple occurrences having the same function

Having multiple NextAssemblyViewUsages between the same parent and child is not supported by all PDM systems. However, in certain use cases, the unique FindNumber/Position is used to support multiple usages of the same child part, for example for mounting reasons (3 identical screws are used to mount one thing together and 4 further of these screws are used to mount another thing together, all of them defined as childs within one assembly node)

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

Preprocessor Recommendations:

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.

If multiple NextAssemblyViewUsages exist between the same parent and child, it is recommended to concatenate the NextAssemblyViewUsage.Id into the NextAssemblyOccurrenceUsage.Id so that those Occurrences involved in one NextAssemblyViewUsage can be identified.

Postprocessor Recommendations:

If the target PDM system has also an aggregation object as defined above, it may import the properties associated to the NextAssemblyViewUsageobjects (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, of if it gets created automatically out of the sum of all NextAssemblyOccurrenceUsages, the postprocessor shall ignore the NextAssemblyViewUsage.

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

The postprocessor shall recognize multiple NextAssemblyViewUsages 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 NextAssemblyViewUsages and NextAssemblyOccurrenceUsages with the appropriate semantic, ISO Jira Tasks TCSC410303-1398 and TCSC410303-661 have been submitted.

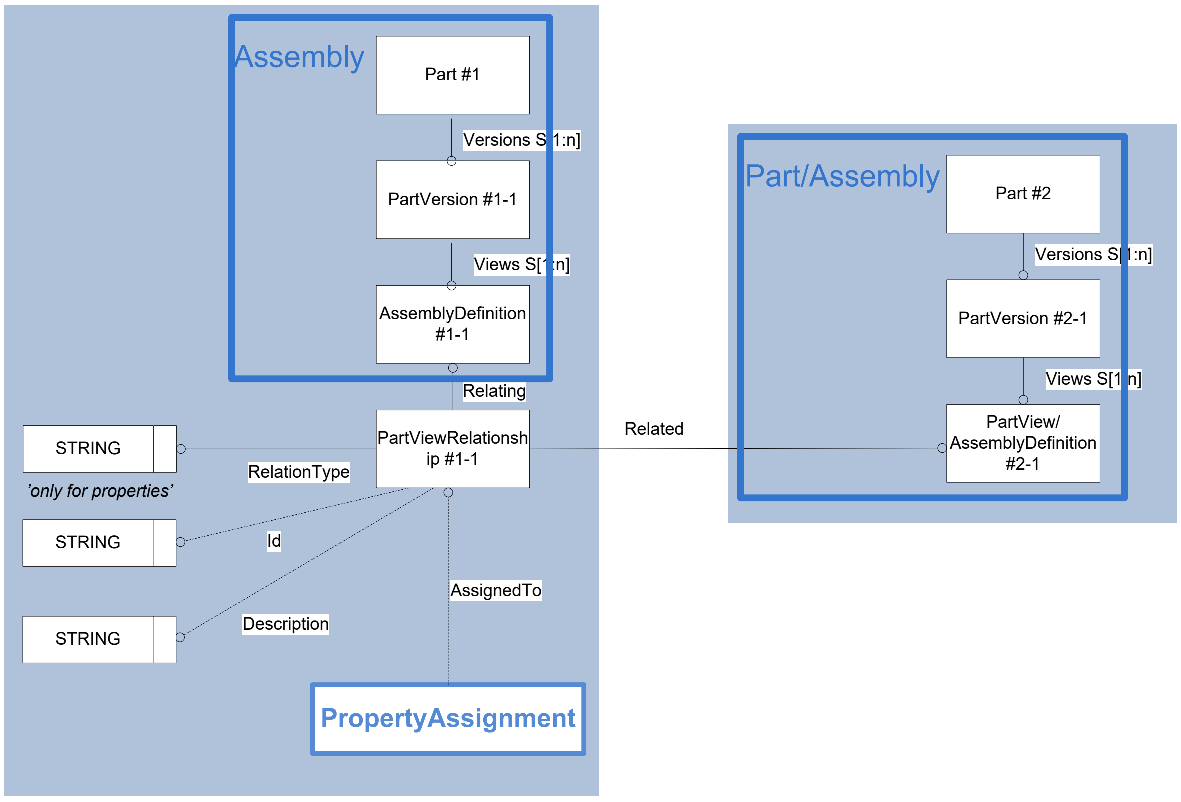


Figure 39: Template "PartViewRelationship" for properties

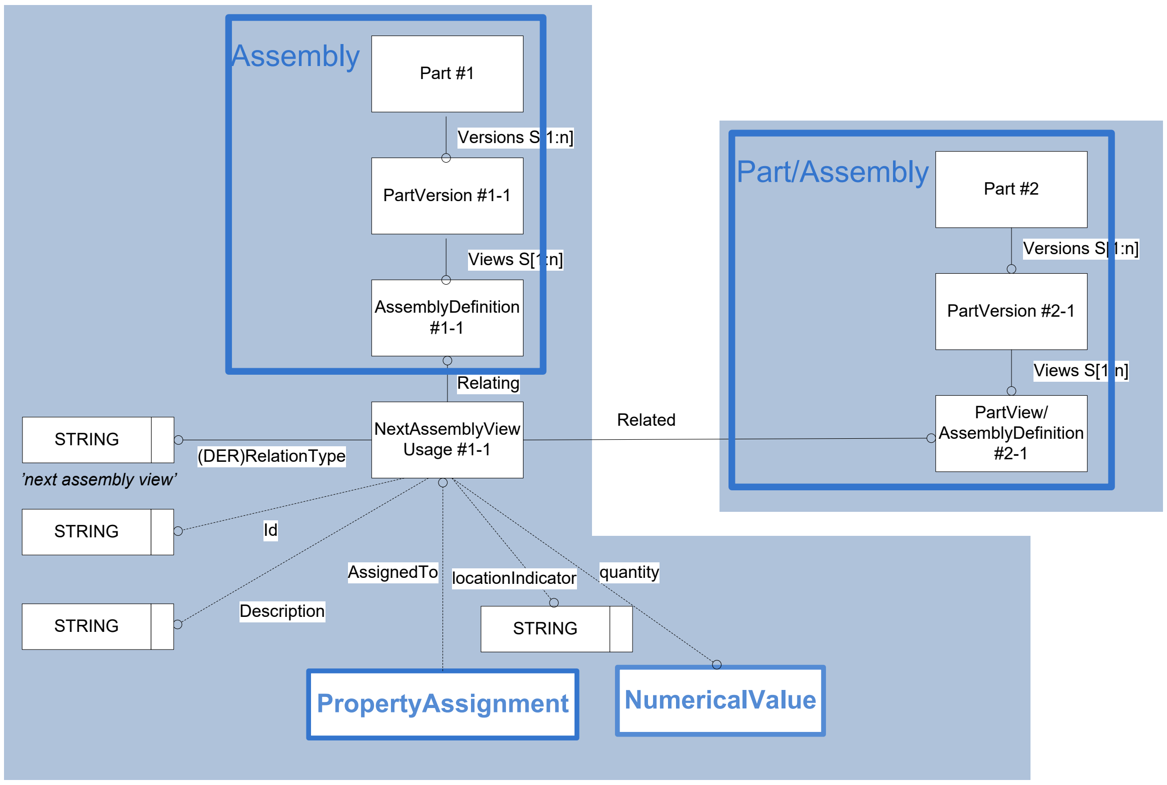


Figure 40: Template "PartViewRelationship" for quantity (+ properties)

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain 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" idRoleRef="rl--ii"
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"> </PropertyDefini-
                        tion>
                  </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" idRoleRef="rl--
ii" 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 Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
InterfaceConnection	OPTIONAL SET[1:?] of InterfaceConnection
MaterialIdentification	OPTIONAL SET[1:?] of MaterialIdentification
Related	PartView
RelationType	ClassSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
SurfaceCondition	OPTIONAL SET[1:?] of SurfaceCondition
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment

ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
AnalysisAssignment	OPTIONAL SET[1:?] of AnalysisAssignment
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
StateAssignment	OPTIONAL SET[1:?] of StateAssignment
StateDefinitionAssignment	OPTIONAL SET[1:?] of StateDefinitionAssignment
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssignment

Table 46: "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. Recommended values are:

RelationType	
'next assembly view'	The quantity to be stored in NextAssemblyViewUsage deviates from the number of Occurrences given by NextAssemblyOccurrenceUsage (see above in this section), or some substitute parts are defined via AssemblyViewRelationshipSubstitution (see section 7.5.3) According to the definition of NextAssemblyViewUsage (subtype of PartViewRelationship), only this value of RelationType is allowed (DERIVE attribute with fix value)
'only for properties'	Some properties are exchanged along the assembly structure on PartViewRelationship level
'mirrored'	The subtype GeometricalRelationship is used for mirroring (see section 7.7)
'tracking'	The PartViewRelationship relates different views of a complex product structure (see section 15.2)

  According to the above recommendation not to use multiple PartViewRelationships between the same parent and child, properties and substitutes for example shall be mapped on the same instance. In such a case, use 'next assembly view' as soon as NextAssemblyViewUsage is engaged.

- ***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 Domain Model XML syntax)

```
<Part uid="p--0000000017086CB0">
  <Id>
    <Identifier uid="pid--0000000017086CB0--id1" id="as1" idRoleRef="rl--ii"
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" idRoleRef="rl--
ii" 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 PartViewRela-tionship)
ClassifiedAs	OPTIONAL SET[1:?] of Classification
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 ActivityMethodAssign-ment

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
AssemblyViewRelationshipSubstitution	OPTIONAL SET[1:?] of AssemblyViewRelationshipSubstitution
DefiningGeometry	OPTIONAL GeometricModel
LocationIndicator	OPTIONAL IdentifierSelect
Quantity	OPTIONAL ValueWithUnit
ShapeDependentProperty	OPTIONAL SET[1:?] of ShapeDependentProperty
ShapeElement	OPTIONAL SET[1:?] of ShapeElement
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment

Table 47: "NextAssemblyViewUsage" Attributes

Attribute recommendations

- **ClassifiedAs:** the classifications of the NextAssemblyViewUsage. The value of this attribute need not be specified. Use “Classification” template (see 4.6.5) with value ‘alternative’ if referenced by AssemblyViewRelationshipSubstitution.Related (see 7.5.3).

- **AssemblyViewRelationshipSubstitution:** to assign (optionally) that this NextAssemblyViewUsage may be substituted for another NextAssemblyViewUsage. For more details, refer to section 7.5.3.
- ***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. Since the semantic is handled here by the attribute naming, use PropertyDefinitionString='quantity' for Quantity.Definition (see section 4.6.9) and leave NumericalValue.Name unset.
- 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.

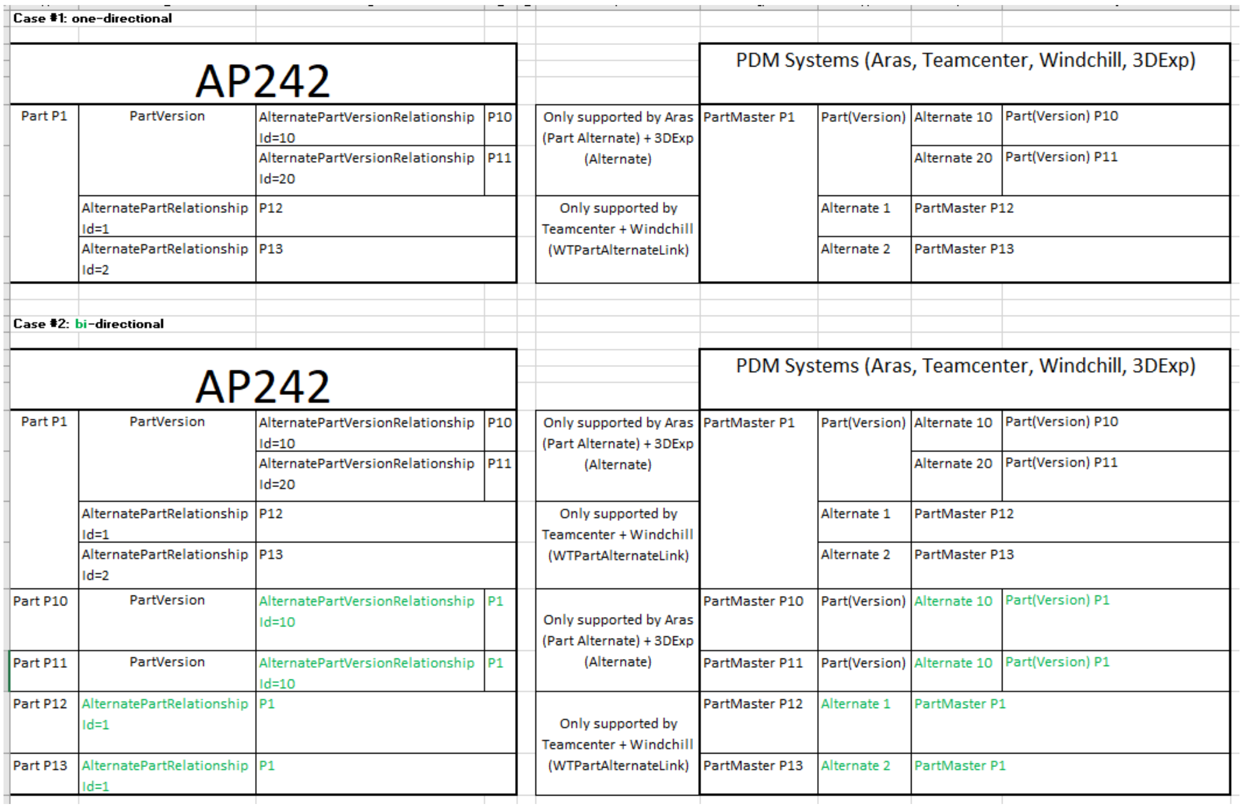
Preprocessor Recommendation:

- The usage of ValueRange, ValueLimit, ValueWithTolerances and LimitsAndFits for Quantity is not recommended.

7.5 *Alternate and Substitute Parts*

This chapter describes the handling of alternate and substitute parts.

This table gives an overview of all alternate/substitute mechanisms, as well as their support by the PDM systems:



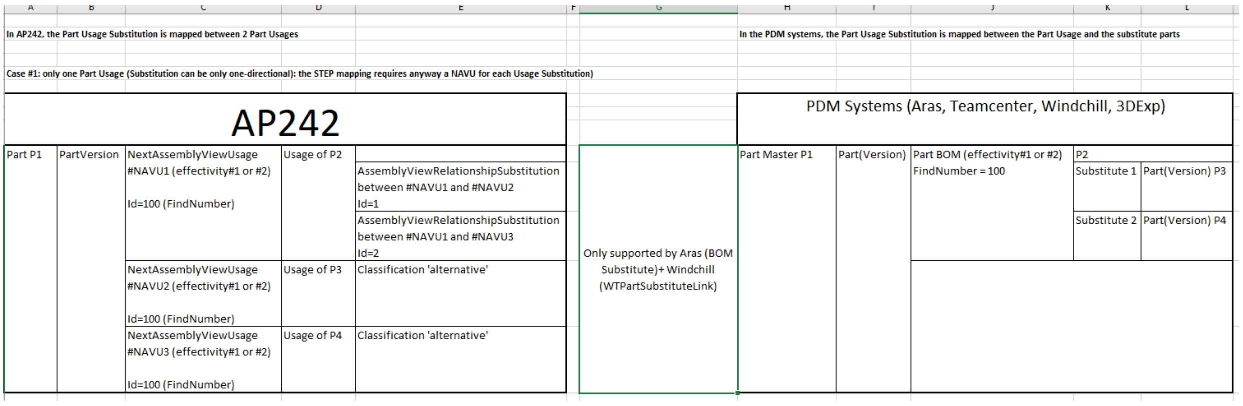
Case #1: one-directional

AP242					PDM Systems (Aras, Teamcenter, Windchill, 3DExp)			
Part P1	PartVersion	AlternatePartVersionRelationship Id=10	P10	Only supported by Aras (Part Alternate) + 3DExp (Alternate)	PartMaster P1	Part(Version)	Alternate 10	Part(Version) P10
		AlternatePartVersionRelationship Id=20	P11				Alternate 20	Part(Version) P11
	AlternatePartRelationship Id=1	P12		Only supported by Teamcenter + Windchill (WTPartAlternateLink)		Alternate 1	PartMaster P12	
	AlternatePartRelationship Id=2	P13				Alternate 2	PartMaster P13	

Case #2: bi-directional

AP242					PDM Systems (Aras, Teamcenter, Windchill, 3DExp)			
Part P1	PartVersion	AlternatePartVersionRelationship Id=10	P10	Only supported by Aras (Part Alternate) + 3DExp (Alternate)	PartMaster P1	Part(Version)	Alternate 10	Part(Version) P10
		AlternatePartVersionRelationship Id=20	P11				Alternate 20	Part(Version) P11
	AlternatePartRelationship Id=1	P12		Only supported by Teamcenter + Windchill (WTPartAlternateLink)		Alternate 1	PartMaster P12	
	AlternatePartRelationship Id=2	P13				Alternate 2	PartMaster P13	
Part P10	PartVersion	AlternatePartVersionRelationship Id=10	P1	Only supported by Aras (Part Alternate) + 3DExp (Alternate)	PartMaster P10	Part(Version)	Alternate 10	Part(Version) P1
Part P11	PartVersion	AlternatePartVersionRelationship Id=10	P1		PartMaster P11	Part(Version)	Alternate 10	Part(Version) P1
Part P12	AlternatePartRelationship Id=1	P1		Only supported by Teamcenter + Windchill (WTPartAlternateLink)	PartMaster P12	Alternate 1	PartMaster P1	
Part P13	AlternatePartRelationship Id=1	P1			PartMaster P13	Alternate 2	PartMaster P1	

Table 48: Overview of all Alternate mechanisms

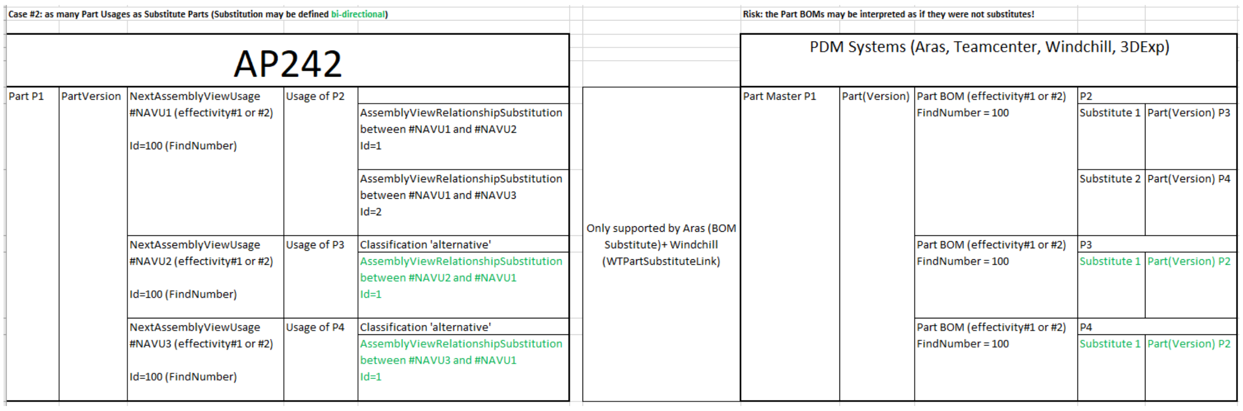


In AP242, the Part Usage Substitution is mapped between 2 Part Usages

In the PDM systems, the Part Usage Substitution is mapped between the Part Usage and the substitute parts

Case #1: only one Part Usage (Substitution can be only one-directional): the STEP mapping requires anyway a NAVU for each Usage Substitution)

AP242						PDM Systems (Aras, Teamcenter, Windchill, 3DExp)				
Part P1	PartVersion	NextAssemblyViewUsage #NAVU1 (effectivity#1 or #2) Id=100 (FindNumber)	Usage of P2		Only supported by Aras (BOM Substitute)+ Windchill (WTPartSubstituteLink)	Part Master P1	Part(Version)	Part BOM (effectivity#1 or #2) FindNumber = 100	P2	
				AssemblyViewRelationshipSubstitution between #NAVU1 and #NAVU2 Id=1					Substitute 1	Part(Version) P3
				AssemblyViewRelationshipSubstitution between #NAVU1 and #NAVU3 Id=2					Substitute 2	Part(Version) P4
		NextAssemblyViewUsage #NAVU2 (effectivity#1 or #2) Id=100 (FindNumber)	Usage of P3	Classification 'alternative'						
		NextAssemblyViewUsage #NAVU3 (effectivity#1 or #2) Id=100 (FindNumber)	Usage of P4	Classification 'alternative'						



Case #2: as many Part Usages as Substitute Parts (Substitution may be defined bi-directional)

Risk: the Part BOMs may be interpreted as if they were not substitutes!

AP242				
Part P1	PartVersion	NextAssemblyViewUsage #NAVU1 (effectivity#1 or #2) Id=100 (FindNumber)	Usage of P2	

				AssemblyViewRelationshipSubstitution
				between #NAVU1 and #NAVU2
				Id=1

				AssemblyViewRelationshipSubstitution
				between #NAVU1 and #NAVU3
				Id=2

		NextAssemblyViewUsage #NAVU2 (effectivity#1 or #2) Id=100 (FindNumber)	Usage of P3	Classification 'alternative'

				AssemblyViewRelationshipSubstitution
				between #NAVU2 and #NAVU1
				Id=1

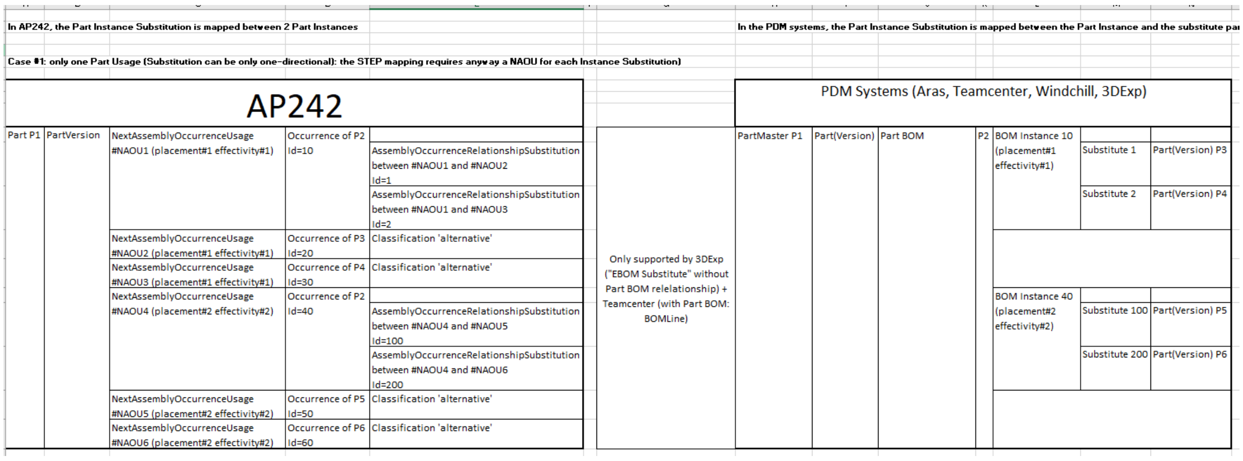
		NextAssemblyViewUsage #NAVU3 (effectivity#1 or #2) Id=100 (FindNumber)	Usage of P4	Classification 'alternative'

				AssemblyViewRelationshipSubstitution
				between #NAVU3 and #NAVU1
				Id=1

Only supported by Aras (BOM Substitute)+ Windchill (WTPartSubstituteLink)

PDM Systems (Aras, Teamcenter, Windchill, 3DExp)				
Part Master P1	Part(Version)	Part BOM (effectivity#1 or #2) FindNumber = 100	P2	
			Substitute 1	Part(Version) P3
			Substitute 2	Part(Version) P4
		Part BOM (effectivity#1 or #2) FindNumber = 100	P3	
			Substitute 1	Part(Version) P2
		Part BOM (effectivity#1 or #2) FindNumber = 100	P4	
			Substitute 1	Part(Version) P2

Table 49: Overview of all View Usage Substitution mechanisms



In AP242, the Part Instance Substitution is mapped between 2 Part Instances

In the PDM systems, the Part Instance Substitution is mapped between the Part Instance and the substitute pai

Case #1: only one Part Usage (Substitution can be only one-directional): the STEP mapping requires anyway a NAOU for each Instance Substitution)

AP242				
Part P1	PartVersion	NextAssemblyOccurrenceUsage #NAOU1 (placement#1 effectivity#1)	Occurrence of P2 Id=10	

				AssemblyOccurrenceRelationshipSubstitution
				between #NAOU1 and #NAOU2
				Id=1

				AssemblyOccurrenceRelationshipSubstitution
				between #NAOU1 and #NAOU3
				Id=2

		NextAssemblyOccurrenceUsage #NAOU2 (placement#1 effectivity#1)	Occurrence of P3 Id=20	Classification 'alternative'
		NextAssemblyOccurrenceUsage #NAOU3 (placement#1 effectivity#1)	Occurrence of P4 Id=30	Classification 'alternative'
		NextAssemblyOccurrenceUsage #NAOU4 (placement#2 effectivity#2)	Occurrence of P2 Id=40	

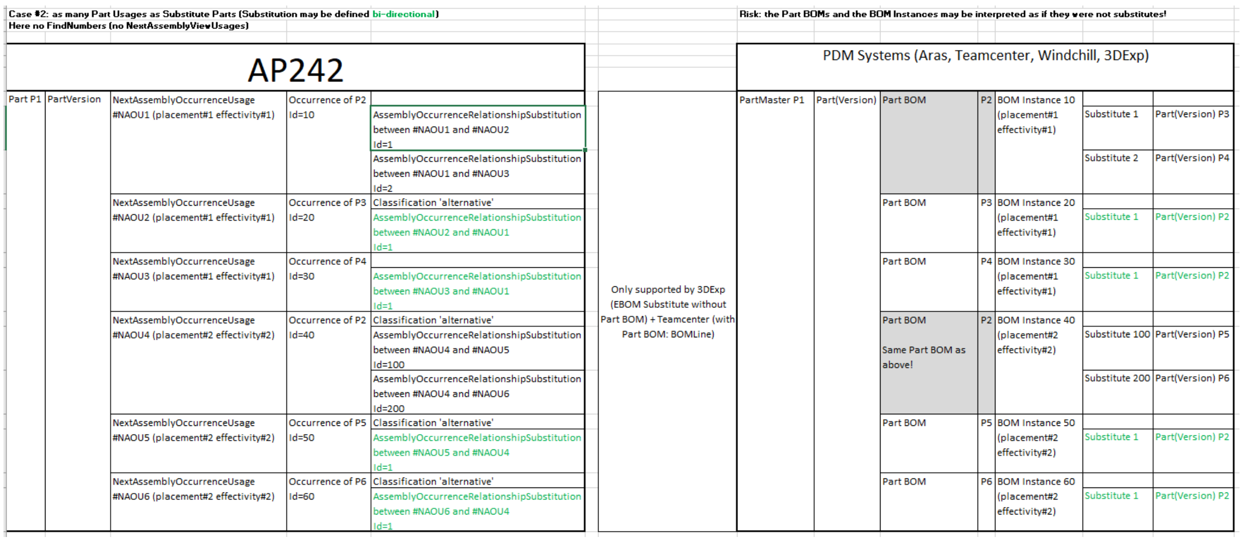
				AssemblyOccurrenceRelationshipSubstitution
				between #NAOU4 and #NAOU5
				Id=100

				AssemblyOccurrenceRelationshipSubstitution
				between #NAOU4 and #NAOU6
				Id=200

		NextAssemblyOccurrenceUsage #NAOU5 (placement#2 effectivity#2)	Occurrence of P5 Id=50	Classification 'alternative'
		NextAssemblyOccurrenceUsage #NAOU6 (placement#2 effectivity#2)	Occurrence of P6 Id=60	Classification 'alternative'

Only supported by 3DExp ("EBOM Substitute" without Part BOM relelationship) + Teamcenter (with Part BOM: BOMLine)

PDM Systems (Aras, Teamcenter, Windchill, 3DExp)						
PartMaster P1	Part(Version)	Part BOM	P2	BOM Instance 10 (placement#1 effectivity#1)		
					Substitute 1	Part(Version) P3
					Substitute 2	Part(Version) P4
				BOM Instance 40 (placement#2 effectivity#2)		
					Substitute 100	Part(Version) P5
					Substitute 200	Part(Version) P6



Case #2: as many Part Usages as Substitute Parts (Substitution may be defined bi-directional)
Here no FindNumbers (no NextAssemblyViewUsages)

Risk: the Part BOMs and the BOM Instances may be interpreted as if they were not substitutes!

AP242				
Part P1	PartVersion	NextAssemblyOccurrenceUsage #NAOU1 (placement#1 effectivity#1)	Occurrence of P2 Id=10	

				AssemblyOccurrenceRelationshipSubstitution
				between #NAOU1 and #NAOU2
				Id=1

				AssemblyOccurrenceRelationshipSubstitution
				between #NAOU1 and #NAOU3
				Id=2

		NextAssemblyOccurrenceUsage #NAOU2 (placement#1 effectivity#1)	Occurrence of P3 Id=20	Classification 'alternative'

				AssemblyOccurrenceRelationshipSubstitution
				between #NAOU2 and #NAOU1
				Id=1

		NextAssemblyOccurrenceUsage #NAOU3 (placement#1 effectivity#1)	Occurrence of P4 Id=30	

				AssemblyOccurrenceRelationshipSubstitution
				between #NAOU3 and #NAOU1
				Id=1

		NextAssemblyOccurrenceUsage #NAOU4 (placement#2 effectivity#2)	Occurrence of P2 Id=40	Classification 'alternative'

				AssemblyOccurrenceRelationshipSubstitution
				between #NAOU4 and #NAOU5
				Id=100

				AssemblyOccurrenceRelationshipSubstitution
				between #NAOU4 and #NAOU6
				Id=200

		NextAssemblyOccurrenceUsage #NAOU5 (placement#2 effectivity#2)	Occurrence of P5 Id=50	Classification 'alternative'

				AssemblyOccurrenceRelationshipSubstitution
				between #NAOU5 and #NAOU4
				Id=1

		NextAssemblyOccurrenceUsage #NAOU6 (placement#2 effectivity#2)	Occurrence of P6 Id=60	Classification 'alternative'

				AssemblyOccurrenceRelationshipSubstitution
				between #NAOU6 and #NAOU4
				Id=1

Only supported by 3DExp (EBOM Substitute without Part BOM) + Teamcenter (with Part BOM: BOMLine)

PDM Systems (Aras, Teamcenter, Windchill, 3DExp)						
PartMaster P1	Part(Version)	Part BOM	P2	BOM Instance 10 (placement#1 effectivity#1)		
					Substitute 1	Part(Version) P3
					Substitute 2	Part(Version) P4
		Part BOM	P3	BOM Instance 20 (placement#1 effectivity#1)		
					Substitute 1	Part(Version) P2
		Part BOM	P4	BOM Instance 30 (placement#1 effectivity#1)		
					Substitute 1	Part(Version) P2
		Part BOM Same Part BOM as above!	P2	BOM Instance 40 (placement#2 effectivity#2)		
					Substitute 100	Part(Version) P5
					Substitute 200	Part(Version) P6
		Part BOM	P5	BOM Instance 50 (placement#2 effectivity#2)		
					Substitute 1	Part(Version) P2
		Part BOM	P6	BOM Instance 60 (placement#2 effectivity#2)		
					Substitute 1	Part(Version) P2

Table 50: Overview of all Occurrence Usage Substitution mechanisms

It helps to understand in the following sections why it might be necessary to map between them (if the source or target PDM system does not support it), and why effectivities, 3D placements and properties should be the same for the base part and all its alternate/substitute parts.

7.5.1 Template "AlternatePartRelationship"

An AlternatePartRelationship specifies that any version of the alternate part, may be used in place of any version of the base part, regardless of its usages and of its versions. The relationship established by the AlternatePartRelationship is not symmetric: if part B is an alternate part for part A, part A is not implied to be an alternate part for part B.

Here are some additional semantical aspects defined in the AP242 ISO document:

NOTE 1 If a part is an alternate for another part, it is understood that there is no interest in keeping track of which part, the base or any alternates specified, is used as a particular instance of the base part within a part structure.

NOTE 2 An organization may track design changes for a base part and establish effectivity conditions for the use of that base part in various assemblies to be manufactured. The use of an alternate part implies that an organization does not specify any particular version of the alternate part nor establish effectivities relating to it.

NOTE 4 This concept usually refers to form, fit, function, and quality. Additional properties such as performance, noise, endurance, or reliability may also be considered as a prerequisite for the replacement

EXAMPLE Two bolts of the same size are parts. One bolt has a square head and the other has a hexagonal head. The two bolts are considered equivalent with respect to form, fit, and function: they both have sufficiently close physical shape, they take up the same space when used, and they both serve to fasten two things together. Thus, one of these two bolts could be considered to be an alternate part for the other bolt.

Preprocessor Recommendations:

- If the alternate relationship shall apply between two part versions, use template PartVersionRelationship with relationType ‘alternative’ (see section 5.1.5).
- The relating and the related Parts shall be different objects.
- The value of PartTypes for the base and alternate part shall be the same.
- If the relation is symmetric in the source PDM system, then a reverse-relationship shall state that it applies in both directions.
- If there are more than one alternate part involved, a ‘star’ structure from the base part to each alternate part shall be mapped (this star maps the semantic, that there is a base part and a number of alternate parts. All the alternate parts are on the same semantical level, but not on the same semantical level than the base part.
- Alternate parts shall have no revision effectivities and their usages shall have no occurrence effectivities, since the ones defined on the base part also apply to the alternate parts.
- The combination of the base part and the alternate part shall be unique.

Postprocessor Recommendations:

- If the target system does not support AlternatePartRelationship, but only PartVersionRelationship (see section 5.1.5), it shall duplicate it between all versions of the base Part and all versions of the alternate Part that are mentioned in the XML file and map each of them as PartVersionRelationship.
  If no part versions are mentioned, the newest part version shall be considered.

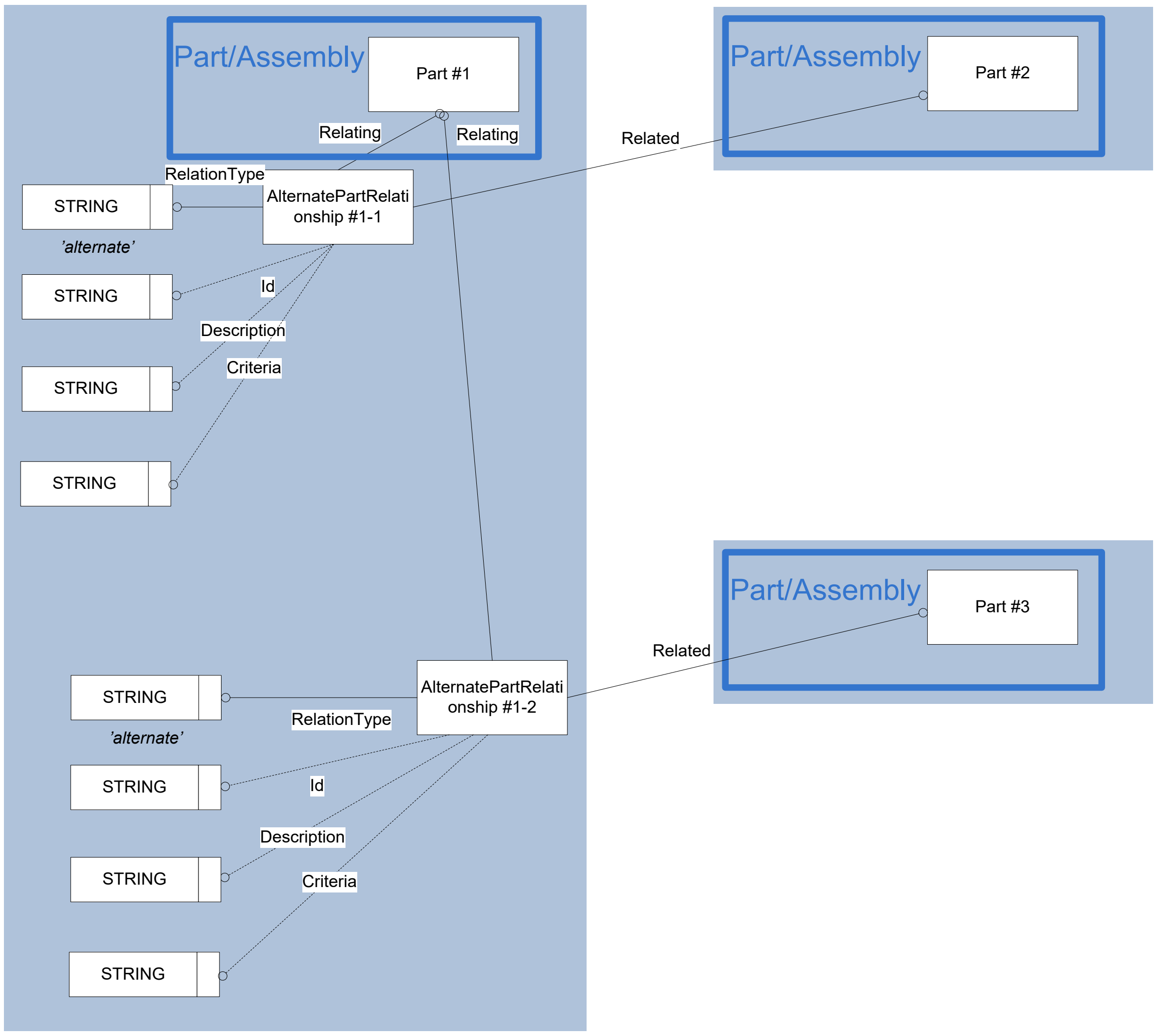


Figure 41: Template "AlternatePartRelationship"

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

```
<Part uid="p--000000001EAA8110">
…
      <Versions>
…
      </Versions>
      <PartRelationship uid="apr--1" xsi:type="n0:AlternatePartRelationship">
            <Related uidRef="p--0000000017D36B80"/>
            <RelationType>
                  <ClassString>alternate</ClassString>
            </RelationType>
            <Criteria>
                  <CharacterString>test</CharacterString>
            </Criteria>
      </PartRelationship>
</Part>

<Part uid="p--0000000017D36B80">
…
```

```
</Part>
```

Entity AlternatePartRelationship	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
Related	Part
RelationType	ClassSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
AnalysisAssignment	OPTIONAL SET[1:?] of AnalysisAssignment
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
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
Criteria	OPTIONAL MultiLingualStringSelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
FrozenAssignment	OPTIONAL SET[1:?] of FrozenAssignment
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment

Table 51: "AlternatePartRelationship" Attributes

Attribute recommendations

- ***Description***: the text or the set of texts that provide further information about the AlternatePartRelationship. The value of this attribute need not be specified. Use “Description” template.

- ***Id:*** stores the Identifier for the AlternatePartRelationship. The value of this attribute need not be specified. Use IdentifierString type.
- ***Related***: Reference to an alternate Part.
- ***RelationType***: the meaning of the relationship. Use ClassString type. Mandatory value: 'alternate'.
- ***Criteria:*** the word or set of words describing the requirements that are covered by both the base part and the alternate part and therefore are the basis for the statement regarding the capability of replacing the base part by the alternate par. Use "Description" template.
- ***PropertyValueAssignment:*** adds the value of a property to the AlternatePartRelationship. 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.

7.5.2 Template "AssemblyOccurrenceRelationshipSubstitution"

An AssemblyOccurrenceRelationshipSubstitution specifies that one NextAssemblyOccurrenceUsage may be substituted by another NextAssemblyOccurrenceUsage. This relationship is not symmetric: if the one NextAssemblyOccurrenceUsage is a substitution for another NextAssemblyOccurrenceUsage, this does not imply the other way around.

The subjects of the substitution are the related Occurrence objects (i.e. occurrence of parts) of both NextAssemblyOccurrenceUsage objects.

The relating AssemblyDefinition shall be the same in both NextAssemblyOccurrenceUsage objects.

Occurrence effectivities and the 3D placement associated to both NextAssem-blyOccurrenceUsage objects should be identical or the substitute NextAssemblyOccurren-ceUsage should have none of them.

In some PDM systems which relate the base NextAssemblyOccurrenceUsage directly to the substitute parts => there is only one NextAssemblyOccurrenceUsage and also no way to store properties on the substitute parts occurences. In such a case, the properties should be the same or there should be none of them.

An AssemblyOccurrenceRelationshipSubstitution for which the first NextAssemblyOccurrenceUsage refers to an occurrence of a part which is an assembly involves that the entire part structure of the substituted part may be used in place of the first part and its part structure.

This concept usually refers to form, fit, function, and quality. Additional properties such as performance, noise, endurance, or reliability may also be considered as a prerequisite for the replacement.

As defined in section 13.1, the assembly validation properties shall consider all the NextAssemblyOccurrenceUsages, no matter if some of them are defined as a substitution or if Effectivities apply to them.

Preprocessor Recommendations:

- The **relating** and the **related** NextAssemblyOccurrenceUsage shall be different objects.

- For better interpretation by the postprocessor, which Part occurrence is the main one and which ones are the alternatives, it is recommended to set NextAssemblyOccurrenceUsage.ClassifiedAs to 'alternative' for the occurrence referenced by AssemblyOccurrenceRelationshipSubstitution.Relating.
- If the relation is symmetric in the source PDM system, then a reverse-relationship shall state that it applies in both directions.
  In this case, NextAssemblyOccurrenceUsage ClassifiedAs='alternative' shall be set only in one direction.
- If there are more than one substitute part involved, a 'star' structure from the base part to each substitute part shall be mapped (this star maps the semantic, that there is a base part and a number of substitute parts. All the substitute parts are on the same semantical level, but not on the same semantical level than the base part.
- The combination of the base Part and the substitute Part shall be unique.
- The substitute Part Ocurrences may have a different 3D placement, but this may not be supported by all target systems.

Postprocessor Recommendation:

- If the target system does not support AssemblyOccurrenceRelationshipSubstitution, but only AssemblyViewRelationshipSubstitution (see chapter 7.5.3), it shall create a usage substitution relationship for all occurrences having the same occurrence substitution rule and do so for each different substitution rule.
- If the target system supports neither AssemblyViewRelationshipSubstitution nor AssemblyOccurrenceRelationshipSubstitution, the alternative NextAssemblyOccurrenceUsages (those having ClassifiedAs='alternative' and/or being referenced by AssemblyOccurrenceRelationshipSubstitution.Related) shall not be interpretated in addition to each other; otherwise, an assembly with too many components would be imported.
- If the 3D placement of the alternative Part Occurrences are different, and this is not supported by the target system, an error should be returned.

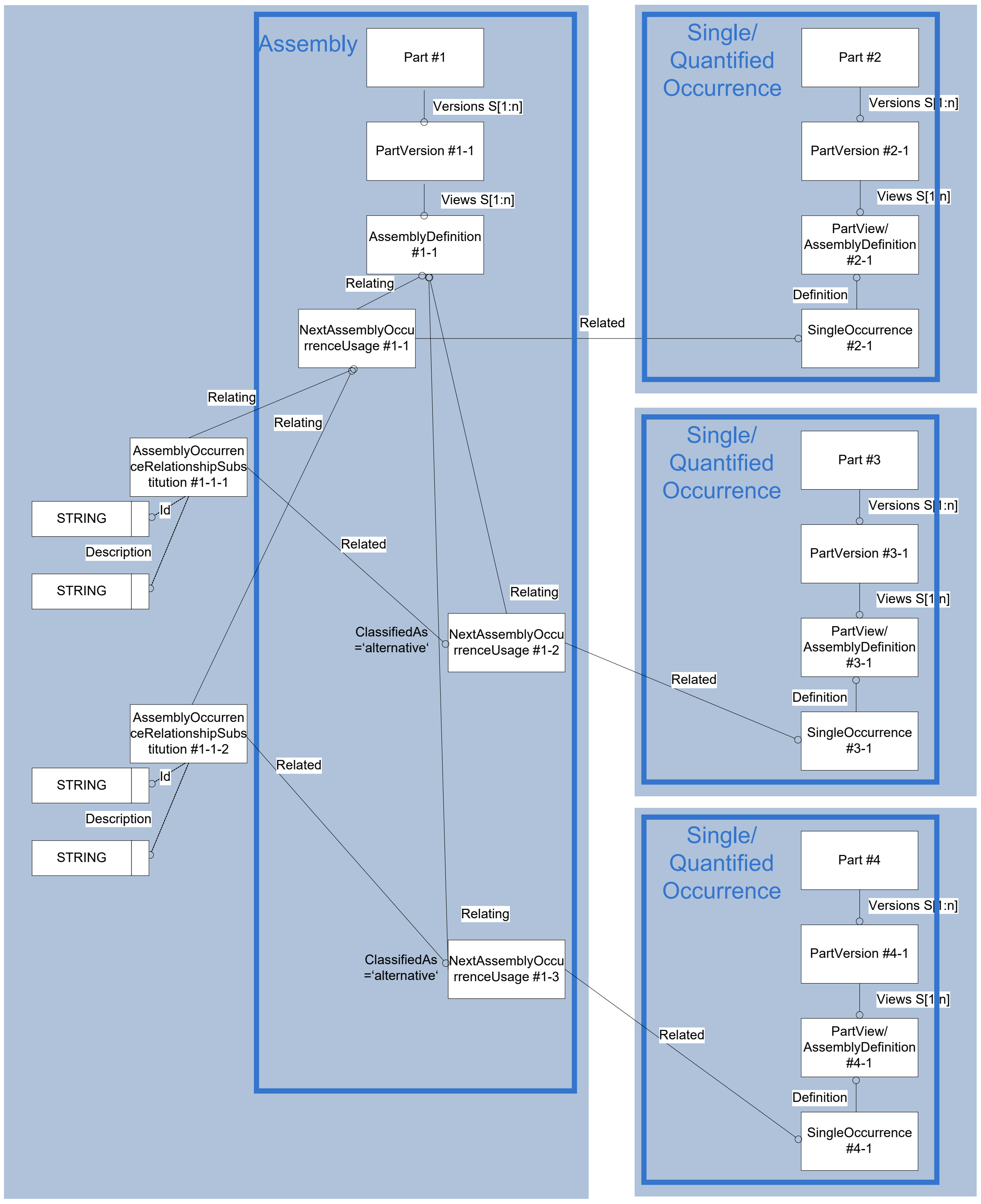


Figure 42: Template "AssemblyOccurrenceRelationshipSubstitution"

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

```
<Classification uid="pcs--alt">
      <Class>
            <ClassString>alternative</ClassString>
      </Class>
      <Role>alternative</Role>
</Classification>

<Part uid="p--0000000017D374A0">
…
  <Versions>
    <PartVersion uid="pv--0000000017D374A0--id1">
…
      <Views>
        <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--
0000000017D374A0--id1">
…
            <ViewOccurrenceRelationship uid="pvvid--0000024E5E5F3870--1987"
xsi:type="bom:NextAssemblyOccurrenceUsage">
…
                  <AssemblyOccurrenceRelationshipSubstitution uid="aors--
0000024E5E5F3870--1987-1">
                        <Id id="10"/>
                        <Related uidRef="pvvid--0000024E5E5F3870-2--1987"/>
                  </AssemblyOccurrenceRelationshipSubstitution>
                  <AssemblyOccurrenceRelationshipSubstitution uid="aors--
0000024E5E5F3870--1987-2">
                        <Id id="20"/>
                        <Related uidRef="pvvid--0000024E5E5F3870-3--1987"/>
                  </AssemblyOccurrenceRelationshipSubstitution>
            </ViewOccurrenceRelationship>
            <ViewOccurrenceRelationship uid="pvvid--0000024E5E5F3870-2--1987"
xsi:type="bom:NextAssemblyOccurrenceUsage">
                  <ClassifiedAs>
                        <Classification uidRef="pcs--alt"/>
                  </ClassifiedAs>
…
                  <AssemblyOccurrenceRelationshipSubstitution uid="aors--
0000024E5E5F3870-2--1987-1">
                        <Id id="101"/>
                        <Related uidRef="pvvid--0000024E5E5F3870--1987"/>
                  </AssemblyOccurrenceRelationshipSubstitution>
            </ViewOccurrenceRelationship>
            <ViewOccurrenceRelationship uid="pvvid--0000024E5E5F3870-3--1987"
xsi:type="bom:NextAssemblyOccurrenceUsage">
                  <ClassifiedAs>
                        <Classification uidRef="pcs--alt"/>
                  </ClassifiedAs>
…
                  <AssemblyOccurrenceRelationshipSubstitution uid="aors--
0000024E5E5F3870-3--1987-1">
                        <Id id="201"/>
                        <Related uidRef="pvvid--0000024E5E5F3870--1987"/>
                  </AssemblyOccurrenceRelationshipSubstitution>
            </ViewOccurrenceRelationship>
…
          <AssemblyType>
            <ClassString>design assembly</ClassString>
          </AssemblyType>
        </PartView>
```

```
         </Views>
…
      </PartVersion>
   </Versions>
</Part>
```

Entity AssemblyOccurrenceRelationshipSub-stitution	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
Related	NextAssemblyOccurrenceUsage
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssign-ment
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 InformationUs-ageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssignment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPer-sonInOrganizationAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssign-ment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassifica-tionAssignment
StateAssignment	OPTIONAL SET[1:?] of StateAssignment
StateDefinitionAssignment	OPTIONAL SET[1:?] of StateDefinitionAssignment
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 52: "AssemblyOccurrenceRelationshipSubstitution" Attributes

Attribute recommendations

- ***Description***: the text or the set of texts that provide further information about the AssemblyOccurrenceRelationshipSubstitution. The value of this attribute need not be specified. Use “Description” template.

- ***Id:*** stores the Identifier for the AssemblyOccurrenceRelationshipSubstitution. The value of this attribute need not be specified. Use IdentifierString type.
- ***Related***: Reference to a substitution NextAssemblyOccurrenceUsage.
- ***PropertyValueAssignment:*** adds the value of a property to the AssemblyOccurrenceRelationshipSubstitution. 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.

7.5.3 Template "AssemblyViewRelationshipSubstitution"

An AssemblyViewRelationshipSubstitution specifies that one PartViewRelationship may be substituted by another PartViewRelationship. This relationship is not symmetric: if the one PartViewRelationship is a substitution for another PartViewRelationship, this does not impliy the other way around.

The subjects of the substitution are the related View objects (i.e. applying to the usage (i.e. all occurrences) of a component part in a given assembly part) of both PartViewRelationship objects.

The relating AssemblyDefinition shall be the same in both PartViewRelationship objects.

View Effectivities associated to both PartViewRelationship objects should be identical or the substitute PartViewRelationship should have none of them.

In some PDM systems which relate the base NextAssemblyViewUsage directly to the substitute parts => there is only one NextAssemblyViewUsage and also no way to store properties on the substitute parts views. In such a case, the properties should be the same or there should be none of them.

An AssemblyViewRelationshipSubstitution for which the first PartViewRelationship refers to a usage of a part which is an assembly involves that the entire part structure of the substituted part may be used in place of the first part and its part structure.

This concept usually refers to form, fit, function, and quality. Additional properties such as performance, noise, endurance, or reliability may also be considered as a prerequisite for the replacement.

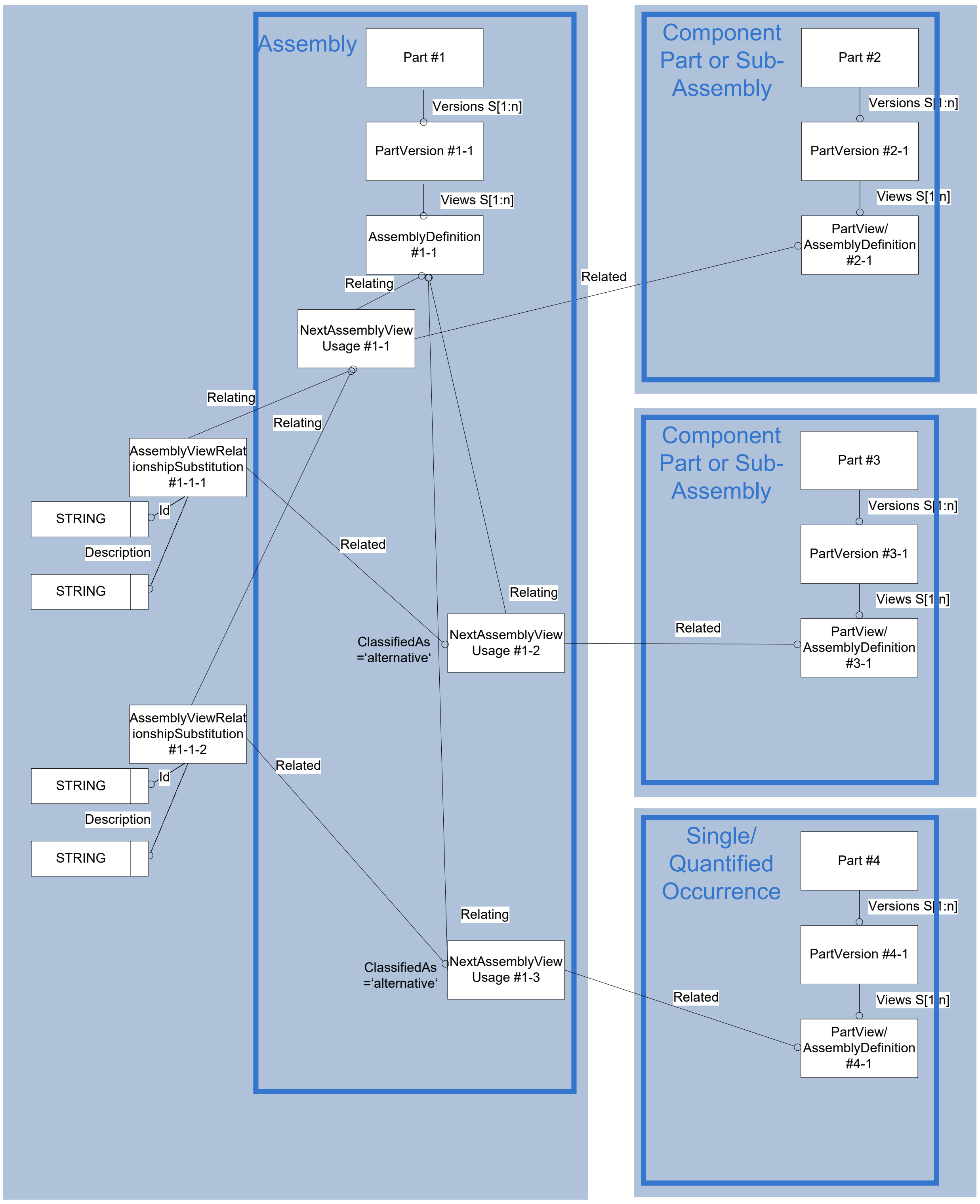

Assembly
Part #1
Versions S[1:n]
PartVersion #1-1
Views S[1:n]
AssemblyDefinition #1-1
Relating
NextAssemblyView Usage #1-1
Related
Relating
Relating
AssemblyViewRelationshipSubstitution #1-1-1
STRING
Id
Description
STRING
Related
Relating
ClassifiedAs ='alternative'
NextAssemblyView Usage #1-2
AssemblyViewRelationshipSubstitution #1-1-2
STRING
Id
Description
STRING
Related
Relating
ClassifiedAs ='alternative'
NextAssemblyView Usage #1-3
Component Part or Sub-Assembly
Part #2
Versions S[1:n]
PartVersion #2-1
Views S[1:n]
PartView/ AssemblyDefinition #2-1
Component Part or Sub-Assembly
Part #3
Versions S[1:n]
PartVersion #3-1
Views S[1:n]
Related
PartView/ AssemblyDefinition #3-1
Single/ Quantified Occurrence
Part #4
Versions S[1:n]
PartVersion #4-1
Related
Views S[1:n]
PartView/ AssemblyDefinition #4-1

Figure 43: Template "AssemblyViewRelationshipSubstitution"

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

```
<Classification uid="pcs--alt">
      <Class>
            <ClassString>alternative</ClassString>
      </Class>
      <Role>alternative</Role>
</Classification>

<Part uid="p--0000000017D374A0">
…
  <Versions>
    <PartVersion uid="pv--0000000017D374A0--id1">
…
      <Views>
        <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--
0000000017D374A0--id1">
…
            <PartViewRelationship uid="pvrid--0000024E5E5F3870--1995"
xsi:type="bom:NextAssemblyViewUsage">
…
                  <AssemblyViewRelationshipSubstitution uid="avrs--
0000024E5E5F3870--1995-1">
                        <Id id="1"/>
                        <Related uidRef="pvrid--0000024E5E5F3870-2--1995"/>
                  </AssemblyViewRelationshipSubstitution>
                  <AssemblyViewRelationshipSubstitution uid="avrs--
0000024E5E5F3870--1995-2">
                        <Id id="2"/>
                        <Related uidRef="pvrid--0000024E5E5F3870-3--1995"/>
                  </AssemblyViewRelationshipSubstitution>
…
            </PartViewRelationship>
            <PartViewRelationship uid="pvrid--0000024E5E5F3870-2--1995"
xsi:type="bom:NextAssemblyViewUsage">
                  <ClassifiedAs>
                        <Classification uidRef="pcs--alt"/>
                  </ClassifiedAs>
…
                  <AssemblyViewRelationshipSubstitution uid="avrs--
0000024E5E5F3870--1995-3">
                        <Id id="11"/>
                        <Related uidRef="pvrid--0000024E5E5F3870--1995"/>
                  </AssemblyViewRelationshipSubstitution>
…
            </PartViewRelationship>
            <PartViewRelationship uid="pvrid--0000024E5E5F3870-3--1995"
xsi:type="bom:NextAssemblyViewUsage">
                  <ClassifiedAs>
                        <Classification uidRef="pcs--alt"/>
                  </ClassifiedAs>
…
                  <AssemblyViewRelationshipSubstitution uid="avrs--
0000024E5E5F3870--1995-4">
                        <Id id="21"/>
                        <Related uidRef="pvrid--0000024E5E5F3870--1995"/>
                  </AssemblyViewRelationshipSubstitution>
…
            </PartViewRelationship>
…
          <AssemblyType>
```

```
            <ClassString>design assembly</ClassString>
          </AssemblyType>
        </PartView>
      </Views>
…
    </PartVersion>
  </Versions>
</Part>
```

Entity AssemblyViewRelationshipSubstitution	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
Related	NextAssemblyViewUsage
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
AnalysisAssignment	OPTIONAL SET[1:?] of AnalysisAssignment
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
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefinitionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssignment
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassificationAssignment
StateAssignment	OPTIONAL SET[1:?] of StateAssignment
StateDefinitionAssignment	OPTIONAL SET[1:?] of StateDefinitionAssignment
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment

Table 53: "AssemblyViewRelationshipSubstitution" Attributes

Attribute recommendations

- ***Description***: the text or the set of texts that provide further information about the AssemblyViewRelationshipSubstitution. The value of this attribute need not be specified. Use “Description” template.
- ***Id:*** stores the Identifier for the AssemblyViewRelationshipSubstitution. The value of this attribute need not be specified. Use IdentifierString type.
- ***Related***: Reference to a substitution NextAssemblyViewUsage. For more details, refer to Section 7.4.
- ***PropertyValueAssignment:*** adds the value of a property to the AssemblyViewRelationshipSubstitution. Use “PropertyAssignment” template (see 6.2)
- Other attributes than these are not covered by this document; their use is discouraged as it would depend on mutual agreements between data exchange partners.

Preprocessor Recommendation:

- The relating and the related PartViewRelationships shall be different objects.
- For better interpretation by the postprocessor, which Part View Usage is the main one and which ones are the alternatives, it is recommended to set NextAssemblyViewUsage.ClassifiedAs to ‘alternative’ for the View Usage referenced by AssemblyViewRelationshipSubstitution.Relating.
- If the relation is symmetric in the source PDM system, then a reverse-relationship shall state that it applies in both directions.
  In this case, NextAssemblyViewUsage ClassifiedAs=’alternative’ shall be set only in one direction.
- If there are more than one substitute part involved, a ‘star’ structure from the base part to each substitute part shall be mapped (this star maps the semantic, that there is a base part and a number of substitute parts. All the substitute parts are on the same semantical level, but not on the same semantical level than the base part.
- The combination of the base Part and the substitute Part shall be unique within the direct childs of an assembly node.
- The NextAssemblyViewUsage.RelationType shall be set to ‘next assembly view’ (see section 7.4).
- The substitute Part View Usages may have a different quantity, but this may not be supported by all target systems.

As defined in section 13.1, the assembly validation properties shall not consider the NextAssemblyViewUsages since they have no placement. Only the NextAssemblyOccurrenceUsages are relevant to compute them.

As defined in section 7.4, the same assembly structure shall not be defined redundantly using NextAssemblyOccurrenceUsages and NextAssemblyViewUsage, but the use of NextAssemblyViewUsages is necessary to map AssemblyViewRelationshipSubstitutions. There is no obligation to define NextAssemblyOccurrenceUsages for each NextAssemblyViewUsage, neither for the base part nor for the substitute part of a AssemblyViewRelationshipSubstitution.

Postprocessor Recommendation:

- If the target system does not support AssemblyViewRelationshipSubstitution, it shall duplicate it between all occurrences of the base Part usage and distinct occurrences of the substitute Part usage that are mentioned in the XML file and map each of them as AssemblyOccurrenceRelationshipSubstitution (see chapter 7.5.2).

- If the target system supports neither AssemblyViewRelationshipSubstitution nor AssemblyOccurrenceRelationshipSubstitution, the alternative NextAssemblyViewUsages (those having ClassifiedAs='alternative' and/or being referenced by AssemblyViewRelationshipSubstitution.Related) shall not be interpreted in addition to each other; otherwise, an assembly with too many components would be imported.
- If the quantity or the 3D placement of the alternative Part View Usages are different, and this is not supported by the target system, an error should be returned.

7.6 Template "OccurrenceRelationship"

An `OccurrenceRelationship` is a relationship between two `Occurrence` objects.

Entity OccurrenceRelationship	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
Related	Occurrence
RelationType	ClassSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
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 54: "OccurrenceRelationship" 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 `OccurrenceRelationship`. The value of this attribute need not be specified. Use IdentifierString type.
- ***Related***: Reference to an `Occurrence` that is the consumed object in the relationship (example AsDesigned, and the `Relating` is the AsPlanned)
- ***RelationType***: the meaning of the relationship. Use ClassString type. Mandatory value: ‘tracking’ (see chapter 15).
- 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 Domain Model XML syntax)

Example with XML code of OccurrenceRelationship

```
<PartView uid="PV__39">
  <InitialContext uidRef="VC__9"/>
  <Occurrence uid="SO__46" xsi:type="n0:SingleOccurrence">
    <Id id="PROPELLER.1"/>
    <OccurrenceRelationship uid="PVR__600387">
      <Description>
        <CharacterString>Tracking objects usage</CharacterString>
      </Description>
      <Related uidRef="SO__229"/>
      <RelationType>
        <ClassString>tracking</ClassString>
      </RelationType>
    </OccurrenceRelationship>
  </Occurrence>
 ...
```

7.7 Template "Mirrored Part"

A mirrored part is an opposite-hand part derived from an original part. This section describes how to keep the derivation relationship (so-called associativity), so that when the original part gets changed, the changes can be detected and overtaken on the opposite-hand part.

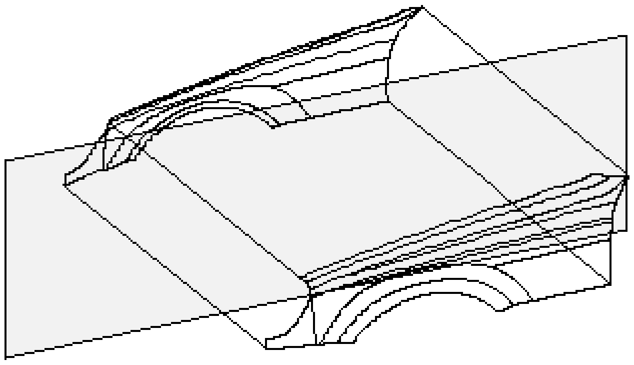


Figure 44: Example of mirrored part

According to the PDM Schema Usage Guide: mirroring (and scaling) is forbidden along the assembly structure (transformation matrix determinant shall be always +1), because in a bill of materials, any Part modified this way can never be the same Part as the original (try mounting a left-side fender on the right side of a car…).

„fully" mirrored Parts may be mapped as

1. two CAD files. In this case, the „derived CAD file" references the geometry of the „original CAD file" (multi model link or similar). This is the most frequent case, for example supported by 3DExperience.
2. one CAD file. In this case the opposite-hand part references the common geometry together with a mirroring matrix

Out of scope:

- „partially" mirrored Parts are mapped as tho disctinct CAD files where all or a portion of the geometry of the original CAD file is referenced from the „derived CAD file", while some further geometry is specific to the original/derived CAD file. Example: the left side fender is mirrored as right side fender and the gas cap is added to the right side geometry.
- Mirrored parts may be non-associative to the original parts. In this case, they have been derived once and stored as an own part like a standalone copy. Changes to the original part have no effect on the copy. Such parts shall be exchanged as normal parts, this section is not relevant for them.
- Some systems support the mirroring of whole assemblies, but currently this is not allowed at most companies.
- Supply package may be imported as a single part with multiple geometries => the handling of mirrored parts within such a supply package is not in scope.

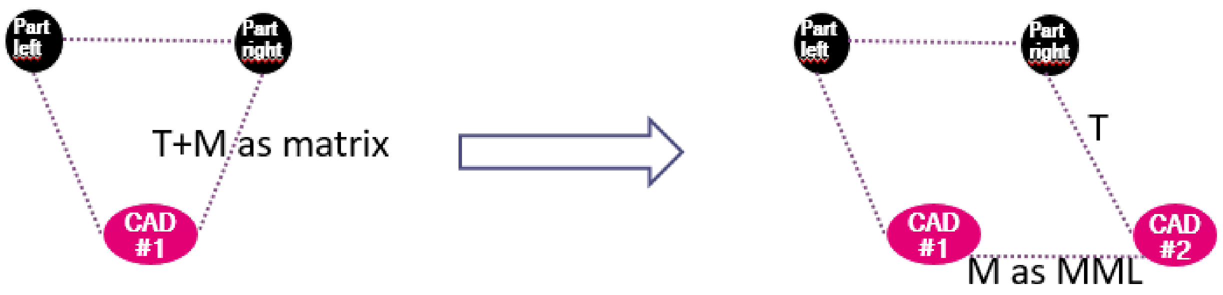
If the original Part is changed (new revision with new CAD model), the mirrored Part also has to be updated (new revision).

Preprocessor Recommendations:

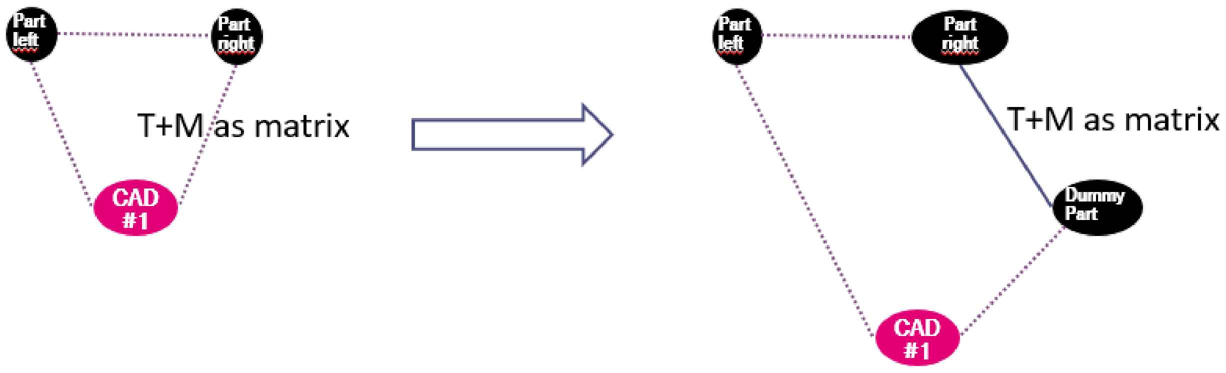
- If the original Part is exchanged, exchanging the mirrored Part is not necessary
- If the mirrored Part is exchanged, exchanging the original Part is necessary

Postprocessor Recommendations:

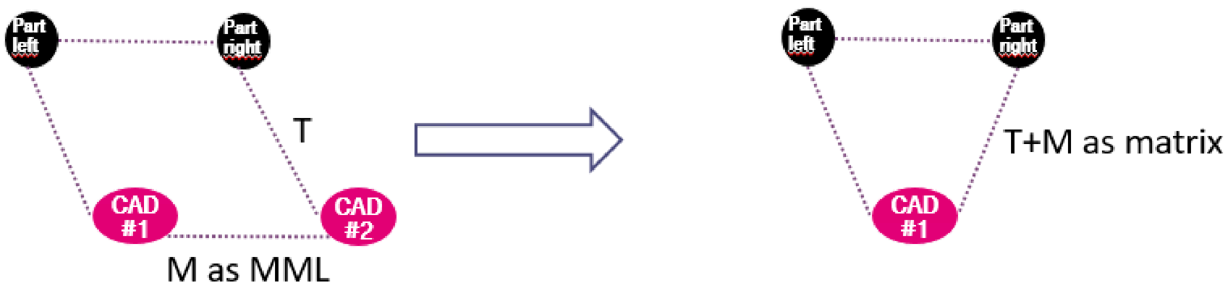
- If some target systems only allow mirroring using distinct CAD models, a mirroring using a common CAD model shall be transformed during import: the second CAD model shall be generated out of the incoming CAD model and mirroring matrix, maintaining the associativity via a multi model link.



- Another possibility is to create a dummy part and mirror it within the right Part (which become a pseudo assembly). The dummy part is not a BoM Part (otherwise a mirroring matrix would not be allowed).

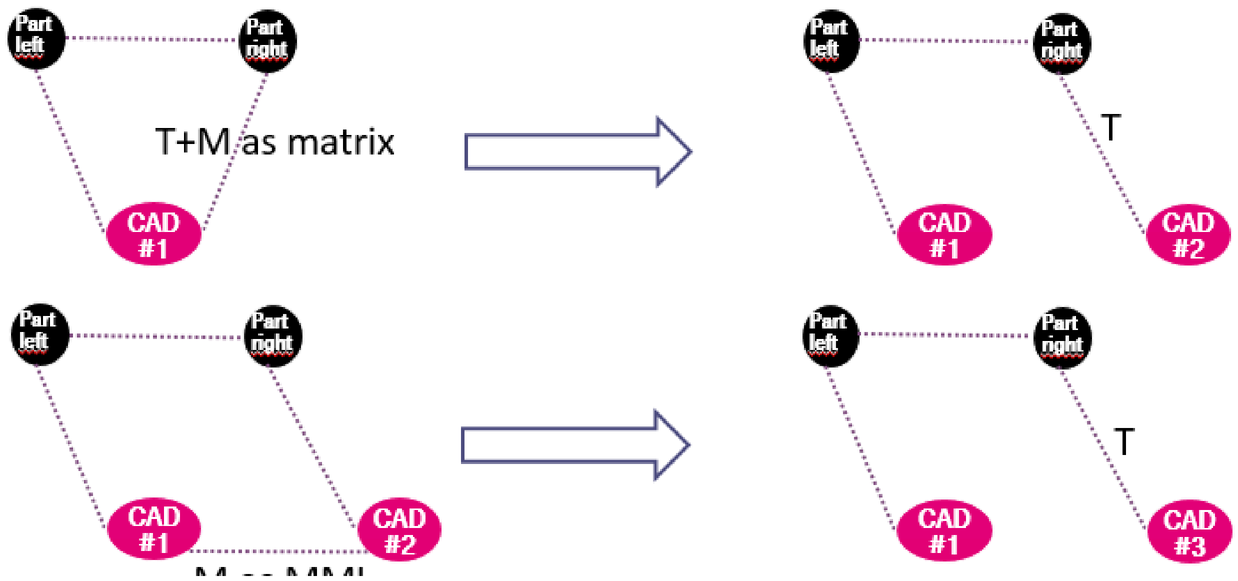


- If some target systems only allow mirroring using a common CAD model, a mirroring using distinct CAD models shall be transformed during import: the multi model link information shall be extracted from the mirrored CAD model and mapped as mirroring matrix between the opposite-hand part and the geometry of the original part. Remark: doing so, since the mirrored CAD model gets discarded, any further information placed in the mirrored CAD model will get lost.



Pre-/Postprocessor Recommendations:

- If the source company does not wish to propagate the associativity to the partner company, or of the target system does not support this associativity, a non-associative mirrored copy of the geometry shall be created during export or during import.



7.7.1 Mirroring using distinct CAD models

The CAD level mirroring (via multi model link or similar) is mapped within the geometry files and not on Domain Model level, however it is recommended to map the fact that the opposite-hand part is derived from an original part as a PartVersionRelationship having RelationType='mirrored' (see section 5.1.5). Doing so, changes in the geometry of the original part can be propagated within the mirrored part, for example by loading the mirrored part into the CAD system (this will fetch the geometry changes of the original part) and storing it under a new PartVersion.

7.7.2 Mirroring using a common CAD model

Like for the assembly structure (see section 7.3), in addition to the usual (geometrical) way of mapping the 3D positioning, a more compact (especially in XML) and simple way has been defined in AP242. This section describes both possible ways.

A JIRA issue TCSC410303-659 has been created to correct the definition of GeometricalRelationship.Placement.

Entity GeometricalRelationship	Attribute type (additionaly to PartViewRelationship)
Placement	TransformationSelect

Table 55: "GeometricalRelationship" Attributes

Attribute recommendations

- ***Relating:*** the PartView of the mirrored part.

- ***Related:*** the PartView of the original part.
- ***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). Where applicable, the following values shall be used:

RelationType	
'mirrored'	The business object defines a relationship where the relating PartView is mirrored from the related (original) PartView

- ***Placement:***.the CartesianTransformation or the GeometricRepresentationRelationshipWithCartesianTransformation that defines a mirroring transformation consisting of rotation and translation to be applied to the relating PartView in order to define the location and the orientation within the context of the related PartView. Translation, rotation, and mirroring, i.e., inversion, is included; scaling is not included.
- 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:

- Even if the mirroring is defined via a symmetry plane within the source system, not the symmetry plane shall be exchanged, but the corresponding mirroring matrix + translation placement.

Postprocessor Recommendations:

- Rather than importing the mirroring matrix, it might be imported as a symmetry plane within the target system, if this is the internal way to manage mirroring.

7.7.2.1 Template "Simplified Mirroring Positioning Representation"

The only instance needed here is a CartesianTransformation.

Preprocessor Recommendations:

- Unlike for the full positioning representations, the use of a GeometricCoordinateSpace is not necessary here.
- 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.
- Since mirroring of assemblies is out of scope, the original and the mirrored parts shall not be of type AssemblyDefinition.

Postprocessor Recommendations:

- None.

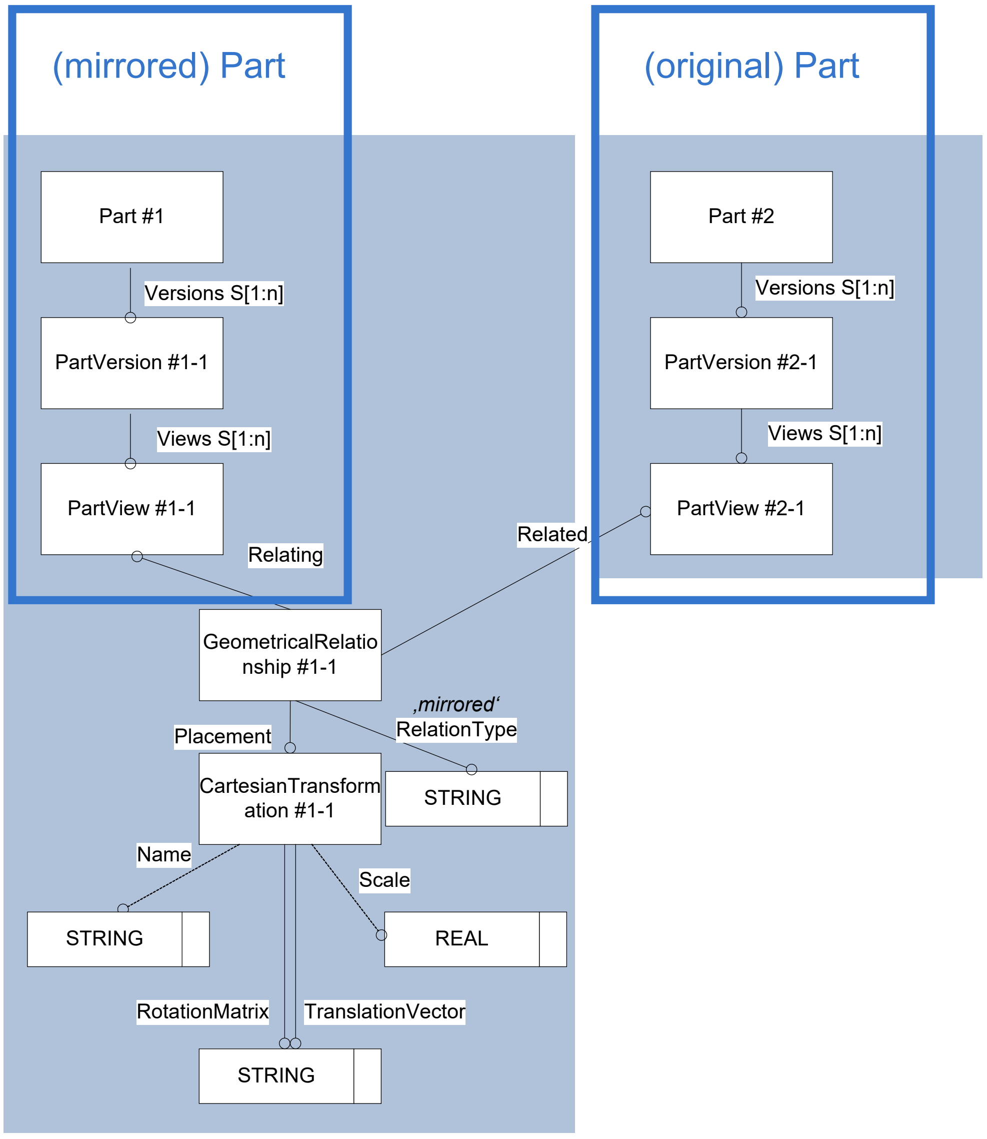


Figure 45: Template "Simplified Mirroring Positioning Representation"

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

T.B.S.

7.7.2.2 Template "Full Mirroring Positioning Representation"

In all these mapping alternative, the original and the mirrored parts are associated to a subtype of GeometricModel and the 3D Positioning information is mapped in a subtype of RepresentationRelationship that references both GeometricModels.

See section 7.3.2 for the recommendation details. Here only the use of GeometricRepresentationRelationshipWithCartesianTransformation is recommended.

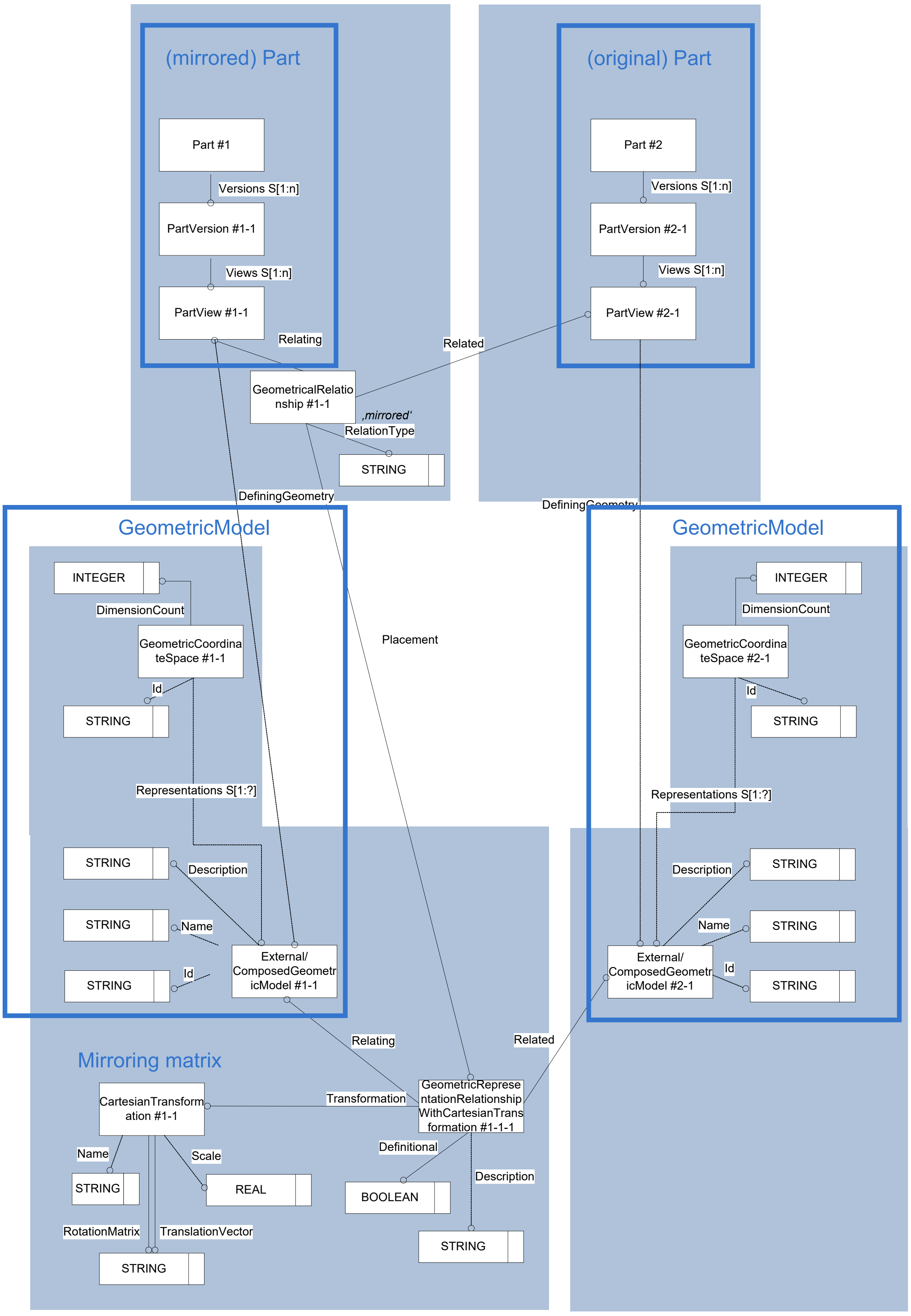


Figure 46: Template "Full Mirroring Positioning Representation"

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

T.B.S.

7.8 Template "Flexible Part"

Flexible Parts are parts that are deformed in the assembly process, such as an O-ring, a rubber gasket, a radiator hose or an electric wire.

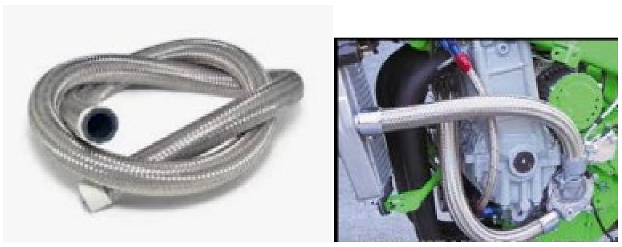


Figure 47: Example of a Flexible Part

According to the CAx-IF Recommended Practices for Alternative Shapes, Flexible Parts are one of the use cases for having Alternative Instance Shapes in an Assembly.

Flexible Parts may be alternatively mapped as

1. Several Parts. In this case, the Parts that define each alternative shape shall have a reference to the non-installed Part.
2. One part. In this case, each Occurrence shall have its own GeometricModel.

In both cases, each installation (with its geometry) has to be uniquely identified in the PDM system.

Out of scope:

- Definition of *assembly-level geometrical features* which modify the shape of the subsequent parts. A typical example is an assembly of two solid plates with a hole through both of them defined at the assembly level. This means the hole does not exist in the original part definition and it may be in different relative locations on the plate for each of the two usages.
- Definition of moving assemblies (no matter if containing flexible parts or not) having a different placement of their assembly components depending on the instance installation.

If the geometry of the non-installed Part is changed (new revision with new CAD model),

- In case of 1., a new revision of all the alternative shape Parts shall be created
- In case of 2., the GeometricModels that define each alternative shape also have to be updated.

Preprocessor Recommendations:

- In case of 1.:
  - If the non-installed Part is exported (new of updated), all the alternative shape Parts relevant for the partner shall also be exported (with their derivation relationship
  - If an Assembly using an alternative shape Part is exported, the non-installed Part shall also be exported in order to keep the derivation relationship between them.

- In case of 2.:
  - Since all alternative shapes are stored on the same part, all the alternative Geometric-Models that are relevant for the partner shall also be exported together with the part.

Postprocessor Recommendations:

- Mapping of 2. to 1.: if some target systems only allow flexible parts using several parts, the GeometricModels defined on occurrence level shall be transformed to distinct parts during import. If the transformation matrices are defined on RepresentationRelationship level, they shall be mapped instead on NextAssemblyOccurrenceUsage level. If Speci-fiedOccurrences are involved, the transformation matrices defined on NAOU level be-tween the root assembly of the SpecifiedOccurrence path and the direct parent assembly have to be considered while computing the transformation matrix on NAOU to the direct parent assembly.
- Mapping from 1. to 2.: if some target systems only allow flexible parts on occurrence level, all the Parts containing an alternative shape shall be transformed to Geometric-Models on Single/SpecifiedOccurrence level. Here the RepresentationRelationship-Placements may be derived from the NextAssemblyOccurrenceUsage.Placements.

7.8.1 Flexible Parts mapped as distinct Parts

Even if no associativity (like for mirrored parts, where updating the geometry of the original part causes the geometry of the mirrored part to be automatically updated) is supported between the non-installed part and the installed parts, it is recommended to map the fact that the installed part is derived from the non-installed part as a PartVersionRelationship having RelationType='al-ternate shape' (see section 5.1.5). Doing so, changes in the geometry of the non-installed part can be detected and an action initiated to overtake this change within each installed part.

Preprocessor Recommendations:

- The uninstalled part may be built into an assembly via NextAssemblyViewUsage (since hav-ing no 3D positioning, for logistic BoM purpose) but shall not be built in any assembly via NextAssemblyOccurrenceUsage (for DMU purpose).
- For certain Users, the Part.Id of the flexible parts may be derived from the Part.Id on the non-installed part.

7.8.2 Flexible Parts mapped as GeometricModels on Occurrence level

According to the section 7.1: If the transformation matrix of the alternate shape is defined within the <u>next</u> parent assembly, each alternate shape shall be mapped to SingleOccurrence.Defin-ingGeometry.

Additionally, it is recommended to map the fact that the installed geometry is derived from the non-installed geometry as a GeneralGeometricRepresentationRelationship having Relation-Type='alternate shape' (see section 7.3.3). Doing so, changes in the non-installed geometry can be detected and an action initiated to overtake this change within each installed geometry.

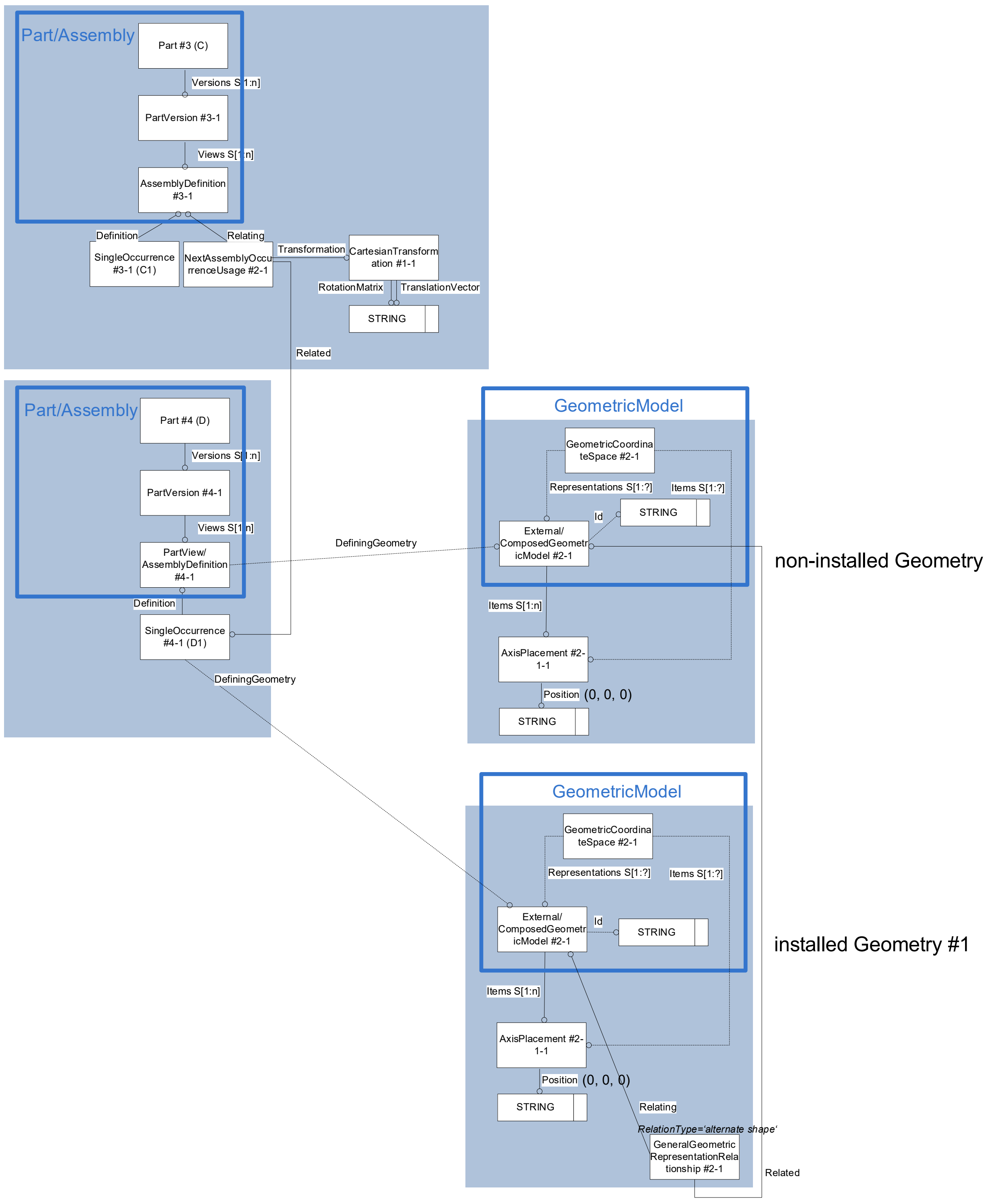


Figure 48: Example of alternative instance shapes based on SingleOccurrences

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

T.B.S.

According to the section 7.2: if the transformation matrix of the alternate shape is defined within an assembly over multiple assembly structure levels, the alternate shape shall be mapped to SpecifiedOccurrence.DefiningGeometry

In this case, according to the Full Positioning Representation defined in section 7.3.2:

- The assembly that builds the SpecifiedOccurrence and the SpecificedOccurrence itself both are associated to a subtype of GeometricModel and the 3D Positioning information is mapped in a subtype of RepresentationRelationship that relates both GeometricModels.
    - The GeometricModel that references the alternative shape (via SpecifiedOccurrence.DefiningGeometry) shall be an ExternalGeometricModel.
    - The GeometricModel that is associated to the AssemblyDefinition via DefiningGeometry shall be a ComposedGeometricModel.
    - Each alternative shape within the assembly shall be mapped as a GeometricRepresentationRelationshipWithPlacementTransformation or a GeometricRepresentationRelationshipWithCartesianTransformation (not GeometricRepresentationRelationshipWithSameCoordinateSpace).
    - RepresentationRelationship.Relating shall reference the ComposedGeometricModel
    - RepresentationRelationship.Related shall reference the ExternalGeometricModel

- Even if GeometricRepresentationRelationshipWithPlacementTransformation is used, no mirroring matrix shall be given.
- According to section 11.2, the non-installed geometry shall be mapped as Document with DocumentTypes='primary geometry'.
- According to section 16.1, the non-installed geometry may be also optionally mapped using PartView.DefiningGeometry.

When viewing the assembly in a CAD viewer, the GeometricModel defined in SingleOccurrence/SpecifiedOccurrence.DefiningGeometry 'overwrites' the one defined on PartView level.

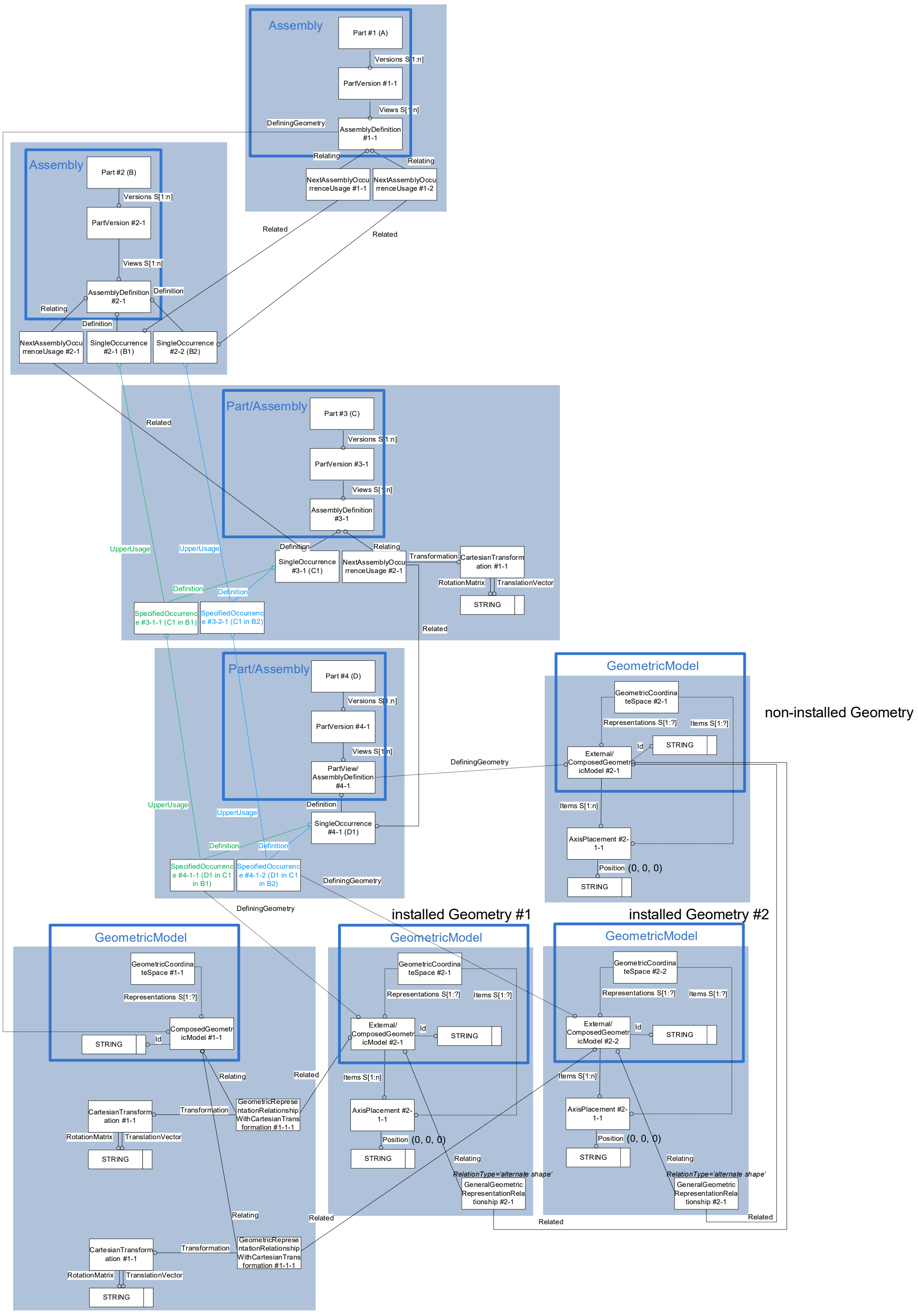


Figure 49: Example of alternative instance shapes based on SpecifiedOccurrences

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

T.B.S.

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 10303-4442 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 Domain 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 10303-4442 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 Domain 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 Domain Model of three structurally distinct data types as also shown in Figure 50:

- 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 50 are root elements and are, thus, outside of this containment block.

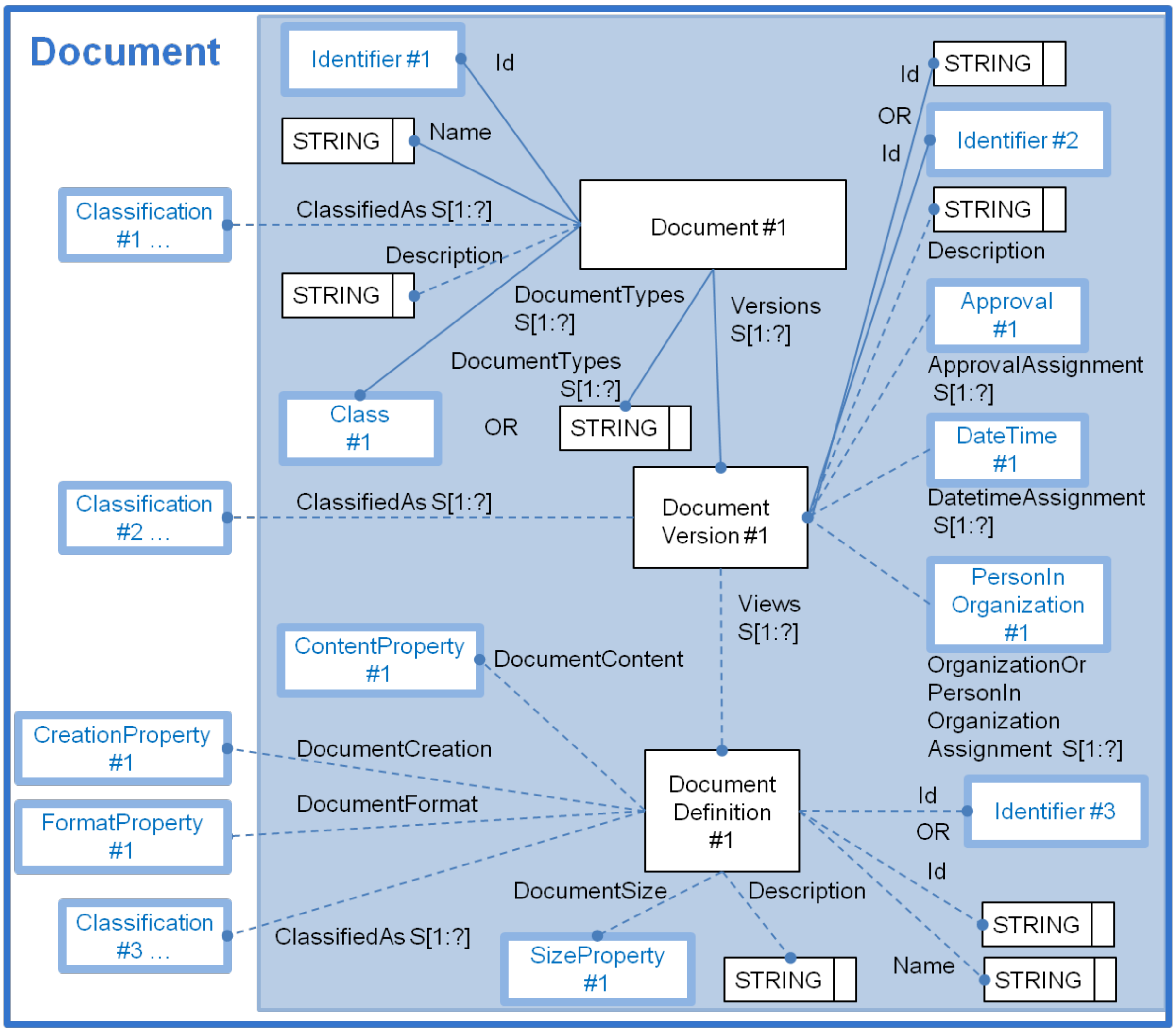


Figure 50: Template "Document"

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 50.

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

```
<Document uid="doc--000000001EAAE870">
  <DocumentTypes>
    <ClassString>geometry</ClassString>
  </DocumentTypes>
  <Id>
    <Identifier uid="docid--000000001EAAE870--id7" id="nut" idRoleRef="rl--
ii" 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:Digi-
talDocumentDefinition">
          <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
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
SameAs	OPTIONAL SET[1:?] of Proxy
Versions	SET[1:?] of DocumentVersion
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssign-ment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
CertificationAssignment	OPTIONAL SET[1:?] of CertificationAssignment
ContractAssignment	OPTIONAL SET[1:?] of ContractAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssign-ment
DatetimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment

Entity Document attributes	Attribute type
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 InformationUs-ageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssign-ment
OrganizationOrPersonInOrganizationAssignment	OPTIONAL SET[1:?] of OrganizationOrPer-sonInOrganizationAssignment
ProjectAssignment	OPTIONAL SET[1:?] of ProjectAssignment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefini-tionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssign-ment
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment
SecurityClassificationAssignment	OPTIONAL SET[1:?] of SecurityClassifica-tionAssignment
StateAssignment	OPTIONAL SET[1:?] of StateAssignment
StateDefinitionAssignment	OPTIONAL SET[1:?] of StateDefinitionAssign-ment
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelation-ship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssign-ment

Table 56: "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 Docu-ment. 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 follow-ing:

DocumentTypes	Explanation
'catalogue'	the Document represents a catalogue. If associated to an object, it is the catalogue in which the associated object is listed

DocumentTypes	Explanation
'manual'	the Document represents a handbook. If associated to an object, it is the handbook that is supplied for the associated object
'specification'	the Document represents a specification. If associated to an object, it 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 related to product structure. This value shall be used for nested external references, when the referenced document relates to another Domain 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.
- ***PropertyValueAssignment:*** to assign a PropertyValue to the Document. Use the “DocumentFileProperty” template (see 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 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.

If the postprocessor has to choose one geometry to process (for example CAD processors), 'primary geometry' shall be the first choice. If no primary, but only secondary geometry is provided, the secondary shall be chosen. Remark: It is not recommended to rely on PartView.DefiningGeometry to find the primary geometry, since it is optional.

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
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
SameAs	OPTIONAL SET[1:?] of Proxy
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

Entity DocumentVersion attributes	Attribute type
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
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 57: "DocumentVersion" Attributes

Attribute recommendations

- ***ActivityAssignment***: the Activities associated to the DocumentVersion. The value of this attribute need not be specified. Use "Activity" template (see Recommended Practices for AP242 Domain Model XML Change Management for details)).
- **ApprovalAssignment**: the level of acceptance of the DocumentVersion. The value of this attribute need not be specified. Use "Approval" template (see 4.6.17).
- **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.

- **DocumentVersionRelationship**: a DocumentVersion of the same Document or of a different Document with a specific relation to the DocumentVersion according to the DocumentVersionRelationship.RelationType attribute. The value of this attribute need not be specified. Use "DocumentVersionRelationship" template (see 8.3).
- **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 4.6.19).
- **Views**: the set of DocumentDefinition objects 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 Domain 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 Proxy
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssign-ment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
BreakdownVersionAssignment	OPTIONAL SET[1:?] of BreakdownVer-sionAssignment
DateAndPersonAssignment	OPTIONAL SET[1:?] of DateAndPersonAssign-ment
DatetimeAssignment	OPTIONAL SET[1:?] of DateTimeAssignment
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
DocumentDefinitionRelationship	OPTIONAL SET[1:?] of DocumentDefinitionRe-lationship
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
EventAssignment	OPTIONAL SET[1:?] of EventAssignment
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
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssign-ment
RequirementAssignment	OPTIONAL SET[1:?] of RequirementAssignment

Entity (Digital)DocumentDefinition attributes	Attribute type
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 58: "(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 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 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 Domain 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).
- 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:

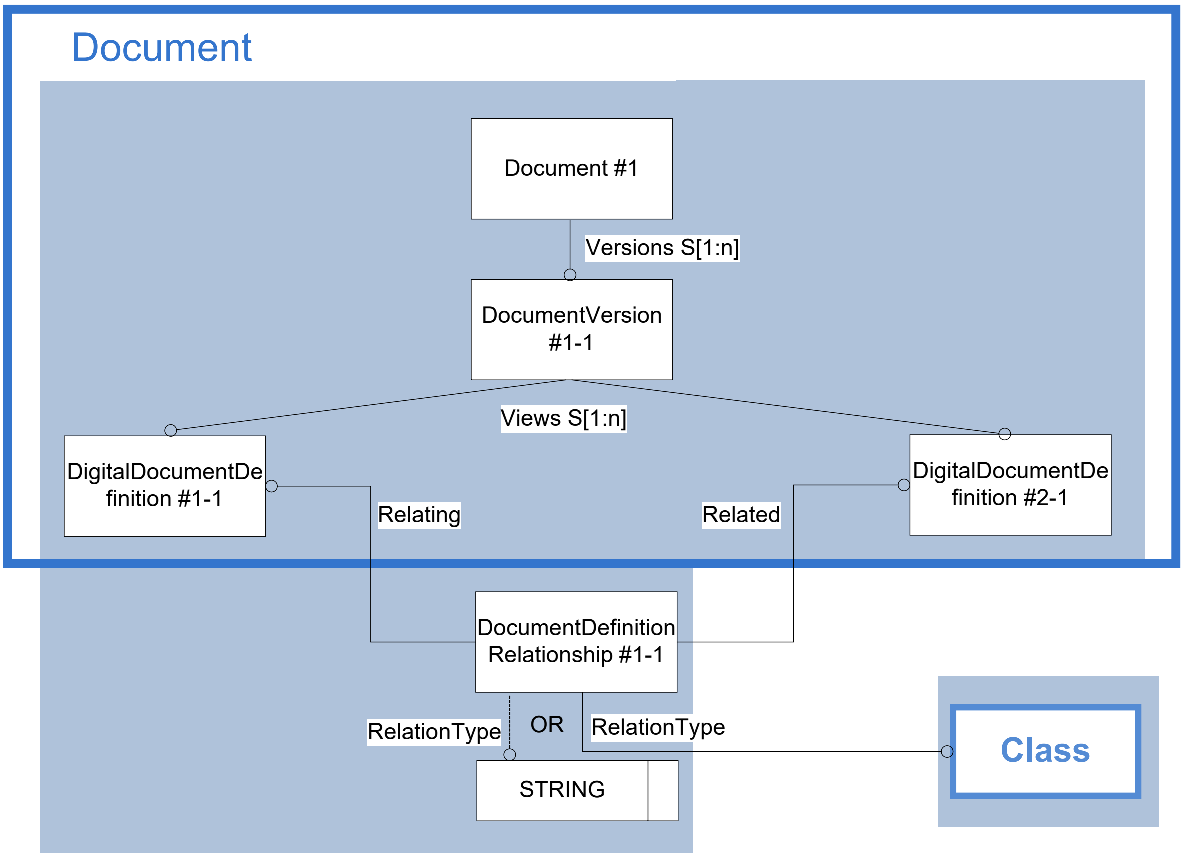
- 'Files' shall reference only one file. in order to manage the versioning of each of them as a distinct document within the PDM system. In case of model splitting or alternate models, please refer to the special cases explained in 11.2.
- In case a DigitalDocumentDefinition has no file, it is recommended to let 'Files' unset, rather than to map a DocumentDefinition without subtype or a DigitalDocumentDefinition referencing a dummy DigitalFile having no Id and (File)Locations.

Postprocessor Recommendations:

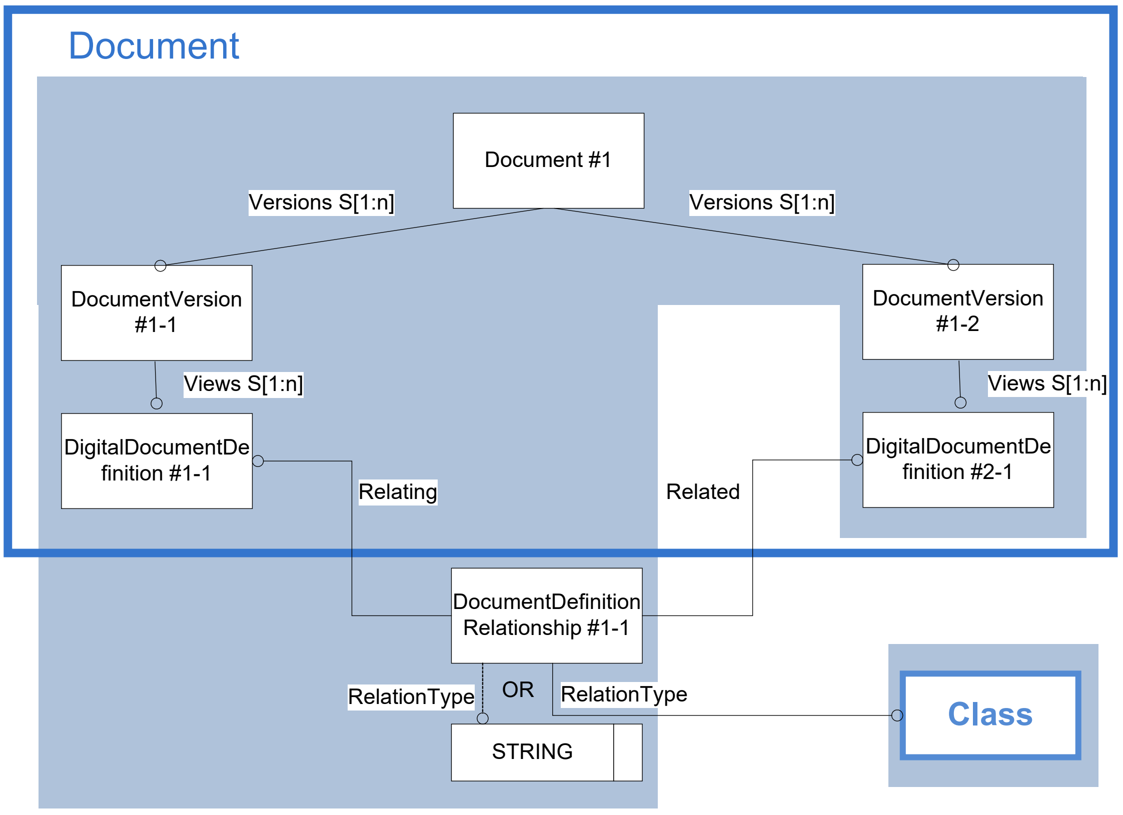
- A DigitalFile having no Id/(File)Locations shall be interpreted as a DigitalDocumentDefinition having no 'Files'.

8.2 Template "DocumentDefinitionRelationship"

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



or of different document versions:



or of different documents:

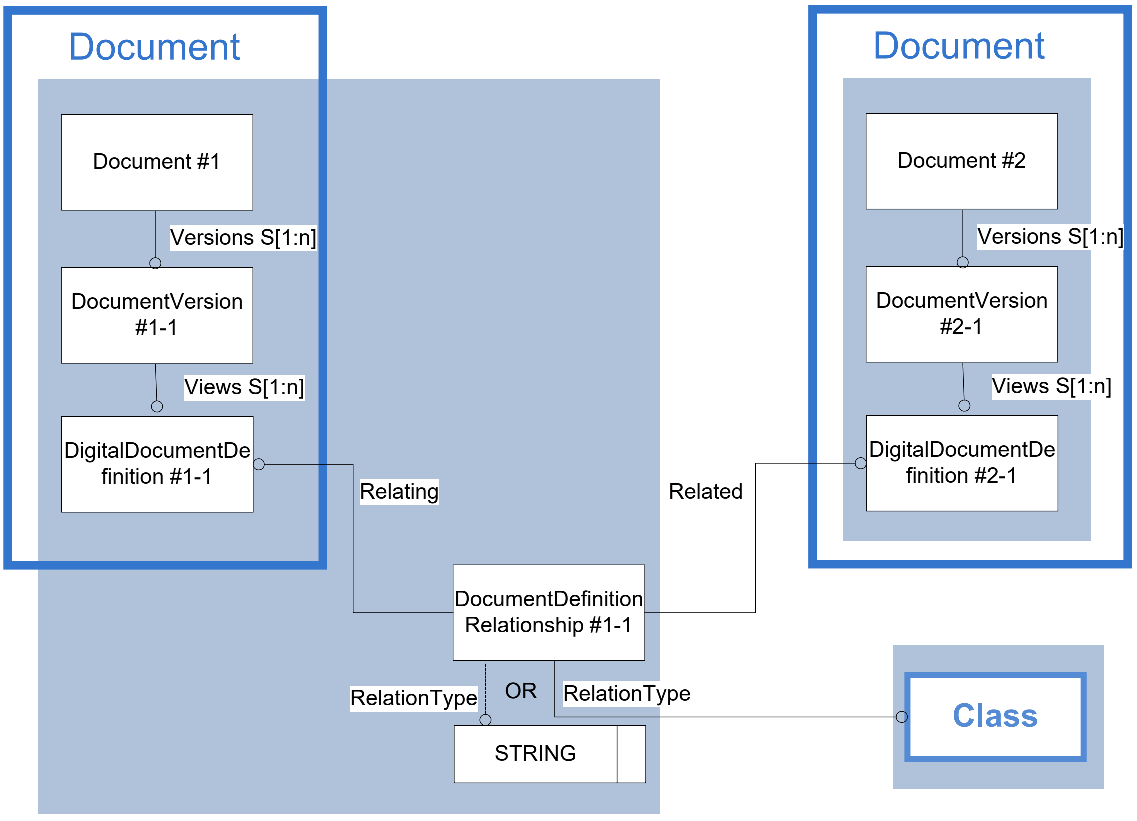


Figure 51: Template “DocumentDefinitionRelationship”

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

```
<Document uid="doc--000000001EB04CF0">
…
  <Id>
    <Identifier uid="docid--000000001EB04CF0--id2" id="plate" idRoleRef="rl--
ii" idContextRef="o--000000178"/>
  </Id>
…
  <Versions>
    <DocumentVersion uid="dv--000000001EB04CF0">
…
      <Id id="A.1"/>
      <Views>
        <DocumentDefinition uid="ddd--000000001EB04CF0" xsi:type="n0:Digital-
DocumentDefinition">
          <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:Digi-
talDocumentDefinition">
          <Id id="/NULL"/>
…
        </DocumentDefinition>
      </Views>
…
    </DocumentVersion>
  </Versions>
</Document>
```

Entity DocumentDefinitionRelationship attributes	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
Related	DocumentDefinition
RelationType	ClassSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
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

Entity DocumentDefinitionRelationship attributes	Attribute type
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 59: "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:

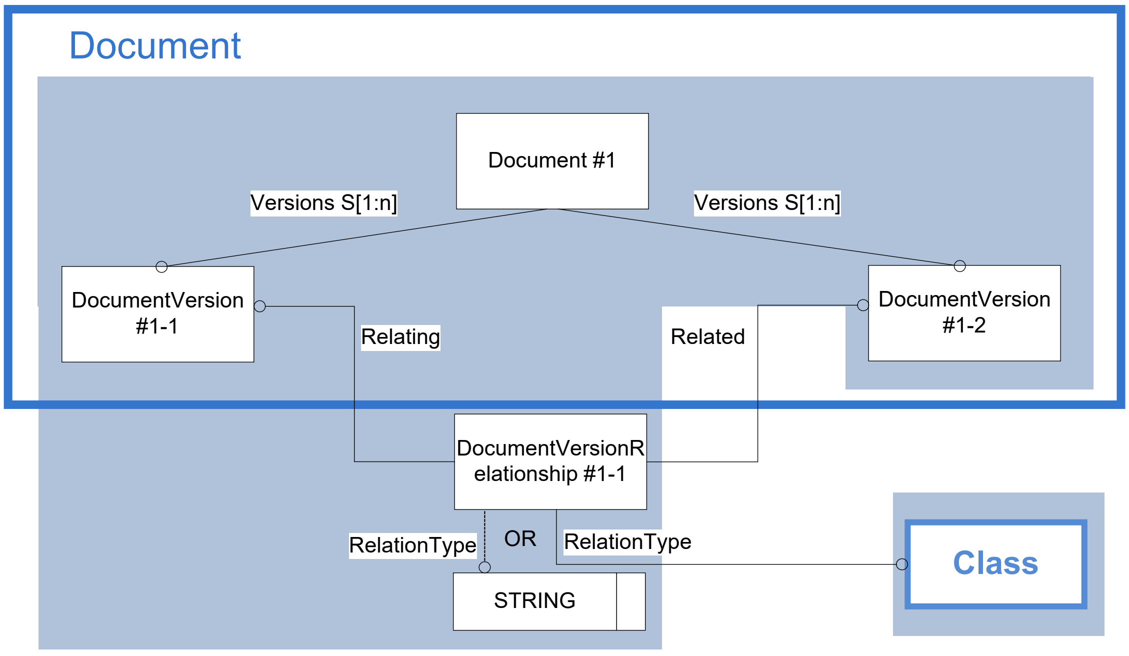
RelationType	
‘addition’	The business object specifies that the related document provides supplementary or collateral information with regard to the information provided by the relating document
‘copy’	The business object defines a relationship where the related DocumentDefinition is a copy of the relating 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.

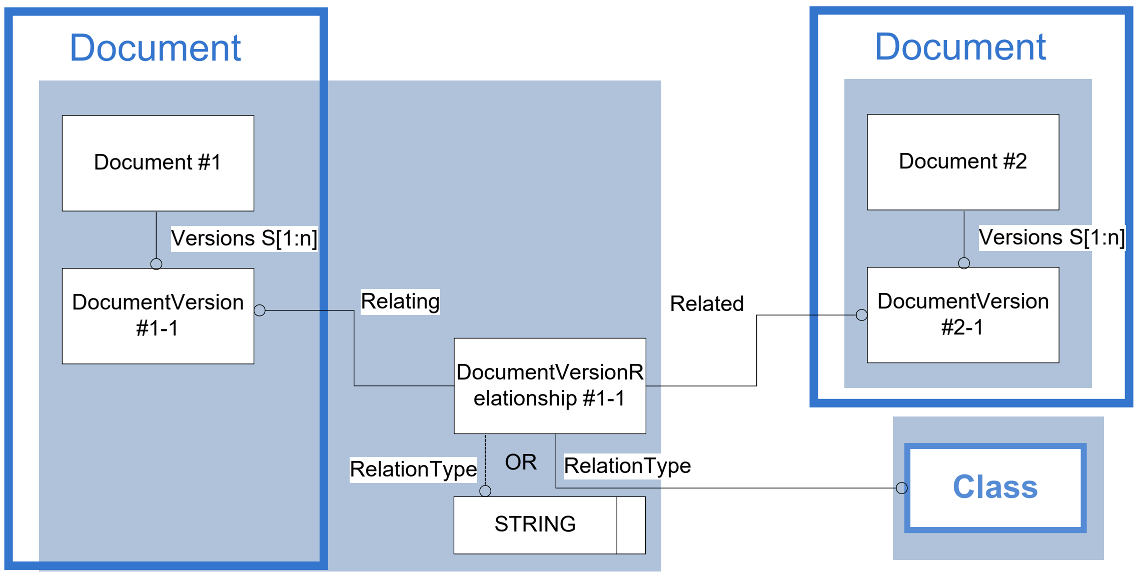


Figure 52: Template "DocumentVersionRelationship"

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain 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 attrib-utes	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
Related	DocumentVersion
RelationType	ClassSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssign-ment
ApprovalAssignment	OPTIONAL SET[1:?] of ApprovalAssignment
CertificationAssignment	OPTIONAL SET[1:?] of CertificationAssignment

Entity DocumentVersionRelationship attributes	Attribute type
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 60: "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'. The Document associated to the Relating/Related DocumentVersions shall be the same.
'supplied document'	The business object defines a relationship between two DocumentVersion objects (both exchanged in the same

RelationType	
	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. It is planned for AP242 Edition 4 in the section 9.6.

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 9.3)

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. Dito 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 post-processor. The postprocessor shall anyway load and import the file correctly.

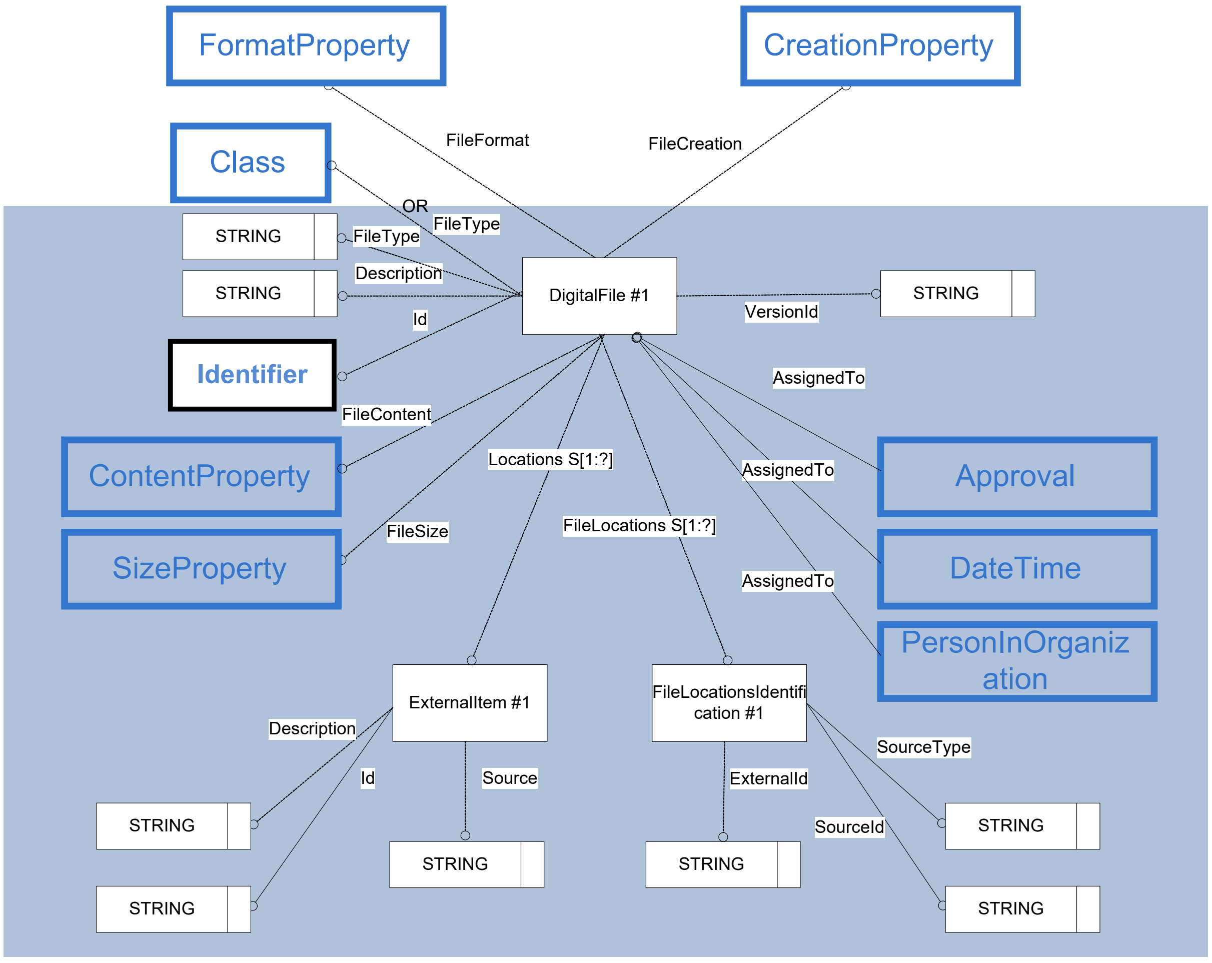


Figure 53: Template "DigitalFile"

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain 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>unspecified</ClassString>
  </Kind>
  <Name>
    <ClassString>byte</ClassString>
  </Name>
  <Prefix>
    <ClassString>kilo</ClassString>
  </Prefix>
      <Quantity>
            <ClassString>data size</ClassString>
      </Quantity>
</Unit>
<Classification uid="gtc--3">
  <Class>
    <ClassString>solid geometry</ClassString>
  </Class>
  <Role>geometry type</Role>
</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" idContex-
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 Classification
Description	OPTIONAL DescriptorSelect
FileContent	OPTIONAL ContentProperty
FileCreation	OPTIONAL CreationProperty
FileFormat	OPTIONAL FormatProperty
FileLocations	OPTIONAL SET[1:?] OF FileLocationIdentification
FileSize	OPTIONAL SizeProperty
FileType	OPTIONAL ClassSelect
Id	OPTIONAL IdentifierSelect
Locations	OPTIONAL SET[1:?] of ExternalItem
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
VersionId	OPTIONAL IdentifierSelect
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
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
FileRelationship	OPTIONAL SET[1:?] of FileRelationship
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
StateAssignment	OPTIONAL SET[1:?] of StateAssignment
StateDefinitionAssignment	OPTIONAL SET[1:?] of StateDefinitionAssignment

Entity DigitalFile	Attribute type
SuppliedObjectRelationship	OPTIONAL SET[1:?] of SuppliedObjectRelation-ship
TimeIntervalAssignment	OPTIONAL SET[1:?] of TimeIntervalAssignment
WorkRequestAssignment	OPTIONAL SET[1:?] of WorkRequestAssign-ment

Table 61: "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. Use “SizeProperty” template (see 10.4).
- ***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 prod-uct 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 Domain Model XML file (see section 9.3)

- ***Id:*** the identifier for the DigitalFile. Although optional in the schema, this attribute shall be specified. Use “Identifier” template (see 4.6.6).
- ***FileLocations:*** location of the DigitalFile. The value of this attribute need not be specified. If empty or unset, the file shall be located in the same directory as the Domain Model XML file referencing to it.
- ***Locations:*** alternatives of the identifier and location of the file. The value of this attribute need not be specified. If empty or unset, the file shall be located in the same directory as the Domain 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.
- ***ActivityAssignment***: the Activities associated to the DigitalFile. The value of this attribute need not be specified. Use "Activity" template (see Recommended Practices for AP242 Domain Model XML Change Management for details)).
- **ApprovalAssignment**: to assign an Approval to the DigitalFile. Use the “Approval” template; see 4.6.17 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, see9.2.**
- **OrganizationOrPersonInOrganizationAssignment**: to assign an Organization or a PersonInOrganization to the DigitalFile. Use the “PersonInOrganization” template; see 4.6.19 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.

Preprocessor Recommendation:

- Using DigitalFile.Locations and ExternalItem.Source, no cyclic structure shall be instantiated (in case ExternalItem.Source would be of type Document of DigitalFile). This is why the use of IdentifierString is recommended below for ExternalItem.Source).
- The use of Locations rather than FileLocations is recommended. Both should not be set at the same time.If multiple versions of a DigitalFile are needed, since DigitalFile.Id and DigitalFile.VersionId are in the same object, all versionless attributes of DigitalFile (Id, FileFormat, FileType, , …) shall be redundantly mapped to all versions, including the multiple identifiers.

Postprocessor Recommendation:

For upward compatibility reasons: if Locations is not set, interprete FileLocations instead.

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 InformationUsageRightAssign-ment

Table 62: "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 Ex-tenalItem. 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 value of this attribute need not be specified. If unset, '.' is assumed. 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 part-ners.

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.

Entity FileLocationIdentification	Attribute type
ExternalId	OPTIONAL IdentifierString
SourceId	IdentifierString
SourceType	IdentifierString

Table 63: "FileLocationIdentification" Attributes

Attribute recommendations

- ***ExternalId:*** the identifier of the external item (redundant to DigitalFile.Id). Use of IdentifierString type. The value of this attribute need not be specified. If the attribute is not specified, then SourceId completely specifies the file location.
- ***SourceId***: the text that identifies the context in which the ExternalId is specified. In case of SourceType='URL' or 'FTP', the Source shall store 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).
- ***SourceType:*** the text that identifies the identification scheme of the SourceId. The following values are recommended:

  According to the ISO AP242 Specification, where applicable, the following values shall be used:

SourceType	
'URL'	For a web page
'FTP'	For an FTP address
'ISBN'	For physical documents

9.2 *Template "FileRelationship"*

This relationship enables to relate two files.

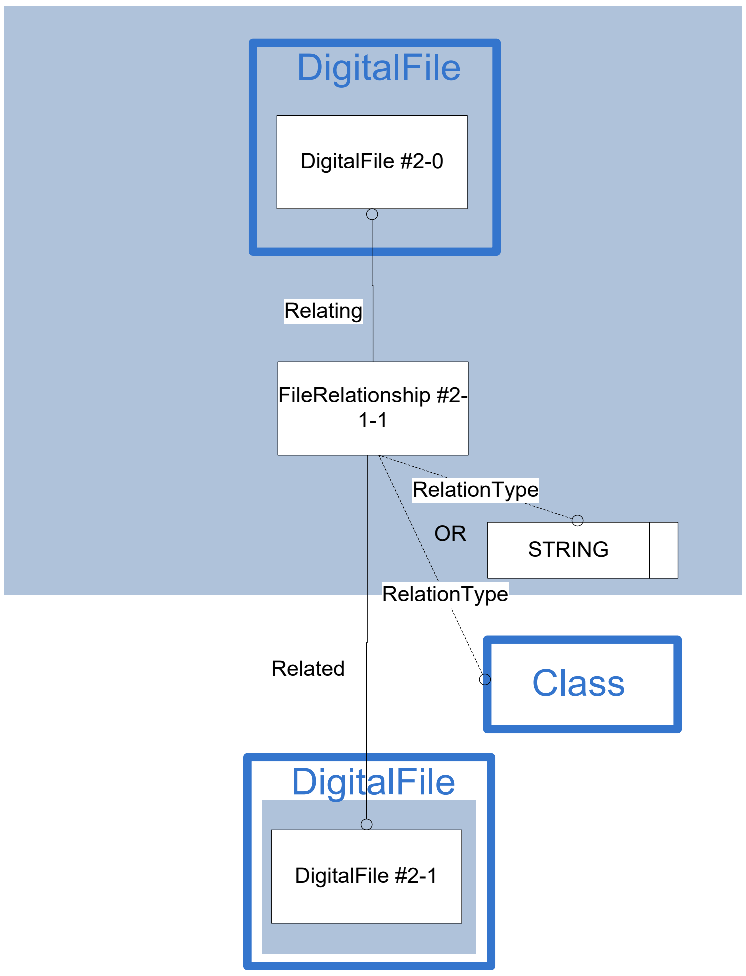


Figure 54: Template "FileRelationship"

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

```
<File xsi:type="n0:DigitalFile" uid="df--000000001EAA8110">
…
  <Id>
    <Identifier uid="dfid--000000001EAA8110--14" id="l-bracket.stp" idRol-
eRef="rl--ii" 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" idRoleRef="rl-
-ii" idContextRef="o--000000178"/>
  </Id>
…
</File>
```

Entity FileRelationship attributes	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
Related	File
RelationType	ClassSelect
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
EffectivityAssignment	OPTIONAL SET[1:?] of EffectivityAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUs-ageRightAssignment
ModelPropertyAssignment	OPTIONAL SET[1:?] of ModelPropertyAssign-ment
PropertyDefinitionAssignment	OPTIONAL SET[1:?] of PropertyDefini-tionAssignment
PropertyValueAssignment	OPTIONAL SET[1:?] of PropertyValueAssign-ment

Table 64: "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 re-lated file is a copy of the relating file
'decomposition'	The business object defines a relationship where the re-lated file is one of potentially more sub documents of the relating file
'derivation'	The business object defines a relationship where the re-lated 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. In case the Id is equal and VersionId of both Files is different, a version sequence is defined.
'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 using the so-called 'Reference' Mechanism of the above VDA recommendation. This mechanism is described in the next chapter.

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 55 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.

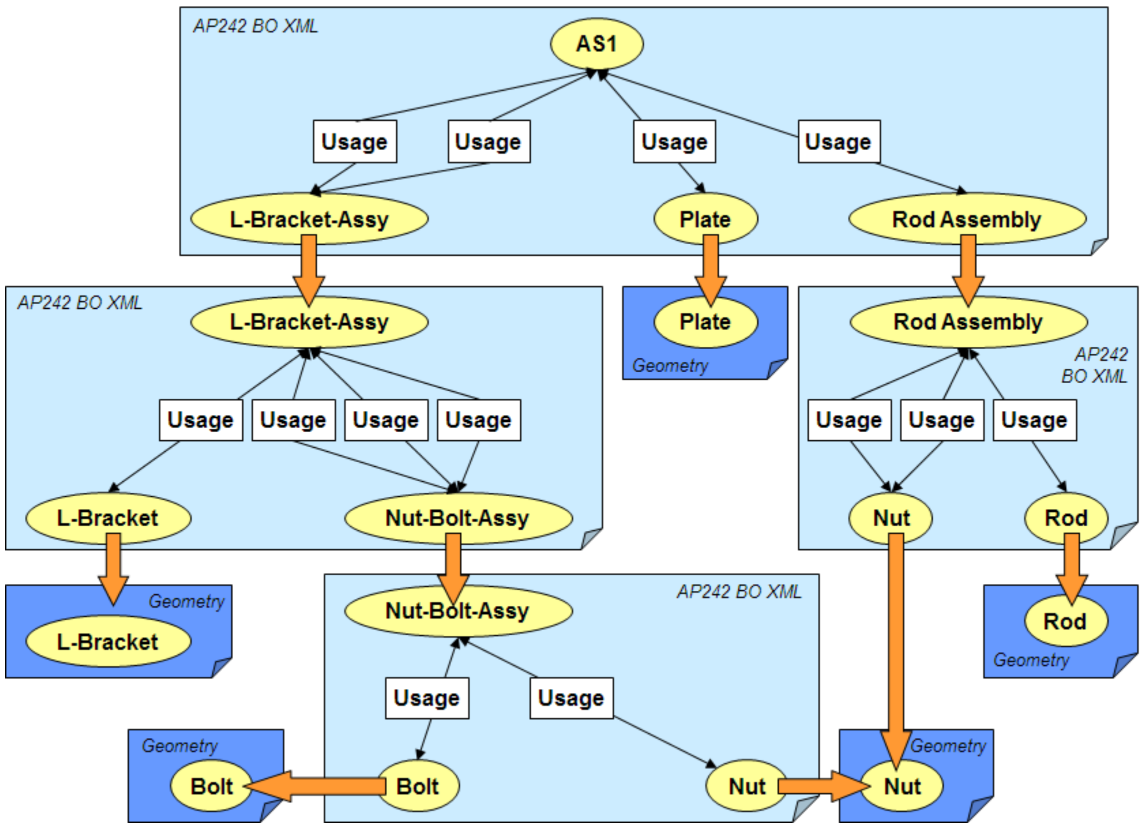


Figure 55: Example for not recommended Nested Structure

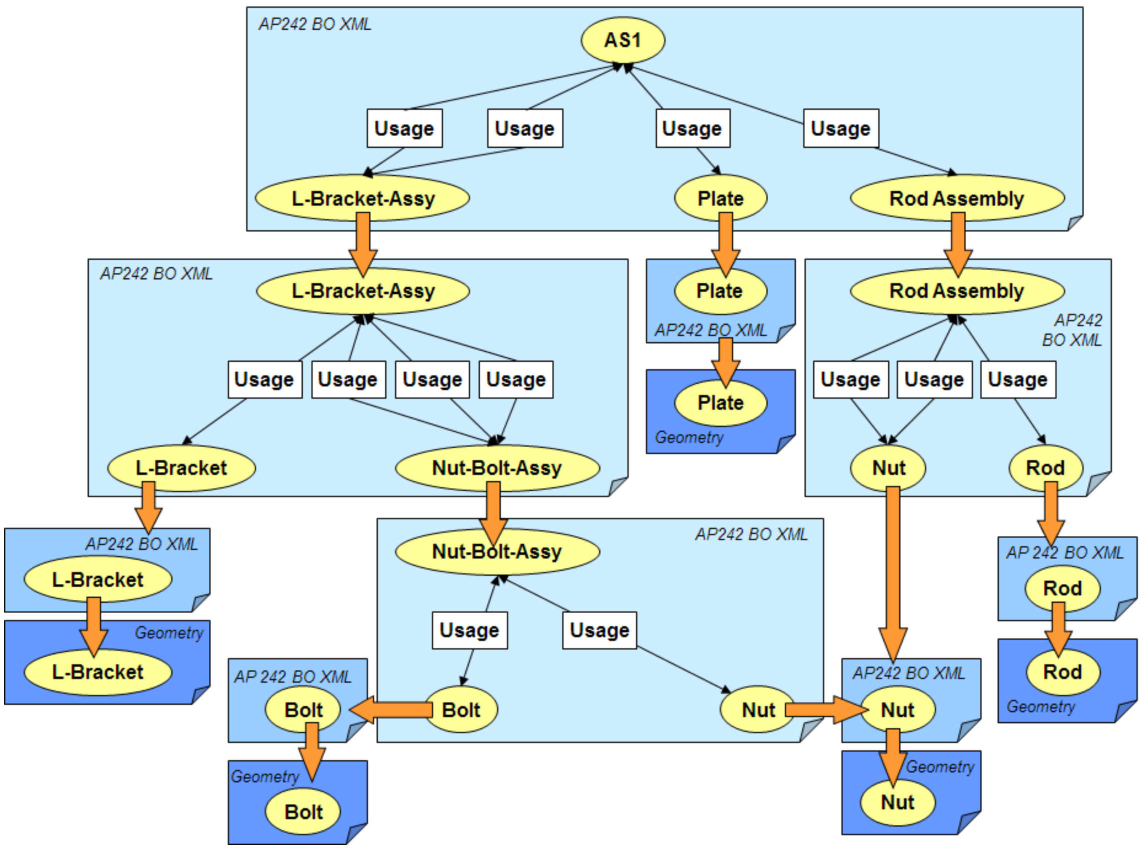


Figure 56: Example for Nested Structure with additional part-level XML files

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 56 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. Name and PartTypes (since mandatory) shall be set respectively to dummy values '/NULL' and 'piece part',
   - PartVersion.id (or '/ANY' if the right version get computed at runtime by the PDM application or during the merge process of the nested files),
   - PartView.id (only if multiple views are handled), PartView.InitialContext (since mandatory)
   - Occurrence.id (to build the referenced part as NextAssemblyOccurrenceUsage into the assembly node)
2. No further attributes (like descriptions, classifications (except the one mentioned in 4.), no assignments of any kind (except to the DigitalFile referencing to the component file (see 2.), i.e., no validation properties, … mapping a reference to the component file (AP242 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 (AP242 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.
6. If no NextAssemblyOccurrenceUsage instance is available for a given NextAssemblyViewUsage, the nested file referencing shall be performed over NextAssemblyViewUsage.
7. If both NextAssemblyOccurrenceUsage and NextAssemblyViewUsage are available for a given assembly link, the nested file referencing shall be consistent (i.e. referencing the same nested file).
8. Further candidates for nesting are PartRelationship (for example AlternatePartRelationship), PartVersionRelationship and PartViewRelationship.

Remark: in case multiple ids are available for a referenced component, it is sufficient here to mention at least the id having the idRoleRef ‘exchange identification information’ and to exchange all the multiple ids (including the one having the idRoleRef ‘exchange identification information’) only inside the component XML file.

If the identifier (with role ‘identification information’) gets changed to the role ‘exchange identification information’ (for the same idContextRef), for example if a ‘made’ part becomes a ‘bought’ part, the referencing mechanism should be able to evaluate both roles. Otherwise the role would have to be updated in all usages of this part.

It is assumed that even if a part gets more identifiers during his life cycle, the file name of the nested reference shall not change (except regarding the version number).

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 along the attribute ‘Definition’ shall also be defined in the superordinate assembly file. Management of references along 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, of 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 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 is described in the Recommended Practices for AP242 Domain Model XML Kinematics. Definitions applicable to the product structure in general made in this context will be reflected here.

5. ProductConfiguration: This is described in the Recommended Practices for AP242 Domain Model XML Configuration Management. Definitions applicable to the product structure in general made in this context will be reflected here.

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.

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

All the nested files that describe a product structure shall have the same value in ExchangeContext.IdentificationContext.

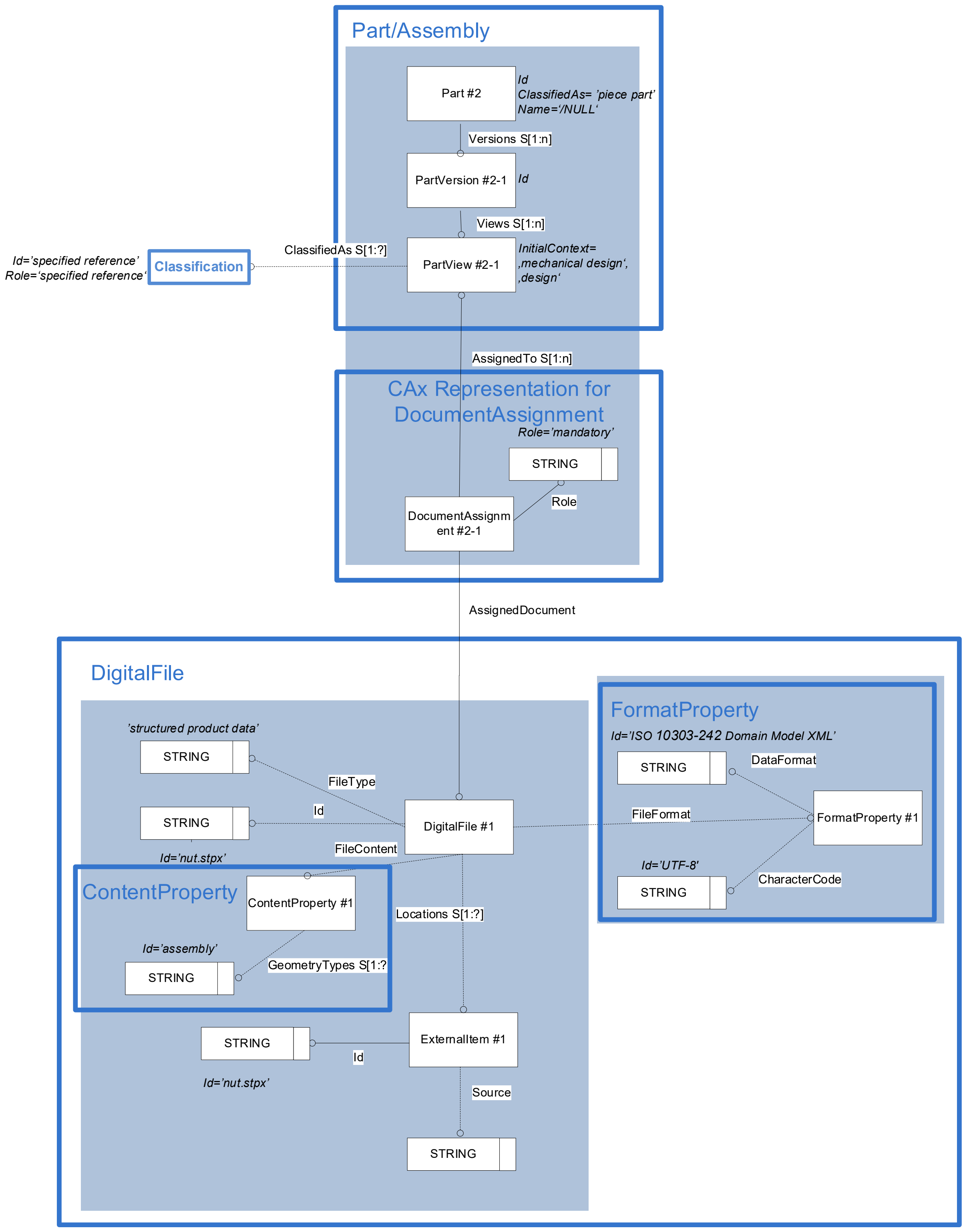


Figure 57: Element Structure for Nested XML File

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

```
<Classification uid="gtc—1">
  <Class>
    <ClassString>specified reference</ClassString>
  </Class>
  <Role>specified reference</Role>
</Classification>
<Part uid="p—000000001E720B30">
  <Id>
    <Identifier uid="pid—000000001E720B30—id7" id="nut" idRoleRef="rl—ii"
idContextRef="o—000000178"/>
  </Id>
…
  <Versions>
    <PartVersion uid="pv—000000001E720B30—id7">
      <Id id="/NULL"/>
      <Views>
        <PartView uid="pvv—000000001E720B30—id7">
          <ClassifiedAs>
            <Classification uidRef="gtc—1"/>
          </ClassifiedAs>
…
          <DocumentAssignment xsi:type= "n0:DocumentAssignment" uid="da—
000000001A304330—id7">
            <AssignedDocument uidRef="df—000000001E720B30"/>
            <Role>
              <ClassString>mandatory</ClassString>
            </Role>
          </DocumentAssignment>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>


<FormatProperty uid="ffp—AP242BOMODEL">
  <CharacterCode>
    <ClassString>UTF-8</ClassString>
  </CharacterCode>
  <DataFormat>
    <ClassString>ISO 10303-242 Domain Model XML</ClassString>
  </DataFormat>
</FormatProperty>
<Classification uid="gtc—2">
  <Class>
    <ClassString>assembly</ClassString>
  </Class>
  <Role>geometry type</Role>
</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" idRoleRef="rl—
ii" 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 document in the PDM system, but are only created during the export process.

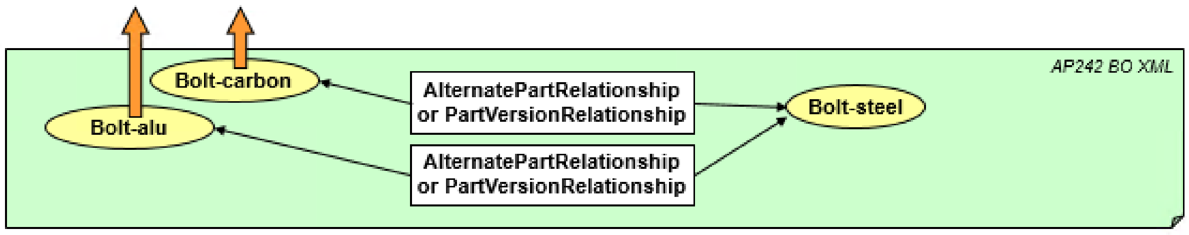

AP242 BO XML
Bolt-carbon
Bolt-alu
AlternatePartRelationship or PartVersionRelationship
AlternatePartRelationship or PartVersionRelationship
Bolt-steel

Figure 58: Example for Nested Structure with altenate parts

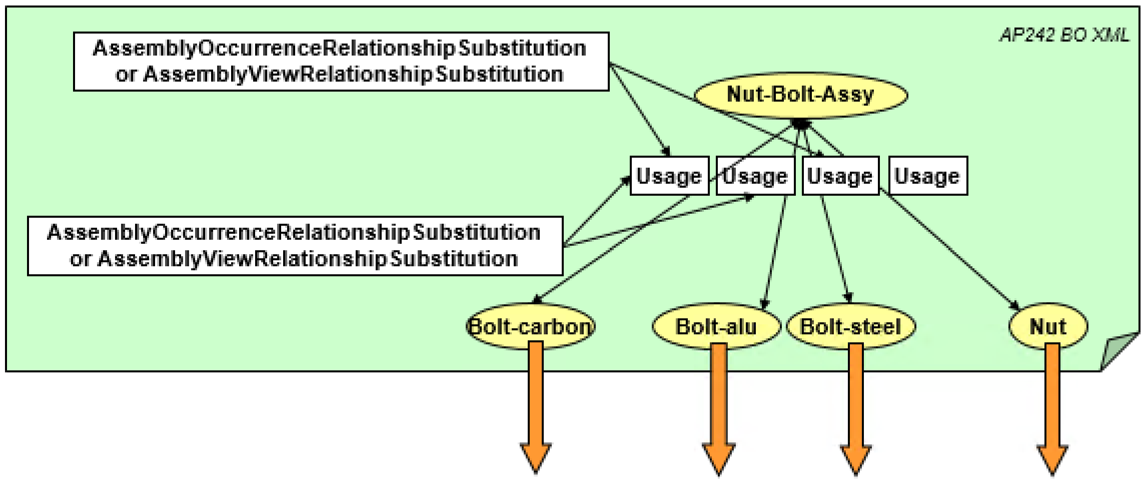

AP242 BO XML
AssemblyOccurrenceRelationshipSubstitution or AssemblyViewRelationshipSubstitution
Nut-Bolt-Assy
Usage
Usage
Usage
Usage
AssemblyOccurrenceRelationshipSubstitution or AssemblyViewRelationshipSubstitution
Bolt-carbon
Bolt-alu
Bolt-steel
Nut

Figure 59: Example for Nested Structure with substitute parts

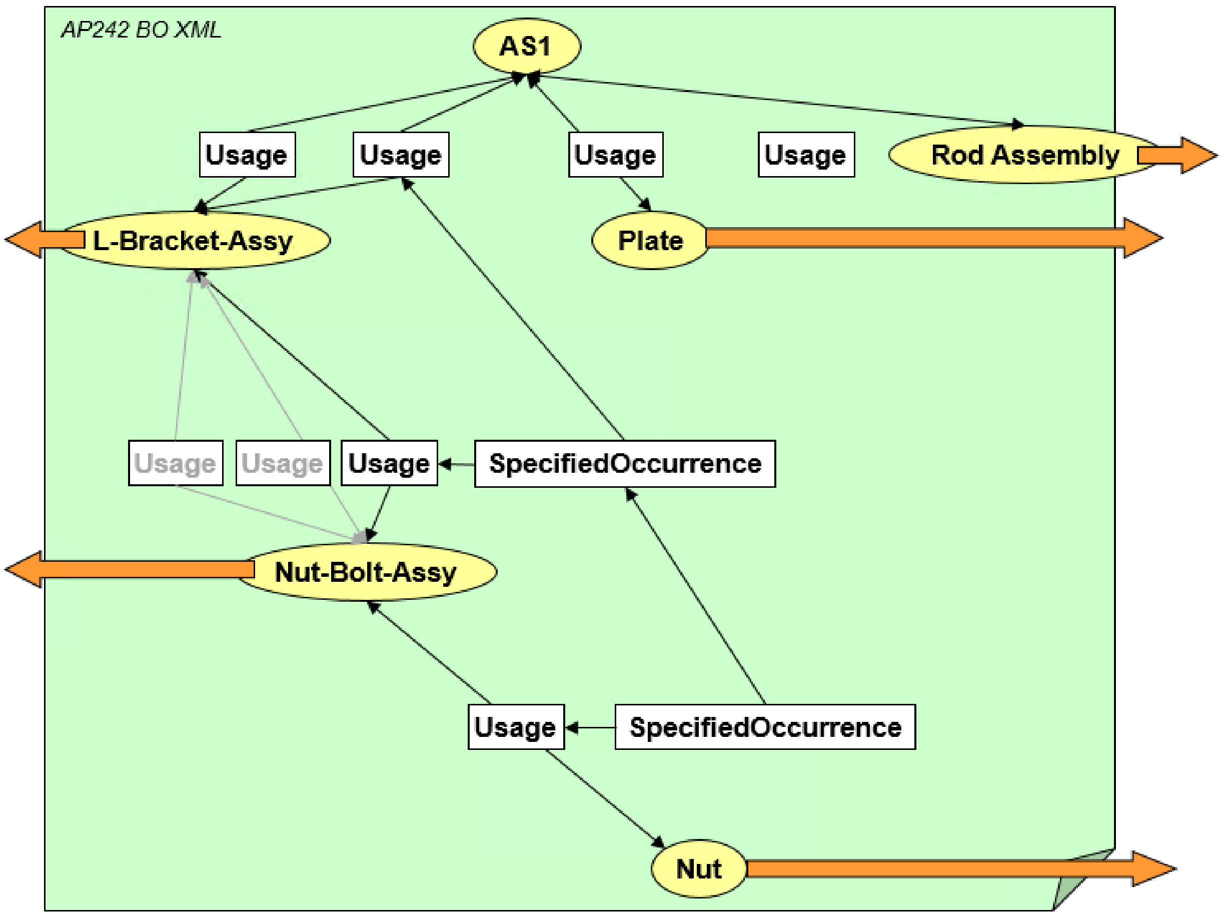


Figure 60: Example for Nested Structure with SpecifiedOccurrences

Preprocessor Recommendations:

- If multiple PartVersions of a Part are provided, each PartVersion shall be mapped to a separate file
- It is assumed that even if a part gets more identifiers during his life cycle, the file name of the nested reference shall not change.
- There is no general recommendation for using the Part.Id to build the file name, but it might be the most straightforward way to do it (e.g. <Part.Id>.stpx). **W A R N I N G**:Using the Part.Id and the PartVersionId to build the file name implies some limitations:
    - If the identifier with role 'exchange identification information' gets changed from one idContextRef to another), for example if a 'made' part becomes a 'bought' part, or if the role 'exchange identification information' is introduced on the identifier of an external company, the file name will now use the other identifier having the role 'exchange identification information'. This should be avoided in the context of Long Term Archiving, where each change in an assembly structure gets archived individually: changing the name of the filename will cause a broken reference from all assemblies where the changed component is referenced.
    - If /ANY is used for referencing the PartVersion.Id (for example to reference a standard part, in case of unprecise structure or for AlternatePartRelationship), the referenced filename shall not use the PartVersion.Id. The file name containing each version of the part shall also not use the PartVersion.Id, since this would result in a broken reference. Inconvenient: such a Part cannot be used simultaneously in a precise structure (using PartVersion.Id in the Filename) an in an unprecise structure (not using the PartVersion.Id in the Filename)…
    - Even for precise structures, where a given version of the component gets built into an assembly node, it may be meaningful not to use the PartVersion.Id in the file name. For example, the Long Term Archiving tool can overtake the choice of the right version, storing all of them under the same file name. Another advantage is that it allows a part to be used 'precise' in one product structure and 'unprecise' in another one.
    - Ditto for the naming of the file for the root node: it may be used later in a precise / unprecise structure (not being a root part anymore).
- The ExternalGeometricModel.Id, ExternalGeometricModel.ExternalFile[DigitalFile].Id and DigitalFile.Locations[ExternalItem.]Id or DigitalFile.FileLocations.ExternalId shall have the same value than the name of the AP242 XML file where the component file is described. For example:
    - NextAssemblyOccurrenceUsage structure is imprecise (use of /ANY): nut.stpx without PartVersion.Id
    - NextAssemblyOccurrenceUsage structure is precise or multiple versions are provided for a given Part: nut-A.1.stpx with PartVersion.Id
- What characters are valid for use in the file name depends on the operating system. A typical (but non exhaustive) list of invalid characters is */\<>|:?". It is recommended to replace invalid characters with an underscore ('_') when creating the file name. Some CAD systems have additional restrictions, but CAD files are not .stpx files.
- The location path of the nested files shall be relative paths to the directory of the file that contains the nested file reference.
- When re-exporting the model, e.g., after an update, it needs to be ensured that the file names are created consistently, in particular for delta-exchange scenarios.

- 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.
- 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 out 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.
- No nested stpx file shall be created for a part where the so-called 'Reference' Mechanism (as defined in chapter 9.4.1) is used.

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 whole data. If not, the top-level assembly node(s) have to be found out through scanning all the stpx(Z) files.
- The import status of each file shall be logged in the result report, so that in case of problems, the files not imported before can be imported during the next attempt, instead of importing the whole structure again.
- In order to detect that any file present in the import directory hasn't been processed (for example because its reference from another file was not evaluated or ignored), it is recommended to get some statistics like: nb of files found, nb of files processed (in use also at the CAx-IF)
- Any file reference that couldn't be resolved (for example because the referenced file couldn't be found) should be logged in the result report.

9.3.1 Handling of nested Relationships

Let's take the example of PartVersionRelationships: according to the ISO definition, the 'sequence' relationship shall be mapped with Relating as precedessor and Related as successor (see section 5.1.5).

The 'Related' (successor) PartVersion is not known when releasing/archiving a 'Relating' (predecessor) PartVersion. The 'sequence' relationship between one PartVersion and its successor is only created when the successor PartVersion is created.

In this case, an additional 'Relating' object may be mapped within the nested file of the 'Related' object. The 'Relating' object shall be mapped as a nested reference, like the component parts of an assembly. Here in the nested file of the successor Part, the predecessor Part is a nested reference (minimal instantiation) and maps only the PartVersionRelationship of kind 'sequence' to the successor Part (fully instantiated).

Other example: if a (non-installed) part gets released together with 5 alternative shape parts (Flexible Parts), according to section 5.1.5, these PartVersionRelationships shall be mapped within the nested file of each of the Flexible Parts. When releasing an additional Flexible Part, the nested file of the (non-installed) Part may not be edited, nor is it necessary to release it again => the additional PartVersionRelationship shall be mapped to the nested file of the 6th Flexible Part like:

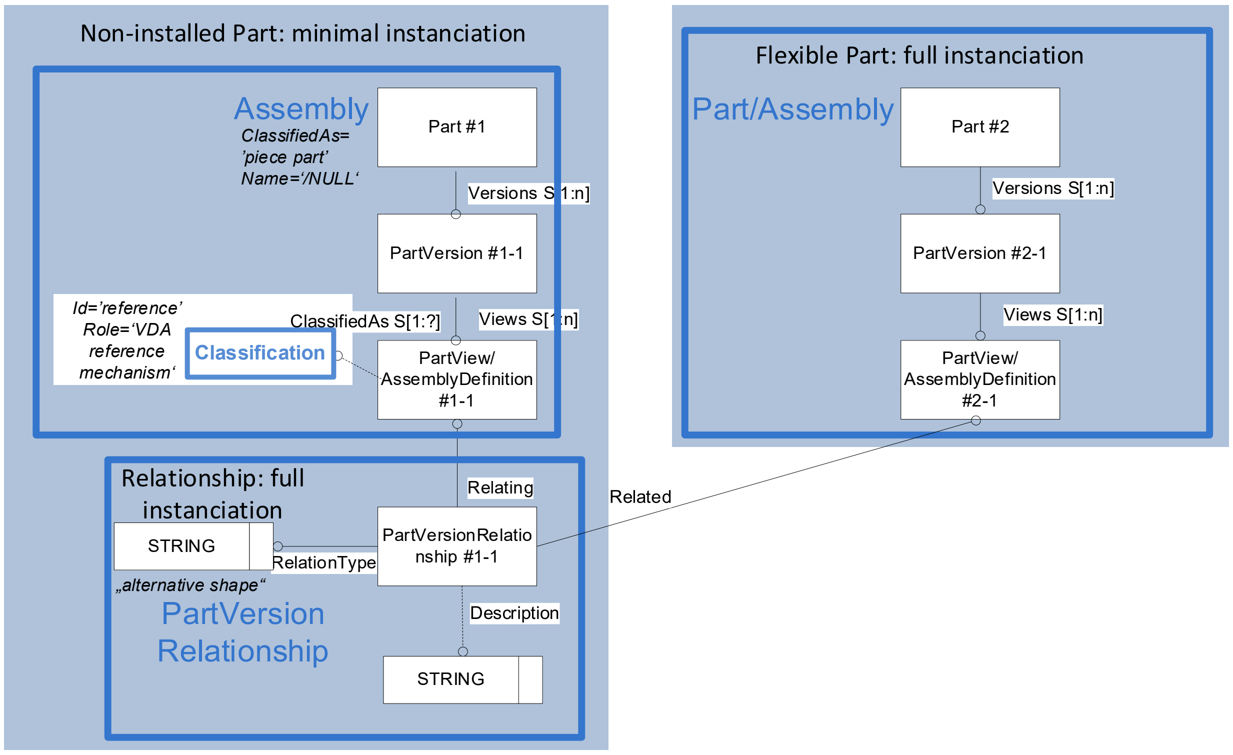

Non-installed Part: minimal instanciation
Assembly
ClassifiedAs= 'piece part' Name='/NULL'
Part #1
Versions S[1:n]
PartVersion #1-1
Id='reference' Role='VDA reference mechanism'
Classification
ClassifiedAs S[1:?]
Views S[1:n]
PartView/ AssemblyDefinition #1-1
Relationship: full instanciation
Relating
STRING
RelationType
PartVersionRelationship #1-1
„alternative shape"
PartVersion Relationship
Description
STRING
Related
Flexible Part: full instanciation
Part/Assembly
Part #2
Versions S[1:n]
PartVersion #2-1
Views S[1:n]
PartView/ AssemblyDefinition #2-1

Figure 61: Special handling of additional 'Related' Parts after archiving/releasing the 'Relating' Part (here example of an additional shape as Flexible Part)

The decision to map a Relationship to the relating or to the related nested file depends on the documentation process of each relationship:

- Does the relationship get changed together with the Relating object? => map the relationship to the nested file of the Relating object
- Does the relationship get changed together with the Related object? => map the relationship to the nested file of the Related object

9.4 Incremental Data Exchange

The key feature of incremental data exchange is that only a subset of available product data is exchanged at a certain time. Incremental data exchange reduces the amount of data to be exchanged. This approach supports actual practice in product data exchange where commonly after the first data exchange only subsets or additional portions of product data are exchanged.

The following examples give an idea of incremental data exchange.

Example: An OEM sends or makes available changes of a particular subset of product data to be communicated, e.g., new versions of items, documents, geometry, positioning information or item properties. Usually, the data set has links to product data, which has been already sent before or which will be sent later.

Example: A supplier sends an update of data. Usually, the supplier only has a specific subset of product data from the OEM's viewpoint that he is allowed to modify.

In these cases, it is feasible to exchange only the modified portion of product data. Another reason to exchange only a particular portion of product data is that a supplier should only modify and send back the shaft of a gearbox, but he has received the complete gearbox with all connected components in order to provide the environmental context.

Incremental data exchange also includes the exchange of administrative product data without CAx files and the exchange of single items and their administrative product data without item structure.

Note : An “incremental data exchange” is different from a "delta exchange". A delta exchange implies an exchange of differences only usually with respect to a set of data previously sent. That means it is the exchange of results of actions, which have been performed at the sender’s side and are to be communicated, e.g., "add component 3 and 4 to an assembly" or "remove document xyz from the set of describing documents of an item". A delta exchange has to be harmonized between two communication partners because change processes usually are company specific.

9.4.1 Reference mechanism

The so-called ‘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.

Note: this mechanism is not meant to be used for nested structures (see previous chapter),

> **Example**: An assembly structure is exchanged, but (some) components are only referenced and not exchanged including their complete definition at the same time.
>
> **Example**: A part may be exchanged, but its assigned document is only referenced.

A reference is a means to identify a single instance or a group of related instances at object level. It is applicable to an existing object at sender’s and/or receiver’s site or represents an object as a ‘placeholder’, i.e., an instance may be referenced at a certain time of data exchange if it was exchanged before or if it will be exchanged later on.

A reference object gets classified as ‘reference’:

part:

- Part: classifiedAs (only if the sender PDM system has a part master object)
- PartView: classifiedAs

document:

- Document: classifiedAs (only if the sender PDM system has a document master object)
- DigitalDocumentDefinition: classifiedAs

A reference object is mapped in the XML File of its assembly(s) by mapping a minimum set of entities and attributes. The application of the reference mechanism is recommended for:

part:

- Part: id with idRoleRef, idContextRef. Name and PartTypes (since mandatory) shall be set respectively to dummy values ‘/NULL’ and ‘piece part’,
- PartVersion: id (or ‘/ANY‘ if the right version get computed at runtime by the PDM application),
- PartView: id (optional), initialContext (mandatory)

document:

- Document: id with idRoleRef, idContextRef, Name and DocumentTypes (since mandatory) shall be set respectively to dummy values '/NULL and 'geometry',
- DocumentVersion: id (or '/ANY' if the right version get computed at runtime by the PDM application),
- DigitalDocumentDefinition: id (optional)

Preprocessor Recommendations:

- Remark: in case multiple ids are available for a referenced component, it is sufficient here to mention at least the id having the idRoleRef 'exchange identification information'.
- Each reference shall be explicitly classified as 'reference'
- The minimum set of entities, attributes and attribute values (see above) shall identify uniquely the referenced object. Otherwise, a receiver will interpret the same object as a new one because of different identifiers.
- The number of relationships (between the assembly and referenced components and between part and a referenced document) has to represent the complete list (of components of the assembly, of documents of the part), independently whether an individual component/document is referenced or its complete definition is exchanged.
- Other data associated to the referenced instance or group of instances (properties, dates, persons, approval...) shall not be instantiated, since they will be ignored by the postprocessor. This applies also for documents associated to a reference part and for files associated to a reference document, as well as to the assembly links and effectivities from a referenced assembly to its direct components:
  - For example, if bounding box geometry has been sent previously, but contains changed components, the assembly nodes of the bounding box geometry shall be sent completely *with* the assembly links to the changed components and with 'reference' links to the unchanged components.
  - For example, if some components have been sent previously but their positioning within the assembly changed, the unchanged components would be sent as 'reference' *without* their underlying assembly links; only their positioning would be exchanged.
- In case of AlternatePartRelationship, one version of the part shall be provided and referenced

Postprocessor Recommendations:

- If the target PDM system hasn't got a part master object:
  - the attributes/properties of a non-reference Part(master) shall be mapped only to those part versions / part views that are mentioned for this part in the XML file and that are not 'referenced'. If all PartVersions/PartViews are 'referenced', the part master changes will be ignored.

    Dito for Document/DocumentVersion,
  - the attributes/properties (present in the target PDM system) of a reference Part(master) shall apply to the import of the PartVersions/PartViews
- If the target PDM system hasn't got a part view object, the attributes/properties (present in the target PDM system) of a reference PartView shall apply to the import of the PartVersion.

Dito for DocumentVersion/DigitalDocumentDefinition,

- It is assumed that all attributes/properties of a referenced part/document are unchanged. This applies also to the documents of a referenced part and to the files of a referenced document, but not necessarily to the assembly links and effectivities from a referenced assembly to its direct underlying components: if they are provided, they should be imported (either none or all the direct assembly links under the assembly node will be provided).
- If the reference mechanism is used on non-supported objects (i.e. other than Part and Document), an error shall be reported

Here are some examples:

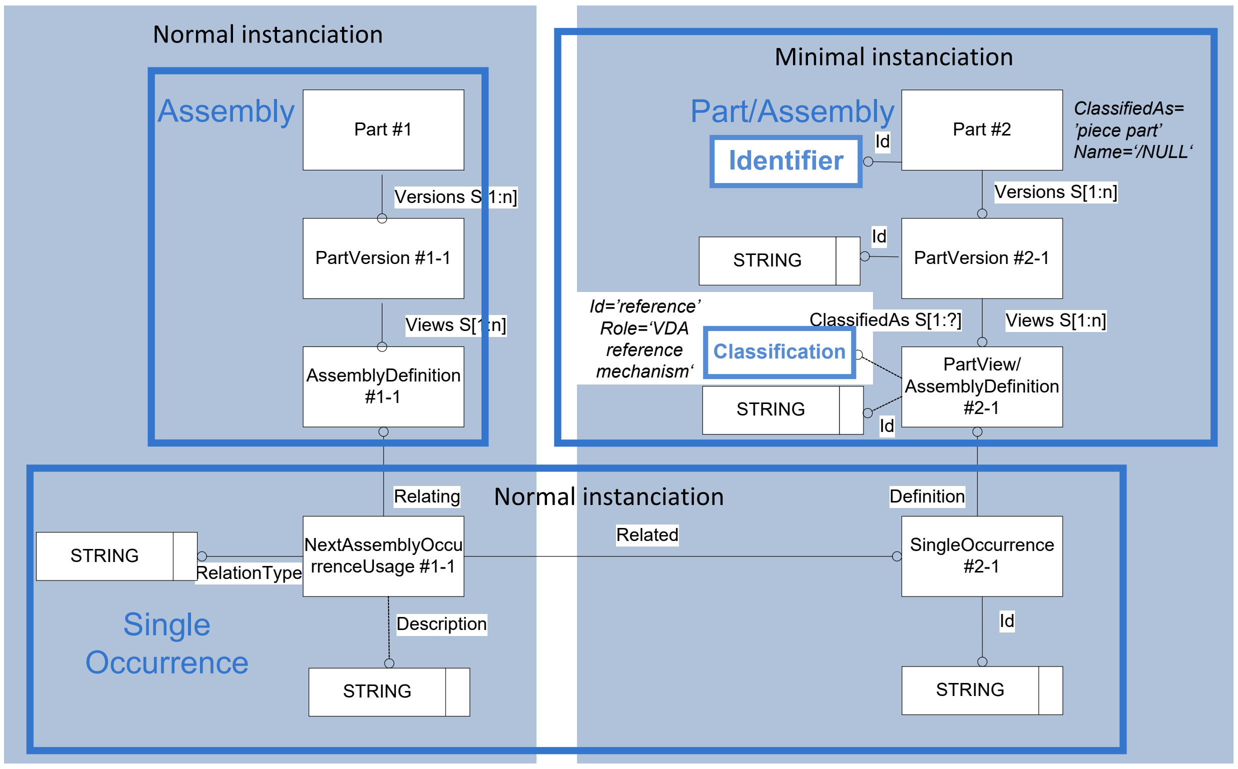


Figure 62: Template “Component reference”

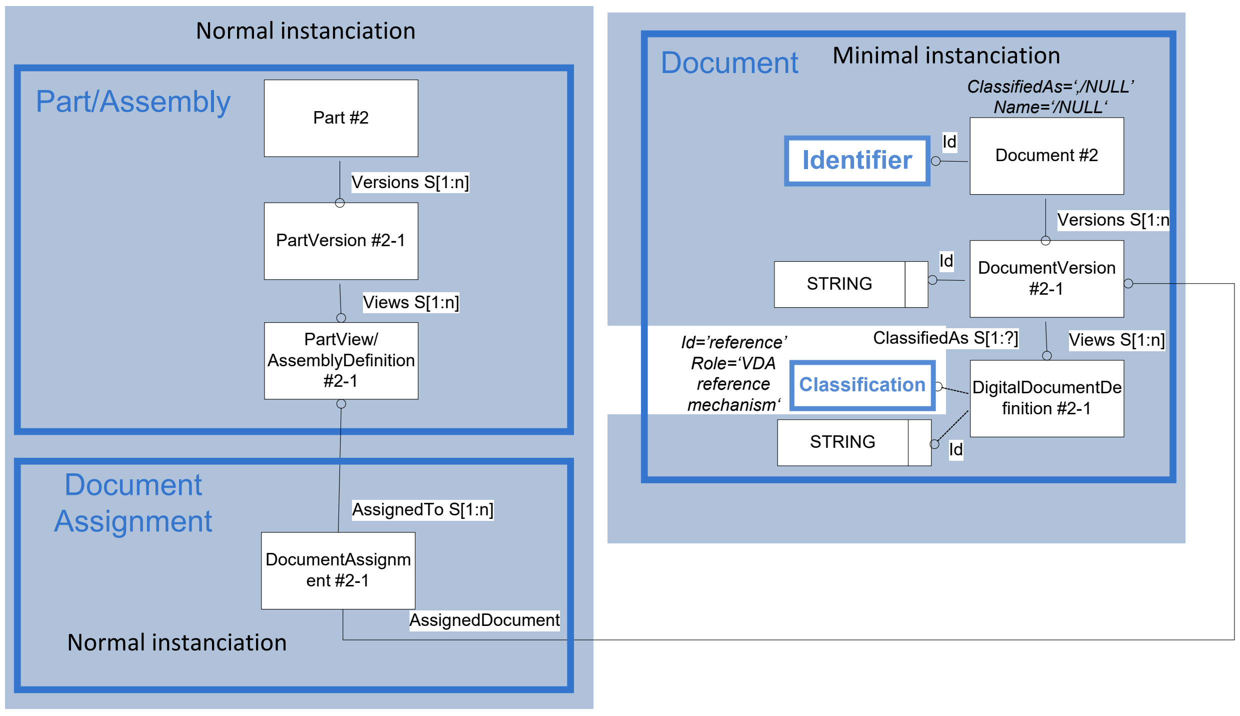


Figure 63: Template "Document reference"

9.5 Delta Exchange

For some special cases, the Incremental Exchange (see section 9.4) is not enough granular. For example, in an assembly node having a very large number of components (NextAssemblyOccurrenceUsages (NAOU))

- if one NextAssemblyOccurrenceUsage shall be deleted, one updated and one added, all the NAOUs have to be exchanged
- the receiver detects the deleted/updated/added NAOU by comparing all the NAOUs with the content of his PDM application

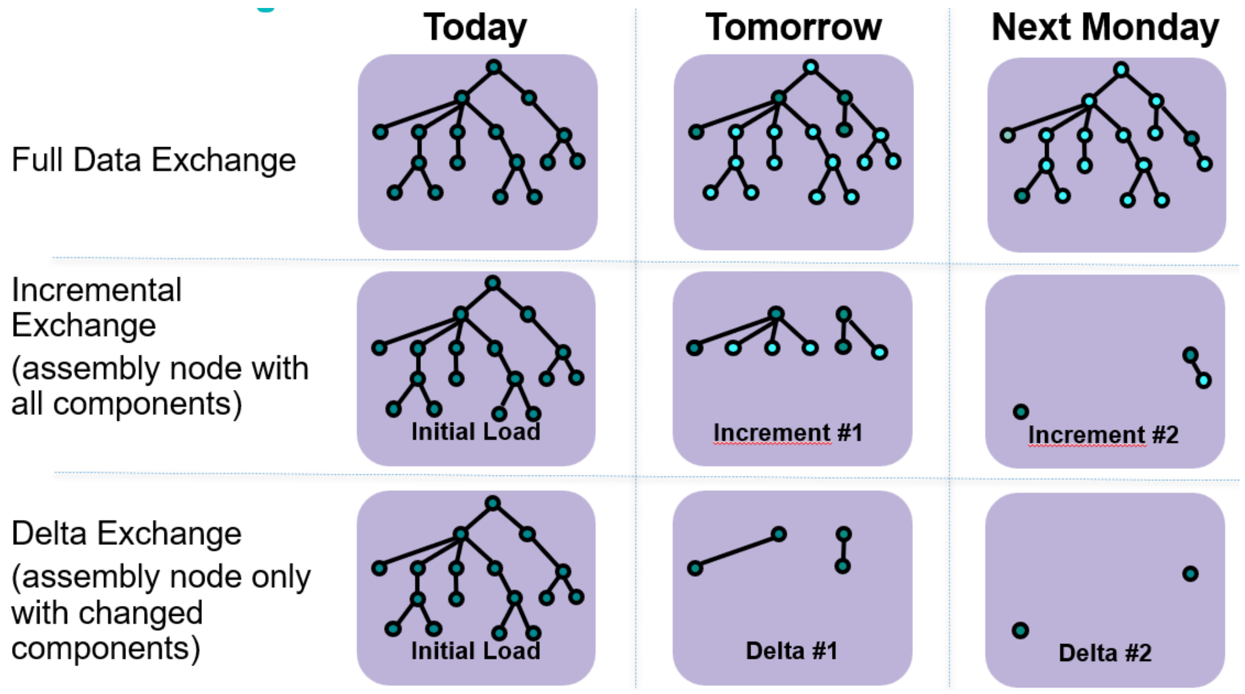

Today
Tomorrow
Next Monday
Full Data Exchange
Incremental Exchange (assembly node with all components)
Initial Load
Increment #1
Increment #2
Delta Exchange (assembly node only with changed components)
Initial Load
Delta #1
Delta #2

Figure 64: Differences between full data exchange, incremental exchange and delta exchange

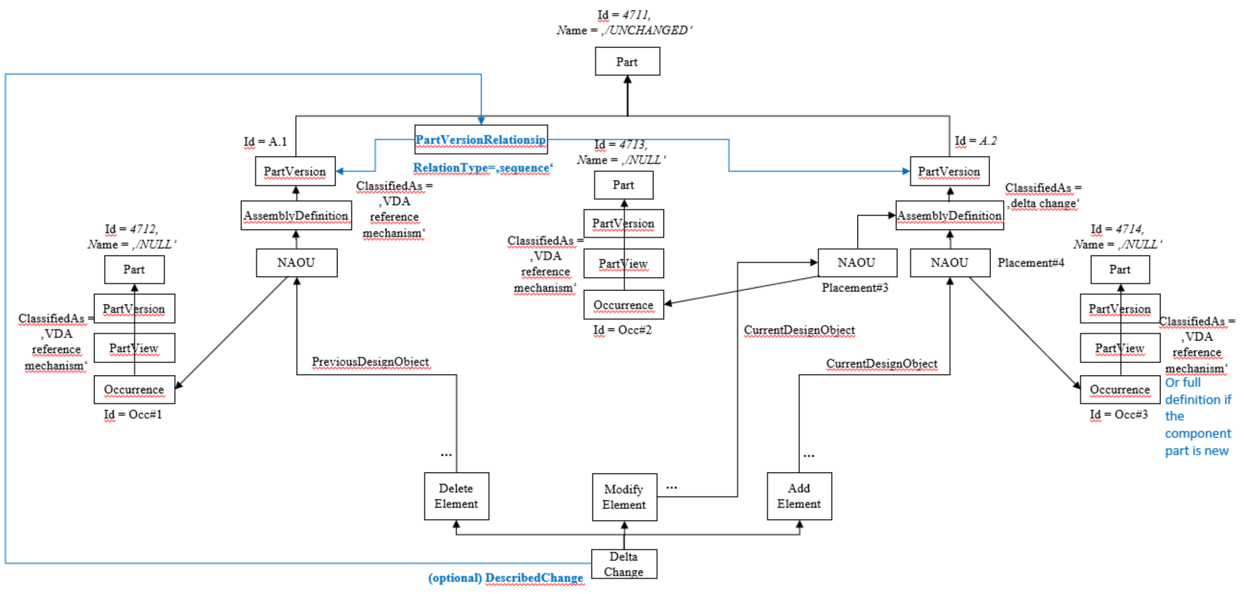


Figure 65: Example of a Delta Exchange on an assembly part with one component deleted, one added and one modified

In a changed object, beside the changed attributes, only a minimum set of entities and attributes are mapped. The application of the delta exchange mechanism is recommended for:

Part:

- Part: id with idRoleRef, idContextRef. Name and PartTypes (since mandatory) shall be set respectively to dummy values '/UNCHANGED' and the type value before/after the change,

- PartVersion: id (or '/ANY' if the right version get computed at runtime by the PDM application),
- PartView: id (optional), initialContext (mandatory)

Document:

- Document: id with idRoleRef, idContextRef, Name and DocumentTypes (since mandatory) shall be set respectively to the dummy value '/UNCHANGED' and the type value before/after the change,
- DocumentVersion: id (or '/ANY' if the right version get computed at runtime by the PDM application),
- DigitalDocumentDefinition: id (optional)

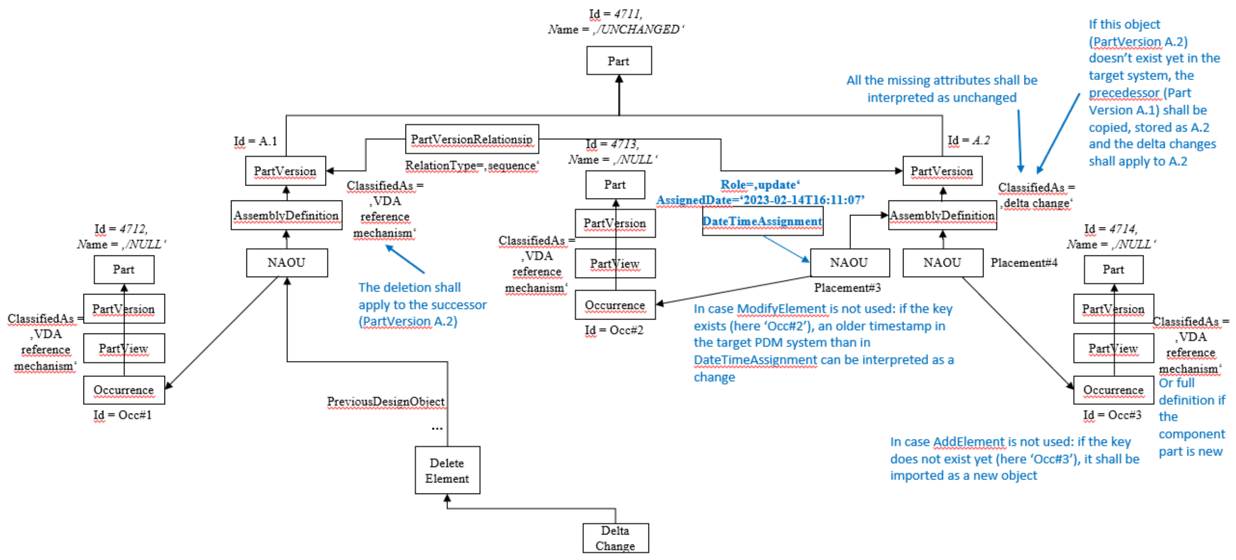

Id = 4711,
Name = ,/UNCHANGED'
Part
All the missing attributes shall be interpreted as unchanged
If this object (PartVersion A.2) doesn't exist yet in the target system, the precedessor (Part Version A.1) shall be copied, stored as A.2 and the delta changes shall apply to A.2
Id = A.1
PartVersionRelationsip
RelationType=,sequence'
Id = 4713,
Name = ,/NULL'
Id = A.2
PartVersion
ClassifiedAs = ,VDA reference mechanism'
Part
Role=,update'
AssignedDate='2023-02-14T16:11:07'
PartVersion
ClassifiedAs = ,delta change'
AssemblyDefinition
DateTimeAssignment
AssemblyDefinition
Id = 4712,
Name = ,/NULL'
PartVersion
ClassifiedAs = ,VDA reference mechanism'
Id = 4714,
Name = ,/NULL'
NAOU
PartView
NAOU
NAOU
Placement#4
Part
Placement#3
Part
The deletion shall apply to the successor (PartVersion A.2)
Occurrence
In case ModifyElement is not used: if the key exists (here 'Occ#2'), an older timestamp in the target PDM system than in DateTimeAssignment can be interpreted as a change
PartVersion
ClassifiedAs = ,VDA reference mechanism'
PartVersion
ClassifiedAs = ,VDA reference mechanism'
Id = Occ#2
PartView
PartView
Occurrence
Or full definition if the component part is new
Occurrence
Id = Occ#1
PreviousDesignObject
Id = Occ#3
...
In case AddElement is not used: if the key does not exist yet (here 'Occ#3'), it shall be imported as a new object
Delete Element
Delta Change

Figure 66: Example of the minimal instanciation of Delta Exchange of a new assembly version with one component deleted, one added and one modified

Preprocessor Recommendation:

- Remark: in case multiple ids are available for a delta changed object, and none id is added nor suppressed, it is sufficient here to mention at least the id having the idRoleRef 'exchange identification information'.
- Each changed object shall be explicitly classified as 'delta change'.
- Only the changed attributes need to be mapped (for example: some properties of an assembly node were updated), all the missing attributes are considered as unchanged.
- References to unchanged objects (for example updated part with unchanged Document) shall be mapped as defined for Incremental Exchange with ClassifiedAs.Class.ClassString=,reference'. Refer to the section 9.4 for more details.
- If an object referenced by DeleteElement.PreviousDesignObject has ClassifiedAs.Class.ClassString=,delta change', this means that the change is performed in the PreviousDesignObject itself. In this case, this object shall be already existing in the target system (according to its key attributes).

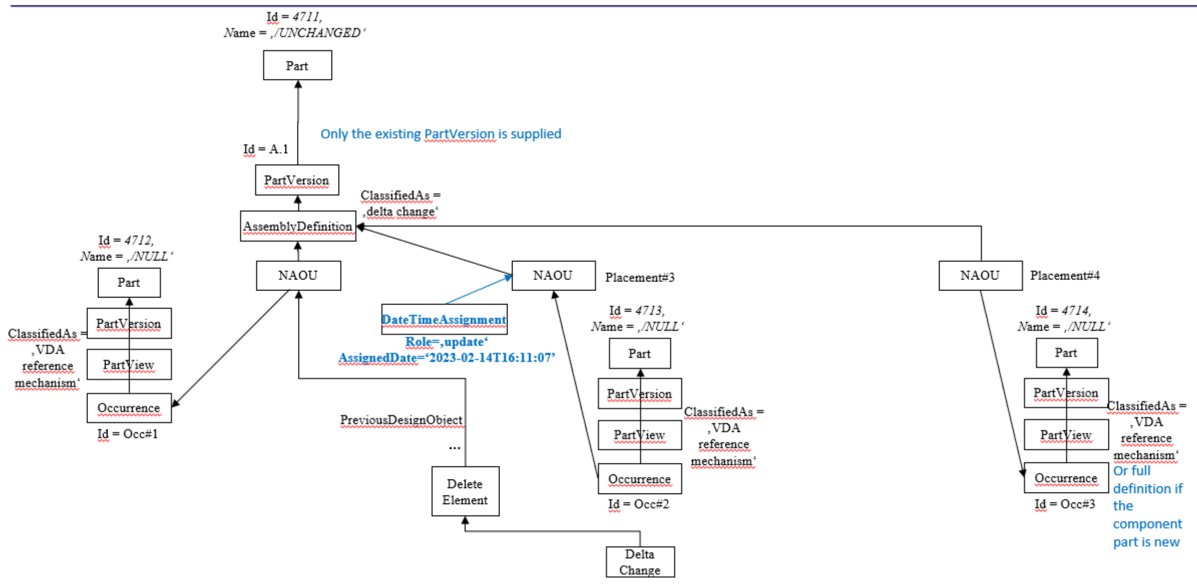


Figure 67: Same example as update of an existing PartVersion

- If an object referenced by DeleteElement.PreviousDesignObject has ClassifiedAs.Class.ClassString=‚reference', this means that the change is performed in the successor version of the object and not in the PreviousDesignObject itself. In this case, its successor shall be related to it via a Part/DocumentVersionRelationship with Relationship='sequence'. Refer to the section 5.1.5 or 8.3 for more details.
- An object having ClassifiedAs.Class.ClassString=‚delta change' and having no DeleteElement.PreviousDesignObject that reference to it may either:
  - be already existing in the target system (according to its key attributes),
  - or be a new one. In this case, its predecessor shall be related to it via a Part/DocumentVersionRelationship with Relationship='sequence'. Refer to the section 5.1.5 or 8.3 for more details.
- Mixing full/incremental exchange with delta exchange is not recommended

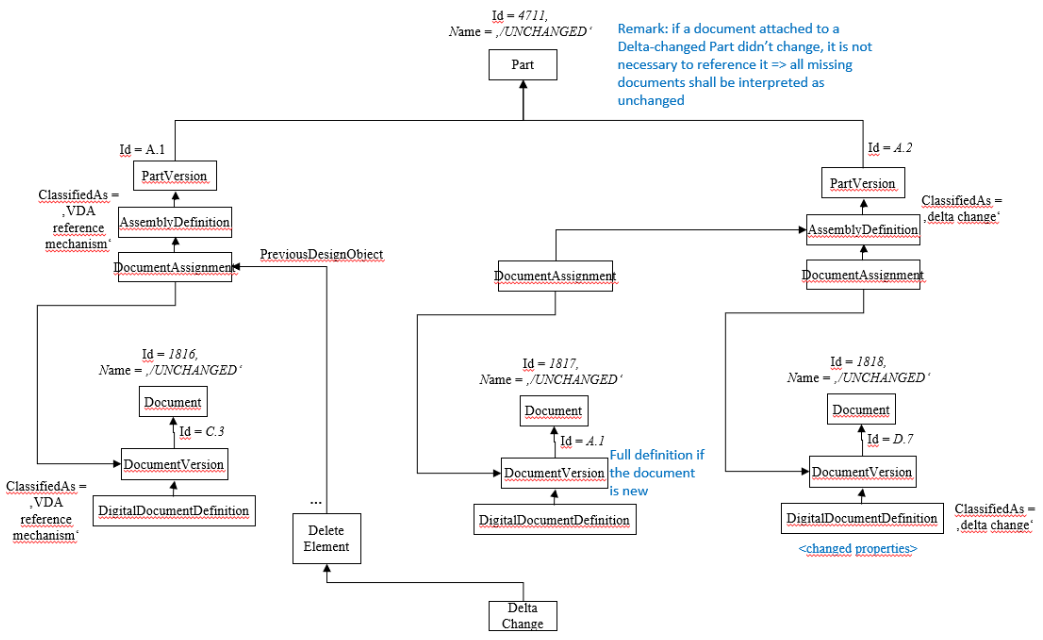

Id = 4711,
Name = 'UNCHANGED'
Part
Remark: if a document attached to a Delta-changed Part didn't change, it is not necessary to reference it => all missing documents shall be interpreted as unchanged
Id = A.1
PartVersion
ClassifiedAs = 'VDA reference mechanism'
AssemblyDefinition
DocumentAssignment
PreviousDesignObject
Id = 1816,
Name = 'UNCHANGED'
Document
Id = C.3
DocumentVersion
ClassifiedAs = 'VDA reference mechanism'
DigitalDocumentDefinition
...
Delete Element
Id = A.2
PartVersion
ClassifiedAs = 'delta change'
AssemblyDefinition
DocumentAssignment
DocumentAssignment
Id = 1817,
Name = 'UNCHANGED'
Document
Id = A.1
DocumentVersion
Full definition if the document is new
DigitalDocumentDefinition
Id = 1818,
Name = 'UNCHANGED'
Document
Id = D.7
DocumentVersion
DigitalDocumentDefinition
ClassifiedAs = 'delta change'
<changed properties>
Delta Change

Figure 68: Example of the minimal instanciation of Delta Exchange of a new part version with one document deleted, one added and one modified

Postprocessor Recommendation:

T.B.S.

9.5.1 Template “DeltaChange”

A DeltaChange is a mechanism to describe changes in product data. It collects the change activities that describe the differences between two (or more) objects.

The single changes on attribute level are collected by the attribute ChangeActivities.

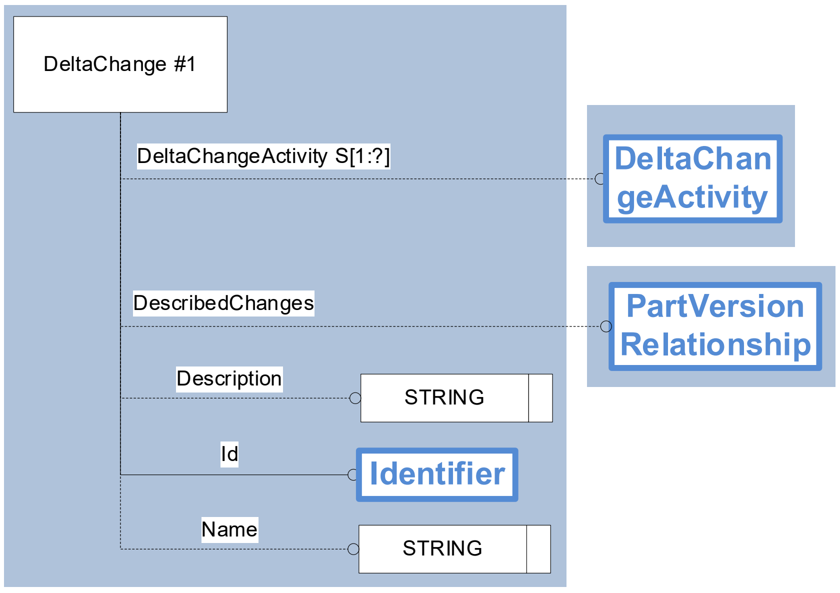


Figure 69: Template "DeltaChange"

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

```
<Classification uid="delta">
  <Class>
    <ClassString>delta change</ClassString>
  </Class>
  <Role>delta change</Role>
</Classification>
<Classification uid="ref_pv">
  <Class>
    <ClassString>reference</ClassString>
  </Class>
  <Role>VDA reference mechanism</Role>
</Classification>
…
<DeltaChange uid="dc-1">
  <ChangeActivities>
    <DeltaChangeActivity uid="dca-1" xsi:type="bom:AddElement">
      <CurrentDesignObject>
        <ViewOccurrenceRelationship uidRef="ID_1708"/>
      </CurrentDesignObject>
    </DeltaChangeActivity>
    <DeltaChangeActivity uid="dca-2" xsi:type="bom:DeleteElement">
      <PreviousDesignObject>
        <ViewOccurrenceRelationship uidRef="ID_1738"/>
      </PreviousDesignObject>
    </DeltaChangeActivity>
    <DeltaChangeActivity uid="dca-3" xsi:type="bom:ModifyElement">
      <AttributeName>Placement</AttributeName>
      <CurrentDesignObject>
        <ViewOccurrenceRelationship uidRef="ID_1726d"/>
      </CurrentDesignObject>
    </DeltaChangeActivity>
  </ChangeActivities>
  <DescribedChange uidRef="pvr-1"/>
  <Id>
    <Identifier uid="dc-id1" id="Sequence#1" idRoleRef="ID_804" idContex-
tRef="ID_805"/>
  </Id>
</DeltaChange>
…
<Part uid="ID_1998">
  <Id>
    <Identifier uid="ID_1995" id="120241-TSI-VDA" idRoleRef="ID_804" idCon-
textRef="ID_805"/>
  </Id>
  <Name>
    <CharacterString>/UNCHANGED</CharacterString>
  </Name>
  <PartTypes>
    <ClassString>assembly</ClassString>
  </PartTypes>
  <Versions>
    <PartVersion uid="ID_1997">
      <Id>
        <Identifier uid="ID_1996" id="A.1" idRoleRef="ID_804" idContex-
tRef="ID_1995"/>
      </Id>
      <Views>
        <PartView uid="ID_899" xsi:type="bom:AssemblyDefinition">
          <ClassifiedAs>
            <Classification uidRef="delta"/>
          </ClassifiedAs>
          <InitialContext uidRef="ID_829"/>
          <ViewOccurrenceRelationship uid="ID_1738" xsi:type="bom:NextAssem-
blyOccurrenceUsage">
            <Related uidRef="ID_1740"/>
            <RelationType>
              <ClassString>next assembly occurrence</ClassString>
            </RelationType>
          </ViewOccurrenceRelationship>
          <ViewOccurrenceRelationship uid="ID_1726" xsi:type="bom:NextAssem-
blyOccurrenceUsage">
            <Related uidRef="ID_1728"/>
            <RelationType>
              <ClassString>next assembly occurrence</ClassString>
            </RelationType>
            <Placement>
              <CartesianTransformation uid="ID_1727">
<RotationMatrix>1.000000000000080E+00 -7.462707997618930E-13 -
1.763715855220600E-12 7.404873296037771E-13 1.000000000000090E+00 -
5.210185581589930E-13 1.753736316324390E-12 5.239827668972550E-13
1.000000000000090E+00</RotationMatrix>
<TranslationVector>6.437129533499321E-01 2.800987384232940E-01 -
5.673703299180340E-02</TranslationVector>
              </CartesianTransformation>
            </Placement>
          </ViewOccurrenceRelationship>
9
      </Views>
      <PartVersionRelationship uid="pvr-1">
        <Related uidRef="ID_1997d"/>
        <RelationType>
          <ClassString>sequence</ClassString>
        </RelationType>
      </PartVersionRelationship>
    </PartVersion>
    <PartVersion uid="ID_1997d">
      <Id>
        <Identifier uid="ID_1996d" id="A.2" idRoleRef="ID_804" idContex-
tRef="ID_1995"/>
      </Id>
      <Views>
        <PartView uid="ID_899d" xsi:type="bom:AssemblyDefinition">
          <ClassifiedAs>
            <Classification uidRef="delta"/>
          </ClassifiedAs>
          <InitialContext uidRef="ID_829"/>
          <ViewOccurrenceRelationship uid="ID_1726d" xsi:type="bom:NextAssem-
blyOccurrenceUsage">
            <Related uidRef="ID_1728"/>
            <RelationType>
              <ClassString>next assembly occurrence</ClassString>
            </RelationType>
            <Placement>
              <CartesianTransformation uid="ID_1727d">
```

```
<RotationMatrix>1.000000000000080E+00 -7.462707997618930E-13 -
1.763715855220600E-12 7.404873296037771E-13 1.000000000000090E+00 -
5.210185581589930E-13 1.753736316324390E-12 5.239827668972550E-13
1.000000000000090E+00</RotationMatrix>
<TranslationVector>6.437129533499321E-00 2.800987384232940E-00 -
5.673703299180340E-01</TranslationVector>
              </CartesianTransformation>
            </Placement>
          </ViewOccurrenceRelationship>
          <ViewOccurrenceRelationship uid="ID_1708" xsi:type="bom:NextAssem-
blyOccurrenceUsage">
            <Related uidRef="ID_1710"/>
            <RelationType>
              <ClassString>next assembly occurrence</ClassString>
            </RelationType>
            <Placement>
              <CartesianTransformation uid="ID_1709">
<RotationMatrix>1 -5.423864602588700E-13 -1.360506705124480E-12
5.423864602583161E-13 1 -4.042063721496210E-13 1.360506705124780E-12
4.042063721488900E-13 1</RotationMatrix>
<TranslationVector>6.437129533580320E-01 2.800987384249200E-01 -
5.673703299222880E-02</TranslationVector>
              </CartesianTransformation>
            </Placement>
          </ViewOccurrenceRelationship>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
…
<Part uid="ID_2002">
  <Id>
    <Identifier uid="ID_1999" id="120242-TSI-VDA" idRoleRef="ID_804" idCon-
textRef="ID_805"/>
  </Id>
  <Name>
    <CharacterString>/NULL</CharacterString>
  </Name>
  <PartTypes>
    <ClassString>piece part</ClassString>
  </PartTypes>
  <Versions>
    <PartVersion uid="ID_2001">
      <Id>
        <Identifier uid="ID_2000" id="A.1" idRoleRef="ID_804" idContex-
tRef="ID_1999"/>
      </Id>
      <Views>
        <PartView uid="ID_1602">
          <ClassifiedAs>
            <Classification uidRef="ref_pv"/>
          </ClassifiedAs>
          <Id id="design"/>
          <InitialContext uidRef="ID_829"/>
          <Occurrence uid="ID_1740" xsi:type="bom:SingleOccurrence">
            <Id id="Lower piston_i"/>
          </Occurrence>
        </PartView>
      </Views>
    </PartVersion>
  </Versions>
</Part>
…
```

Entity DeltaChange	Attribute type
ChangeActivities	OPTIONAL SET[1:?] OF DeltaChangeActivity
ClassifiedAs	OPTIONAL SET[1:?] of Classification
DescribedChange	OPTIONAL DeltaChangeRelationshipSelect
Description	OPTIONAL DescriptorSelect
Id	IdentifierSelect
Name	OPTIONAL DescriptorSelect
DocumentAssignment	OPTIONAL SET[1:?] of DocumentAssignment
InformationUsageRightAssignment	OPTIONAL SET[1:?] of InformationUs-ageRightAssignment

Table 65: "DeltaChange" Attributes

Attribute recommendations

- ***ChangeActivities***: the additions, deletions, or modifications caused by the DeltaChange. The value of this attribute need not be specified. Use “DeltaChangeActivity” template.
- ***DescribedChange***: a relationship that identifies the previous group of objects and current group of objects. For example, consider two PartVersion objects A and B which are related by a PartVersionRelationship with RelationType 'sequence'. The attribute DescribedChange of DeltaChange references this PartVersionRelationship. The value of this attribute need not be specified but is recommended in order to clearly identify the previous version of the object (for example for AddElement/ModifyElement) and the new version of the object (for example for DeleteElement) since these objects are not referenced explicitly.
- ***Description***: the text or the set of texts that provides further information about the DeltaChange. The value of this attribute need not be specified. Use “Description” template.
- ***Id:*** the identifier for the DeltaChange. Use “Identifier” template (see 4.6.6).
- ***Name***: the words or set of words by which the DeltaChange 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.

Preprocessor Recommendation:

- The use of DeltaChange is mandatory in case of a a DeltaChangeActivity of type DeleteElement (see next section), otherwise it is optional (as long as no ordering of the changes has to be specified, see below).

- If the Delta Exchange is triggered each time a part reaches a given lifecycle state, it may be of interest to map only one instance of DeltaChange per STEP file and to set the DeltaChange.Id as a unique sequence number over all the data supplies between two partners, so that the receiver can make sure that he didn't skip any delta supply files.
- The use of the attribute DescribedChange assumes that one DeltaChange object is mapped for each changed object. In the above example (PartVersion A.1->A.2 and A.2->A.3), one DeltaChange from PartVersion A.1->A.2 and one DeltaChange from PartVersion A.2->A.3). In this case, it is recommended to specify the ordering of changes. For this purpose, a JIRA issue TCSC410303-1509 has been created to introduce an DeltaChangeRelationship between two DeltaChanges.

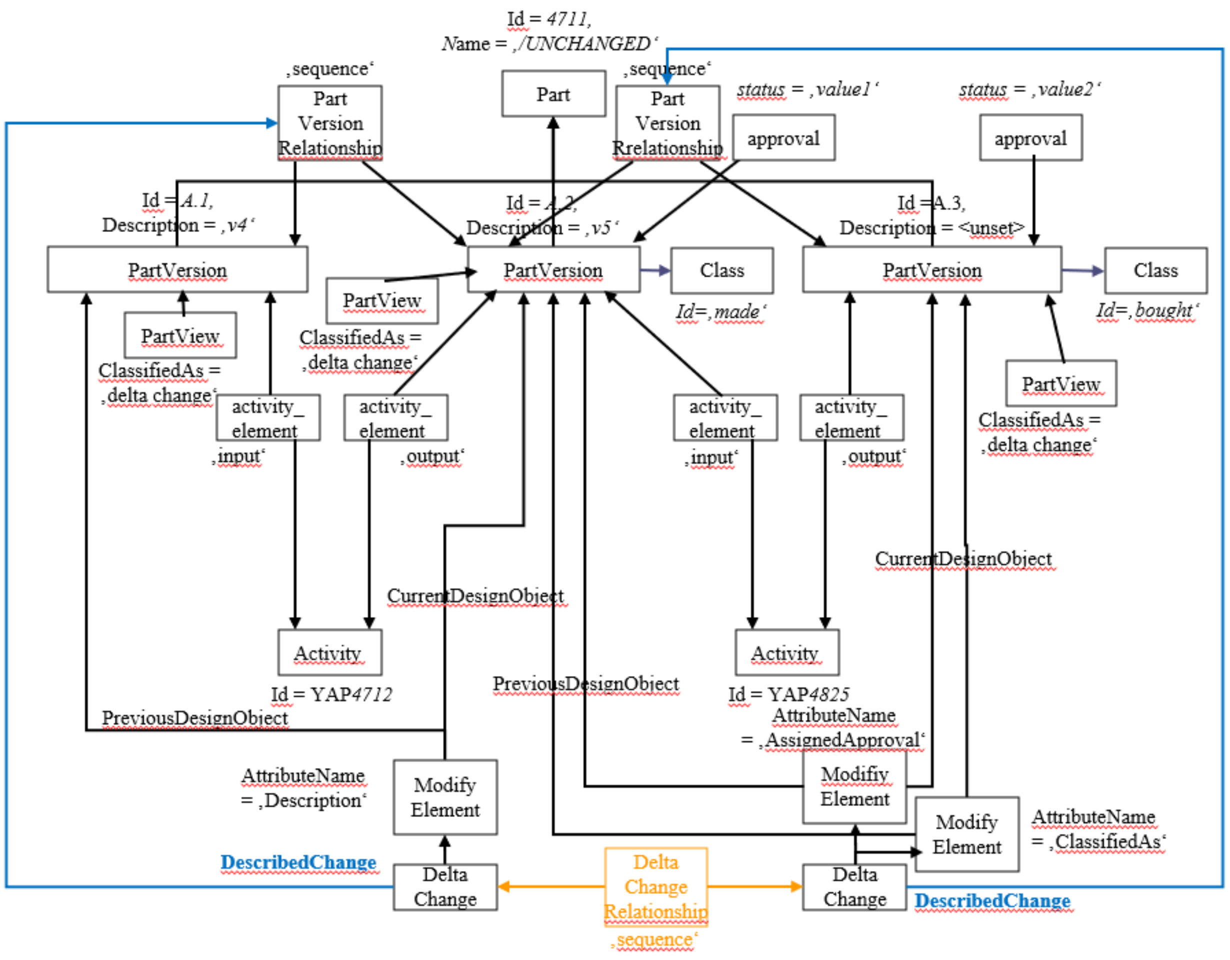


Figure 70: Example of a Delta Exchange for two sequential part changes which order is controlled between two DeltaChanges

- Currently, the Select type used for the definition of DescribedChange is very limited compared to the 100 objects defined in DeltaChangeManagementObjectSelect. For this purpose, a JIRA issue TCSC410303-1511 has been created to add all the relationship objects that are relating objects defined in DeltaChangeManagementObjectSelect.

Postprocessor Recommendation:

T.B.S.

9.5.2 Template “DeltaChangeActivity”

A DeltaChangeActivity is the representation of a detailed change within a DeltaChange.

There are 4 kinds of delta change activities:

- A **DeleteElement** is a type of DeltaChangeActivity that identifies one or several objects that have been deleted.
  EXAMPLE Version A of a part has a subcomponent C1. In version B of the part this subcomponent is deleted. This change can be described by referencing with the attribute PreviousDesignObject of DeleteElement a NAOU representing the deleted subcomponent C1 in version A.
- An **AddElement** is a type of DeltaChangeActivity that identifies one or several objects that have been added.
  EXAMPLE In version B of a part a new subcomponent C1 is added. This change can be described by referencing with the attribute CurrentDesignObject of AddElement a NAOU representing the new subcomponent C1.
- A **ModifyElement** is a type of DeltaChangeActivity that identifies one or several object that have been modified.
  EXAMPLE The Placement of one of the subcomponents C1 has been modified from version A to version B of a part. This change can be described by referencing with the attribute CurrentDesignObject of ModifyElement a NAOU representing the modified Placement value in version B.
- A **ModifySingleElement** is a type of DeltaChangeActivity that identifies an object that has been modified.
  EXAMPLE The Placement of one of the subcomponents C1 has been modified from version A to version B of a part. This change can be described by referencing with the attribute PreviousDesignObject of ModifySingleElement a NAOU representing the original Placement value of version A, and with the attribute CurrentDesignObject of ModifySingleElement a NAOU representing the new Placement value of version B.

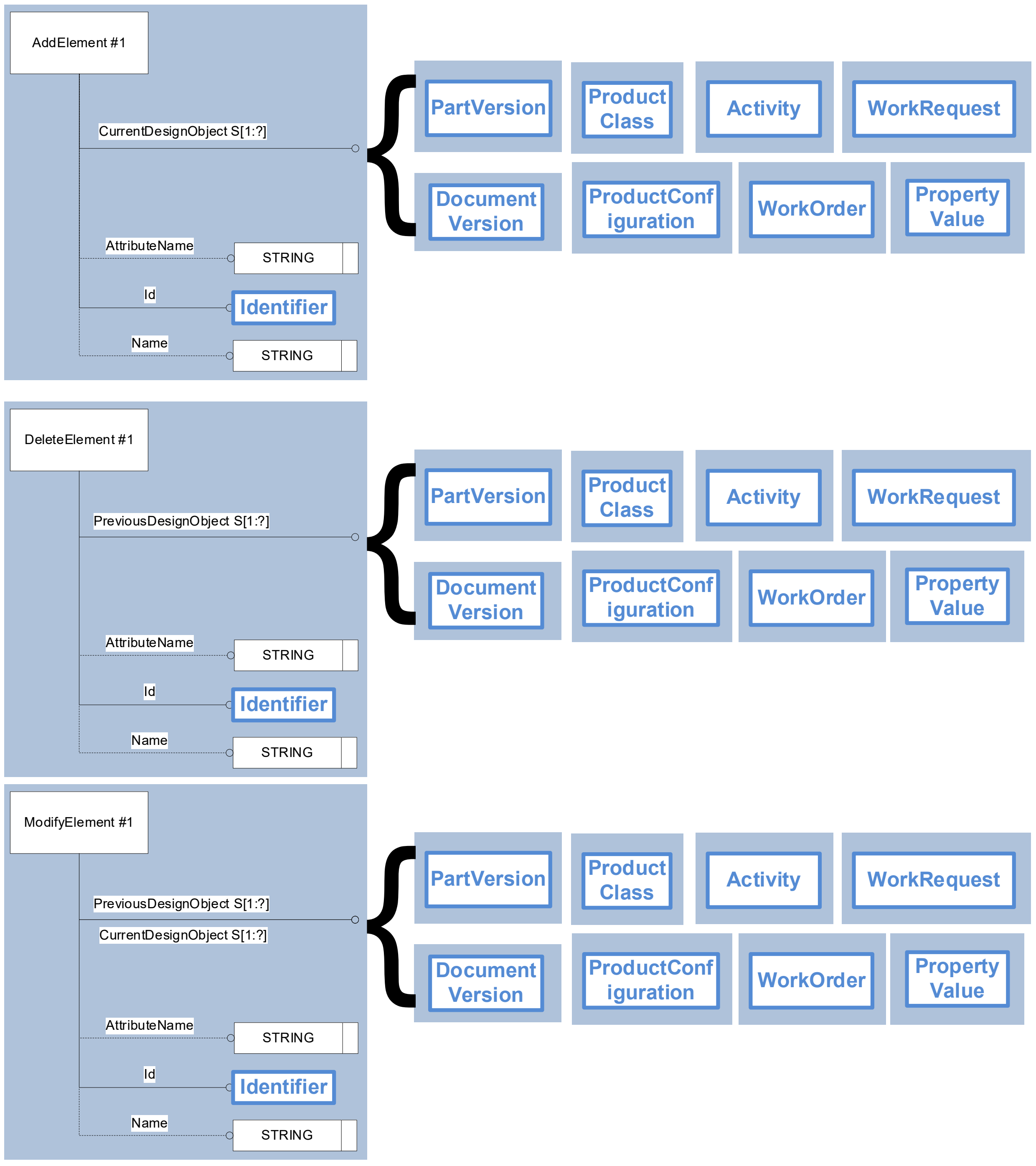


Figure 71: Template "DeltaChangeActivity"

Entity DeltaChangeActivity (and its subtypes)	Attribute type
AttributeName	OPTIONAL STRING
ChangeElementSequence	OPTIONAL SET[1:?] OF ChangeElementSequence
ChangeLocationInAggregateAttribute	OPTIONAL LIST[1:?] OF INTEGER

Entity DeltaChangeActivity (and its subtypes)	Attribute type
Except for DeleteElement: CurrentDesignObject	Except for ModifySingleElement: SET[1:?] OF… DeltaChangeManagementObjectSelect
except for AddElement: PreviousDesignObject	Except for ModifySingleElement: SET[1:?] OF… DeltaChangeManagementObjectSelect

Table 66: "DeltaChangeActivity" Attributes

Attribute recommendations

- ***AttributeName***: the attribute of the previous and current objects whose values are changed. The value of this attribute need not be specified. Use IdentifierString type.
- ***CurrentDesignObject***: the set of objects that have been added (AddElement) or modified (ModifyElement).
- ***PreviousDesignObject***: the previous object that has been modified (ModifyElement) or deleted (DeleteElement).
- 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:

- Since ModifySingleElement does not enable to bind multiple changes (unlike AddElement, ModifyElement and DeleteElement) and since it obliges to reference both objects (before and after the change) while in most use cases, it is sufficient to exchange only the object after the change, the use of ModifySingleElement is not recommended.
- AddElement may be omitted (an object not being associated to a DesignChange object is implicitly new).
- ModifyElement may be omitted if the file does not contain several sequential changes (PartVersion A.1->A.2 and A.2->A.3). During import, possibly the receiver system may compare DateTimeAssignment having Role='update' (see section 4.6.16) with the timestamp in the target PDM system.
- It is recommended to set AttributeName in case several sequential changes might have occurred on the same object since the last Delta Exchange (PartVersion A.1->A.2 and A.2->A.3). For example, the PartVersionDescription changed between A.1 and A.2, but not between A.2 and A.3 => Description (optional attribute) will not be set for A.3 => without setting AttributeName to 'Description' for the first nor to the second ModifyElement, the receiver interface might interpret that the Description has been emptied in A.3
- For CurrentDesignObject/PreviousDesignObject, it is recommended to reference a PDM business object that gets versioned through the change. For example, PartVersion, DocumentVersion, ProductClass, ProductConfiguration, Activity, WorkOrder, WorkRequest, Observation, … or even more in some PDM systems, like NextAssemblyOccurrenceUsage in 3DExp… More granular AP242 objects are not recommended, except to map the change of customized properties from a non-unset value to an unset value (see section 14.3).

Postprocessor Recommendation:

T.B.S.

9.6 External Element References (EER)

An External Element Reference (EER) consists of a referencing and referenced element.Three kinds of EERs are planned at the moment:

- From AP242 XML file to another AP242 XML file
    - Example of use case: nested files, see section 9.3
    - Here a new referencing mechanism is currently being defined in Part 15 Edition 2 (to be introduced with AP242 Edition 4), less redundant than the current incremental exchange and nesting mechanism and more general than the RepresentationItem.External and SameAs mechanisms
- From AP242 XML file to Part 21 file
    - Example of use case: assembly PMIs, see [AP242-PMI]
    - Here the anchor section mechanism of Part 21 is recommended,
      combined with the new referencing mechanism for AP242 XML mentioned above.
- From AP242 XML file to JT file
    - Example of use case: assembly PMIs, see [AP242-PMI]
    - Here the JT monikers need to be generalized to all JT objects (not only XTBodies and XTFaces),
      combined with the new referencing mechanism for AP242 XML mentioned above.

9.6.1 Part 15 Edition 2 Reference Mechanism

Coming together with AP242 Domain Model Edition 4, using this mechanism, a reference to any object of any type may be defined. There are three XML cases:

XML Case	Everything in one file	External Reference

#1: No containment	`<Part uid="ID_1978">`
	`…`
	`  <Versions>`
	`    <PartVersion uid="ID_1977">`
	`…`
	`      <Views>`
	`        <PartView uid="ID_1976">`
	`…`
	`          <ShapeElement uid="_2667">`
	`…`
	`            <RepresentedGeometry`
	`uidRef=„ID_1908"/>`
	`…`
	`          </ShapeElement>`
	`…`
	`        </PartView>`
	`      </Views>`
	`    </PartVersion>`
	`  </Versions>`
	`</Part>`
	
	`<RepresentationItem xsi:type="n0:AdvancedFace"`
	`uid="ID_1908" uuid="550e8400-e29b-11d4-a716-`
	`446655440000">`
	`…`
	`</RepresentationItem>`
		`<Part uid="ID_1978">`
		`…`
		`  <Versions>`
		`    <PartVersion uid="ID_1977">`
		`…`
		`      <Views>`
		`        <PartView uid="ID_1976">`
		`…`
		`          <ShapeElement uid="_2667">`
		`…`
		`            <RepresentedGeometry`
		`uidRef=„ID_1908"/>`
		`…`
		`          </ShapeElement>`
		`…`
		`        </PartView>`
		`      </Views>`
		`    </PartVersion>`
		`  </Versions>`
		`</Part>`
		
		`<ExternalRefBaseObject uid="ID_869" typeRef="`
		`AdvancedFace" extIdRef="550e8400-e29b-11d4-a716-`
		`446655440000" URL="./120236-TSI-VDA-A.1.stpx">`
		`</ExternalRefBaseObject>`

#2: Simple containment	`<Part uid="ID_1978">`
	`…`
	`  <Versions>`
	`    <PartVersion uid="ID_1977">`
	`…`
	`      <Views>`
	`        <PartView uid="ID_1976" uuid="550e8400-`
	`e29b-11d4-a716-446655440000">`
	`…`
	`          <Occurrence uid="ID_1975"`
	`xsi:type="bom:SingleOccurrence">`
	`            <Id id="Rear Brake"/>`
	`          </Occurrence>`
	`…`
	`        </PartView>`
	`      </Views>`
	`    </PartVersion>`
	`  </Versions>`
	`</Part>`
		`<ExternalRefBaseObject uid="ID_869"`
		`typeRef="bom:PartView" extIdRef="550e8400-e29b-`
		`11d4-a716-446655440000" URL="./120236-TSI-VDA-`
		`A.1.stpx">`
		`  <SubObject uid="ID_870" xsi:type="bom:Single-`
		`Occurrence">`
		`    <Id id="Rear Brake"/>`
		`  </SubObject>`
		`</ExternalRefBaseObject>`

#3: Containment over a complex element	`<Part uid="ID_1978" uuid="550e8400-e29b-11d4-`
	`a716-446655440000">`
	`…`
	`  <Versions>`
	`    <PartVersion uid="ID_1977">`
	`…`
	`      <Views>`
	`        <PartView uid="ID_869">`
	`          <Id id="design"/>`
	`          <InitialContext uidRef="ID_829"/>`
	`          <Occurrence uid="ID_1504"`
	`xsi:type="bom:SingleOccurrence">`
	`            <Id id="Rear Brake"/>`
	`          </Occurrence>`
	`        </PartView>`
	`      </Views>`
	`    </PartVersion>`
	`  </Versions>`
	`</Part>`
		`<ExternalRefBaseObject uid="ID_869"`
		`typeRef="bom:Part" extIdRef="550e8400-e29b-11d4-`
		`a716-446655440000" URL="./120236-TSI-VDA-`
		`A.1.stpx">`
		`  <NamedSubObject elementName="Versions">`
		`    <SubObject uid="ID_870"`
		`xsi:type="bom:PartVersion">`
		`…`
		`      <Views>`
		`        <PartView uid="ID_869">`
		`          <Id id="design"/>`
		`          <InitialContext uidRef="ID_829"/>`
		`          <Occurrence uid="ID_1504"`
		`xsi:type="bom:SingleOccurrence">`
		`            <Id id="Rear Brake"/>`
		`          </Occurrence>`
		`        </PartView>`
		`      </Views>`
		`    </SubObject>`
		`  </NamedSubObject>`
		`</ExternalRefBaseObject>`

In all cases:

- the attribute ‘uuid’ has to be set for at least all the ‘original’ objects that shall be externally referenced (via ‘extIdRef’ of type UUID). It has to comply to the syntax pattern of a UUID v5. The content of uuid depends on the company’s agreement and is independent from the usage of the reference mechanism

Entity ExternalRefBaseObject	Attribute type
SubObject	OPTIONAL SET[1:?] OF BaseObject
NamedSubObject	OPTIONAL SET[1:?] OF NamedSubObject
extIdRef	xsd:string or UUID
typeRef	xsd:string
URL	xsd:string

Table 67: "ExternalRefBaseObject" Attributes

Attribute recommendations

- ***SubObject***: an embedded object (any BaseObject) in case of a simple containment. The value of this attribute need not be specified. All the XSD restrictions and all recommendations defined for the BaseObject shall apply.
- ***NamedSubObject***: an embedded object in case of a containment over a complex element. The value of this attribute need not be specified. See below.
- ***extIdRef***: the identifier (or UUID) of the referenced object in another file (AP242 Domain Model, AP242 P21, JT, CAD native, …). The content of the identifier or UUID (tool specific, centrally managed like UUID v4/v5) and its persistence (unlike the uid) is not in scope of this document.
- ***typeRef***: the type of the referenced object (without any namespace). It shall be a valid object name within the DomainModel and shall correspond to the type of the referenced object in the referenced file (for example AdvancedFace in case of advanced_face in a P21 file or XTFace in a JT file)
- ***URL***: the location where the referenced object can be found. In some cases like the usage of UUID v5 identifiers, this attribute need not be specified. Otherwise, it shall contain the relative path to the file, or an absolute path (for example in the case of an URL). 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).

Entity NamedSubObject	Attribute type
SubObject	SET[1:?] OF BaseObject
elementName	xsd:string

Table 68: "NamedSubObject" Attributes

Attribute recommendations

- ***SubObject***: an embedded object (any BaseObject) in case of a containment over a complex element. All the XSD restrictions and all recommendations defined for the BaseObject shall apply.
- ***elementName***: the name of the complex element used for the containment.

Preprocessor Recommendations:

- typeRef shall be a valid element name of the AP242 DomainModel, no matter which format and data model the referenced file has (see next sections for more details)
- If both referencing and referenced files have the same format (AP242 DomainModel XML), the value of typeRef and the type of the referenced Object shall be identical. If the referenced file has another format (where the referenced element can be uniquely identified), the value of typeRef and the type of the referenced object shall be equivalent (see next sections for more details)
- Using the supertype or one of its subtype for typeRef depends on the situation.

  Example: ActivityRelationship.Related for example can either reference <Activity uid="xxx"> or <Activity uid="xxx" xsi:type="ns:PlannedActivity">. The same applies to typeRef: if the referenced activity is a planned one, the subtype PlannedActivity shall be used
- SubObject shall not be a BaseRootObject, since is shall be only used in case of containments.
- The type of SubObject contained in ExternalRefBaseObject shall be allowed as containment of an element defined by typeRef or shall correspond to the type of corresponding elementName in the type typeRef (in case of a containment over a complex element)

  Example: SingleOccurrence (rsp. Occurrence) is contained in PartView
- Attribute elementName of NamedSubObject shall be the name of an embracing element in the type given by typeRef

  Example: Versions is the name of an embracing element in Part
- The type of the SubObject contained in NamedSubObject shall correspond to the type of corresponding elementName in the type typeRef

  Example: The type of the elements in NamedSubObject with elementName Versions in typeRef Part is PartVersion

  This mechanism is not needed recursively, e.g., if a new PartView is added to a PartVersion, this PartVersion should be externally referenced rather than referencing the Part => always and only the direct container of a new contained object shall be referenced
- Within one stpx file, the same object shall not be provided as ExternalRefBaseObject and as ‚original' object (since it does not make sense), even not if they have different UUIDs (it would violate the uniqueness rule of the UUID).
  This applies also in case of simple/complex containment

  Example: a PartView x shall not be present as ExternalRefBaseObject and as (Named)SubObject

For this reason, within one stpx file, all the containments of kind SubObject and NamedSubObject that apply to the same referenced object shall be placed in the one ExternalRefBaseObject

- In some special cases, many elements of a referenced object are needed. In this case, each referenced element shall be mapped as a distinct ExternalRefBaseObject.

  Example: new PartVersion to a referenced Part:

  - the Part itself shall be mapped as ExternalRefBaseObject, containing the new PartVersion via NamedSubObject

  - the Part.Id shall be mapped as a disctinct ExternalRefBaseObject, referenced by PartVersion.Id.idContextRef according to section 4.6.6

- Unlike in the 'original' objects, the order of the elements in an ExternalRefBaseObject is not relevant, but it may be convenient to stick to the order within the 'original' object for manual comparisons

- The upper cardinality of the original XML element shall not be violated

  Example: ‚Versions' may have multiple elements, but ‚InitialContext' may not

- The lower cardinality '1' of the original XML element doesn't have to be fulfilled, since a mandatory element has been already defined in the original object and doesn't need to be confirmed within the referenced object (except if it shall change to a new value).

  Example: Part.Name need not to be provided if it didn't change

- Since ExternalRefBaseObject is a kind of BaseRootObject, it inherits from the attribute 'uuid' => theoretically, a ExternalRefBaseObject could be referenced, but until a use case is identified, this is not recommended

- Even if (Named)SubObject would allow to map some DeltaChange semantic like:
  - In case the upper cardinality of the original XML element is ‚unbounded': previously provided elements in the original object are kept unchanged. The new ones are added to them or replacing them in case they have the same key attributes
  - In case the upper cardinality of the original XML element is 1: the previously provided value gets replaced by it.

  it is not recommended to use it that way

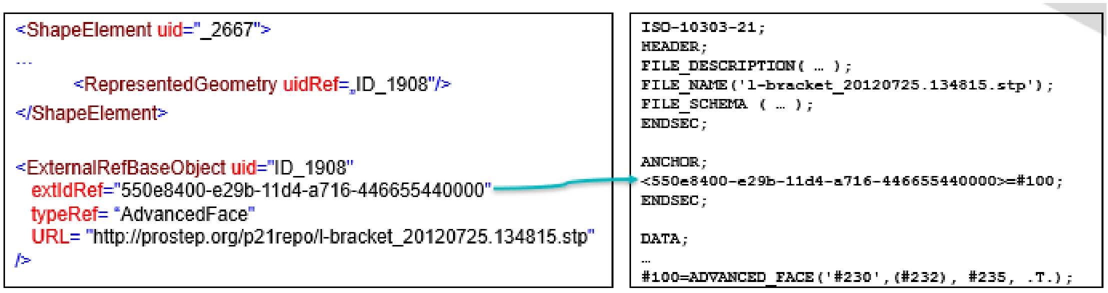
- It is not recommended to use (Named)SubObject if the value is identical to the one provided in the original object, or when the key attributes are the same => (Named)SubObject shall be used only to set a previously unset element or to add an element to a sequence of elements

Postprocessor Recommendation:

T.B.S.

9.6.2 References into a Part 21 Geometry File

The recommendation is to use Anchor references as defined in Part 21 Edition 3 (see the "Recommended Practices for Persistent IDs for Design Iteration and Downstream Exchange" for more details [PID]).



```
<ShapeElement uid="_2667">
...
        <RepresentedGeometry uidRef=„ID_1908"/>
</ShapeElement>

<ExternalRefBaseObject uid="ID_1908"
   extIdRef="550e8400-e29b-11d4-a716-446655440000"
   typeRef= "AdvancedFace"
   URL= "http://prostep.org/p21repo/l-bracket_20120725.134815.stp"
/>
```

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION( … );
FILE_NAME('l-bracket_20120725.134815.stp');
FILE_SCHEMA ( … );
ENDSEC;

ANCHOR;
<550e8400-e29b-11d4-a716-446655440000>=#100;
ENDSEC;

DATA;
…
#100=ADVANCED_FACE('#230',(#232), #235, .T.);
```

Figure 72: *Example of an EER to an element within a Part 21 geometry file*

Preprocessor Recommendations:

- Part 21 specifies: “The identified ANCHOR_ITEM shall meet the requirements for an entity or value instance belonging to the file_schema of this exchange structure.”

  The object type given in ExternalRefBaseObject.typeRef (according to the AP242 DomainModel) shall be equivalent to the type of the referenced object in the AP242 MIM.

  For example, if the AP242 XML references an AdvancedFace into a P21 file, the referenced UUID shall be an Anchor which points to an ADVANCED_FACE’.

9.6.3 References into a JT Geometry File

Since Monikers are currently not (yet) available on all JT elements (only on XTBodies and XTFaces, but not on XTEdges, PMI Annotations, ..), the current workaround is to:

- maintain a user defined string property called ‘Anchor’ as JtkProperty (would have to be set with a stable Persistent ID as defined in [PID]). This works for all XTEntities, but nor for PMI annotations
- for PMI annotations, maintain a user defined property called ‘Anchor’ set to ‘invisible’

An issue has been submitted to Siemens in order to enhance the usage of Monikers to all necessary JT elements. This is planned to be available until the end of 2024.

9.6.4 Handling of updates in the referencing or referenced file

In all cases, the UUID of an ‘original’ object shall be set prior to (or at the same time as) its usage as reference. This sounds logical, but implies clear rules when creating, keeping stable and updating references.

3 scenarios are identified:

a. Referencing and referenced file are created together
   - Here the stability of the Ids is not critical
   - But this is not practicable for most use cases, like assembly level PMIs
b. Referencing file is created later (possibly via another tool) and references further elements within an already existing file
   - requires that all the referenced elements have <u>already</u> a stable Id (before they ever get referenced)
   - or that it may be easily generated out of it (Anchor, Moniker), but this requires to enrich/update the referenced file. Even using Anchors (separate section within

the P21 file => can be added without editing the data section), the Part 21 file has to be edited.

c. Referenced file gets modified

- requires that the referenced elements keep the same Id (as long as the changes do not impact the definition of the referenced items)
- if the change is too big to keep the references unchanged, the referencing files need to be updated (not only the extIdRef, but also their semantical content)

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 Domain Model XML entities and attributes

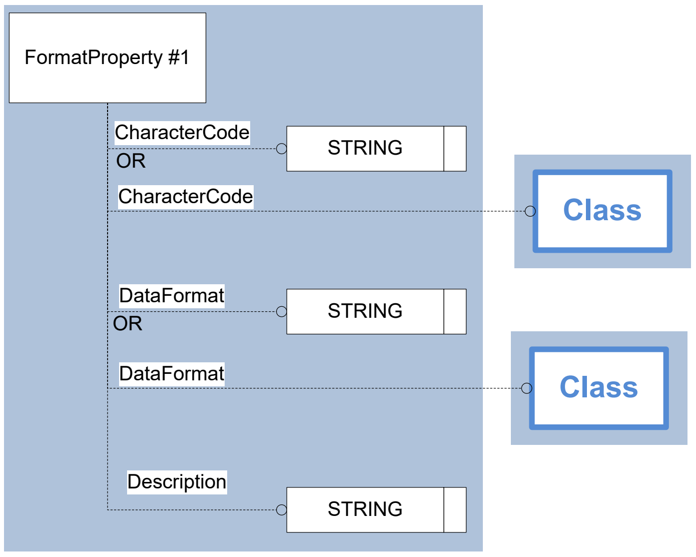


Figure 73: Template "FormatProperty"

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 69: "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

CharacterCode	Explanation
'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[1]
'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 Domain 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[1]
'ISO 10646'	The coded character set used to encode the document data according to ISO/IEC 10646.

- ***DataFormat***: the convention that was used to structure the information in the characterized object. 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

[1] **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	Explanation
'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 Recommendations:

- DataFormat shall be always set, since a FormatProperty without DataFormat has no added value.
- In case of a non-native proprietary format (for example CATIA CGR), the DataFormat shall refer to a Class and ExternalClassSystem (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 Domain 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
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

DataFormat	Explanation
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 Domain Model XML syntax)

```
<FormatProperty uid="ffp--AP242DOMAINMODEL">
  <CharacterCode>
    <ClassString>UTF-8</ClassString>
  </CharacterCode>
  <DataFormat>
    <ClassString>ISO 10303-242 Domain Model XML</ClassString>
  </DataFormat>
</FormatProperty>
```

10.2 Template "ContentProperty"

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

The Instance Model: AP242 Domain Model XML entities and attributes

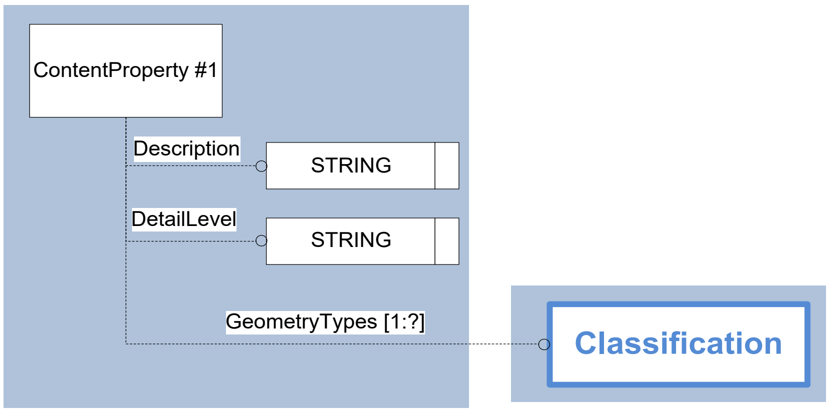


Figure 74: Template "ContentProperty"

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

RealWorldScale	OPTIONAL NumericalValue

Table 70: "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 “Classification” template (see 4.6.5). Use ClassString type for Classification.Class if one of the values below is used:

  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
‘exact geometry’	The document contains a precise three-dimensional B-Rep model made of either wireframe, surface and/or solid geometry. This value can be used if the PDM system doesn’t get more details about the kind of B-Rep geometry from the CAD system.
‘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 Informatrion 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 (laminate table)
‘explicit composite’	The document contains the explicit representation of a solid composite part. This value is typically used together with ‘tessellated geometry’.

STEP geometry files as well as many native formats typically contain either exact geometry (STEP B-Rep, or native CAD format such as e.g. CATPart from CATIA), or tessellated geometry (STEP Tessellated, or derived CAD format such as e.g. CGR from CATIA), though they may contain both in some cases. Other native or neutral formats, such as e.g. JT, usually combine exact and tessellated geometry into one file. In any case, the references files need to be characterized appropriately to ensure the correct file for the current use case or application will be provided.

If the DigitalFile contains another AP242 Domain 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 overlayed 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.

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

```
<Classification uid="gtc--1">
  <Class>
    <ClassString>surface geometry</ClassString>
  </Class>
  <Role>geometry type</Role>
</Classification>
<Classification uid="gtc--3">
  <Class>
    <ClassString>solid geometry</ClassString>
  </Class>
  <Role>geometry type</Role>
</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 Domain Model XML entities and attributes

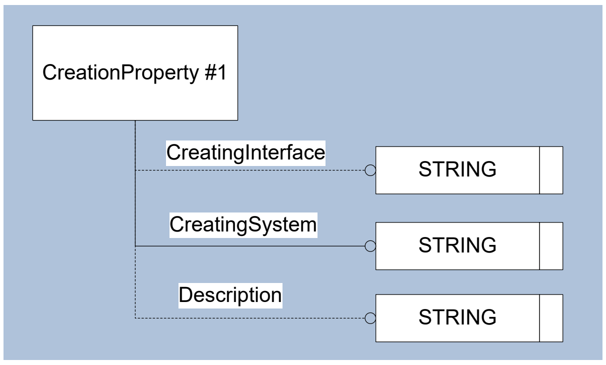


Figure 75: Template "CreationProperty"

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

Table 71: "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 unset.
- ***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 Domain 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.

The Instance Model: AP242 Domain Model XML entities and attributes

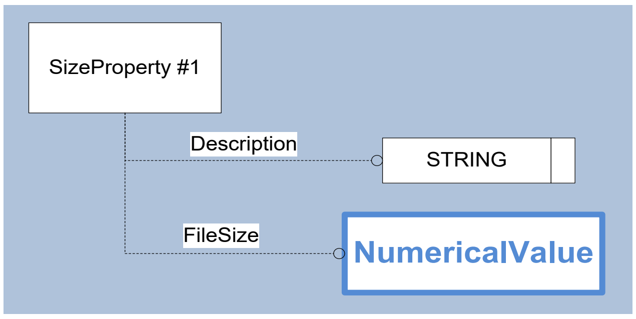


Figure 76: Template "SizeProperty"

Entity SizeProperty	Attribute type
Description	OPTIONAL DescriptorSelect
FileSize	OPTIONAL ValueWithUnit
PageCount	OPTIONAL ValueWithUnit

Table 72: "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). Since the semantic is handled here by the attribute naming, use PropertyDefinitionString='file size' for FileSize.Definition and leave NumericalValue.Name unset.

Preprocessor Recommendation:

- The usage of ValueRange, ValueLimit,ValueWithTolerances and LimitsAndFits for FileSize is not recommended.
- 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 Domain Model XML syntax)

```
<Unit uid="u--000000003">
  <Kind>
    <ClassString>unspecified</ClassString>
  </Kind>
  <Name>
```

```
    <ClassString>byte</ClassString>
  </Name>
  <Prefix>
    <ClassString>kilo</ClassString>
  </Prefix>
      <Quantity>
            <ClassString>data size</ClassString>
      </Quantity>
</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 </PropertyDefinitionString>
      </Definition>
      <Unit uidRef=" u--000000003"/>
      <ValueComponent>2.3</ValueComponent>
    </FileSize>
  </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 attachement of the DocumentDefinition or File to the value represented via the "NumericalValue" (see 4.6.9), "StringValue" templates (see 4.6.10), ValueRange (see 4.6.11), ValueWithTolerances (see 4.6.12), ValueList (see 4.6.13) or ValueSet (see 4.6.15).

The Instance Model: AP242 Domain Model XML entities and attributes

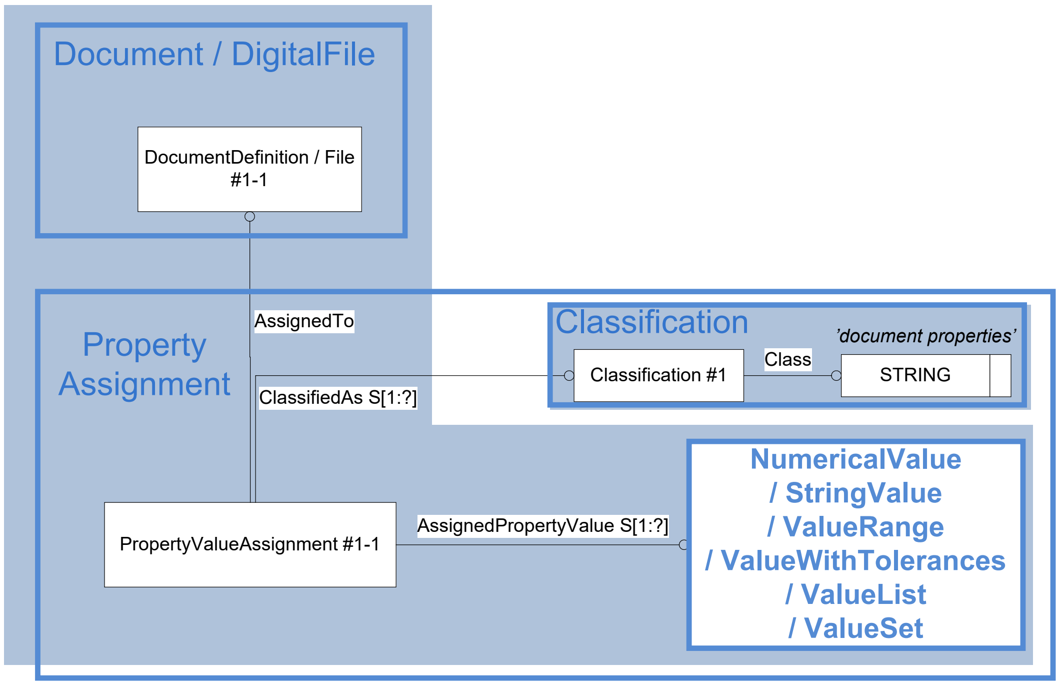

Document / DigitalFile
DocumentDefinition / File #1-1
AssignedTo
Property Assignment
Classification
'document properties'
Classification #1
Class
STRING
ClassifiedAs S[1:?]
PropertyValueAssignment #1-1
AssignedPropertyValue S[1:?]
NumericalValue / StringValue / ValueRange / ValueWithTolerances / ValueList / ValueSet

Figure 77: 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 Domain 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>
  <Role>kind of properties</Role>
</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 Classification
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 73: "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. **This value shall be used for the file that contains the geometry of the part.**
'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

1. 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.

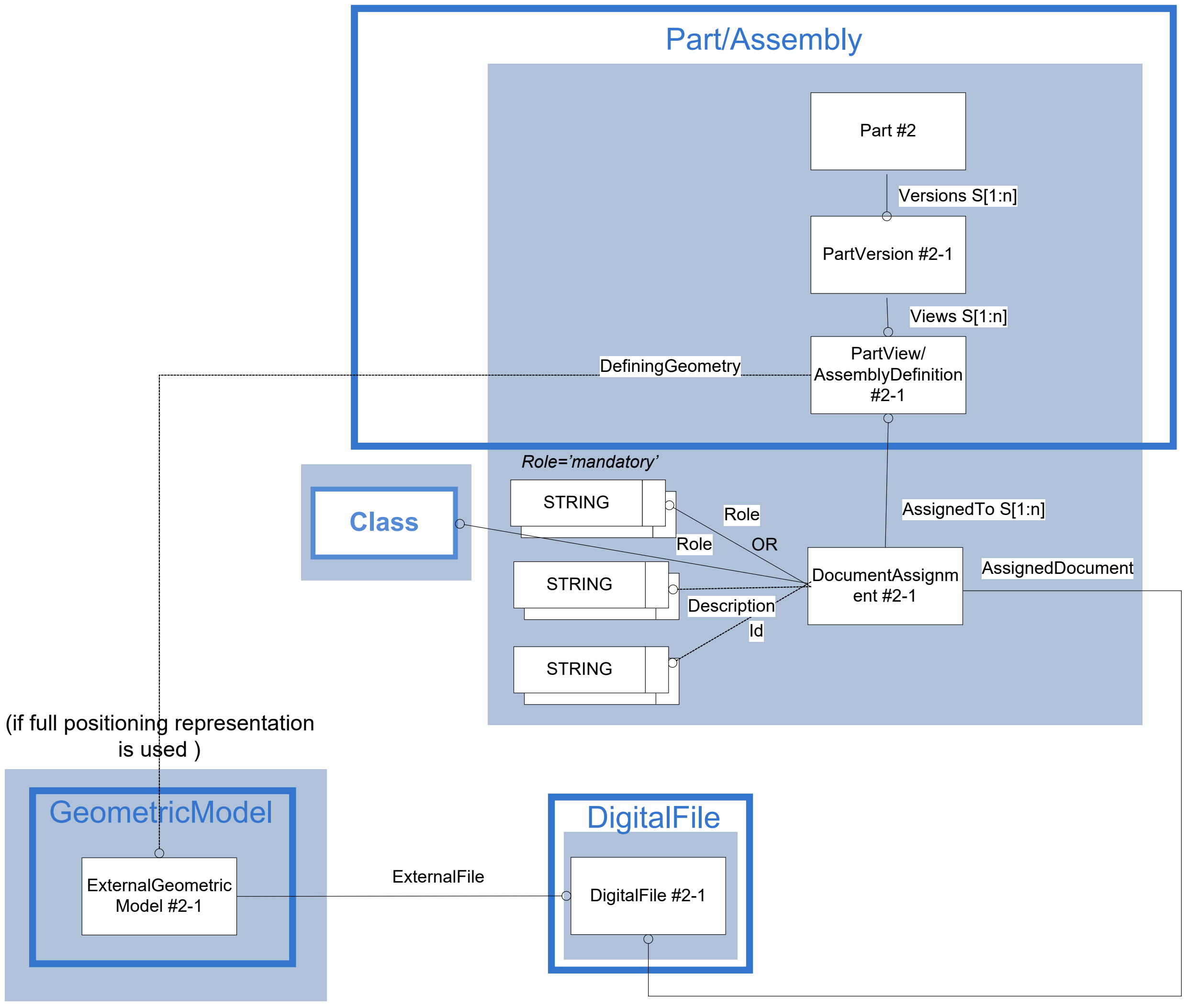


Figure 78: Template "CAx Representation for DocumentAssignment"

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

```
<Part uid="p--000000001E720B30">
  <Id>
    <Identifier uid="pid--000000001E720B30--id7" id="nut" idRoleRef="rl--ii"
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>

<RepresentationContext uid="ccs—origin-nut" xsi:type="n0:GeometricCoordi-
nateSpace">
  <Id id="/NULL"/>
  <Representations>
    <Representation uid="egm--000000001E720B30"
xsi:type="n0:ExternalGeometricModel">
      <Id id="nut.stp"/>
…
      <ExternalFile uidRef="df--000000001E720B30"/>
    </Representation>
…
  </Representations>
…
</RepresentationContext>

<File xsi:type="n0:DigitalFile" uid="df--000000001E720B30">
…
  <Id>
    <Identifier uid="dfid--000000001E720B30--19" id="nut.stp" idRoleRef="rl--
ii" idContextRef="o--000000178"/>
  </Id>
...
</File>
```

Preprocessor Recommendations:

A DigitalFile shall always be referenced by at least a PartView via DocumentAssignment.

Special cases: mapped compliantly to the recommendations of the PDM Usage Guide

1. Model splitting (see Figure 79 on the next page):
   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’.

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 (see Figure 80):

   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.

   For flexible parts having several geometries, depending on their occurrences, AuxiliaryGeometry shall not be used, but rather Occurrence.DefiningGeometry

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 [AP242-PMI]).

Postprocessor Recommendation:

In case a DigitalFile is not referenced by any PartView via DocumentAssignment, an error shall be returned and the DigitalFile shall be ignored.

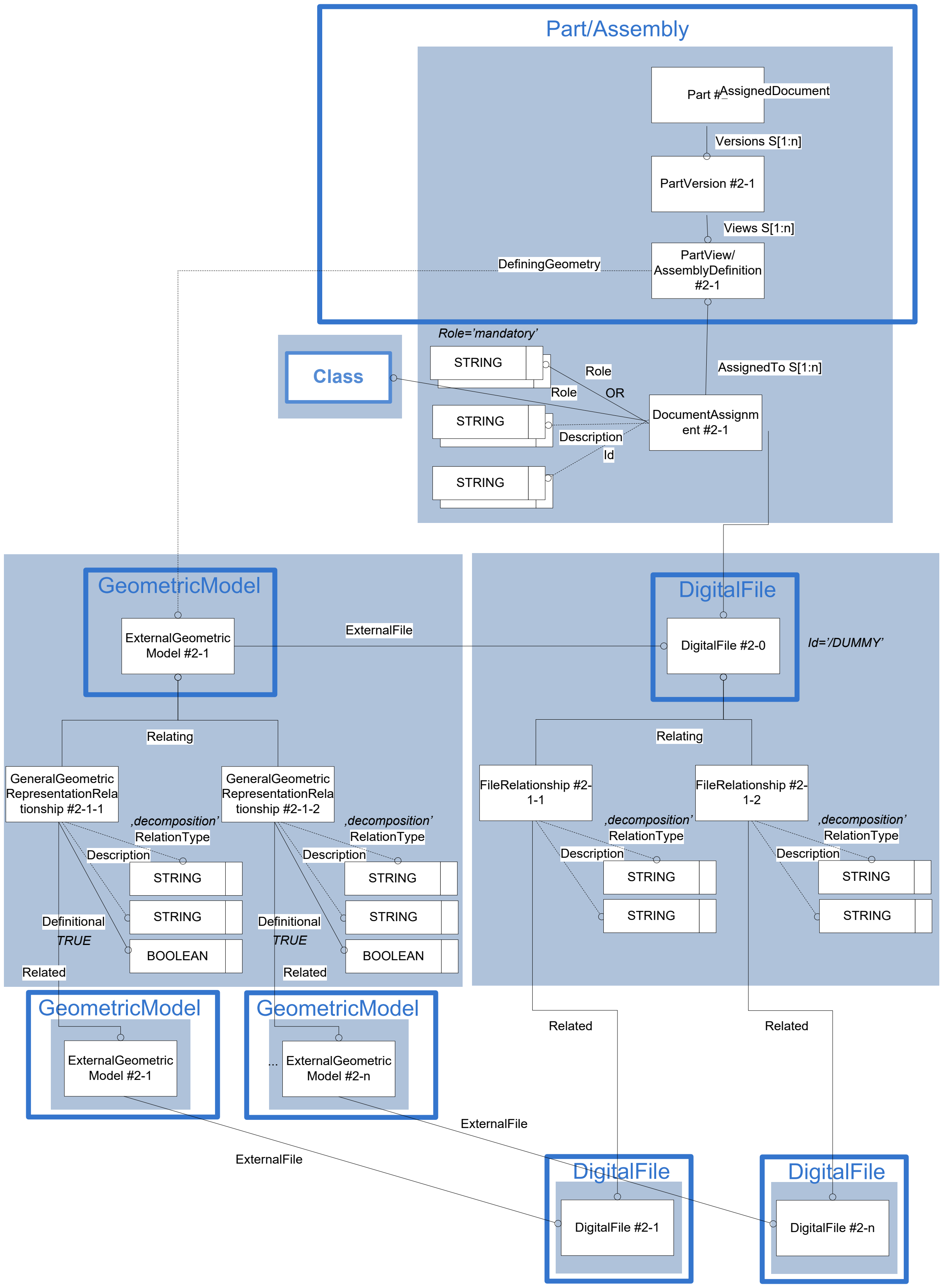


Figure 79: Model splitting for "CAx Representation for DocumentAssignment"

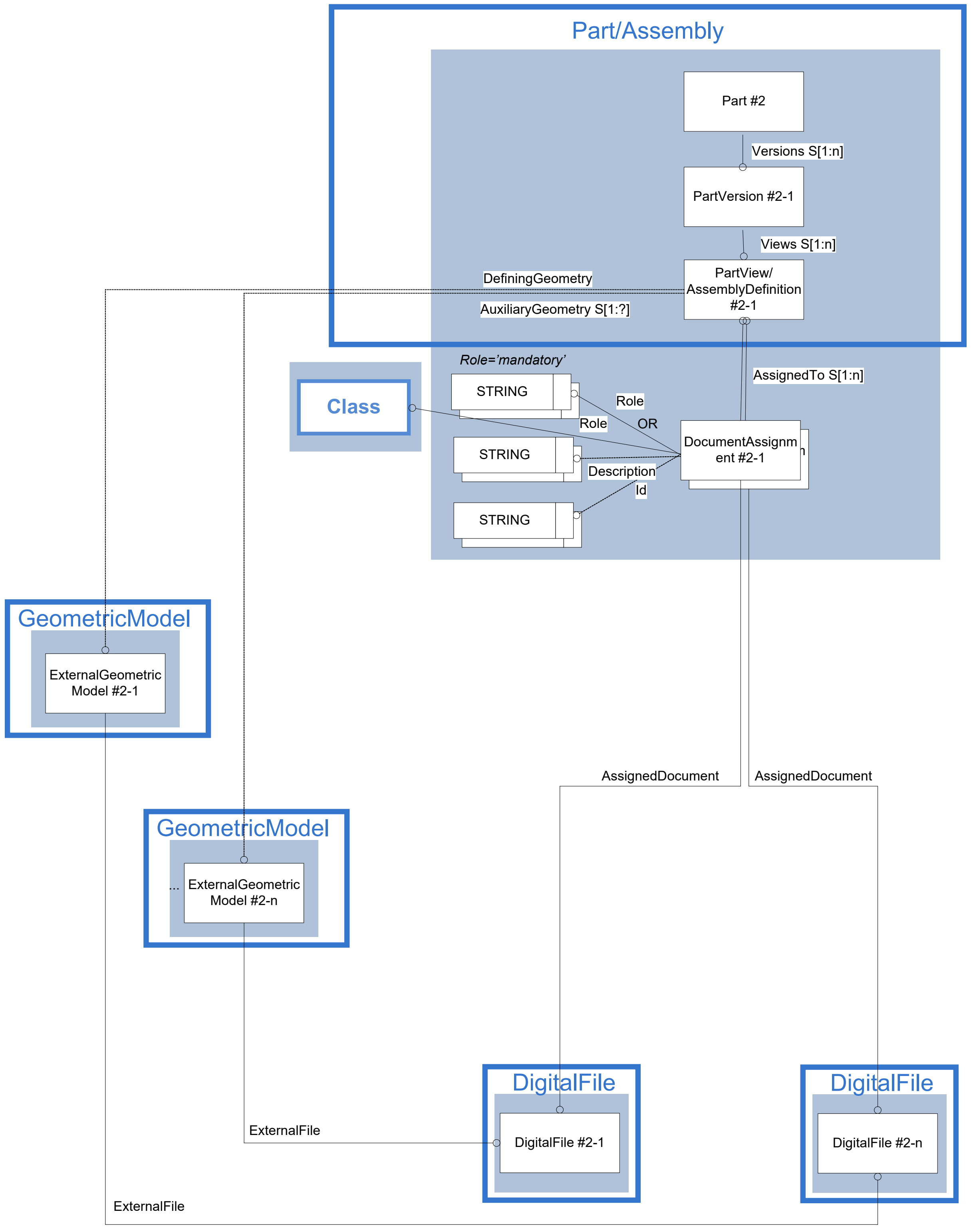


Figure 80: Alternate models for "CAx Representation for DocumentAssignment"

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.

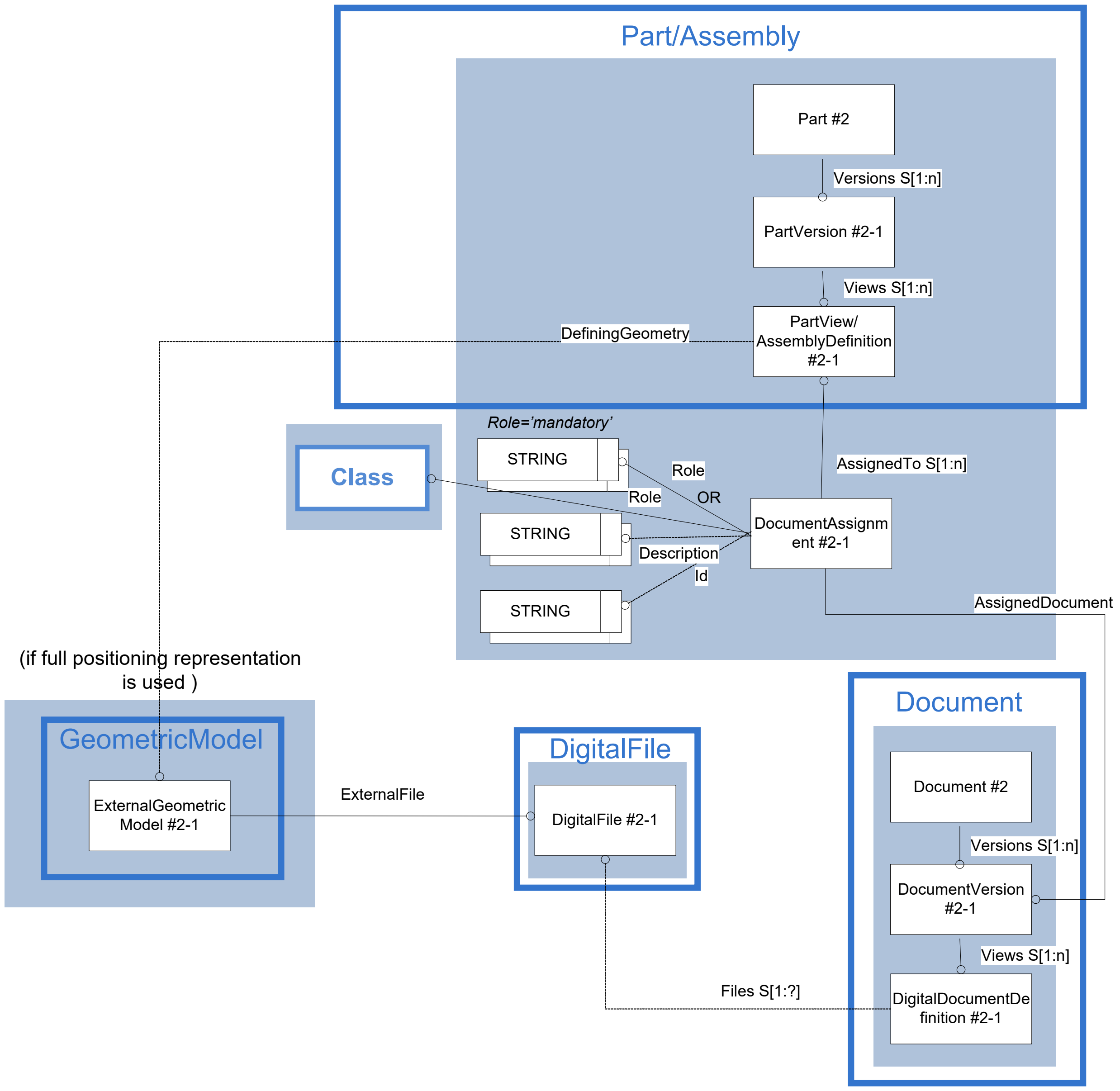


Figure 81: Template "PDM Representation for DocumentAssignment"

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

```
<Part uid="p--000000001EAAE870">
  <Id>
```

```
    <Identifier uid="pid--000000001EAAE870--id7" id="nut" idRoleRef="rl--ii"
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>

<RepresentationContext uid="ccs--origin-nut" xsi:type="n0:GeometricCoordi-
nateSpace">
  <Id id="/NULL"/>
  <Representations>
     <Representation uid="egm--000000001AA415B0"
xsi:type="n0:ExternalGeometricModel">
      <Id id="nut.stp"/>
…
      <ExternalFile uidRef="df--000000001EAAE870"/>
    </Representation>
…
  </Representations>
…
</RepresentationContext>

<Document uid="doc--000000001EAAE870">
…
  <Id>
    <Identifier uid="docid--000000001EAAE870--id7" id="nut" idRoleRef="rl--
ii" 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>
<File xsi:type="n0:DigitalFile" uid="df--000000001EAAE870">
…
  <Id>
    <Identifier uid="dfid--000000001EAAE870--19" id="nut.stp" idRoleRef="rl--
ii" 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.
- A DigitalFile shall always be referenced by at least a DocumentDefinition via the attribute “Files”.

Postprocessor Recommendation:

- In case a DigitalFile is not referenced by any DocumentDefinition, an error shall be returned and the DigitalFile shall be ignored.

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’.

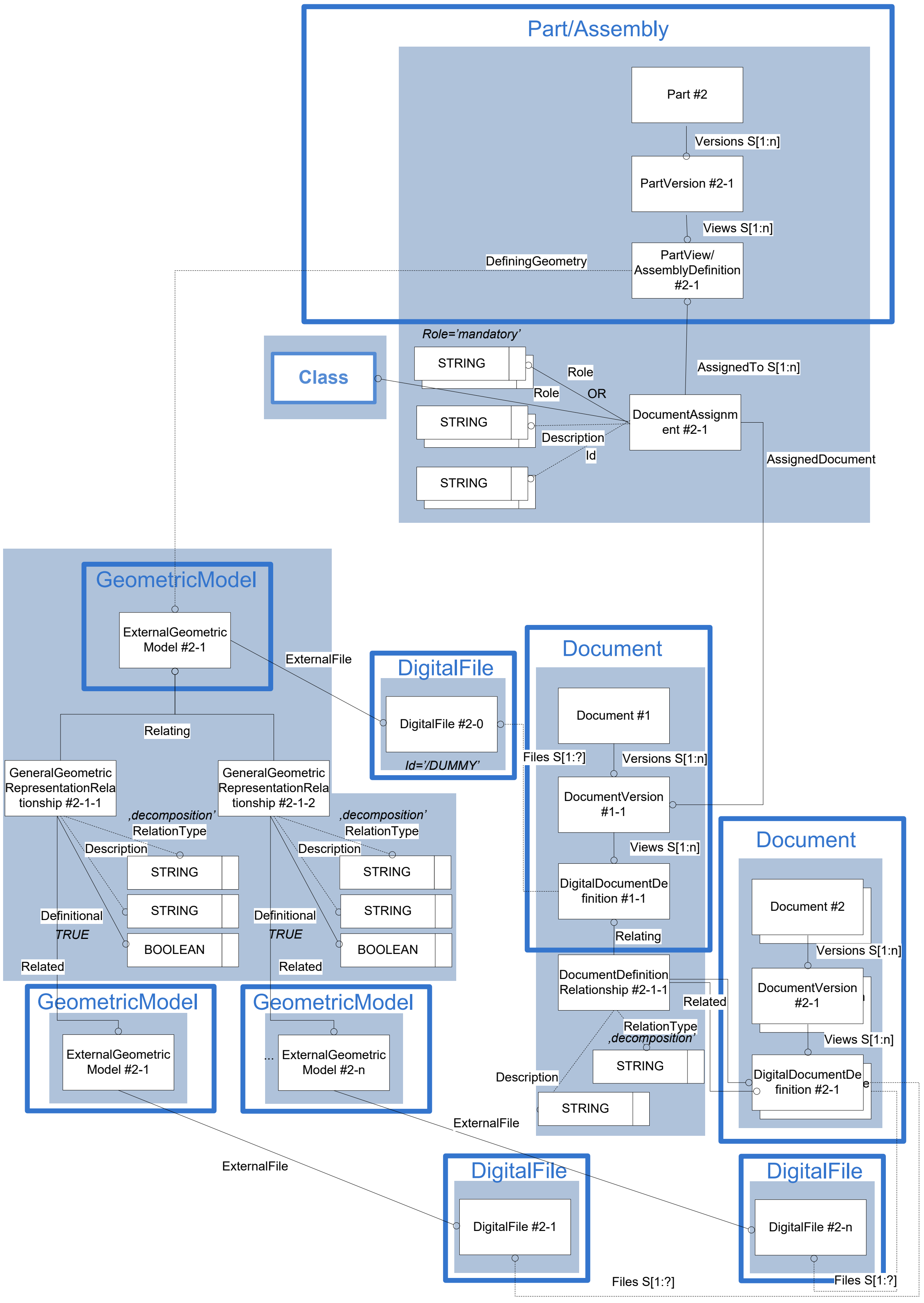


Figure 82: Model splitting for "PDM Representation for DocumentAssignment"

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[AP242-PMI]).

2. Document sharing (shared geometry):
   In case a document is shared by several parts, the Document and the GeometricModel "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.

   The document containing the master geometry and the document containing the alternative geometry shall be related via a DocumentDefinitionRelationship with RelationType='derivation'.

   The DocumentType of the document containing the master geometry shall be 'primary geometry' while for the alternative geometries, the DocumentType shall be 'secondary geometry'.

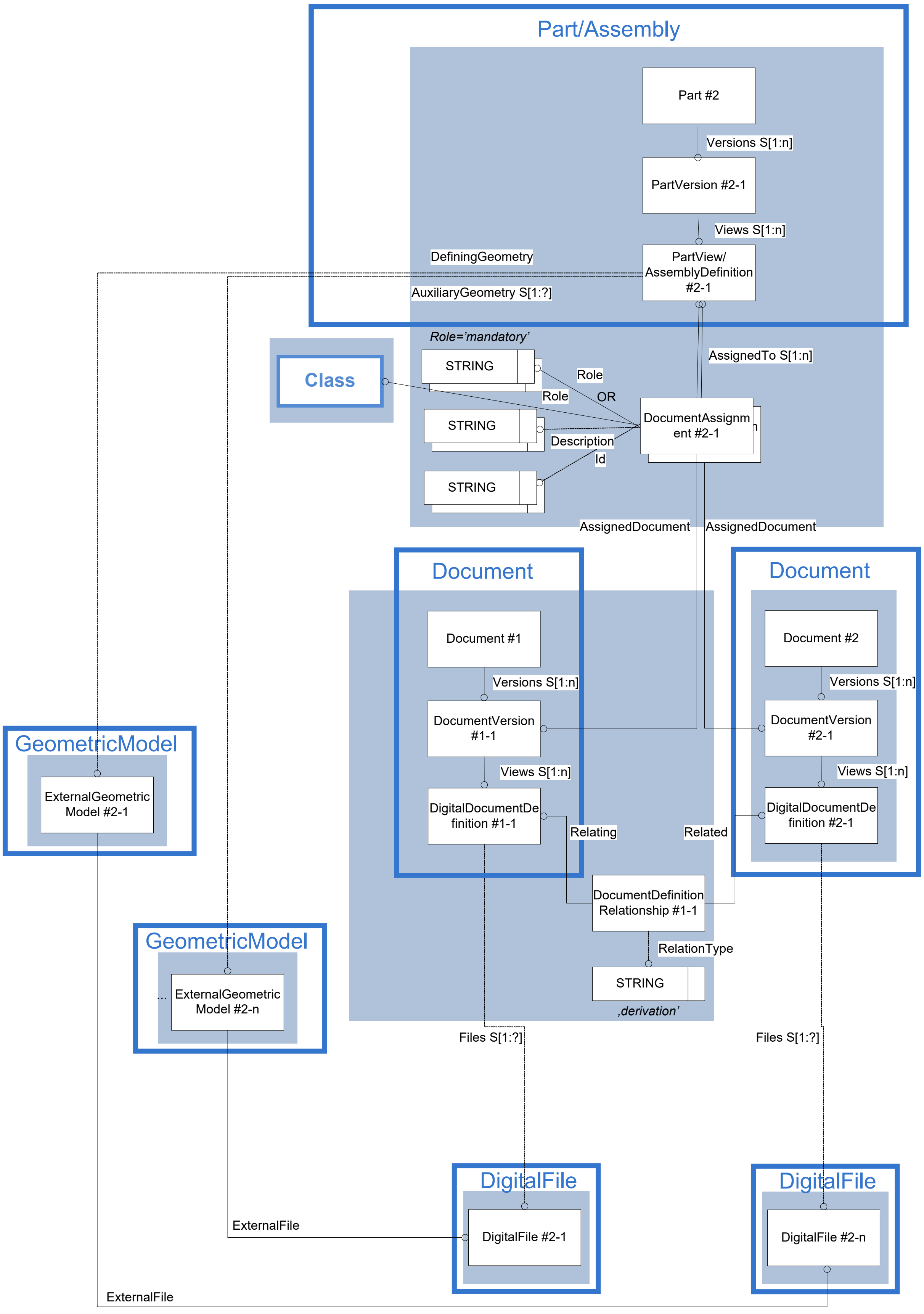


Figure 83: Alternate models for "PDM Representation for DocumentAssignment"

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 "PropertyValueAssignment" template (see 6.2)
  Part(master) is not applicable to UDAs, since CAD systems don't have a business object "PartMaster" like some of the PDM systems.

- 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)
  This is not applicable to UDAs, since CAD systems only consider occurrences.

- a portion of the shape defining the part (`Part`, `AssemblyDefinition or PartView`) (see[AP242-PMI])

- the entire document (DocumentDefinition) according to section 8.1
  This is not applicable to UDAs, since CAD systems don't have a business object "Document" like the PDM systems.

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.

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 value
- simple measure values
  - name and value ("Values without unit" template for integer or real see 12.5.3)
  - name, value and unit ("NumericalValue" template see 4.6.9)
- boolean values ("Values without unit" template for boolean/logical see 12.5.3)

  - name and value
- value ranges (“ValueRange” template see 4.6.11)
  - name, lower limit, upper limit and unit
- values with tolerances (“ValueWithTolerances” template see 4.6.12)
  - name, lower limit, upper limit, nominal value and unit
- aggregation of values (list or set) (“ValueList”/“ValueSet” templates see 4.6.13 and 4.6.15)
  - name and StringValue or NumericalValue for each value element

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), “StringValue” templates (see 4.6.10), ValueRange (see 4.6.11), ValueWithTolerances (see 4.6.12), ValueList (see 4.6.13) or ValueSet (see 4.6.15).
- in the property value template, link the `PropertyValue` to the `PropertyDefinition` with `PropertyValue.Definition` attribute.

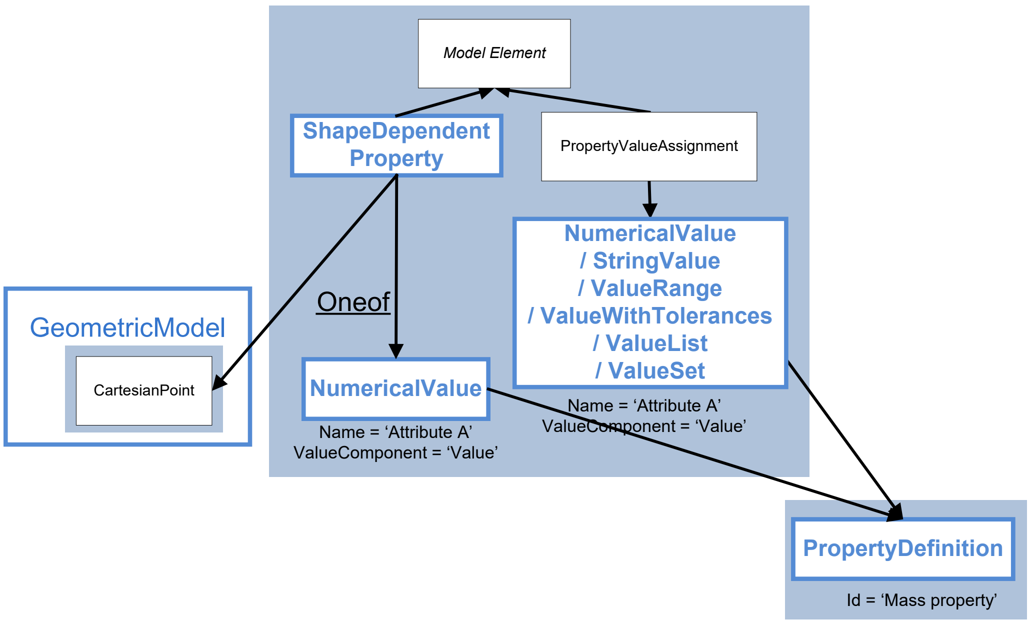


Figure 84: Definition of an attribute name and its usage

The Instance Model: AP242 Domain Model XML entities and attributes

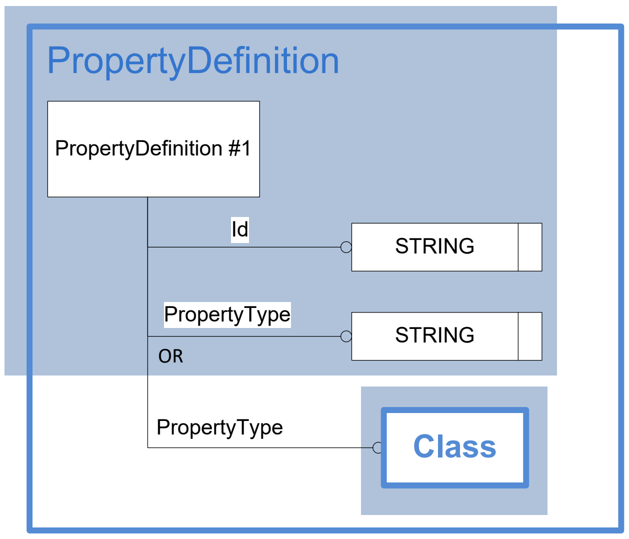


Figure 85: Template "PropertyDefinition"

ENTITY PropertyDefinition	**Attribute Type**
AllowedUnits	OPTIONAL SET[1:?] of UnitSelect
ClassifiedAs	OPTIONAL SET[1:?] of Classification
DefinedIn	OPTIONAL ExternalClassSystem
Definition	OPTIONAL ProxyItemSelect
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
PropertyType	OPTIONAL ClassSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
SameAs	OPTIONAL SET[1:?] of Proxy
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 74: "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, 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 reuse 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 and (if the CAx system can handle them) for CAx data exchange. Example: software characteristics, …
‘user defined attribute’	The property is defined by the 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 + 13.1.2

- For the semantic texts involved in PMIs [AP242-PMI], the following values shall be used:

PropertyType	Explanation
'semantic text'	See [AP242-PMI] section 6.3

- 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.
- ***VersionId***: the identification or set of identifications of a specific version of the PropertyDefinition. The value of this attribute need not be specified.
- ***PropertyDefinitionRelationship***: to relate to another PropertyDefinition. Use the "PropertyDefinitionRelationship" template, see 12.3.
- 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.

Preprocessor Recommendations:

- Even if PropertyType becomes optional with Ed2, for upward compatibility reasons, it is still recommended to set it as if it was mandatory. A valid exception to this rule is in case of ValueList or ValueSet: if the PropertyValues within the ValueList/ValueSet have the same PropertyDefinition, it is sufficient to set it for ValueList/ValueSet and to leave it unset in the embedded PropertyValues.
- If multiple versions of a PropertyDefinition are needed, since PropertyDefinition.Id and PropertyDefinition.VersionId are in the same object, all versionless attributes of PropertyDefinition (Id, Name, , …) shall be redundantly mapped to all versions, including the multiple identifiers.

Postprocessor Recommendations: None specified.

Related Entities: There are no specific related entities.

The Instance Model: STEP exchange file format (ISO10303 AP242 Domain 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>
  <VersionId id="A.1"/>
 </PropertyDefinition>
```

12.3 *Template “PropertyDefinitionRelationship”*

This relationship enables to relate two PropertyDefinitions.

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

```
<PropertyDefinition uid="ID_60">
      <Id id="PDMIFPartBool_TSI"/>
      <PropertyType>
            <ClassString>customized PDM property</ClassString>
      </PropertyType>
      <VersionId id="A.1"/>
      <PropertyDefinitionRelationship uid="pdr--1">
            <Related>
                  <PropertyDefinition uidRef="ID_60-1"/>
            </Related>
            <RelationType>
                  <ClassString>sequence</ClassString>
            </RelationType>
      </PropertyDefinitionRelationship>
</PropertyDefinition>
<PropertyDefinition uid="ID_60-1">
      <Id id="PDMIFPartBool_TSI"/>
      <PropertyType>
            <ClassString>customized PDM property</ClassString>
      </PropertyType>
      <VersionId id="B.1"/>
</PropertyDefinition>
```

Entity PropertyDefinitionRelationship attributes	Attribute type
ClassifiedAs	OPTIONAL SET[1:?] of Classification
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL IdentifierSelect
Related	PropertyDefinitionSelect
RelationType	ClassSelect
RetentionPeriod	OPTIONAL SET[1:?] of RetentionPeriod
ActivityAssignment	OPTIONAL SET[1:?] of ActivityAssignment

Entity PropertyDefinitionRelationship attributes	Attribute type
ActivityMethodAssignment	OPTIONAL SET[1:?] of ActivityMethodAssignment
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
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 75: "PropertyDefinitionRelationship” 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	
‘decomposition’	The PropertyDefinitionRelationship defines a relationship where the Related PropertyDefinition is a member of a group of PropertyDefinition objects that is established by the Relating PropertyDefinition
‘dependency’	The related PropertyDefinition is dependent upon the Relating PropertyDefinition
‘hierarchy’	The business object defines a hierarchical relationship where the Related PropertyDefinitionn is on a lower level than the Relating PropertyDefinition
‘peer’	The related PropertyDefinition shall not be used without the Relating PropertyDefinitionn and vice versa
‘substitution’	The PropertyDefinitionRelationship defines a relationship where the Related PropertyDefinition replaces the Relating PropertyDefinition

'value domain'	The PropertyDefinitionRelationship defines a relationship where the values assigned to the Relating PropertyDefinition shall be within the limits indicated by the values assigned to the Related PropertyDefinition

- Additionally, the following value is recommended:

'sequence'	The business object defines a logical sequence where the related PropertyDefinition comes after the relating PropertyDefinition. In case the Id is equal and VersionId of both PropertyDefinitions is different, a version sequence is defined.

- ***Related***: the other object of **PropertyDefinition** 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.

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.

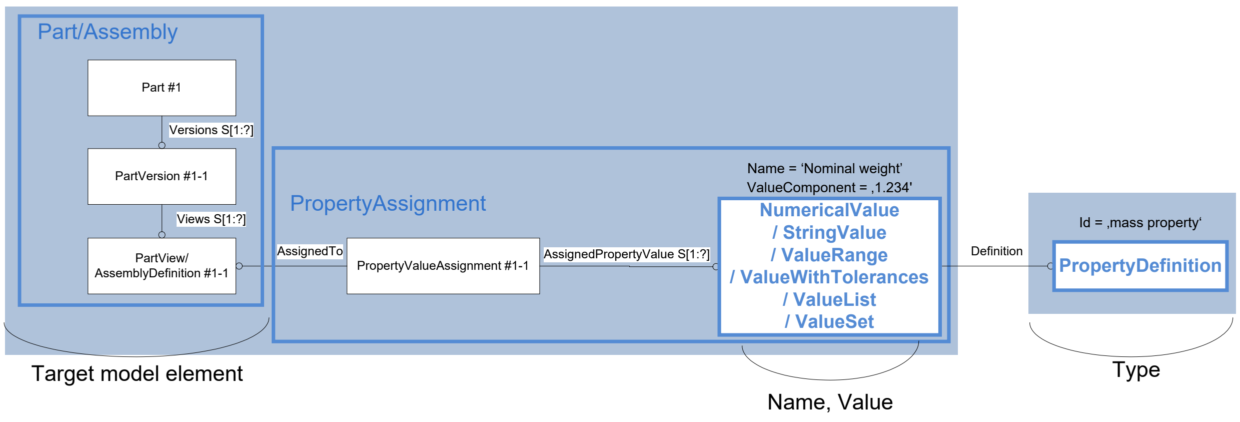


Figure 86: User defined attribute at the part/assembly level

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 `NextAssemblyOccurenceUsage`, it is recommended to use "PropertyAssignment" template defined in chapter 6.2 directly in the `NextAssemblyOccurenceUsage` entity.

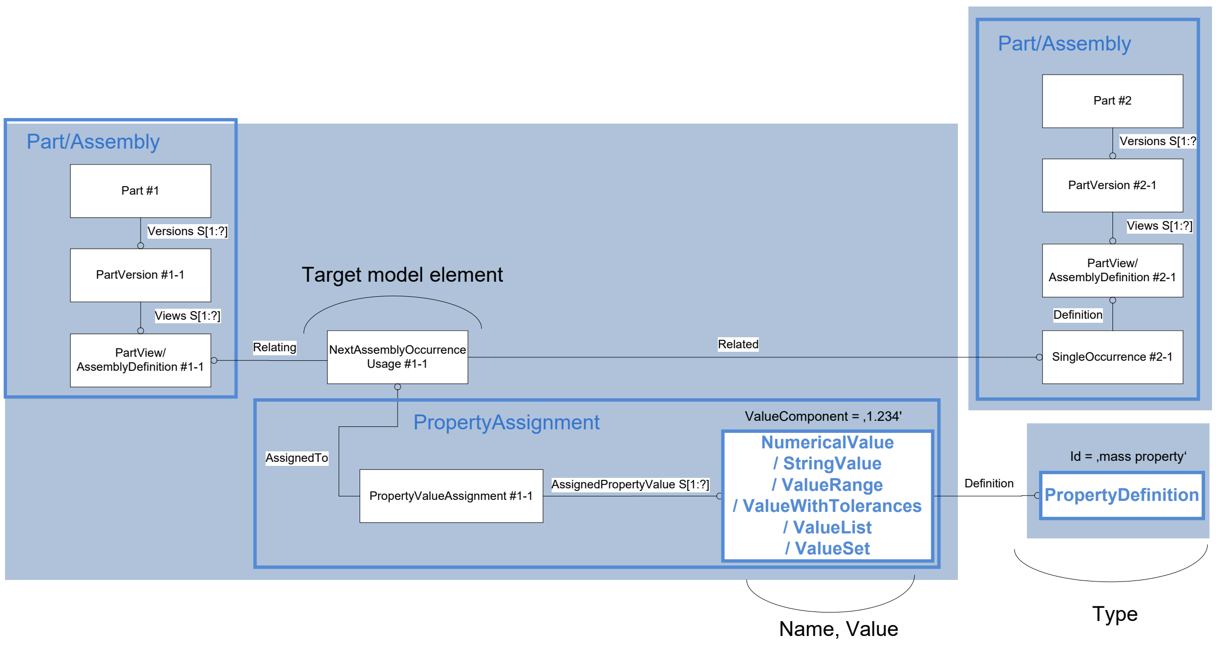


Figure 87: User defined attribute for a simple component instance in an assembly

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.

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 (String)

Values without applicable unit convey all kinds of information, in particular in the context of PDM Properties.

To define such an attribute (string), the “StringValue” template defined in chapter 4.6.10. shall be used.

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.

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 the following entity type for the transfer of string values:

- `descriptive_representation_item`

12.5.3 Values without unit (Integer, Real, Boolean, Logical)

Values without applicable unit convey all kinds of information, in particular in the context of PDM Properties.

To define such an attribute (Integer, Real, Boolean, Logical), the “NumericalValue” template defined in chapter 4.6.10. shall be used.

Three main types can be distinguished, which shall be handled as recommended below.

Preprocessor Recommendations:

- A dedicated `Name` shall be set with the value ‘number without unit’.
- An *integer attribute* stores a whole number (and nothing else) as a double in the `ValueComponent` attribute. Examples for this include counts and sequences.
- A *real attribute* stores a decimal number (and nothing else) as a double in the `ValueComponent` attribute. Examples for this include ratios and percentages.
- A *boolean* or *logical attribute* stores a pre-defined value as a double in the `ValueComponent` attribute. The value shall be either 1 for ‘TRUE’, 0 for ‘FALSE’, or 0.5 for ‘UNKNOWN’.

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 three dedicated entity types for the transfer of values without unit:

- `integer_representation_item`
- `real_representation_item`
- `boolean_representation_item`

Having such explicit counterparts in the AP242 Domain Model would improve the stability of PDM Property exchange. ISO Jira Task TCSC410303-1399 has been created to add these elements in a future revision of the standard. Target for STEP AP242 ed3 to improve the model.

Example of Integer representation with the “NumericalValue” template:

Use “number without unit” as the `NumericalValue Name` and “integer representation” as `Unit Name`.

```
<PropertyValue uid="id1" xsi:type="ap:NumericalValue">
  <Definition>
    <PropertyDefinition uidRef="pd--000000320"/>
  </Definition>
  <Name>
    <CharacterString>number without unit</CharacterString>
  </Name>
  <Unit uidRef="u--000000002"/>
  <ValueComponent>45</ValueComponent>
</PropertyValue>

<Unit uid="u--000000002">
  <Kind>
    <ClassString>unspecified</ClassString>
  </Kind>
  <Name>
    <ClassString>integer representation</ClassString>
  </Name>
</Unit>
```

Example of Real representation with the “NumericalValue” template:

Use “number without unit” as the `NumericalValue Name` and “real representation” as `Unit Name`.

```
<PropertyValue uid="id1" xsi:type="ap:NumericalValue">
  <Definition>
    <PropertyDefinition uidRef="pd--000000320"/>
  </Definition>
  <Name>
    <CharacterString>number without unit</CharacterString>
  </Name>
  <Unit uidRef="u--000000002"/>
  <ValueComponent>0.56</ValueComponent>
</PropertyValue>

<Unit uid="u--000000002">
  <Kind>
    <ClassString>unspecified</ClassString>
  </Kind>
  <Name>
    <ClassString>real representation</ClassString>
  </Name>
</Unit>
```

Example of Boolean / Logical representation with the "NumericalValue" template:

For the Boolean representation, the ValueComponent element have to be set following the list hereafter because the XSD used an XSD:double where string value are not authorized.

Based on the Logical, Boolean definition from the 10303 Part11: FALSE < UNKNOWN < TRUE

Use "number without unit" as the `NumericalValue Name` and "logical representation" as `Unit Name`.

- FALSE : ValueComponent = 0 (for Logical and Boolean)

```
<PropertyValue uid="id1" xsi:type="ap:NumericalValue">
  <Definition>
    <PropertyDefinition uidRef="pd--000000320"/>
  </Definition>
  <Name>
    <CharacterString>number without unit</CharacterString>
  </Name>
  <Unit uidRef="u--000000002"/>
  <ValueComponent>0</ValueComponent>
</PropertyValue>

<Unit uid="u--000000002">
  <Kind>
    <ClassString>unspecified</ClassString>
  </Kind>
  <Name>
    <ClassString>logical representation</ClassString>
  </Name>
</Unit>
```

- UNKNOWN : ValueComponent = 0.5 (only for Logical and NOT for Boolean)

```
<PropertyValue uid="id1" xsi:type="ap:NumericalValue">
  <Definition>
    <PropertyDefinition uidRef="pd--000000320"/>
  </Definition>
  <Name>
    <CharacterString>number without unit</CharacterString>
  </Name>
  <Unit uidRef="u--000000002"/>
  <ValueComponent>0.5</ValueComponent>
</PropertyValue>

<Unit uid="u--000000002">
  <Kind>
    <ClassString>unspecified</ClassString>
  </Kind>
  <Name>
    <ClassString>logical representation</ClassString>
  </Name>
</Unit>
```

- TRUE : ValueComponent = 1 (for Logical and Boolean)

```
<PropertyValue uid="id1" xsi:type="ap:NumericalValue">
  <Definition>
    <PropertyDefinition uidRef="pd--000000320"/>
  </Definition>
  <Name>
```

```
      <CharacterString>number without unit</CharacterString>
    </Name>
    <Unit uidRef="u--000000002"/>
    <ValueComponent>1</ValueComponent>
  </PropertyValue>

  <Unit uid="u--000000002">
    <Kind>
      <ClassString>unspecified</ClassString>
    </Kind>
    <Name>
      <ClassString>logical representation</ClassString>
    </Name>
  </Unit>
```

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 sug-gested to enable validation of exchanged information. It is optional to apply validation properties in ex-change 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 proper-ties that reflect the main semantics of the product data. For the purpose of this docu-ment 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 sec-tion 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 sur-face 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 Prop-erties (see reference in Annex C) and section 13, below for details.

These validation properties enable the verification of geometry and assembly information of re-ceived 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 com-pared, 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 rec-orded.
- *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 guar-anteed by this methodology, but the chance of an incorrect position and orientation com-bining 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 Domain 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 88, below.

Note: Number of Children properties shall be computed without taking configuration man-agement information into account, since not all target applications support configuration man-agement information. Configuration management information may include concepts like ef-fectivity and versioning.

The property value shall be instantiated according to the following description, which is depicted in the instance diagram in Figure 88 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 PropertyDefinition.PropertyType shall be 'assembly validation property' and the PropertyDefinition.Id shall be 'quality property'. The Unit of the NumericalValue shall be 'each'. The format of the NumericalValue should be Integer (i.e. not a real with comma or scientific notation).

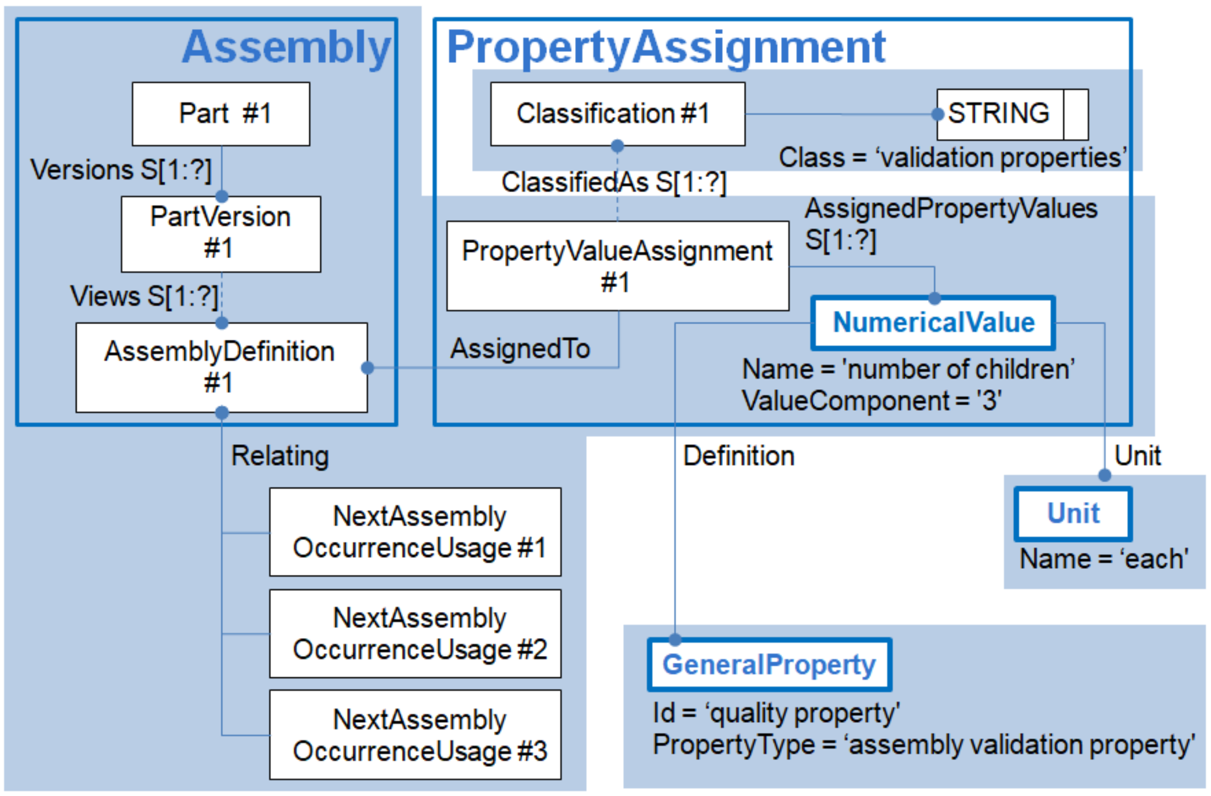


Figure 88: Instantiation of AVP 'number of children' for 3 children

13.1.2 Notional Solids Centroid Position

This property is similar to the geometric validation property "centroid" (see section**Fehler! Verweisquelle konnte nicht gefunden werden.**): 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 as-sumed 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 as-sumed 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 property value shall be instantiated according to the following description, which is de-picted in the instance diagram in Figure 89, below.

All instances of type AssemblyDefinition that are referenced as relating AssemblyDefinition in-stances by one or more NextAssemblyOccurrenceUsage instances will have a single Cen-tre-OfMass assigned to it. The CentreOfMass represents the notional solid centroid position. The child nodes that this centroid position is valid for are those Occurrences that are refer-enced by the NextAssemblyOccurrenceUsage.related attributes.

The role of the CentreOfMass shall be ‘assembly validation property’. Its id shall be ‘notional solids centroid’. An AssemblyDefinition instance shall be assigned at maximum one instance of CentreOfMass with this id.

The CentreOfMass.centrePoint is a CartesianPoint with exactly three coordinate values of type REAL. The CartesianPoint defines the calculated centroid for the notional solids as-sumed for each child node. See section 6.1 for the definition of CartesianPoint.

To denote the coordinate space of the CartesianPoint a GeometricCoordinateSpace is in-stanti-ated 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 or at least in ExchangeContext.DefaultUnit.

In case of relative positioning, it is up to the pre-processor to provide a distinct instance of Ge-ometricCoordinateSpace for each CentreOfMass.

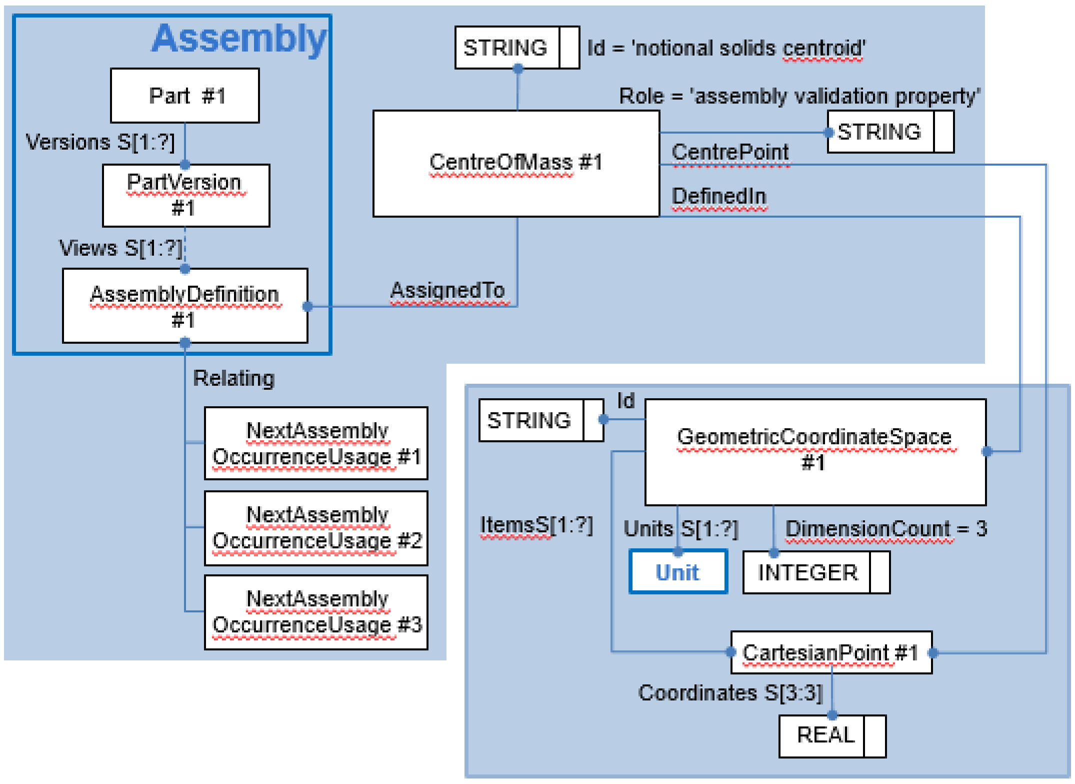


Figure 89: Instantiation of AVP ‘notional solids centroid’

ENTITY CentreOfMass	**Attribute Type**
ClassifiedAs	OPTIONAL SET[1:?] of Classification
DefinedIn	GeometricCoordinateSpace
Description	OPTIONAL DescriptorSelect
Id	OPTIONAL Id
Role	OPTIONAL ClassSelect
ValueDetermination	OPTIONAL ClassSelect
CentrePoint	CartesianPoint

Table 76: "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 Centre-OfMass.
- ***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 CentreOf-Mass in a GeometricCoordinateSpace. The CentreOfMass.centrePoint is a Cartesian-Point 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 part-ners.

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 GeometricCoordi-nateSpace.

Postprocessor Recommendations:

- According to the section 4.13.2 of the Recommended Practices for Geometric and As-sembly Validation Properties, the value of the validation property (CentrePoint) and the centroid value calculated using the AP242 XML file shall not deviate by more than 1 millimeter three-dimensional distance to be considered as 'green'.

Related Entities:

- See Table 35 in section 6.1 for the definition of CartesianPoint.
- See Table 33 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.

It is recommended that the Geometrical Validation Properties are mapped within the CAD mod-els, so that the single parts can be validated using the GVPs.

Additionally, the assembly structure can be validated using the Assembly Validation Properties. If both is correct, everything is OK: There is no need to store the GVPs in the Domain Model.

13.3 LOTAR Product Structure Validation Properties (PSVP)

Note: Some concepts in this section, namely the definition and transfer of the hash attribute definitions, are included for completeness but have not yet been fully tested. Hence, these defi-nitions might change in the future.

In order to check the integrity of the exchanged/archived product structure data, LOTAR defines a way to build hash values out of multiple attribute values (see LOTAR document referenced in Annex C). The LOTAR specification is not specific to a data format, so here the implementation recommendation is defined for usage with the AP242 Domain Model.

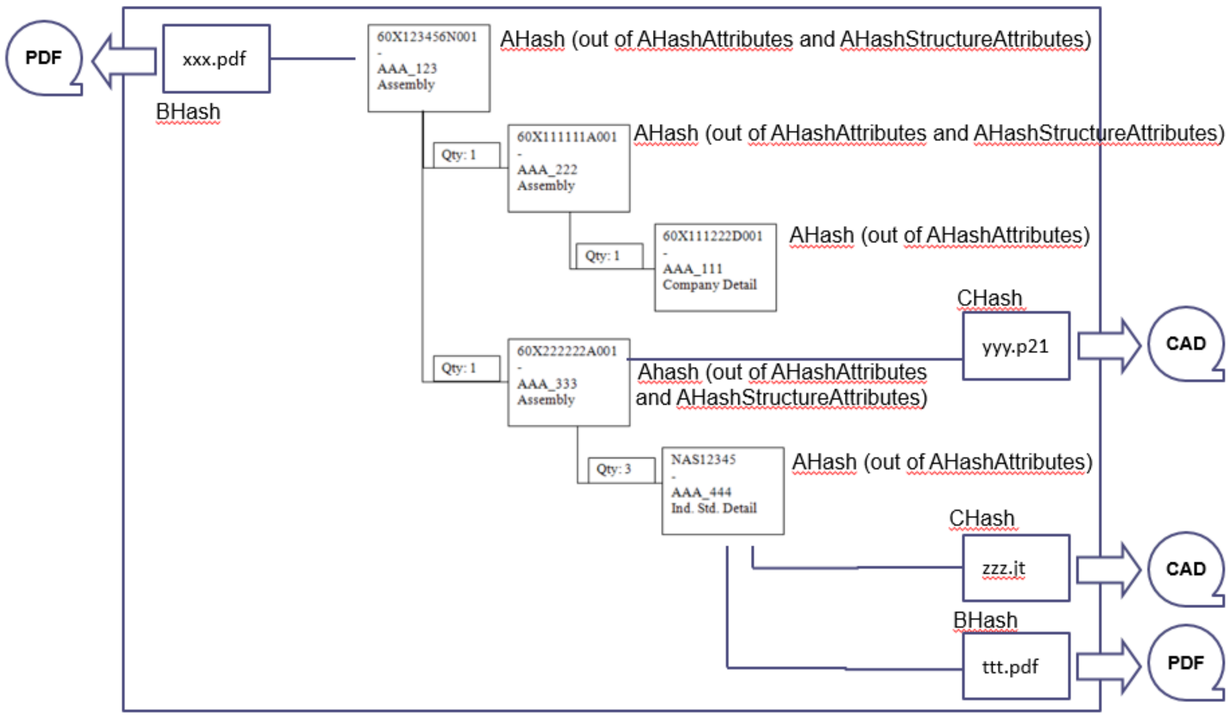

PDF
xxx.pdf
BHash
60X123456N001
-
AAA_123
Assembly
AHash (out of AHashAttributes and AHashStructureAttributes)
Qty: 1
60X111111A001
-
AAA_222
Assembly
AHash (out of AHashAttributes and AHashStructureAttributes)
Qty: 1
60X111222D001
-
AAA_111
Company Detail
AHash (out of AHashAttributes)
CHash
yyy.p21
CAD
Qty: 1
60X222222A001
-
AAA_333
Assembly
Ahash (out of AHashAttributes and AHashStructureAttributes)
Qty: 3
NAS12345
-
AAA_444
Ind. Std. Detail
AHash (out of AHashAttributes)
CHash
zzz.jt
CAD
BHash
ttt.pdf
PDF

Figure 90: Example of LOTAR Product Structure Validation Properties at the single File, Part and Assembly level

The following Validation Properties are defined:

- AHash: is the hash value of the associated node based on the attributes defined by the AHashAttributes + (in case structure relationships underneath the node shall also be taken into consideration) based on the attributes defined by the AHashStructureAttributes
- BHash: is the hash value of a non-geometrical file, like PDF
- CHash: is the hash value of a geometrical file, like AP242 P21, JT, …
- The mechanisms defined in this section allow any further Hash (DHash, EHash, …) defined on further objects, project specific and agreed between the exchange partners.

A hash value must be calculated using a standard cryptographic algorithm to generate a unique signature The Hash algorithm used to compute the hash values is also mapped, so that the consumers of the data can regenerate the hash after loading to validate that the data integrity. Recommended are:

- SHA1 for AHash
- SHA512 for BHash and CHash, due to its greater length but any industry standard algorithm that ensures file uniqueness are possible if agreed between the exchange partners.

Remark: even if stored as a string property, a hash is a hexadecimal value, therefore it is *not* case sensitive.

Two main use cases are identified:

1. Import the STEP data in the target PDM system, compute the validation properties out of the imported data and compare them with the values given in the STEP data
2. Validation of the STEP data itself, before importing/archiving the STEP data

These two use cases have slightly different requirements. These differences are mentioned along the following sections.

13.3.1 Definition of AHashAttributes

AHashAttributes is a comma separated list of the object attributes, that have their values included in the AHash calculation.
This is a subset of the attributes in the AP242 Domain Model XML file.
However, it is possible to define another separator in the PropertyValues AHashAttributesSeparatorBegin (instead of ‘,’) and AHashSeparatorEnd (empty in case of ‘,’, used only with different begin and end separators like ‘<’ and ‘>’).
For use case 1, it shall be ensured that all attributes of this subset shall be imported into the target system, for example

- overtake the partner revision number
- no truncation of large string attribute values
- enough precision in order to overtake all the significant digits of integers/doubles
- etc…

The attributes shall be uniquely identified using their case-sensitive spelling defined in the XSD and a path definition, how to come from the current object to the attribute.

The EXPRESS path operators provide a powerful mechanism to traverse forward/backward any path, but would be a rupture with the XML data and XSD definition and would require specific development due to the lack of wide tool support.
Rather than defining a proprietary syntax, the possibilities of Xpath have been analyzed and seem to be sufficient.
For better understanding, the following examples are given both in XPath syntax and (for information) in EXPRESS syntax. AHashAttributes shall contain only the XPath syntax.

Example of AHashAttributes: the line feeds (replacing the ‘,’) are here only for better readability but shall not be part of the real data:

```
./parent::Views/parent::PartVersion/parent::Versions/parent::Part/Id/Identifier

./parent::Views/parent::PartVersion/parent::Versions/parent::Part/Name


//PersonInOrganization
   [@uid=./parent::Views/parent::PartVersion/OrganizationOrPersonInOrganiza-
tionAssignment
      [Role/ClassString='creator']
    /AssignedPersonOrOrganization/@uidRef
   ]/Id/Identifier/@id


./parent::Views/parent::PartVersion/DateTimeAssignment
   [Role/ClassString='creation']
   /AssignedDate


./Id/Identifier


//Approval
   [@uid=./parent::Views/parent::PartVersion/ApprovalAssignment
      [not(boolean(./Role))]
      /AssignedApproval/@uidRef
   ]/Status
```

```
./PropertyValueAssignment/AssignedPropertyValues/PropertyValue
   [Name/CharacterString='PDMIFPartTol_xxx']:7E


./PropertyValueAssignment/AssignedPropertyValues/PropertyValue
   [Name/CharacterString='PDMIFPartVers_ValueList']


//Class
   [@uid=//Classification
      [@uid=./parent::Views/parent::PartVersion/parent::Versions/parent::Part/Classi-
fiedAs/Classification/@uidRef
      ]
      /Class/Class/@uidRef
   ]


./parent::Views/parent::PartVersion/parent::Versions/parent::Part/PartTypes[PartCate-
goryEnum='assembly']/PartCategoryEnum/text()
```

Here the value without line feeds (with ',' as separator):

```
./parent::Views/parent::PartVersion/parent::Versions/parent::Part/Id/Identi-
fier,./parent::Views/parent::PartVersion/parent::Versions/par-
ent::Part/Name,//PersonInOrganization[@uid=./parent::Views/parent::PartVer-
sion/OrganizationOrPersonInOrganizationAssignment[Role/ClassString='crea-
tor']/AssignedPersonOrOrganization/@uidRef]/Id/Identifier/@id,./par-
ent::Views/parent::PartVersion/DateTimeAssignment[Role/ClassString='crea-
tion']/AssignedDate,./Id/Identifier,//Approval[@uid=./parent::Views/par-
ent::PartVersion/ApprovalAssignment[not(boolean(./Role))]/AssignedAp-
proval/@uidRef]/Status/ClassString,./PropertyValueAssignment/AssignedProper-
tyValues/PropertyValue[Name/CharacterString='PDMIFPartTol_xxx']:7E,./Proper-
tyValueAssignment/AssignedPropertyValues/PropertyValue[Name/Character-
String='PDMIFPartVers_ValueList'],//Class[@uid=//Classification[@uid=./par-
ent::Views/parent::PartVersion/parent::Versions/parent::Part/Classi-
fiedAs/Classification/@uidRef]/Class/Class/@uidRef],./parent::Views/par-
ent::PartVersion/parent::Versions/parent::Part/PartTypes[PartCatego-
ryEnum='assembly']/PartCategoryEnum/text()
```

Equivalence using EXPRESS path operators (just for information):

```
SELF::ViewOf::VersionOf::Id,

SELF::ViewOf::VersionOf::Name,

SELF::Viewof<-AssignedTo{OrganizationOrPersonInOrganizationAssignment | Role = 'crea-
tor'}[1]::AssignedPersonOrOrganization::Id,

SELF::Viewof<-AssignedTo{DateTimeAssignment | Role = 'creation'}[1]::AssignedDate,

SELF::Id,

SELF::Viewof<-AssignedTo{ApprovalAssignment | Description = 'disposition' AND Role =
?}[1]::Status,

SELF<-AssignedTo{PropertyValueAssignment}::AssignedPropertyValues{PropertyValue |
Name = 'PDMIFPartTol_xxx'},


SELF<-AssignedTo{PropertyValueAssignment}::AssignedPropertyValues{PropertyValue |
Name = 'PDMIFPartVers_ValueList'},
SELF::ViewOf::VersionOf::ClassifiedAs::Class

SELF::ViewOf::VersionOf{PartTypes = PartCategoryEnum.assembly}::PartTypes[1]
```

The order of the attributes in the list defines the order of their string value concatenation.

Each attribute value is assumed to be an XML string (use of UTF-8 as stated in the first line of the XML file).

In order to handle empty values or unset optional attribute, each attribute value shall be separated. An appropriate separator shall not occur in the attribute values.
The use of XML tag delimiter '<' and '>' is recommended (while these characters are mapped as &lt; and &gt; if they appear within the attribute values) since it will not collide with an attribute value.
However, it is possible to define another separator in the PropertyValues AHashSeparatorBegin and AHashSeparatorEnd.

Example of string concatenation of the AHash attribute values before computing the AHash: the line feeds and blanks between two separators are here only for better readability but shall not be part of the real data

```
<
 <
  <as1-A>
  <identification information>
  <company-a.com>
 >
 <
  <as1-B>
  <exchange identification information>
  <company-b.com>
 >
<
  <as1-C>
  <identification information>
  <company-c.com>
 >
>

<
<de-DE><as 1 deutsch>
 <en-US><as 1 english>
 <fr-FR><as 1 francais>
>

<005-TP/EVD-Mustermann>

<2021-10-16T09:08:07>

<&>

<in progress>

<

 <3e-3>
 <8.901>
 <4e-3>
>

<
```

```
 <PDMIFPartVers_ValueList#1>
 <PDMIFPartVers_ValueList#2>
 <PDMIFPartVers_ValueList#3>
>

<
 <customized part type>
 <PDMIFPartType_xxx>
>

<assembly>
```

Here the value without line feeds:

```
<<<as1-A><identification information><company-a.com>><<as1-B><exchange iden-
tification information><company-b.com>><<as1-C><identification infor-
mation><company-c.com>>><<de-DE><as 1 deutsch><en-US><as 1 english><fr-FR><as
1 francais>><005-TP/EVD-Mustermann><2021-10-16T09:08:07><&><in progress><<3e-
3><8.901><4e-3><<PDMIFPartVers_ValueList#1><PDMIFPartVers_Val-
ueList#2><PDMIFPartVers_ValueList#3>><<customized part type><PDMIFPart-
Type_xxx>><assembly>
```

The AHash value for this string (used as CPAH in the next section) is 69601e8d1534d46214141a1461c966ff96699a66 (computed with http://www.sha1-online.com/).

13.3.1.1 Date/Time

For use case 1, all dates and times shall be converted to UTC (Time Zone Designation ‘Z’, ‘Z’ being given in the string value) during computing the hash value, so the hash value is reproduceable.

13.3.1.2 ValueWithUnit

Each value (NumericalValue) of combination of values (ValuesWithTolerances, ValueRange, ValueList, ValueSet, LimitsAndFits) shall be enclosed in a start and end separator so it is clear that they belong to one PropertyValue.

For use case 1, complex values like ValueRange, ValueWithTolerances, the value components shall be concatenated in alphabetical order.

For use case 2, complex values like ValueRange, ValueWithTolerances, the value components shall be concatenated in the order defined by the XSD.

If the value has a unit (except Integer, Real, Boolean, Logical as defined in section 12.5.3), and if the unit is different from the default unit defined in ExchangeContext.DefaultUnit, the ValueWithUnit.Unit.Prefix + Unit.Name shall be considered for building the hash value (especially in use case 1 after import/re-export of the dat), but does not need to be part of the hashed value since part of the definition of the PropertyValue in the original stpx file.

Remark: the xPath needs only to point ot the element of type ValueWithUnit => xPath only returns the ValueWithUnit node without content (since it is not of type string) and the algorithm used to build the AHash shall retrieve the content of ValueWithUnit, depending on which kind of ValueWithUnit applies.

In case of ValueList, redundant values shall not be pruned (like xPath would do it).

13.3.1.3 Double

For use case 1, double values need to be formatted in a unique way, so the hash value is reproduceable:

- Zero should simply be represented as 0.
- The first non zero digit should be to the left of the decimal.
- Remove trailing zeros from the decimal.
- Remove leading zeros from the exponent.
- When the exponent is zero remove the exponent portion altogether.
- The negative symbol is only included when required, positive signs are always suppressed.

For use case 1, a “format” specification (for the number of relevant digits of double attributes, see below) has to be defined, since the interfaces may provide more digits than the relevant precision, which may cause different roundings and would lead to different hash values. Its syntax shall be `[width]type` placed directly at the end of the attribute path, using the separator ‘:’.

- `width` gives the number of relevant digits after the dot in an exponential notation having only one digit before the dot
- `type` (E)xponential

It is possible to define another separator in the PropertyValues AHashAttributeFormatSeparatorBegin (instead of ‘:’) and AHashAttributeFormatSeparatorEnd (empty in case of ‘:’, used only with different begin and end separators like ‘<’ and ‘>’).

If no format is provided, all provided digits are assumed to be relevant. This can be applied for use case 2.

Examples: `-1e4, 1.43233e12, 1.278e-3, 1.2e1, 0, 3.4e1, 1.75`

This also applies to vector and matrix attributes like in CartesianTransformation or AxisPlacement.

Using xPath, the format and the number of relevant digits can only be defined on each element of type `xsd:double` using the XSLT function `format-number(LowerLimit, '#.#########')`. Therefore, a proprietary extension to xPath is necessary: add :xE where x is the integer size of the display including the digit left to the decimal and excluding the negative and decimal signs. The format applies to all elements of type `xsd:double` present within the Xpath element where it is defined. For example for a ValueWithTolerances, `PropertyValue:7E` applies to TolerancedValue, LowerLimit and UpperLimit. In case of RotationMatrix:7E and TranslationVector:7E, it applies to each element of the Matrix/Vector.

=> For the computation of the AHash, the double value shall be rounded. For example with 7E, +01.2345678000e012 would be formatted and truncated to 1.234568e12 when building the AHash value.

For use case 2, neither the format nor the number of relevant digits need to be defined since the values provided in the STEP file shall be taken as is (as strings).

13.3.1.4 Boolean/Logical

According to section 12.5.3, 1 for ‘TRUE’, 0 for ‘FALSE’, and 0.5 for ‘UNKNOWN’ shall be mapped using their decimal value.

13.3.1.5 ValueList, ValueSet

For use case 1, the SET value components shall be concatenated in the alphabetical order while the LIST value components shall be ordered as they are stored in the target PDM system.

For use case 2, the value components shall be concatenated in the order given in the stpx File.

The List/Set of values shall be enclosed in a start and end separator so it is clear that they belong to one PropertyValue.

13.3.1.6 Identifier(s)

In case an Id is not of type IdentifierString, but of type Identifier, all three XML attributes (id, idRoleRef.Id and idContextRef.Id shall be concatenated in the order defined in the XSD.

Each triple of id, idRoleRef.Id and idContextRef.Id shall be enclosed in a start and end separator so it is clear that they belong to one PropertyValue.

For use case 1, in case of multiple Identifiers, all the Identifiers shall be concatenated in alphabetical order.

For use case 2, in case of multiple Identifiers, all the Identifiers shall be concatenated in the order given in the stpx File.

Multiple identifiers shall be enclosed in a start and end separator so it is clear that they belong to one identifier attribute.

Remark: the xPath needs only to point ot the element of type IdentifierSelect => xPath only returns the Identifier node without content (since it is not of type string) and the algorithm used to build the AHash shall retrieve the IdentifierString, the Identifier or the multiple Identifiers elements, depending on which applies.

13.3.1.7 Class

In case a classification attribute is not of type ClassString, but of type Class, both XML attributes DefinedIn.Id and Id shall be concatenated in the order given in the stpx File and enclosed in a start and end separator so it is clear that they belong to one class attribute.

Remark: the xPath needs only to point ot the element of type ClassSelect => xPath only returns the Class node without content (since it is not of type string) and the algorithm used to build the AHash shall retrieve the ClassString or the Class elements, depending on which applies.

13.3.1.8 LocalizedString

In case a string attribute is not of type CharacterString, but LocalizedString, all the localized strings shall be concatenated and for each of them, both attributes `lang` and the string value shall be concatenated and enclosed in a start and end separator so it is clear that they belong to one string attribute.

For use case 1, all the LocalizedStrings shall be concatenated in alphabetical order of `lang`.

For use case 2, all the LocalizedStrings shall be concatenated in the order given in the stpx File.

Remark: the xPath needs only to point ot the element of type MultiLingualStringSelect => xPath only returns the MultiLingualStringSelect node without content (since it is not of type string) and the algorithm used to build the AHash shall retrieve the CharacterString or the LocalizedStrings, depending on which applies.

13.3.1.9 Unset

If the given path for an attribute as no match, either because the path is not fulfilled or because the attribute value ist unset, the value of the attribute shall be unset

In order to distinguish between empty and unset attribute, empty shall be mapped as an empty value and unset as ‘&’. Since & is mapped as “&amp;” if it appears within the attribute values, it will not collide with an attribute value.

However, it is possible to define another unset character in the PropertyValue AHashUnsetFlag.

13.3.1.10 Subtypes

A given subtype may be restricted in xPath using `[@xsi:type='bom:ValueList']`, but this is not supported by all xPath engines (like XMLSpy).

13.3.1.11 Enumerations

A given enumeration value may be restricted in xPath using `[PartCategoryEnum='assembly']` and `PartCategoryEnum/text()` to retrieve the value itself.

13.3.2 Definition of AHashStructureAttributes

AHashStructureAttributes is a comma separated list of the structure relationship attributes underneath the object, that have their values included in the AHash calculation.
However, it is possible to define another separator in the PropertyValues AHashAttributesSeparatorBegin (instead of ‘,’) and AHashSeparatorEnd (empty in case of ‘,’, used only with different begin and end separators like ‘<’ and ‘>’).

xPath prunes all duplicate and empty values from its result. Therefore, a two step approach is needed to get the relevant structure attributes for each child of the structure relationship:

1. Set AHashStructureAttributes[1] with the relative path from the object to the chosen relationship
2. Set AHashStructureAttributes[2] to [n] with the xPaths to the relevant attribute, expressed relatively to AHashStructureAttributes[1].

All aspects of the previous section apply, except that in this case, the AHash computation is done in three steps:

1. Hash computation based on AHashAttributes, called the Current Part Attribute Hash (CPAH), as defined in the previous section
2. Computation of AHashStructureAttributes[1]
   For each result found, computation/concatenation based on AHashStructureAttributes[2] to [n]
3. String concatenation of the CPAH value with the result of step 2
4. Hash computation of the whole string

In order to handle multiple relationships, the attribute list related to each relationship shall be separated. An appropriate separator shall not occur in the attribute values. The use of XML tag delimiter ‘<’ and ‘>’ is recommended (while these characters are mapped as &lt; and &gt; if they appear within the attribute values) since it will not collide with an attribute value.
However, it is possible to define another separator in the PropertyValues AHashStructureSeparatorBegin and AHashStructureSeparatorEnd

Example of AHashStructureAttributes: the line feeds (replacing the ‘,’) are here only for better readability but shall not be part of the real data:

```
<    -- first relationship
 ./ViewOccurrenceRelationship[xsi:type="bom:NextAssemblyOccurrenceUsage"]   -- rela-
tive path of the relationship

 //Occurrence[@uid=./Related/@uidRef]/Id/@id

 ./Placement/CartesianTransformation/RotationMatrix:7E
 ./Placement/CartesianTransformation/TranslationVector:7E

 //Occurrence[@uid=./Related/@uidRef]/parent::PartView/parent::Views/parent::PartVer-
sion/parent::Versions/parent::Part/Id/Identifier

 ./PropertyValueAssignment/AssignedPropertyValues/PropertyValue[Name/Character-
String='PDMIFNaou_Boolean']
>
<   -- second example of relationship, not used in the following example
 ./parent::Views/parent::PartVersion/PartVersionRelationship[Relation-
Type/ClassString='alternative']
 …
>
```

Here the value without line feeds (with ‘,’ as separator):

```
<./ViewOccurrenceRelationship[xsi:type="bom:NextAssemblyOccurrenceUsage"],//Occur-
rence[@uid=./Related/@uidRef]/Id/@id,./Placement/CartesianTransformation/RotationMa-
trix:7E,./Placement/CartesianTransformation/TranslationVector:7E,//Occur-
rence[@uid=./Related/@uidRef]/parent::PartView/parent::Views/parent::PartVersion/par-
ent::Versions/parent::Part/Id/Identifier,./PropertyValueAssignment/AssignedProper-
tyValues/PropertyValue[Name/CharacterString='PDMIFNaou_Boolean']>
```

Equivalence using EXPRESS path operators (just for information):

```
SELF<-Relating{NextAssemblyOccurrenceUsage}::Related::Id,
SELF<-Relating{NextAssemblyOccurrenceUsage}::Placement::RotationMatrix,
SELF<-Relating{NextAssemblyOccurrenceUsage}::Placement::TranslationVector,
SELF<-Relating{NextAssemblyOccurrenceUsage}::Related::Definition::ViewOf::Ver-
sionOf::Id,
SELF<-Relating{NextAssemblyOccurrenceUsage}<-AssignedTo{PropertyValueAssignment}::As-
signedPropertyValues{PropertyValue | Name = ‘PDMIFNaou_Boolean’},
SELF::ViewOf<-Relating{PartVersionRelationship}…
```

Example of string concatenation of the AHash attribute values before computing the AHash: the line feeds and blanks are here only for better readability but shall not be part of the real data

```
<69601e8d1534d46214141a1461c966ff96699a66>
<
<
  <left bracket>
  <0 -0.866025 -0.5 1 0 0 0 -0.5 0.866025>
  <5e1 -1.6967263e1 1.680486e1>
  <
   <
    <l-bracket-A>
    <identification information>
    <company-a.com>
   >
   <
    <l-bracket-B>
    <exchange identification information>
    <company-b.com>
   >
   <
    <l-bracket-C>
    <identification information>
    <company-c.com>
   >
  >
  <
   <logical representation>
   <0>
  >
 >
```

```
<
  <plate.1>
  <-1 0 0 0 0.5 -0.866025 0 -0.866025 -0.5>
  <1e2 -3.25e1 -5.629165e1>
  <
   <
    <plate-A>
    <identification information>
    <company-a.com>
   >
   <
    <plate-B>
    <exchange identification information>
    <company-b.com>
   >
   <
    <plate-C>
    <identification information>
    <company-c.com>
   >
  >
  <
   <logical representation>
   <0>
  >
 >
 <
  <prod4.1>
  <0 -0.5 0.866025 0.5 0.75 0.433013 -0.866025 0.433013 0.25>
  <4e1 2.5 -1.16913e2>
  <
   <
    <nut and rod-A>
    <identification information>
    <company-a.com>
   >
   <
    <nut and rod-B>
    <exchange identification information>
    <company-b.com>
   >
   <
    <nut and rod-C>
    <identification information>
    <company-c.com>
   >
  >
  <
   <logical representation>
   <0>
  >
 >
 <
  <right bracket>
  <0 0.866025 0.5 1 0 0 0 0.5 -0.866025>
  <5e1 -4.803274e1 -1.293882e2>
  <
   <
    <l-bracket-A>
    <identification information>
    <company-a.com>
   >
   <
    <l-bracket-B>
    <exchange identification information>
    <company-b.com>
   >
   <
    <l-bracket-C>
    <identification information>
    <company-c.com>
   >
  >
  <
   <logical representation>
   <0>
  >
 >
>
```

Here the value without line feeds:

```
<69601e8d1534d46214141a1461c966ff96699a66><<<left bracket><0 -0.866025 -0.5 1
0 0 0 -0.5 0.866025><5e1 -1.696726e1 1.680486e1><<<l-bracket-A><identifica-
tion information><company-a.com>><<l-bracket-B><exchange identification in-
formation><company-b.com>><<l-bracket-C><identification information><company-
c.com>>><<logical representation><0>>><<plate.1><-1 0 0 0 0.5 -0.866025 0 -
0.866025 -0.5><1e2 -3.25e1 -5.629165e1><<<plate-A><identification infor-
mation><company-a.com>><<plate-B><exchange identification information><com-
pany-b.com>><<plate-C><identification information><company-c.com>>><<logical
representation><0>>><<prod4.1><0 -0.5 0.866025 0.5 0.75 0.433013 -0.866025
0.433013 0.25><4e1 2.5 -1.169134e2><<<nut and rod-A><identification infor-
mation><company-a.com>><<nut and rod-B><exchange identification infor-
mation><company-b.com>><<nut and rod-C><identification information><company-
c.com>>><<logical representation><0>>><<right bracket><0 0.866025 0.5 1 0 0 0
0.5 -0.866025><5e1 -4.803274e1 -1.293882e2><<<l-bracket-A><identification in-
formation><company-a.com>><<l-bracket-B><exchange identification infor-
mation><company-b.com>><<l-bracket-C><identification information><company-
c.com>>><<logical representation><0>>>>
```

The AHash value for this string is 800278a65702afbf845477d4cee7f1798913127a (computed with http://www.sha1-online.com/).

In order to handle structure relationship has multiple childs, the AHashStructureAttribute values for each child shall be enclosed in a start and end separator so it is clear that they belong to different childs. For more details, see AHashSeparatorBegin and AHashSeparatorEnd defined ins the previous section.

Since especially the backward path expression (similar to the USEDIN EXPRESS function) returns an unordered SET back, it is important that the order is kept over all the AHashStructure-attributes that use the same path, so that the multiple childs are concatenated consistently.

For use case 1, the order of the backward SET-values shall be the alphabetical order of the internal ids of the referenced PDM objects.

For use case 2, the order of the backward SET-values shall be the alphabetical order of the uids of the referenced XML objects.

In order to be able to exchange/archive the product structure in nested files (see section 9.3), the AHashStructureAttributes shall contain only key attributes of the child parts that are used to reference it in a nested file. Otherwise, every local changes in the child parts would oblige to compute bottom up all the AHashes of the whole product structure.

13.3.3 Template “AHash”

This section specifies what and how to attach the AHash, AHashAttributes, AHashStructure-Attributes and AHashAlgorithm properties to an object.

Currently, it is recommended to attach a AHash only to a PartView or to an AssemblyDefinition.

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

```
<PropertyDefinition uid="ID_216">
 <Id id="quality property"/>
 <PropertyType>
  <ClassString>assembly validation property</ClassString>
 </PropertyType>
</PropertyDefinition>
<PropertyDefinition uid="ID_216-1">
 <Id id="quality property"/>
 <PropertyType>
  <ClassString>product structure validation property</ClassString>
 </PropertyType>
</PropertyDefinition>

<Part uid="p--0000000017D374A0">
…
 <Id>
  <Identifier uid="pid--0000000017D374A0--id1" id="as1-A" idRoleRef="rl--ii"
idContextRef="o--000000178"/>
  <Identifier uid="pid--0000000017D374A0--id2" id="as1-B" idRoleRef="rl--eii"
idContextRef="o--000000179"/>
  <Identifier uid="pid--0000000017D374A0--id3" id="as1-C" idRoleRef="rl--ii"
idContextRef="o--000000180"/>
 </Id>
…
 <Versions>
  <PartVersion uid="pv--0000000017D374A0--id1">
…
   <Views>
    <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--0000000017D374A0--
id1">
…
     <PropertyValueAssignment uid="ID_225">
      <AssignedPropertyValues>
       <PropertyValue uid="ID_224" xsi:type="n0:NumericalValue">
        <Definition>
         <PropertyDefinition uidRef="ID_216"/>
        </Definition>
        <Name>
         <CharacterString>number of children</CharacterString>
        </Name>
        <Unit uidRef="ID_x543008816"/>
        <ValueComponent>4</ValueComponent>
       </PropertyValue>
       <PropertyValue uid="ID_224-1" xsi:type="n0:StringValue">
        <Definition>
         <PropertyDefinition uidRef="ID_216-1"/>
        </Definition>
        <Name>
         <CharacterString>AHash</CharacterString>
        </Name>
        <ValueComponent>
         <CharacterString>800278a65702afbf845477d4cee7f1798913127a</Charac-
terString>
        </ValueComponent>
       </PropertyValue>
       <PropertyValue uid="ID_224-2" xsi:type="n0:StringValue">
        <Definition>
         <PropertyDefinition uidRef="ID_216-1"/>
        </Definition>
        <Name>
         <CharacterString>AHashAttributes</CharacterString>
        </Name>
        <ValueComponent>
         <CharacterString>./parent::Views/parent::PartVersion/parent::Ver-
sions/parent::Part/Id/Identifier,./parent::Views/parent::PartVersion/par-
ent::Versions/parent::Part/Name,//PersonInOrganization[@uid=./par-
ent::Views/parent::PartVersion/OrganizationOrPersonInOrganizationAssign-
ment[Role/ClassString='creator']/AssignedPersonOrOrganiza-
tion/@uidRef]/Id/Identifier/@id,./parent::Views/parent::PartVer-
sion/DateTimeAssignment[Role/ClassString='creation']/AssignedDate,./ Id/Iden-
tifier,//Approval[@uid=./parent::Views/parent::PartVersion/ApprovalAssign-
ment[not(boolean(./Role))]/AssignedApproval/@uidRef]/Sta-
tus/ClassString,./PropertyValueAssignment/AssignedPropertyValues/Proper-
tyValue[Name/CharacterString='PDMIFPartTol_xxx']:7E,./PropertyValueAssign-
ment/AssignedPropertyValues/PropertyValue[Name/Character-
String='PDMIFPartVers_ValueList'],//Class[@uid=//Classification[@uid=./par-
ent::Views/parent::PartVersion/parent::Versions/parent::Part/Classi-
fiedAs/Classification/@uidRef]/Class/Class/@uidRef],./parent::Views/par-
ent::PartVersion/parent::Versions/parent::Part/PartTypes[PartCatego-
ryEnum='assembly']/PartCategoryEnum/text()</CharacterString>
        </ValueComponent>
       </PropertyValue>
       <PropertyValue uid="ID_224-3" xsi:type="n0:StringValue">
        <Definition>
         <PropertyDefinition uidRef="ID_216-1"/>
        </Definition>
        <Name>
         <CharacterString>AHashStructureAttributes</CharacterString>
        </Name>
        <ValueComponent>
         <CharacterString>&lt;./ViewOccurrenceRelation-
ship[xsi:type="bom:NextAssemblyOccurrenceUsage"],//Occurrence[@uid=./Re-
lated/@uidRef]/Id/@id,./Placement/CartesianTransformation/RotationMa-
trix:7E,./Placement/CartesianTransformation/TranslationVector:7E,//Occur-
rence[@uid=./Related/@uidRef]/parent::PartView/parent::Views/parent::PartVer-
sion/parent::Versions/parent::Part/Id/Identifier,./PropertyValueAssign-
ment/AssignedPropertyValues/PropertyValue[Name/CharacterString='PDMIF-
Naou_Boolean']&gt;</CharacterString>
        </ValueComponent>
       </PropertyValue>
       <PropertyValue uid="ID_224-4" xsi:type="n0:StringValue">
        <Definition>
         <PropertyDefinition uidRef="ID_216-1"/>
        </Definition>
        <Name>
         <CharacterString>AHashAlgorithm</CharacterString>
        </Name>
        <ValueComponent>
         <CharacterString>SHA1</CharacterString>
        </ValueComponent>
       </PropertyValue>
      </AssignedPropertyValues>
      <ClassifiedAs>
       <Classification uidRef="ID_217"/>
      </ClassifiedAs>
     </PropertyValueAssignment>
…
    </PartView>
   </Views>
…
  </PartVersion>
 </Versions>
…
</Part>
```

The Instance Model: AP242 Domain Model XML entities and attributes

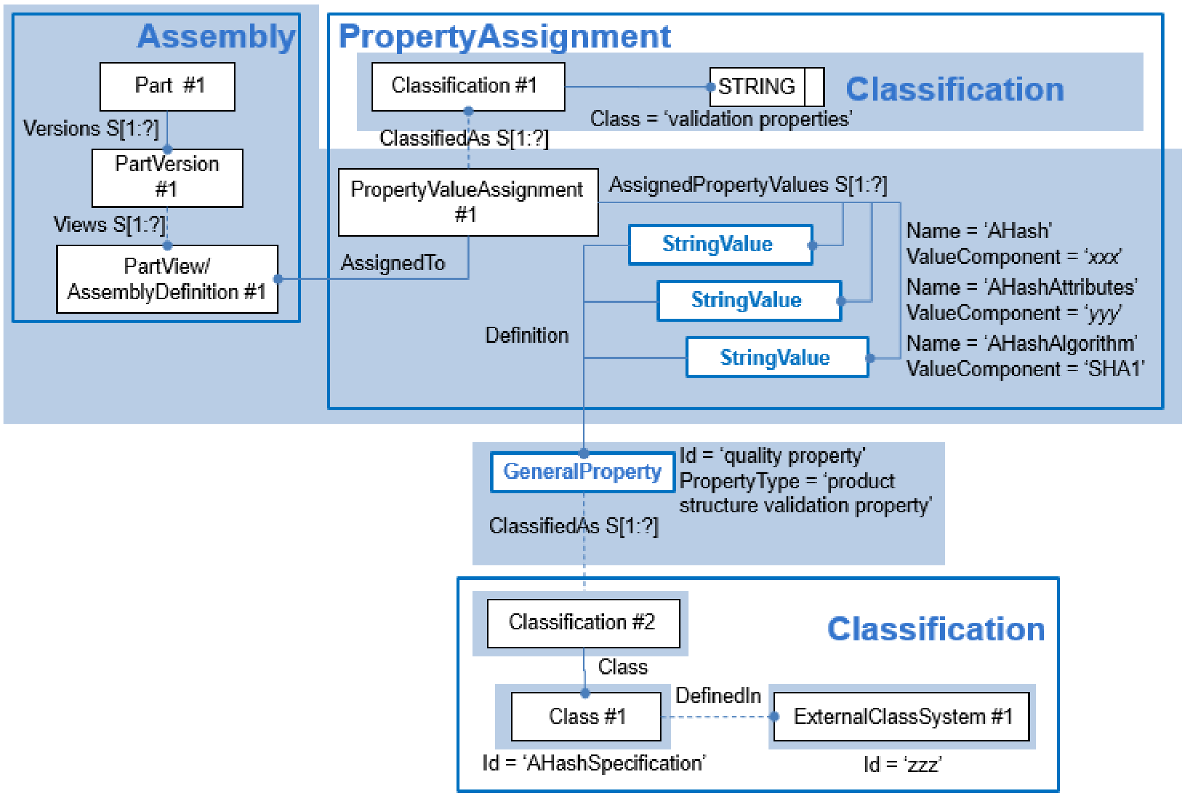


Figure 91: Template "AHash"

List of attributes and recommendation are similar to the PropertyAssignment template defined in chapter 6.2.

Preprocessor Recommendations:

- It is recommended that all the validation properties (AVPs and PSVP) use the same PropertyValueAssignment.
- It is recommended to include the specification used to dictate the calculation of the hash value ('AHashSpecification'). This may be a reference to the LOTAR specification, or it may be a company specific document which provides the recipe for calculation.
- If AHashAttributes, AHastStructureAttributes and AHashAlgorithm have the same values for all objects for which a AHash gets calculated, it is recommended to instantiate

them only in one object.
For this purpose, a JIRA issue TCSC410303-660 has been created to attach them to the ExchangeContext. Since a StringValue cannot be reused by multiple PropertyValueAssignments, but would have to be copied each time, the above recommendation keeps the datasize as small as possible.

Postprocessor Recommendations: None specified.

Related Entities: There are no specific related entities.

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

T.B.S.

13.3.4 Template “BHash and CHash”

This section specifies what and how to attach the BHash and BHashAlgorithm properties to a non-geometrical DigitalDocumentDefinition, as well as the CHash and CHashAlgorithm properties to a geometrical DigitalDocumentDefinition.

The Instance Model: AP242 Domain Model XML entities and attributes

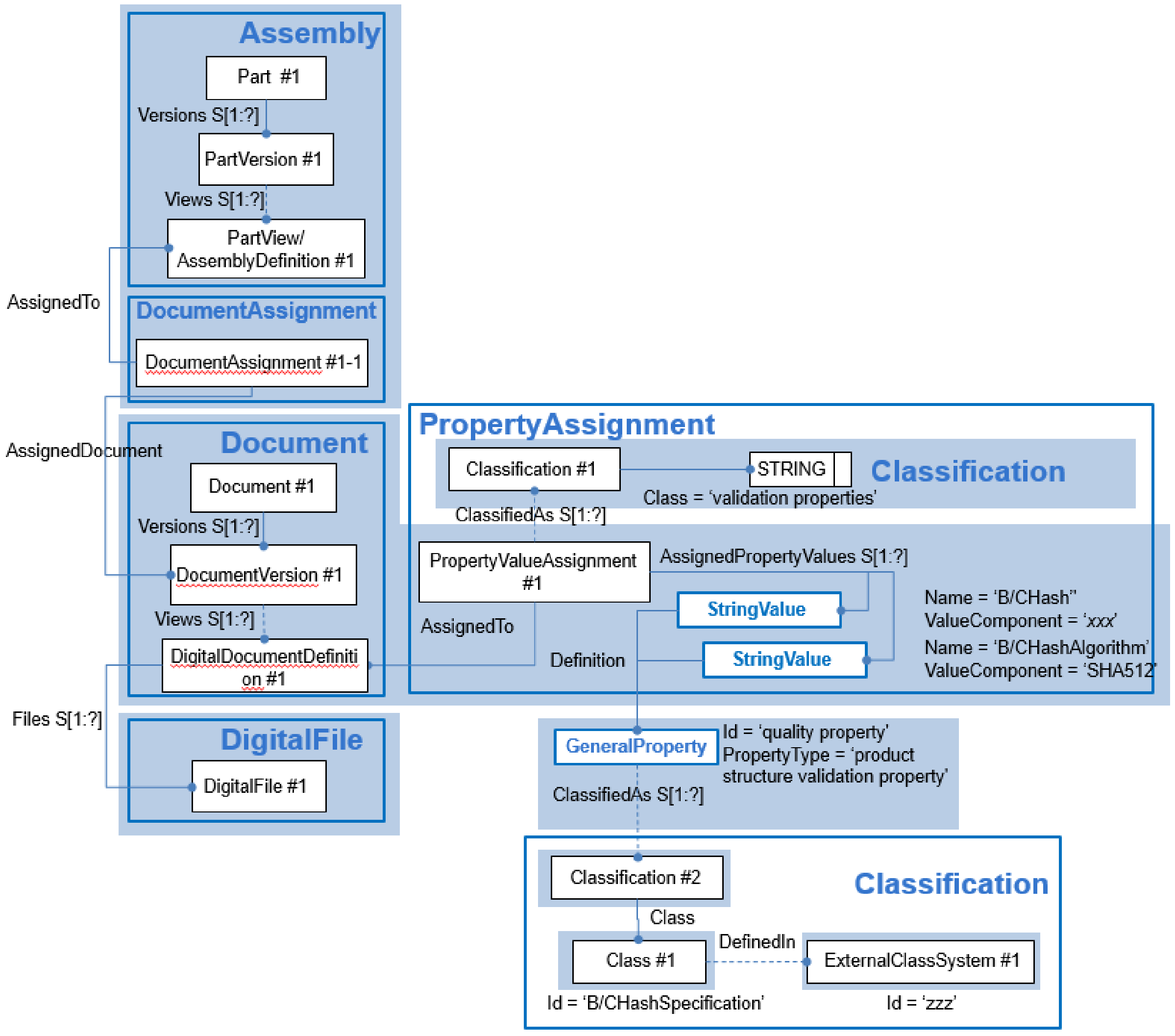


Figure 92: Template "BHash and CHash"

Preprocessor Recommendations:

- It is recommended to include the specification used to dictate the calculation of the hash value ('BHashSpecification' or 'CHashSpecification'). This may be a reference to the LOTAR specification, or it may be a company specific document which provides the recipe for calculation.
- If BHashAlgorithm has the same values for all objects for which a BHash gets calculated, it is recommended to instantiate it only in one object.
  For this purpose, a JIRA issue TCSC410303-660 has been created to attach them to the ExchangeContext. Since a StringValue cannot be reused by multiple PropertyValueAssignments, but would have to be copied each time, the above recommendation keeps the datasize as small as possible.
- Ditto for CHashAlgorithm.

Postprocessor Recommendations: None specified.

Related Entities: There are no specific related entities.

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

T.B.S.

14 Exchange of Customized PDM Information

This section describes how, additionally to the standard objects, structures and properties defined in the previous sections, the customized objects, structures and properties defined in the PDM system may be exchanged.

14.1 Supported Customization Features of the PDM Systems

Current PDM system offer a wide range of customization features. General concepts such as the definition of specialized objects, attributes and relationships, can easily be mapped to the AP242 Domain Model and exchanged between the different systems. The following sections provide an introduction to managing these concepts.

There are, however, concepts where it is currently not clear if and how they can be mapped not only to the AP242 Domain Model, but also between different PDM systems. Enumerations, sequences and formulas are a few examples.

These advanced concepts will be worked on based on user requirements and added to future releases of these Recommended Practices when an agreed solution has been found.

14.2 Customized Objects & Structures

14.2.1 Without the use of property groups

All the AP242 Domain Model entities have an attribute ‘ClassifiedAs’, which may reference one or many instances of Classification (for more details, refer to chapter 4.6.5).

Preprocessor Recommendations:

In case an object or structure is customized, its name shall be mapped to a Class (not a ClassString) defined in an ExternalClassSystem called ‘customized part type’ (for Parts), ‘customized document type’ (for Documents), ‘customized work request type’ (for WorkRequests), customized work order type’ (for WorkOrders or ‘customized activity type’ (for Activities). This class shall be referenced through ClassifiedAs via an instance of Classification with Role=’customized type’.

The idContextRef of the Class and ExternalClassSystem references the organization from which the customization is coming from.

It is not recommended to use the attributes Part.PartTypes and Document.DocumentTypes for this purpose. PartTypes/DocumentTypes shall contain recommended or individual values that indicate what kind of part/document it is, but not the name of customized PDM type.

If the PDM customization defines herarchies of customized types, like for example:

```
PDMIFPart_MB
  PDMIFElectricalPart_MB
     PDMIFCapacitor_MB
     …
   PDMIFMechanicalPart_MB
     PDMIFMechStandardPart_MB
```

…

it is recommended that only the leaf class gets referenced by ClassifiedAs.

The mapping of this hierarchy via ClassificationRelationship is currently out of scope of this document.

PropertyValue.Name shall be unique within all PropertyValues assigned by a PropertyValueAssignment.

Postprocessor Recommendations:

The properties of each customized object/structure shall be mapped to properties of the corresponding (standard or customized) object/structure of the target PDM system.

If PropertyValue.Name is not unique within a PropertyValueAssignment, an error shall be returned and the duplicate PropertyValue shall be ignored.

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

```
<ExternalClassSystem uid="ecs--cpp">
  <Id>
    <Identifier uid="cpp--id" id="customized part type" idRoleRef="C__5" id-
ContextRef="O__3"
  </Id>
</ExternalClassSystem>

<Class uid="cpp--pdmifmb">
  <DefinedIn uidRef="ecs--cpp"/>
  <Id id="PDMIFPart_MB"/>
</Class>

<Classification uid="cl--pdmifmb">
    <Class>
       <Class uidRef="cpp--pdmifmb"/>
    </Class>
  <Role>customized type</Role>
</Classification>

<Part uid="p--0000000017086CB0">
    <ClassifiedAs>
        <Classification uidRef="cl--pdmifmb"/>
    </ClassifiedAs>
...
</Part>
```

14.2.2 With the use of property groups

Note: Some concepts in this section, namely the definition and transfer of property groups, are included for completeness but have not yet been tested. Hence, these definitions might change in the future.

Some PDM systems like 3DExperience or Teamcenter group the customized properties in so-called Extension or Form objects.

Several extensions/forms may occur, if many groups of property are involved (for example energy consumption property group, water consumption property group and hazardous waste mass measure property group).

An extension/form may occur multiple times, if the same group of property is needed multiple times, for example the approval information (role, who, from several actors).

The same extension/from may be used by several PDM object types (part, document, …)

Some customized properties may be defined in the customized subtype of the PDM object (called 'Type' in the following picture).

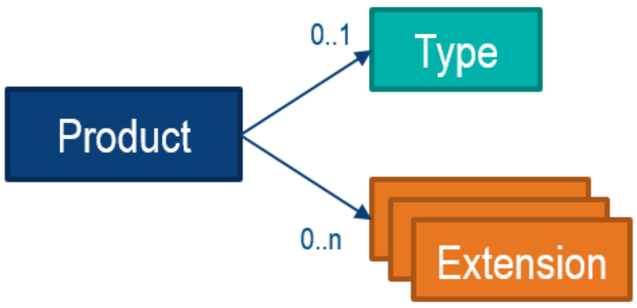


Figure 93: Use of extensions/forms in PDM systems

Preprocessor Recommendations:

- Refer to the mapping defined in section 14.2.1 for Classification, Class and ExternalClassSystem to map the customized PDM object (may be any object on which a PropertyValueAssignment is recommended, for example a part, a document, etc…).
- For each extension/form related to this PDM object:
    - Create a PropertyValueAssignment having a unique Id (maybe the name of the extension/form + #1, #2, etc… if the same extension/form is used multiple times on the same PDM object)
- additionally to the PropertyValueAssignment for the customized properties directly stored within the PDM object or stored in a default extension/form: this PropertyvalueAssignment has no Id
- Within a property group, properties of kind Date, Approval or Person/Organization shall be assigned to the PropertyValueAssignment that maps the group.

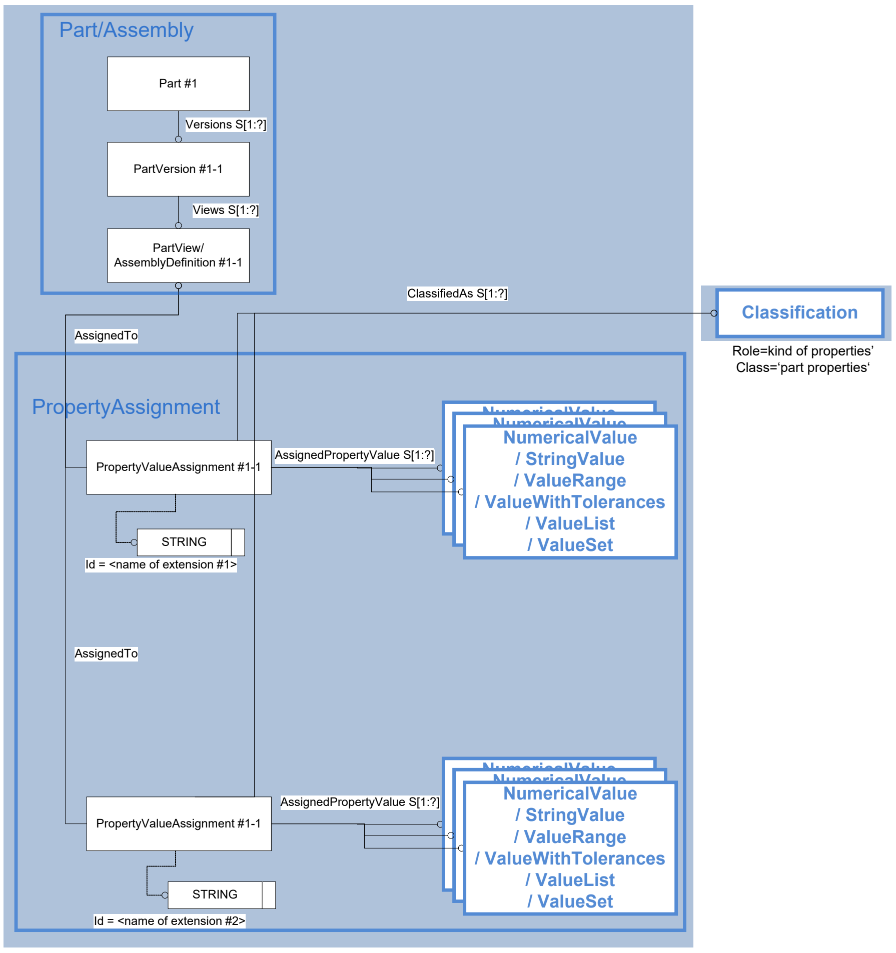

Part/Assembly
Part #1
Versions S[1:?]
PartVersion #1-1
Views S[1:?]
PartView/ AssemblyDefinition #1-1
ClassifiedAs S[1:?]
Classification
Role=kind of properties'
Class='part properties'
AssignedTo
PropertyAssignment
PropertyValueAssignment #1-1
AssignedPropertyValue S[1:?]
NumericalValue / StringValue / ValueRange / ValueWithTolerances / ValueList / ValueSet
STRING
Id = <name of extension #1>
AssignedTo
PropertyValueAssignment #1-1
AssignedPropertyValue S[1:?]
NumericalValue / StringValue / ValueRange / ValueWithTolerances / ValueList / ValueSet
STRING
Id = <name of extension #2>

Figure 94: Example of property groups on a Part having properties on the PartView.

Postprocessor Recommendations:

- The groups of properties are only for information. The target system may group the properties in another way than the source system.
- If the target system does not support groups of properties, they can be ignored during import => all PropertyValueAssignments of type 'part properties' shall be processed.
- If the user scenario requires that the original groups of properties are not lost during bidirectional exchange even if the target system does not support groups of properties, the postprocessor shall at least save this schema information (not necessary at value level) in order to re-export the original groups of properties.
- If PropertyValue.Name is not unique because two property groups have the same property name, the target system shall be able to distinguish the values of these properties having the same name.

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

This example maps a Part with:

- Three properties called 'PDMIFPartVers_Tol', 'PDMIFPartVers_Range' and 'PDMIFPartVers_String' on a subtype of part called 'PDMIFPartType'

- Three properties called 'PDMIFPartVers_Numerical ', 'PDMIFPartVers_Real' and 'PDMIFPartVers_Integer' on an Extension/Form called 'PDMIFForm1'
- Three properties 'PDMIFPartVers_Boolean', 'PDMIFPartVers_ValueList' and 'PDMIFPartVers_ValueSet' on an Extension/Form called 'PDMIFForm2'

All the other instances of 'PDMIFPartType' within the same AP242 XML file would reference the same instance of Classification/Class and its PropertyValues would reference the same instances of PropertyDefinition (grouped in the three groups mentioned above).

Other instances of Part belonging to another subtype of part would reference another instance of Classification/Class with other PropertyDefinitions, except for those properties coming from the same Extensions/Forms than in 'PDMIFPartType'.

```
<ExternalClassSystem uid="ID_240">
  <Id>
    <Identifier uid="ID_241" id="customized part type" idRoleRef="C__7" id-
ContextRef="O__3"> </Identifier>
  </Id>
</ExternalClassSystem>
<ExternalClassSystem uid="ID_381">
  <Id>
    <Identifier uid="ID_382" id="customized document type" idRoleRef="C__7"
idContextRef="O__3"> </Identifier>
  </Id>
</ExternalClassSystem>
…
<Class uid="ID_242">
  <ClassAttribute uid="ca_1">
    <AttributeDefinition uidRef="ID_g1"/>
    <Id id="PDMIFPartType"/>
  </ClassAttribute>
  <ClassAttribute uid="ca_2">
    <AttributeDefinition uidRef="ID_g2"/>
    <Id id="PDMIFForm1"/>
  </ClassAttribute>
  <ClassAttribute uid="ca_3">
    <AttributeDefinition uidRef="ID_g3"/>
    <Id id="PDMIFForm2"/>
  </ClassAttribute>
  <DefinedIn uidRef="ID_240"/>
  <Id id="PDMIFPartType"/>
</Class>
…
<PropertyDefinition uid="PD__35">
  <Id id="general property"/>
  <PropertyType>
    <ClassString>system property</ClassString>
  </PropertyType>
</PropertyDefinition>
<PropertyDefinition uid="ID_57_1">
  <Id id="general property"/>
  <PropertyType>
    <ClassString>customized PDM property</ClassString>
  </PropertyType>
</PropertyDefinition>
…
<Classification uid="Cl__36">
  <Class>
    <ClassString>part properties</ClassString>
  </Class>
  <Role>kind of properties</Role>
</Classification>
<Classification uid="ID_243">
  <Class>
    <Class uidRef="ID_242"/>
  </Class>
  <Role>customized type</Role>
</Classification>
…
<Part uid="Pa__4">
  <ClassifiedAs>
      <Classification uidRef="ID_243"/>
  </ClassifiedAs>
…
  <Versions>
    <PartVersion uid="PVe__9">
…
      <Views>
        <PartView uid="PV__13" xsi:type="n0:AssemblyDefinition">
…
          <PropertyValueAssignment uid="PVA__33">
            <AssignedPropertyValues>
              <PropertyValue uid="ID_159" xsi:type="n0:ValueWithTolerances">
                <Definition>
                  <PropertyDefinition uidRef="ID_57_1"/>
                </Definition>
                <Name>
                  <CharacterString>PDMIFPartVers_Tol</CharacterString>
                </Name>
…
              </PropertyValue>
              <PropertyValue uid="ID_162" xsi:type="n0:ValueRange">
                <Definition>
                  <PropertyDefinition uidRef="ID_57_2"/>
                </Definition>
                <Name>
                  <CharacterString>PDMIFPartVers_Range</CharacterString>
                </Name>
…
              </PropertyValue>
              <PropertyValue uid="ID_165" xsi:type="n0:StringValue">
                <Definition>
                  <PropertyDefinition uidRef="ID_57_3"/>
                </Definition>
                <Name>
                  <CharacterString>PDMIFPartVers_String</CharacterString>
                </Name>
            <ClassifiedAs>
              <Classification uidRef="Cl__36"/>
            </ClassifiedAs>
          </PropertyValueAssignment>
          <PropertyValueAssignment uid="PVA__33-1">
            <AssignedPropertyValues>
              </PropertyValue>
              <PropertyValue uid="ID_166" xsi:type="n0:NumericalValue">
                <Definition>
                  <PropertyDefinition uidRef="ID_57_4"/>
                </Definition>
                <Name>
                  <CharacterString>PDMIFPartVers_Numerical</CharacterString>
                </Name>
…
              </PropertyValue>
              <PropertyValue uid="ID_167" xsi:type="n0:NumericalValue">
                <Definition>
                  <PropertyDefinition uidRef="ID_57_5"/>
                </Definition>
                <Name>
                  <CharacterString>PDMIFPartVers_Real</CharacterString>
                </Name>
…
              </PropertyValue>
              <PropertyValue uid="ID_168" xsi:type="n0:NumericalValue">
                <Definition>
                  <PropertyDefinition uidRef="ID_57_6"/>
                </Definition>
                <Name>
                  <CharacterString>PDMIFPartVers_Integer</CharacterString>
                </Name>
            <ClassifiedAs>
              <Classification uidRef="Cl__36"/>
            </ClassifiedAs>
            <Id id="PDMIFForm1"/>
          </PropertyValueAssignment>
          <PropertyValueAssignment uid="PVA__33-1">
            <AssignedPropertyValues>
              </PropertyValue>
              <PropertyValue uid="ID_169" xsi:type="n0:NumericalValue">
                <Definition>
                  <PropertyDefinition uidRef="ID_57_7"/>
                </Definition>
                <Name>
                  <CharacterString>PDMIFPartVers_Boolean</CharacterString>
                </Name>
…
              </PropertyValue>
              <PropertyValue uid="ID_170" xsi:type="n0:ValueList">
                <Definition>
                  <PropertyDefinition uidRef="ID_57_8"/>
                </Definition>
                <Name>
                  <CharacterString>PDMIFPartVers_ValueList</CharacterString>
                </Name>
…
              </PropertyValue>
              <PropertyValue uid="ID_171" xsi:type="n0:ValueSet">
                <Definition>
                  <PropertyDefinition uidRef="ID_57_9"/>
                </Definition>
                <Name>
                  <CharacterString>PDMIFPartVers_ValueSet</CharacterString>
                </Name>
…
              </PropertyValue>
              <PropertyValue uid="SV__34" xsi:type="n0:StringValue">
                <Definition>
                  <PropertyDefinition uidRef="PD__35"/>
                </Definition>
                <Name>
                  <CharacterString>project</CharacterString>
                </Name>
…
              </PropertyValue>
            </AssignedPropertyValues>
            <ClassifiedAs>
              <Classification uidRef="Cl__36"/>
            </ClassifiedAs>
            <Id id="PDMIFForm2"/>
          </PropertyValueAssignment>
          </PartView>
      </Views>
…
    </PartVersion>
  </Versions>
…
</Part>
```

14.3 Customized Properties

Preprocessor Recommendations:

The customized properties of a customized object or structure shall be mapped as described in the chapter 12 using the value ‘customized PDM property’ for PropertyDefinition.PropertyType.

They are defined in the customized PDM object or structure, either as classification properties (like in SAP) or as subtype of the standard PDM object/structure (like in Teamcenter, Windchill, …) or as additional properties of the standard PDM object/structure (like in Aras).

Customized properties of type Date can be mapped beside the standard date properties (4.6.11) with Role=<name of the customized date property>.

Dito for Persons (4.6.18) and PersonInOrganizations (4.6.19): using OrganizationOrPersonInOrganizationAssignment with dedicated role = <name of the customized person or organization property. The use of DateAndPersonAssignment is not recommended.

Dito for Approvals (4.6.17): using ApprovalAssignment with dedicated role = <name of the customized approval property>.

In order to exchange a lifecycle workflow, i.e. the supported approval status transitions (often defined as a graph in the PDM systems), the use of ApprovalRelationship with RelationType=’workflow successor’ is recommended. An approval may have zero, one or multiple predecessors/successors.

If more than one lifecycle workflow is exchanged (for example one for Parts and one for Documents, or even one for CAD Documents and one for non-CAD Documents), the Approvals of each workflow shall be assigned to a distinct ExternalClassSystem.

Here is an example:

```
<ExternalClassSystem uid="ecs--lc-part">
  <Id>
    <Identifier uid="fd--lc-part-id1" id="part lifecycle" idRoleRef="rl--ii"
idContextRef="o--000000178"/>
  </Id>
</ExternalClassSystem>
<Class uid="cl--lc-part-1">
```

```
  <DefinedIn uidRef="ecs--lc-part"/>
  <Id id="part lifecycle value 1"/>
</Class>
<Class uid="cl--lc-part-2">
  <DefinedIn uidRef="ecs--lc-part"/>
  <Id id="part lifecycle value 2"/>
</Class>

<Approval uid="app--1">
  <Description>
    <CharacterString>disposition</CharacterString>
  </Description>
  <Status>
    <Class uidRef="cl--lc-part-1"/>
  </Status>
</Approval>

<Approval uid="app--2">
  <Description>
    <CharacterString>disposition</CharacterString>
  </Description>
  <Status>
    <Class uidRef="cl--lc-part-2"/>
  </Status>
</Approval>
```

The following values shall be used if a customized property has an empty value:

- ‘’.for STRING properties
- [‘’] for Set/List properties
- 2147483647 (MAX_LONG) for INTEGER or REAL properties with unit
- 1970-01-01T00:00:00 for Date properties
- UNKNOWN for Boolean Properties
- ‘/NULL’ for Approval, Organization and Person properties

Only the customized properties having a value (even empty) shall be exchanged and shall be imported/updated in the target system. Unset customized properties shall not be exchanged and shall not be updated in the target system.

Some PLM systems (Out of the Box) do not differentiate properly between unset values and empty values in customized properties of kind. In this case:

- unset string properties may be imported/exported as empty string values
- unset boolean properties may be imported/exported as FALSE

Removed customized properties (update from a non-unset value to an unset value) shall be mapped as a DeltaChangeActivity of type DeleteElement (see section 9.5.2). For example:

```
<DeltaChange uid="dc-1">
  <ChangeActivities>
    <DeltaChangeActivity uid="dca-1" xsi:type="bom:DeleteElement">
      <PreviousDesignObject>
        <PropertyValue uidRef="ID_1708"/>
      </PreviousDesignObject>
    </DeltaChangeActivity>
    <DeltaChangeActivity uid="dca-2" xsi:type="bom:DeleteElement">
      <PreviousDesignObject>
        <ApprovalAssignment uidRef="ID_1738"/>
      </PreviousDesignObject>
    </DeltaChangeActivity>
    <DeltaChangeActivity uid="dca-3" xsi:type="bom:DeleteElement">
      <PreviousDesignObject>
        <DateTimeAssignment uidRef="ID_1727"/>
      </PreviousDesignObject>
    </DeltaChangeActivity>
    <DeltaChangeActivity uid="dca-3" xsi:type="bom:DeleteElement">
      <PreviousDesignObject>
        <OrganizationOrPersonInOrganizationAssignment uidRef="ID_1728"/>
      </PreviousDesignObject>
    </DeltaChangeActivity>
  </ChangeActivities>
…
</DeltaChange>
```

Assignments to empty customized properties of kind Approvals shall be exchanged as a dummyApproval having an Approval.Status='/NULL'.

Assignments to empty customized properties of kind Person/Organizations shall be exchanged as a dummy PersonInOrganization having Organization.Id='/NULL' and Person.LastName='/NULL'.

In case of Part and Document, to simplify the postprocessors:

- the customized DateTimeAssignments, ApprovalAssignments and OrganizationOrPersonInOrganizationAssignments shall be associated to the PartVersion or DocumentVersion (and not to PartView or DocumentDefinition), just as described in the sections 5.1.2 and 8.1.2.
- the customized PropertyAssignments shall be associated to the PartView or DocumentDefinition (and not to PartVersion or DocumentVersion), just as described in the sections 5.1.3 and 8.1.3.

In case the sender PDM system has distinct part/document master/version objects, customized PropertyAssignments, DateTimeAssignments, ApprovalAssignments and OrganizationOrPersonInOrganizationAssignments shall only be associated to Part/Document if they apply to all PartVersions or DocumentVersions

Since the object Occurrence (SingleOccurrence, SpecifiedOccurrence, …) is not a distinct object in most PDM systems, but rather mapped together with the ViewOccurrenceRelationship (as NextAssemblyOccurrenceUsage), no customized properties shall be associated to the Occurrences. If needed, they should be all associated to the NextAssemblyOccurrenceUsage

Postprocessor Recommendations:

Each relevant customized property shall be mapped to the corresponding (standard or customized) property of the target PDM system.

If the target PDM system does not support multi-value properties (ValueSet/ValueList), ValueLimit, ValueRange, ValueWithtolerances or LimitsAndFits, these properties shall be either ignored or splitted into simple properties by the postprocessor.

DateTimeAssignments, ApprovalAssignments and OrganizationOrPersonInOrganizationAssignments associated to PartView or DocumentDefinition may be ignored. PropertyAssignments associated to PartVersion or DocumentVersion also.

If some customized PropertyAssignments, DateTimeAssignments, ApprovalAssignments or OrganizationOrPersonInOrganizationAssignments are associated to Part/Document, but the target PDM system hasn't got distinct part/document master/version objects, they shall be mapped only to those PartVersions or DocumentVersions that are coming along with this part/document in the XML file.

Customized properties associated to object Occurrence (SingleOccurrence, SpecifiedOccurrence, …) shall be mapped as if they were associated to the corresponding NextAssemblyOccurrenceUsage

For the special case, where some customized properties need to be exchanged on PartViewRelationship, see chapter 7.4 for more details how to map and interprete them

15 Multi-View Product Structure Representation

This section describes how to represent multiple views of the product structure and how to keep relationship between these views.

The different views covered are:

- AsDesigned,
- AsPlanned,
- AsDocumented / AsCertified,
- AsPlanned,
- AsBuild,
- AsDelivered, and
- AsSupported / AsFlying / AsMaintained / AsOperated.

The following topics are described in this chapter:

- Multiple view concept
- Product structure with Part and Occurrence
- Product structure with IndividualPart
- Relationship for tracking usage

15.1 Multiple view concept

The view concept is managed via the Template "ViewContext" (see section 4.6.8). These views relate to virtual / physical instances of the products. Each instance can be identified by unique (within the organization) serial number (SN).

The 'multi-SN keyword' describes product structures defined with Part and Occurrence of Part.

The 'one per SN' keyword describes product structures defined with IndividualPart.

Mapping between the views introduced above and the LifeCycleStage concept:

- AsDesigned (multi-SN) => 'design'
- AsPlanned (multi SN) => 'manufacturing planning'
- AsDocumented / AsCertified (multi-SN) => 'documentation' or 'certification'
- AsPlanned (one per SN) => 'manufacturing realization'
- AsBuild (one per SN) => 'built' or 'manufactured'
- AsDelivered (one per SN) => 'delivery'
- AsSupported / AsFlying / AsMaintained / AsOperated (one per SN) => 'support'

The following table show values that should be used for the management of complex product structure with differents views.

ViewContext.ApplicationDomain	ViewContext.LifeCycleStage
'mechanical design' 'electrical design'	'design'
'mechanical design' 'electrical design'	'manufacturing planning'
'product delivery'	'documentation' or 'certification'
'mechanical design' 'electrical design'	'manufacturing realization'
'mechanical design' 'electrical design'	'built" or 'manufactured'
'product delivery'	'delivery'
'product support'	'support'

15.2 Product structure with Part and Occurrence

This chapter is for the representation of AsDesigned, AsPlanned (engineering), and AsDocumented/AsCertified.

This product structure is composed of Part and Occurrences. Each product structure contains a full set of information and supports multiple effectivies / MSN:

- Part/PartVersion/PartView (Section 5.1)
- Properties (Section 6)
- Occurrence and relationship with matrix positioning (Section 7.1)

Even if the STEP information model allows multiple PartViews under a PartVersion, for use cases with multiple views each product structure must be duplicated for compatibility with tools, i.e. one Part/PartVersion/PartView for AsDesigned products and a different Part/PartVersion/PartView for AsPlanned products. There are no restrictions for a STEP file to have multiple root nodes, one per view.

Depending on the use case, the same part geometry may be shared by the AsDesigned and AsPlanned parts, or each of them may have his own geometry.

Example of "As designed and As planned" duplicate product structure

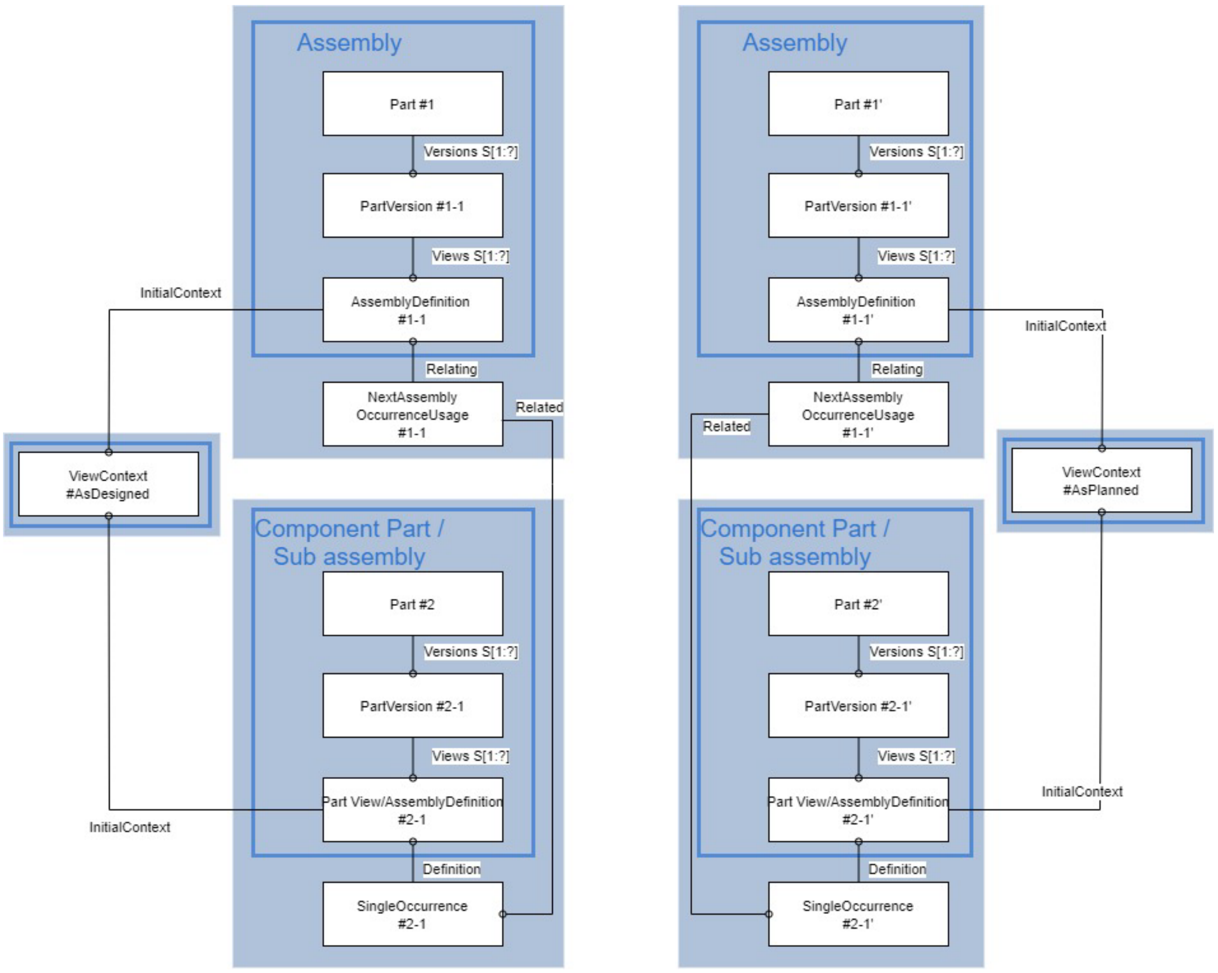


Figure 95: Representation of As Designed and As Planned views

15.3 Product structure with IndividualPart

This chapter is for the representation of AsPlanned (manufacturing), AsBuild, AsDelivered and AsSupported/AsFlying/AsMaintained/AsOperated product structure.

This representation based on `IndividualPart` objects is currently out of scope.

15.4 Relationship between views based on product structure with Part and Occurrence

To keep track of the usage of product between several views use the `PartViewRelationship` (section 7.4) only for the root nodes relation and `OccurrenceRelationship` for the other structure nodes.

Example of “As designed and As planned” product structure with their usage relationship on `PartView` and `Occurrence`

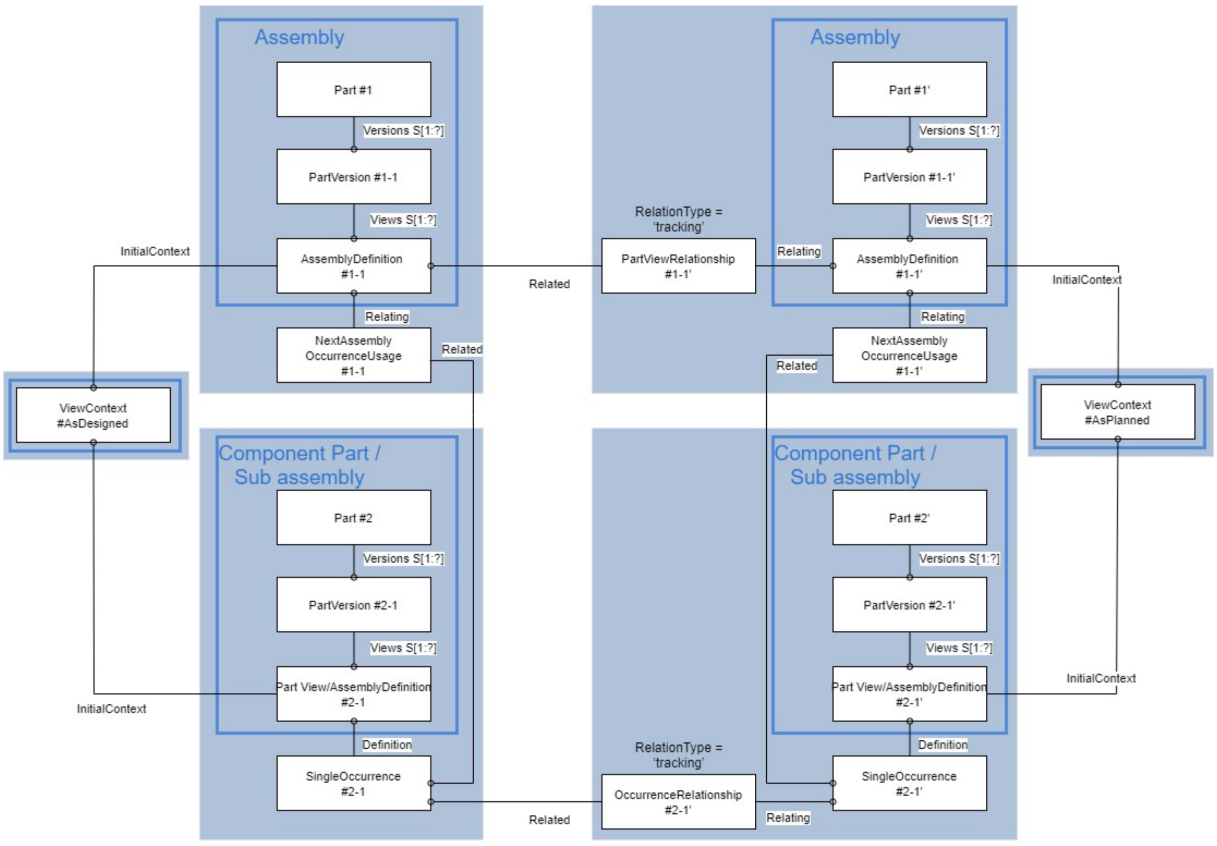
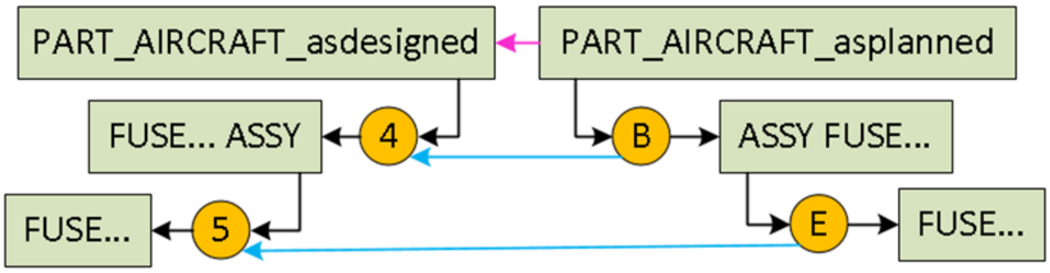


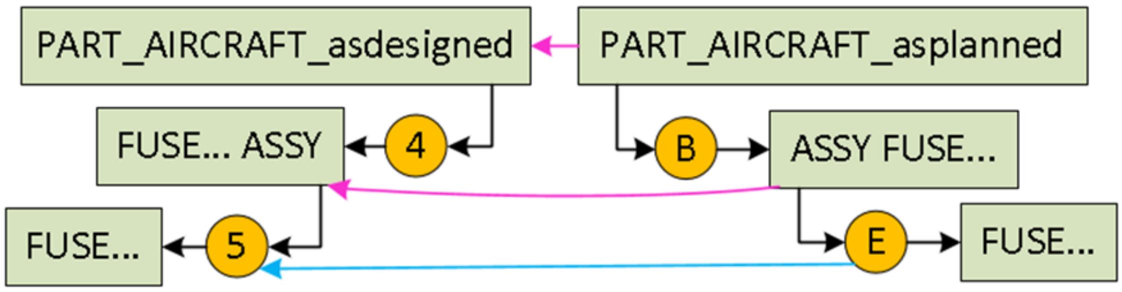
Figure 96: Representation of tracking relationship

Preprocessor Recommendations:

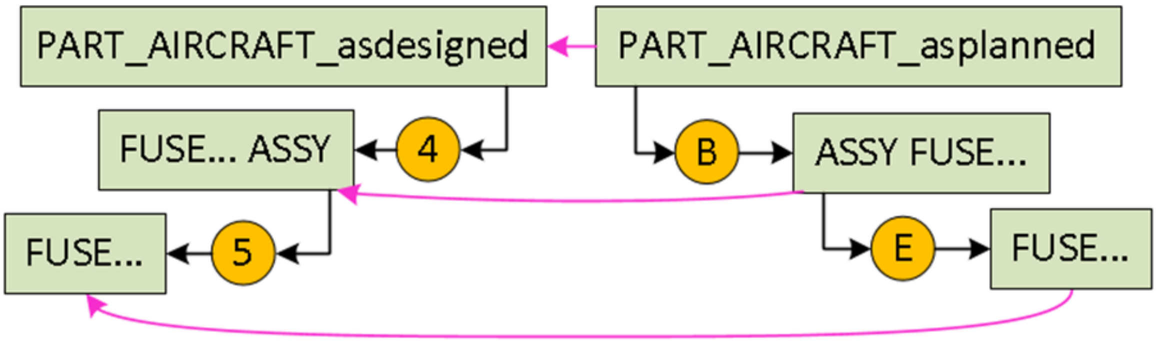
- Tracking relationships on sub-assembly / single part level shall build upon tracking relationships on parent assembly levels, so they are fully defined within the whole assembly. Otherwise, the import in some PDM systems (like 3DExperience) will not be possible:
  - Here a valid example of PartView Relationship on level 1 and Occurrence Relationships on level 2 and 3, saying that the occurrence E the as-planned Fuse within Assy Fuse… is tracked to the occurrence 5 of as-designed Fuse, but only in the context of the occurrence B of as-planned Assy Fuse… tracked to occurrence 4 of as-designed Fuse…Assy.
    Using this tracking, the as-planned Assy Fuse… may have only one occurrence within the as-planned Part_Aircraft.



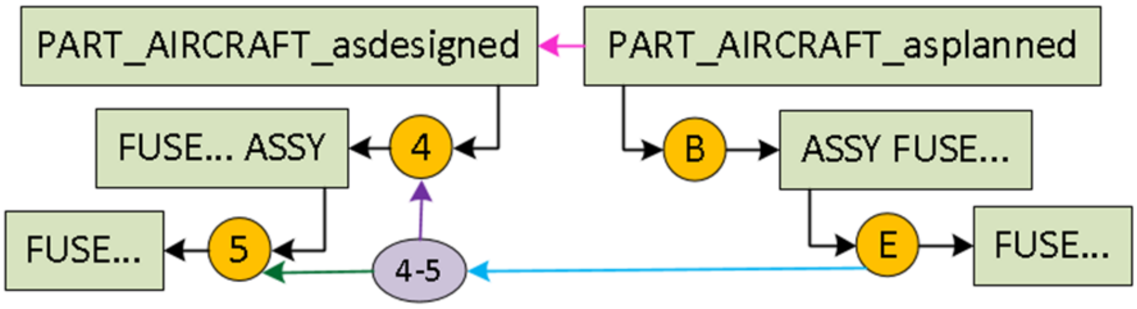
  - Here a valid example of PartView Relationships on level 1 and 2, and Occurrence Relationship on level 3, saying that the occurrence E of as-planned Fuse within Assy Fuse… is tracked by occurrence 5 of as-designed Fuse, applying to all the occurrences of as-planned Assy Fuse… anywhere in the whole as-planned assembly.



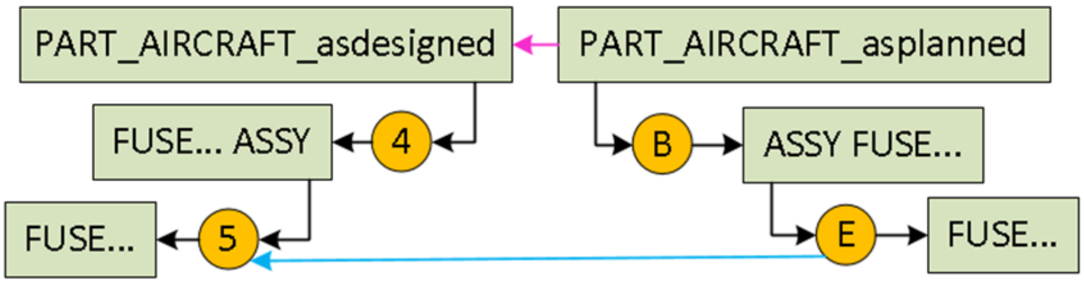
- Here a valid example of PartView Relationships on level 1, 2 and 3, saying that all the occurrences of as-planned Fuse anywhere within the whole assy are tracked to the as-designed Fuse.



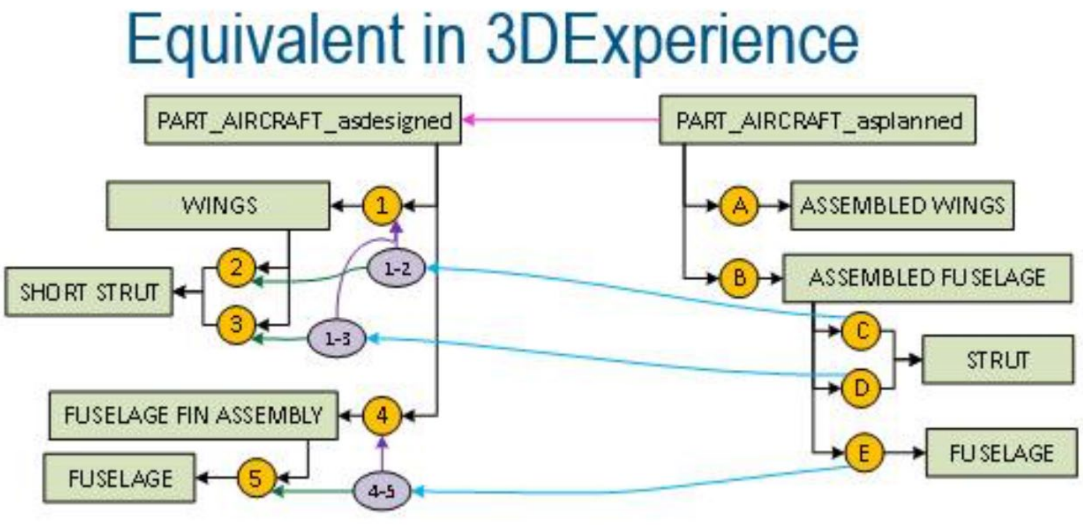
- Here a valid example of PartView Relationship on level 1 and Occurrence Relationship on SpecifiedOccurrence on level 3 which refers to a SingleOccurrence on level 2, saying that the occurrence E the as-planned Fuse within Assy Fuse… is tracked to the occurrence 5 of as-designed Fuse, but only in the context of the occurrence 4 of as-designed Fuse…Assy.
  Here no tracking is defined from as-planned Assy Fuse… => other components of as-planned Assy Fuse… could be tracked anywhere else within the as-designed assembly structure and not only within the as-designed Fuse…Assy.



- Here an invalid example of View Relationship on level 1 and Occurrence Relationships on level 3 without tracking on level 2 => it is not clear which for which occurrence of the as-planned Assy Fuse… the tracking of occurrence E applies.



- To do so, if the AsDesigned depth of the product structure is >1 and there is no tracking relationship defined on each parent level, the tracking relationships on the child levels shall reference SpecifiedOccurrences of the SingleOccurence on the parent level (see chapter 7.2):



- A purchased part in the AsPlanned view (i.e. having ClassifiedAs.Role='make or buy' with ClassifiedAs.Class='bought' shall be a single part. In the AsDesigned view, it may also be an assembly part.

Postprocessor Recommendations:

If OccurrenceRelationship is not supported by the target PDM system to map the tracking relationship, it shall be mapped as PartViewRelationship and return a warning (especially if there are many OccurrenceRelationships of the same part that have tracking relationships to different parts (not possible to map using PartViewRelationship).

If tracking relationships of SpecifiedOccurrences are not supported by the target PDM system, they shall be mapped as SingleOccurrences and return a warning (since some complex cases cannot be mapped).

15.4.1 PartViewRelationship usage for multi view

See chapter 7.4.

The `RelationType` attribute should be set with ‘tracking’ value with ClassString type.

The `Relating` attribute represents the consumer in the relationship (example AsPlanned)

The `Related` attribute represents the consumed object in the relationship (example AsDesigned).

15.4.2 OccurrenceRelationship usage for multi view

See chapter 7.6.

The `RelationType` attribute should be set with ‘tracking’ value with ClassString type.

The `Relating` attribute represents the consumer in the relationship (example AsPlanned)

The `Related` attribute represents the consumed object in the relationship (example AsDesigned).

16 CAx-PDM Compatibility Guide

Due to the two different focusses of CAx systems and PDM systems, two mapping flavors have been described in chapter0.

Hence, it is important for CAx systems to be able to read AP242 XML Files from a PDM system and vice versa. This chapter gives some good practices to avoid most common problems.

In addition to the general advises given in chapter 4.2.1 (Entities and Attributes not supported by the Preprocessor), in chapter 4.2.2 (Entities and Attributes not supported by the Postprocessor) and up to chapter 4.5, here are some typical challenges for the exchange between CAx and PDM systems.

16.1 CAx system to read a file coming from a PDM system

Dealing with multiple PartViews within a PartVersion

Since according chapter 5.1.2, the PartView.initialContext.applicationDomain shall contain a unique value over all PartView, the PartView having the value ‘mechanical design’ in applicationDomain shall be evaluated.The other ones can be ignored (exception: Composites Parts, see chapter 5.1.2).

Dealing with multiple PartVersions within a Part:

The product structure in AP242 is relating a particular version of an assembly with a particular version of its component parts. Therefore, it is sufficient to choose one of the versions of the top node and to traverse top-down the assembly structure (ignoring the other part versions).

If several versions of a component part are referenced in a (sub-)assembly, the version with the highest index that has geometry defined for it shall be chosen.

Dealing with multiple documents:

As stated in the chapter 8.1.1: multiple documents may be provided for each part by the PDM system. The CAx system shall always choose the document having DocumentTypes= ‘primary geometry'. If no primary, but only secondary geometry is provided, the secondary shall be chosen.

If Document.DocumentTypes is set neither to primary nor to secondary geometry, some CAx postprocessors might choose the first DocumentAssignment, or the one that points to a known/support format (either by looking at FileFormat or at the suffix of the filename, but this is quite hazardous.

Of course, non-CAx documents shall never be exchanged as primary nor secondary geometry, and only one document per part shall be marked as primary or secondary geometry (except for alternate models, see below).

Remark: It is not recommended to rely on PartView.DefiningGeometry to find the primary geometry, since it is optional.

Dealing with multiple files:

Model splitting (as defined in chapter 11.2) may be supported as if each splitted model was a disctinct part.

Model sharing (as defined in chapter 11.2) should be recognized so that the same CAx model is not being duplicated for each part it is used in.

Alternate models (as defined in chapter 11.2) may be supported, but like with configuration management data (see below), all the models will be cumulated although only one of them is built at a time.

Dealing with QuantifiedOccurrence:

QuantifiedOccurrence with a quantity value 1 and unit 'each' can be handled like a SingleOccurrence.

QuantifiedOccurrence with an integer quantity value other than 1 and unit 'each' shall be mapped using as many SingleOccurrence as the quantity says. In such cases, maybe no 3D placement is provided (NextAssemblyOccurrenceUsage.Placement has no value) => use Identity matrix. In all cases, the real 3D placement is not provided by the PDM system…

QuantifiedOccurrence with a unit other than 'each' or a quantity value other than an integer shall be ignored.

Dealing with configuration management data:

If the product structure has been provided with configuration management (effectivities), they will be ignored by the CAx system, but it means a 150% product structure will be loaded in the CAx system => many alternative components will be built at the same time, which is actually OK for this scenario.

Dealing with PDM properties and with CAD user defined attributes:

Those PropertyValueAssignments recognized using PropertyValue.Name and/or PropertyDefinition.PropertyType (especially 'user defined attribute') and supported by the CAx system will get imported. The other ones can be ignored.

Dealing with organizational data:

Id, version, name, description, approval status, creator/last modifier, creation/last modification date, etc… may be imported as user defined attributes (if supported) or ignored (except Part.Id).

If the CAx system cannot store multiple identifiers, refer to the Postprocessor Recommendation in 4.6.6.

If the CAx system does not support multiple languages for attributes of type Description, refer to the Postprocessor Recommendation in 4.6.7.

Dealing with incremental exchange (as defined in chapter 9.4):

This shall be explicitly avoided during the PDM export, since the CAx postprocessor will probably have no mechanisms to merge this data with previously sent data…

16.2 PDM system to read a file coming from a CAx system

Dealing with kinematics or composites data:

If such information is not supported by the PDM system, it shall be ignored or mapped together with the geometry of the CAx files into the target format of the CAx files.

Dealing with '/NULL':

As stated in chapter 4.2.1, PartVersion.Id='/NULL' means that the CAx system does not support versioning information. In this case, the PDM import shall decide to create a new version of the part or to reuse the highest existing version…

17 Outlook

As written in the introductory sections of this document, the Recommended Practices for AP242 Domain Model XML Product and Assembly Structure have been written to guide implementations of the new Domain 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 Domain 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 Domain 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 Domain 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.

17.1 “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 vocabulaty
- Definition of a building block mechanism called “Core Technical Capabilities”, describing a recommendation of use on a subset of the STEP AP242 Domain 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 18). The next stage will be the creation of DEXs for automatic verification and validation.

18 List of Templates

This chapter gives a summary of the templates described in this document:

Name	Paragraph
Template “ExchangeContext”	4.6.1
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 “ValueRange”	4.6.11
Template “ValueWithTolerances”	4.6.12
Template “ValueList”	4.6.13
Template “ValueSet”	4.6.15
Template “DateTime”	4.6.11
Template “Approval”	4.6.17
Template “Person”	4.6.18
Template “PersonInOrganization”	4.6.19
Template “DateAndPersonOrganization”	4.6.20
Template “Part”	5.1
Template “Assembly”	5.1
Template “GeometricModel”	6.1
Template “PropertyAssignment”	6.2
Template “SingleOccurrence”	7.1
Template “SpecifiedOccurrence”	7.2
Template “Simplified Positioning Representation”	7.3.1
Template “PartViewRelationship”	7.4
Template “AlternatePartRelationship”	7.5.1
Template “AssemblyOccurrenceRelationshipSubstitution	7.5.2
Template “AssemblyUsageRelationshipSubstitution	7.5.3

Name	Paragraph
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 Domain Model EXPRESS Schema

This annex is derived from Annex B of **ISO/TS 10303-4442:2022** ‘**Domain model: Managed model based 3D engineering domain model**’, which is part of ISO 10303-242 edition 3.

Especially examples have been added for the purpose of these recommended practices. To separate such additions clearly from the text of the standard, the added text in these Recommended Practices is formatted in italics.

A.1 General concepts

This section describes the general concepts for the derivation of the XML Schema from the corresponding Domain Model in SysML. These concepts where used to create the XML Schema Definition from the SysML model using dedicated transformation scripts *and the configuration directives of ISO 10303-15 as well as to derive the EXPRESS Schema using the configuration directives of ISO 10303-16*.

A.1.1 Naming conventions

In general, the XML name derived from a SysML model identifier is the SysML model identifier modified with following rules:

- Names of XML 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 Domain Model names.

The structure of the SysML model is preserved in XML:

- *No changes in cardinality*
- *One SysML instance is represented by one XML instance*
  - *No aggregation of several SysML objects into one XML element*
  - *No splitting of one SysML object into several XML elements*
- *Reverting the direction of associations is allowed:*
  - *Does not violate the principle of structure preservation*

A.1.2 Mapping of SysML blocks data types

For each SysML block data type declaration (equivalent to an EXPRESS entity) the XML Schema contain the definition of a new complex type corresponding to that SysML block 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 SysML 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 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>
        …
```

In addition to the declaration of simple data types SysML allows the specification of objects as subtypes of other objects. 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 object declared by using inheritance relationships with supertypes is said to be a complex object.

```
<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 object inherits not only the SysML attributes and rules appearing in the SysML declarations of all of its supertypes, but also all the SysML attributes and rules they inherit. So subtype objects are specialisations of any of their supertypes, where a specialisation means a more constrained form of the original declaration. The mapping of complex objects use the technique of derivation by extension for those objects that do not inherit from multiple supertypes. When a complex object is derived by extension, its effective content model is the content model of the base object plus the content model specified in the object derivation.

SysML allows the declaration of entities that are not intended to be directly instantiated. For each SysML object declared to be ABSTRACT, the Schema shall contain an *XML* element declaration corresponding to the SysML object. 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 SysML object and XML Schema ComplexType:

- SysML blocks that have an equivalent that is 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 SysML object 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

In EXPRESS, a SELECT data type has a select list where each item shall be an entity data type or a defined data type. In SysML, it corresponds to an abstract block, each select being a subtype of the abstract select block. *SELECT data types that are used in SysML are mostly created as cmn:Reference 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 always needed to create SELECT type definitions. In order to make resolving of 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="CartesianTrans-
formation"/>
    <xsd:element name="GeometricRepresentationRelationship"
type="cmn:Reference"/>
  </xsd:choice>
</xsd:group>
```

A.1.5 Mapping of Block properties

For each SysML explicit property of a SysML block data type declaration the corresponding ComplexType in the XML Schema Definition shall contain an element definition. For the following explicit properties, id, idRoleRef, idContextRef, considered as simple semantics, XML attributes shall be used. *In case of SysML attributes that have simple semantics (e.g., name, description, role, relationType, versionId etc.), XML elements shall be used.*

There are four main cases of SysML attribute mappings:

- *Single attribute*
  - *by containment*
  - *by reference*
- *Aggregation attribute*
  - *by containment*
  - *by reference.*

The order of elements is fixed - this shall be implemented by XML Schema sequence grouping.

The order of the elements in the XSD is defined as following:

- *alphabetic, with some rare exceptions*
  a. *first the local attributes (built-in XML types, references and of type Id and Description) in alphabetical order*
  b. *then the local embedded objects, also in alphabetic order. Examples for local embedded objects are Occurrence and MaterialIdentification in PartView, ShapeElement in Occurrence, ElementDelivery in Activity*
  c. *then the assignments and relationships (also mapped as embedded objects), also in alphabetical order*
- *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

Assignments and Relationship objects from A in alphabetical order

Inherited attributes from B in alphabetical order

Embedded objects from B in alphabetical order

Assignments and Relationship objects from B in alphabetical order

Local attributes of C in alphabetical order

Local embedded objects of C in alphabetical order

Assignments and Relationship objects from C in alphabetical order

The name of the XML element shall be the name of the SysML element written in upper camel casing style

SysML block properties that are mapped to XML attributes shall be written in lower camel case.

If the EXPRESS attribute is declared to be OPTIONAL, then the minOccurs pattern of the XML element shall be declared to be "0".

```
<xsd:complexType name="Activity">
  <xsd:complexContent>
    <xsd:extension base="cmn:BaseRootObject">
      <xsd:sequence>
        <xsd:element name="Requestor" type="DateAndPersonOrganization"
minOccurs="0"/>
```

The type of the XML element shall be declared according to the data type of the SysML property.

For each inverse property of a SysML block data type declaration, the associated complexType shall contain an XML element declaration corresponding to the SysML block property. The name of the XML element shall be the name of the data type of the SysML inverse attribute. The type of the XML element shall be declared according to the data type of the block inverse property.

The re-declaration of block properties shall have no effect on XML schema declaration.

A.1.5.1 SysML property types corresponding to XML complex type

The XML element corresponding to a SysML property whose data type is a SysML block data type shall be mapped in one of following ways:

- for a SysML PartProperty, the XSD element type shall be the name of a complexType defined in XML Schema. When instantiated in XML the data representing the element will be represented by containment inside its parent element.
- For a SysML ReferenceProperty, the Xsd element type shall be defined as type="uidRef". When instantiated the uidRef shall reference the uid attribute of an XML complexType corresponding to the SysML block data type of the property.

The XML element corresponding to an EXPRESS attribute whose data type is a SELECT data type shall reference the XML group defined for the SELECT.

```
<xsd:complexType name="AssemblyOccurrenceRelationship">
  <xsd:complexContent>
    <xsd:extension base="ViewOccurrenceRelationship">
      <xsd:sequence>
        <xsd:element name="Placement" minOccurs="0">
          <xsd:group ref=" TransformationAndAssociationSelect" max-
Occurs="unbounded"/>
        </xsd:element>
        …
```

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 SysML property types corresponding to XML simple type

The XML element corresponding to SysML property 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 SysML type declaration, if not otherwise specified in a configuration directive:

EXPRESS attribute type	ISO XML element type
NUMBER, REAL	xs:double
Array of NUMBER or REAL	xs:string
INTEGER	xs:integer
STRING	xs:string
BOOLEAN	xs:boolean

As suggested in ISO 10303-28, 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:enumera-tion 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 to IEEE 754-1985.

A.1.5.3 Attributes with aggregate data types

Domain model in STEP should support four kinds of aggregation data types:

- ARRAY,
- LIST,
- BAG,
- and SET.

The XML element corresponding to an aggregate valued SysML property shall be declared as follow:

- optional attributes are represented by cardinality constraint minOccurs=“0”;
- aggregations (by containment) are represented as a sequence of elements with the SysML property type as name and type;
- aggregations (by reference) are represented as a sequence of elements with the SysML property type as name and "cmn:Reference” as type;
- the cardinality ":?]" is represented by maxOccurs="unbounded".

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"
      maxOccurs="unbounded"/>
                  </xsd:sequence>
                  …
```

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 make use of maxOccurs= "unbounded".

Note: maxOccurs= "unbounded" most match the EXPRESS Type BAG, since redundant values are allowed. The uniqueness of the elements (like in a SET), the indexing/ordering of the elements (like in a LIST or an ARRAY) and OPTIONAL ARRAY values are not supported.

Multi-Dimensional aggregate PropertyValues are supported, but some multi-dimensional aggregate attributes are mapped as one dimensional

Example: MomentsOfInertia.InertiaValue of kind ARRAY[1:3] OF ARRAY[1:3] OF NumericalValue; is mapped to:

```
      <xsd:element name="NumericalValue" type="NumericalValue" max-
      Occurs="9"/>
```

The ordering is in the order of sequence as the aggregates are defined, separated by blanks.

An exception is made for CartesianTransformation.RotationMatrix. This entity is 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 and the ordering of the elements is as follows: 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

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.*

The description between the lines is added for this recommended practices document.

The following criteria apply for the reference type of mapping:

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 shall 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"/>
   …
```

   Instantiated as:

```
<ActivityAssignment uid="ID_747">
  <AssignedActivity uidRef="ID_400"/>
```

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>
```

Or in the case of an aggregate:

```
<xsd:element name="SameAs" minOccurs="0">
  <xsd:complexType>
    <xsd:group ref="ProxySelect" minOccurs="0" maxOccurs="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 on another object and cannot be re-used 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);*
- *A contained element cannot be defined as BaseRootObject.*

The description between the lines is added for this recommended practices document.

The following criteria apply for the containment type of mapping:

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 shall 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>
```

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 the EXPRESS representation of the Buisness Object Model, the directions have been inverted:

- *The original attribute is omitted.*
- *A new attribute is added to the originally reference entity.*

```
ENTITY Approval
   …
   ApprovalScope : OPTIONAL SET[1:?] OF ApprovalScopeSelect;
END_ENTITY;
TYPE ApprovalScopeSelect = SELECT(
   Activity,
   …
END_TYPE;

  <xsd:complexType name="Activity" abstract="true">
    <xsd:complexContent>
      <xsd:extension base="cmn:BaseRootObject">
        <xsd:sequence>
          …
          <xsd:element name="ApprovalAssignment" type="ApprovalAssignment"
minOccurs="0" maxOccurs="unbounded"/>
          …
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
```

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, that is, 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 instances, 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="ActivityRela-
tionship" minOccurs="0" maxOccurs="unbounded"/>
              …
      </xsd:complexType>
        <xsd:complexType name="ActivityRelationship">
          <xsd:complexContent>
            <xsd:extension base="cmn:BaseObject">
              <xsd:sequence>
                <xsd:element name="Related" type="cmn:Reference"/>
                …
```

A.1.9 Representation of Attributes of type IdentifierSelect

For Id and VersionId attributes, IdentifierSelect is mapped to the XML type 'Id'. Here two representations are supported:

- *XML attribute 'id' as compact representation of the most used variant "simple string";*
- *Identifier representation for single identifier with associated context and/or roleMultiple identifiers are mapped as many instances of Identifier*

```
<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 Identifier is set, Id.id shall not be set.*

Examples:

```
<Class uid="rl--ii">
  <Id id="identification information"/>
</Class>
<Organization uid="o-1">
  <Id id="mercedes-benz.com"/>
</Organization>
<Part uid="p-1">
  <Id>
    <Identifier uid="pid-1" id="as1" idRoleRef="rl--ii"  idContex-
tRef="o-1"/>
  </Id>
  …
</Part>
<Part uid="p-1">
  <Id>
    <Identifier uid="pid-1" id="as1" idRoleRef="rl--ii"  idContex-
tRef="o-1"/>
    <Identifier uid="pid-2" id="assy1" idRoleRef="rl--ii"  idContex-
tRef="o-2"/>
  </Id>
  …
</Part>
```

The other attributes of kind IdentifierSelect are mapped to the XML type 'IdentifierSelect':

```
<xsd:group name="IdentifierSelect">
  <xsd:choice>
    <xsd:element name="IdentifierSet">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="Identifier" type="Identifier" minOccurs="2"
maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:group ref="SingleIdentifierSelect" minOccurs="0"/>
  </xsd:choice>
</xsd:group>
<xsd:group name="SingleIdentifierSelect">
```

```
    <xsd:choice>
      <xsd:element name="Identifier" type="Identifier"/>
      <xsd:element name="IdentifierString" type="xsd:string"/>
    </xsd:choice>
  </xsd:group>
```

Here three representations are supported:

- *IdentifierString representation of the most used variant “simple string“*
- *Identifier representation for single identifier with associated context and/or role*
- *IdentifierSet as a set of Identifiers for multiple identifiers*

Examples:

```
<StartNumber>
  <IdentifierString>3</IdentifierString>
</StartNumber>

<StartNumber>
        <Identifier uid="dfid--0000028DA4863110--46" id="3" idRol-
eRef="ct--id--in" idContextRef="org--0xbc66cf8"/>
</StartNumber>

<StartNumber>
  <IdentifierSet>
    <Identifier uid="dfid--0000028DA4863110--46" id="3" idRoleRef="ct--
id--in" idContextRef="org--0xbc66cf8"/>
    <Identifier uid="dfid--0000028DA4863110--47" id="3" idRoleRef="ct--
id--in" idContextRef="org--andere"/>
  </IdentifierSet>
</StartNumber>
```

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 uid="am-3-consequence-0">no consequence</Local-
izedString>
    <LocalizedString uid="am-3-consequence-1" lang="en-GB">no conse-
quence</LocalizedString>
    <LocalizedString uid="am-3-consequence-2" lang="de-DE">keine Konse-
quenz</LocalizedString>
    <LocalizedString uid="am-3-consequence-3" lang="fr-FR">aucune
conséquence</LocalizedString>
  </Consequence>
</ActivityMethod>
```

The business object definition 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

To be 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-2/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/cnf.
xsd urn:iso:std:iso:10303:-28:ed-2:tech:XMLschema:common
../../implementation_resources/iso10303_28_base_xml_schema/exp.xsd ">
</iso_10303_28>
```

Annex B Known Issues

This section lists known issues with the AP242 Domain Model, both related to the assembly structure and to other domains within the Domain Model. These issues concern 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. Many of these have been resolved by the Technical Corrigendum 1 (TC1) of AP242, which was published in 2016, by the Ed.2 (2019) and by the Ed.3 (2022).

Since 2022, the Bugzilla list hosted by PDES Inc. has been transferred to ISO JIRA with new issue numbers. The old Bugzilla issue numbers are still mentioned in the short description.

ISO Jira Issue	Summary	Target Mile-stone	Status January 2024
BS10303-3224	DataEnvironment.name too complex (Bugzilla #4907)	CR12	Confirmed
BS10303-3228	Restrict population in BO for DataEnvironment.characterization (Bugzilla #4911)	CR12	Confirmed
BS10303-3241	CartesianTransformation.scale can't be mapped (Bugzilla #4925)	CR12	Confirmed
BS10303-3245	invalid mapping specification syntax in BO mapping templates. (Bugzilla #4929)	CR12	Confirmed
BS10303-3248	Missing concepts (Bugzilla #4932)	CR12	Confirmed

ISO Jira Issue	Summary	Target Mile-stone	Status January 2024
BS10303-3249	RequirementAssignment.assignedTo mapping: BOM/ARM select type alignment (Bugzilla #4933)	CR12	Confirmed
BS10303-3250	Mapping not possible for: RequirementRelationship, BreakdownElementRelationship, BreakdownRela-tionship and IndividualPartRelationship (Bugzilla #4934)	CR12	Confirmed
BS10303-3251	RequirementSatisfactionAssertion.assertedBy map-ping not possible (Bugzilla #4935)	CR12	Confirmed
BS10303-3252	RequirementSatisfactionAssertion.satisfiedBy map-ping: BOM/ARM select type alignment (Bugzilla #4936)	CR12	Confirmed
BS10303-3253	RequirementSource.source mapping: BOM/ARM select type alignment (Bugzilla #4937)	CR12	Confirmed
BS10303-3255	InformationRight.owners mapping not possible (Bugzilla #4939)	CR12	Confirmed
BS10303-3257	AlternativeSolutionRelationship mapping not possi-ble (Bugzilla #4941)	CR12	Confirmed
BS10303-3259	How to map ComponentPlacement & Occurrence-Placement (Bugzilla #4943)	CR12	Confirmed
BS10303-3260	FinalSolution mapping not possible (Bugzilla #4944)	CR12	Confirmed
BS10303-3261	How to map FunctionalBreakdownEle-mentViewAssociation (Bugzilla #4945)	CR12	Confirmed
BS10303-3262	InZone mapping not possible (Bugzilla #4946)	CR12	Confirmed
BS10303-3265	Incomplete map--AddElement to Del-taChangeManagementObjectSelect (as cur-rentDesignObject) (Bugzilla #4949)	CR12	Confirmed
BS10303-3266	Incomplete map--AddElement to DeltaChange.de-scribedChange incomplete in ARM (Bugzilla #4950)	CR12	Confirmed
BS10303-3272	ConditionalEffectivity missing in SUPERTYPE clause of Effectivity (Bugzilla #4956)	SMRLv6	Closed
BS10303-3362	offset is required for some types of conver-sion_based_unit (Bugzilla #5069)	none	Reopened
BS10303-3369	Definition and Support of UNIQUE Rules (Bugzilla #5079)	none	Confirmed
BS10303-3370	Consider WHERE RULES / OCL RULES con-straints to XSD using schematron (normative ?) (Bugzilla #5080)	242-3-CD	Under Dis-cussion
BS10303-3400	No Mapping specification provided for capability AMW1 (Bugzilla #5123)	CR12	Confirmed

ISO Jira Issue	Summary	Target Mile-stone	Status January 2024
BS10303-3401	No Mapping specification provided for capability C1 (Bugzilla #5124)	CR12	Confirmed
BS10303-3402	No Mapping specification provided for capability SBC4 (Bugzilla #5125)	CR12	Confirmed
BS10303-3403	No Mapping specification provided for capability PP1 (Bugzilla #5126)	CR12	Confirmed
BS10303-3404	No Mapping specification provided for capability GMI1 (Bugzilla #5127)	CR12	Confirmed
BS10303-3405	No Mapping specification provided for capability GMI2 (Bugzilla #5128)	CR12	Confirmed
BS10303-3406	No Mapping specification provided for capability GMI5 (Bugzilla #5129)	CR12	Confirmed
BS10303-3407	No Mapping specification provided for capability GMI8 (Bugzilla #5130)	CR12	Confirmed
BS10303-3408	No Mapping specification provided for capability AMW2 (Bugzilla #5131)	CR12	Confirmed
BS10303-3409	No Mapping specification provided for capability AMW3 (Bugzilla #5132)	CR12	Confirmed
BS10303-3410	No Mapping specification provided for capability DM1 (Bugzilla #5133)	CR12	Confirmed
BS10303-3411	No Mapping specification provided for capability CSS (Bugzilla #5134)	CR12	Confirmed
BS10303-3412	Mapping specification uncomplete for capability SBC2 (Bugzilla #5135)	CR12	Confirmed
BS10303-3413	Mapping specification uncomplete for capability SBC5 (Bugzilla #5136)	CR12	Confirmed
BS10303-3414	Mapping specification uncomplete for capability PPP1 (Bugzilla #5137)	CR12	Confirmed
BS10303-3415	Mapping specification uncomplete for capability C2 (Bugzilla #5138)	CR12	Confirmed
BS10303-3416	Mapping specification uncomplete for capability PP2 (Bugzilla #5139)	CR12	Confirmed
BS10303-3417	Mapping specification uncomplete for capability RR3 (Bugzilla #5140)	CR12	Confirmed
BS10303-3418	Mapping specification uncomplete for capability GMI6 (Bugzilla #5141)	CR12	Confirmed
BS10303-3443	Separation of real Parts, Templates, Parts defined by specification, Designs (Bugzilla #5214)	242-3-CD	Open
BS10303-3457	Date-Time as a string (Bugzilla #5267)	none	Open

ISO Jira Issue	Summary	Target Mile-stone	Status January 2024
BS10303-3458	Configuration_effectivity for a 150% product structure (Bugzilla #5270)	242-3-CD	Open
BS10303-3466	Mapping of DateTimeString to "xsd:dateTime" (Bugzilla #5404)	none	Open
BS10303-3578	Product_view_definition: Why the mapping cross-wise exchange name and description? (Bugzilla #5532)	none	Open
BS10303-3818	Update the ARM for multi identification, naming, descriptions (Bugzilla #5885)	CR_ILS	Confirmed
TCSC410303-909	Reintroduce Make or Buy concept	Ed4	On Hold
TCSC410303-1397	Clarify added-value of GeneralShapeDependentProperty.PropertyType (Bugzilla #6021)	10303-4442 Ed4	In Progress
TCSC410303-1398	Combined use of PartViewRelationship and ViewOccurrenceRelationship (Bugzilla #6022)	10303-4442 Ed4	Done
TCSC410303-1399	Dedicated XML elements for the transfer of values without unit (Bugzilla #6290)	10303-4442 Ed4	In Progress
TCSC410303-1400	Definition of default/allowed values (Bugzilla #6844)	10303-4442 Ed4	In Progress
TCSC410303-1401	Add real length/precision, integer length, optional/mandatory, unique, indexed, string length, string pattern to PropertyDefinition (Bugzilla #6856)	10303-4442 Ed4	In Progress
TCSC410303-1308	Add AlternatePartVersionRelationship (Bugzilla #8058)	10303-4442 Ed4	In Progress
TCSC410303-1307	AxisPlacement: attributes Axis and RefDirection (Bugzilla #8182)	10303-4442 Ed4	Done
TCSC410303-1306	Add missing Relationship to entities having attribute VersionId (Bugzilla #8518)	10303-4442 Ed4	Done
TCSC410303-659	Update definition of GeometricalRelationship.Placement	10303-4442 Ed4	In Progress
TCSC410303-660	Add PropertyValueAssignment to ExchangeContext	10303-4442 Ed4	Done
TCSC410303-661	Enhance AssignmentObjectRelationship to Associations and Relationships (redundant mit #6022?)	10303-4442 Ed4	In Progress
TCSC410303-785	ExternalGeometricModel must allow to reference an external DetailedGeometricModelItem	10303-4442 Ed4	Done
TCSC410303-1472	Identifier Unique and Where rules to be harmonised between 4442 and 4000	10303-4442 Ed4	Todo
TCSC410303-1509	Introduce DeltaChangeRelationship	10303-4442 Ed4	In Progress

ISO Jira Issue	Summary	Target Mile-stone	Status January 2024
TCSC410303-1510	Introduce DeltaChange.ControlledBy	10303-4442 Ed4	In Review
TCSC410303-1511	Enhance DeltaChangeRelationshipSelect	10303-4442 Ed4	In Progress

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 Ed.3 Domain Model XML / EXPRESS Schema
    - Dated 11 May 2021 (version N10590)
- Recommended Practices for STEP AP242 Edition 3 Domain Model XML Configuration Management [242-CFG]
- Recommended Practices for STEP AP242 Edition 4 Domain Model XML Product Manufacturing Information (PMI) [AP242-PMI]
- Recommended Practices for Persistent IDs for Design Iteration and Downstream Exchange [PID]
- 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 (https://www.mbx-if.de/cax/cax_recommPractice.php)
    - Geometric and Assembly Validation Properties: Release 4.6; Apr. 21, 2023
    - External References: Release 3.1; Jan. 20, 2014
    - User Defined Attributes: Release 1.8; Feb. 18, 2021
    - STEP File Compression: Release 1.3; Apr. 12, 2023
    - PDM Schema Usage Guide: Release 4.3; Jan., 2002
    - Alternative Shapes: Release 0.2; Nov. 23, 2021
- 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).
- JT Application Benchmark 2013-2018 Experiences
- MIL-STD-31000A (http://model-based-enterprise.org/mil-standards.html)
- JT Content Harmonization Guideline
    - Version 5.0, 06 November 2019
- LOTAR Part 200-1: Technical Specification „Product Structure Validation“ (TS-9300-200-1) (https://lotar-international.org/wp-content/uploads/2020/05/TS-9300-200-1_R2.2.pdf)
- ISO 13584-42: Industrial automation systems and integration (Part library) Part 42: Description methodology: Methodology for structuring part families

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° rotation around the global X axis and the matrix B gives a 45° 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[2]

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 recommende practices we take the assumption that the partners have agree beforehand wich units to use in a project.
Note: The definitions given in this Annex are valid for AP242 edition 2 Domain Model schema.

E.1 SI Base Unit Definitions

The following is the recommendation for exchange of SI base unit definitions[3]:

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>
```

[2] http://www.nist.gov/pml/wmd/metric/si-units.cfm

[3] https://en.wikipedia.org/wiki/SI_base_unit

[4] This instance is created to support definition of SI derived units and is the formal definition that the kilogram is the SI unit of mass.

```
  <Quantity><ClassString>amount of substance</ClassString></Quantity>
</Unit>
```

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. [5]

The AP242 edition 2 Domain 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 (in parenthesis: Unit.Quantity):

- hertz ('frequency')
- radian ('plane angle')
- degree ('plane angle')
- steradian ('solid angle')
- newton ('force')
- pascal ('pressure' or 'stress')
- joule ('energy')
- watt ('power' or 'radiant flux')
- coulomb ('electric charge')
- volt ('electric potential')
- farad ('capacitance')
- ohm ('resistance')
- siemens ('conductance')
- weber ('magnetic flux')
- tesla ('magnetic flux density')
- henry ('inductance')
- degree Celsius ('thermodynamic temperature')
- lumen ('luminous flux')
- lux ('illuminance')
- becquerel ('radioactivity')
- gray ('absorbed dose')

[5] https://en.wikipedia.org/wiki/SI_derived_unit

- sievert (‘equivalent dose‘)
- katal (‘catalytic activity‘)

Example for kW:

```
<Unit uid="u--100000007">
  <Kind><ClassString>SI derived unit</ClassString></Kind>
  <Name><ClassString>watt</ClassString></Name>
  <Prefix><ClassString>kilo</ClassString></Prefix>
  <Quantity><ClassString>power</ClassString></Quantity>
</Unit>
```

E.2.2 Derived quantities and units

To exchange these coverted units, use the `Unit` element with the `Unit.kind` attribute set to ‘unspecified SI derived unit’.

Examples of SI derived unit (in parenthesis: Unit.Quantity):

- square metre (‘area‘)
- cubic metre (‘volume‘)
- litre (‘volume’)
- metre per second (‘velocity‘)
- cubic metre per second (‘volumetric flow‘)
- metre per second squared (‘acceleration‘)
- metre per second cubed (‘jerk’)
- metre per quartic second (‘snap’)
- radian per second (‘angular velocity‘)
- newton second (‘momentum‘)
- newton metre second (‘angular momentum’)
- newton metre (‘moment of force‘ or ‘torque‘)
- newton per second (‘yank‘)
- reciprocal metre (‘attenuation coefficient‘, ‘curvature‘, ‘propagation coefficient‘ or ‘wave number‘)
- kilogram per square metre (‘area density‘)
- kilogram per cubic metre (‘density’)
- cubic metre per kilogram (‘specific volume‘)
- mole per cubic metre (‘amount of substance concentration’)
- cubic metre per mole (‘molar volume‘)
- joule second (‘action’)
- joule per kelvin (‘entropy‘ or ‘heat capacity‘)
- joule per kelvin mole (‘molar heat capacity‘)
- joule per kilogram kelvin (‘specific heat capacity‘)
- joule per mole (‘molar energy‘)
- joule per kilogram (‘specific energy‘)
- joule per cubic metre (‘energy density‘)
- newton per metre (‘stiffness‘ or ‚surface tension‘)
- watt per square metre (‘heat flux density’ or ‘irradiance’)
- watt per metre kelvin (‘thermal conductivity‘)
- square metre per second (‘kinematic viscosity‘ or ‘thermal diffusivity‘)
- pascal second (‘dynamic viscosity‘)
- coulomb per square metre (‘polarization density‘)
- coulomb per cubic metre (‘electric charge density’)

- ampere per square metre (‘electric current density‘)
- siemens per metre (‘electrical conductivity’)
- siemens square metre per mole (‘molar conductivity‘)
- farad per metre (‘permittivity‘)
- henry per metre (‘magnetic permeability‘)
- volt per metre (‘electric field strength‘)
- ampere per metre (‘magnetic field strength‘ or ‘magnetization‘)
- candela per square metre (‘luminance‘)
- lumen second (‘luminous energy‘)
- lux second (‘luminous exposure‘)
- coulomb per kilogram (‘exposure‘)
- gray per second (‘absorbed dose rate’)
- ohm metre (‘resistivity‘)

Example for 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	10^3	2.40%	Kibibyte	KiB	2^{10}
Megabyte	MB	10^6	4.86%	Mebibyte	MiB	2^{20}
Gigabyte	GB	10^9	7.37%	Gibibyte	GiB	2^{30}
Terabyte	TB	10^{12}	9.95%	Tebibyte	TiB	2^{40}
Petabyte	PB	10^{15}	12.6%	Pebibyte	PiB	2^{50}
Exabyte	EB	10^{18}	15.3%	Exbibyte	EiB	2^{60}
Zettabyte	ZB	10^{21}	18.1%	Zebibyte	ZiB	2^{70}
Yottabyte	YB	10^{24}	20.9%	Yobibyte	YiB	2^{80}

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.3.3 Percent

To represent a rate/number/amount in each hundred.

```
<Unit uid="u--100000010">
  <Kind><ClassString>unspecified</ClassString></Kind>
  <Name><ClassString>percent</ClassString></Name>
  <Quantity><ClassString>amount in each hundred</ClassString></Quantity>
</Unit>
```

E.3.4 Ratio

The relationship between two groups or amounts that expresses how much bigger one is than the other. To make ratio values better homogeneous within data exchange, it is recommended to represent a rate/number/amount in each 1.

```
<Unit uid="u--100000010">
  <Kind><ClassString>unspecified</ClassString></Kind>
  <Name><ClassString>ratio</ClassString></Name>
  <Quantity><ClassString>amount in each 1</ClassString></Quantity>
</Unit>
```

E.4 Imperial Units

The following is the recommendation for exchange of Imperial unit definitions[6], the `Unit` element shall be used with the `Unit.kind` attributes set to ‘imperial:

Length (Unit.Quantity = ‘length’):

- thou
- inch
- foot
- yard
- chain
- furlong
- mile
- league
- fathom

[6] https://en.wikipedia.org/wiki/Imperial_units

- cable
- nautical mile
- link
- rod
- chain

Area (Unit.Quantity = ‘area’):

- perch
- rood
- acre

Volume (Unit.Quantity = ‘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>
```