Recommended Practices
For
STEP AP242 Business Object Model XML
Configuration Management

Release 0.9
- Release Candidate –

July 1, 2019

Contacts:

<table>
<thead>
<tr>
<th>Organizational</th>
<th>Technical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jochen Boy</td>
<td>Guillaume Hirel</td>
</tr>
<tr>
<td>PROSTEP AG</td>
<td>T-Systems</td>
</tr>
<tr>
<td>Dolivostraße 11</td>
<td><a href="mailto:guillaume.hirel@t-systems.com">guillaume.hirel@t-systems.com</a></td>
</tr>
<tr>
<td>64293 Darmstadt / Germany</td>
<td><a href="mailto:jochen@t-systems.com">jochen@t-systems.com</a></td>
</tr>
<tr>
<td><a href="mailto:jochen.boy@prostep.com">jochen.boy@prostep.com</a></td>
<td><a href="mailto:guillaume.hirel@t-systems.com">guillaume.hirel@t-systems.com</a></td>
</tr>
<tr>
<td>Frédéric Darré</td>
<td>Jochen Haenisch</td>
</tr>
<tr>
<td>Cimpa S.A.S.</td>
<td>Jotne EPM</td>
</tr>
<tr>
<td>4 Avenue Didier Daurat</td>
<td><a href="mailto:jochen@jotne.com">jochen@jotne.com</a></td>
</tr>
<tr>
<td>31700 Blagnac / France</td>
<td><a href="mailto:jochen@jotne.com">jochen@jotne.com</a></td>
</tr>
<tr>
<td><a href="mailto:frederic.darre.external@airbus.com">frederic.darre.external@airbus.com</a></td>
<td><a href="mailto:frederic.darre.external@airbus.com">frederic.darre.external@airbus.com</a></td>
</tr>
</tbody>
</table>

© PDM Implementor Forum
# Table of Contents

1 Introduction ................................................................. 7
   1.1 Document Overview .................................................. 7
       1.1.1 Goal and Objectives ............................................. 7
       1.1.2 Scope .................................................................. 7
       1.1.3 Intended Audience ................................................. 9
       1.1.4 Intended Use ........................................................ 9
       1.1.5 Document Style .................................................... 9
       1.1.6 Document Structure .............................................. 10
       1.1.7 Instantiation Diagrams .......................................... 10
   1.2 Organizational Framework .......................................... 14
       1.2.1 Vendor Communities ............................................ 14
       1.2.2 User Communities ................................................. 14
   1.3 Maintenance of this Document ................................. 14

2 Reference to Recommended Practices ............................. 15
   2.1 Reference to Core Document [242BO-PAS] .................. 15
   2.2 Listing of Recommended Practices Version in Exchange Files 15

3 Fundamental Concepts and Assumptions .................... 16
   3.1 The Big Picture ........................................................ 16
   3.2 Use Cases ................................................................. 17
       3.2.1 Configuration management with AP242 for supplier exchange 17
       3.2.2 Configuration management with AP242 for archiving ........ 18
       3.2.3 Configuration management with AP242 for internal PLM .... 18
   3.3 Configuration Management ....................................... 19
       3.3.1 Definition .......................................................... 19
       3.3.2 Configuration Identification .................................... 20
       3.3.3 Configuration ....................................................... 20
       3.3.4 Configuration Baseline ........................................... 23
       3.3.5 Configuration Item (CI) ........................................... 24
       3.3.6 Product Concept .................................................... 24
       3.3.7 Bill of Material (BOM) ............................................ 24
       3.3.8 Configuration Management system .......................... 25
       3.3.9 Effectivity .......................................................... 25
       3.3.10 Part Structure Definition .................................... Fehler! Textmarke nicht definiert.
   3.4 Definitions in ISO 10303-3001 .................................... 27
       3.4.1 Configuration ....................................................... 28
       3.4.2 Assembly structure .............................................. 30
       3.4.3 Breakdown ........................................................ 31
       3.4.4 Part occurrence .................................................... 32
       3.4.5 Part view ........................................................... 32
       3.4.6 Effectivity ......................................................... 32
       3.4.7 Product specification ......................................... 33

4 Configuration and Effectivity Information .................... 34
   4.1 Configuration Identification ....................................... 34
       4.1.1 Default Context .................................................. 34
4.1.2 Template “ProductConceptIdentification” .............................................................. 36
4.1.3 Template “ConfigurationItem” ............................................................................... 41
4.1.4 Template “ProductClassRelationship” ................................................................. 47
4.1.5 General Handling of Product Configurations ....................................................... 49
4.1.6 Example Mapping an Item to AP242 Context ......................................................... 49
4.1.7 Template “ProductConceptIdentification” .............................................................. 36
4.2 Configuration Composition Management .............................................................. 49
4.2.1 Common supertype „Effectivity” ............................................................................. 50
4.2.2 Template „DataValidityEffectivity” ........................................................................ 52
4.2.3 Template „LotEffectivity” ......................................................................................... 56
4.2.4 Template „SerialEffectivity” .................................................................................... 57
4.2.5 Template „TimeIntervalEffectivity” ........................................................................ 58
4.2.6 Template „ConditionalEffectivity” ........................................................................ 60
4.2.7 Template „EffectivityAssignment” .......................................................................... 64
4.2.8 General Handling of Effectivities ........................................................................... 71
4.3 Product Specification ............................................................................................ 73
4.3.1 Template „Specification” ......................................................................................... 73
4.3.2 Template „SpecificationAssignment” ...................................................................... 79
4.3.3 Template „SpecificationCategoryAssignment” ....................................................... 81
4.3.4 Template „SpecificationConditionAssignment” .................................................... 83
4.4 Usage of Option Pool (Dictionary) ........................................................................... 95
5 Managing nested file structure .............................................................................. 96
6 Supported Configuration Management for PDM Systems ...................................... 103
7 References ............................................................................................................ 108
Annex A Known Issues ............................................................................................. 109

List of Figures
Figure 1: AP242 Big Picture ............................................................................................ 17
Figure 2: Configuration Management for supplier exchange ....................................... 18
Figure 3: Configuration Management for archiving ..................................................... 18
Figure 4: Configuration Management for internal PLM ................................................. 19
Figure 5: Configuration Management functions ........................................................... 20
Figure 6: Configuration .................................................................................................. 21
Figure 7: Configuration Table Example ......................................................................... 23
Figure 8: Configuration Baselines ................................................................................. 24
Figure 9: Unit Effectivity Example ................................................................................ 26
Figure 10: Date Effectivity Example .............................................................................. 27
Figure 11: Information requirement domains in ISO 10303-3001 .................................. 28
Figure 12: Simplified overview of the AP242 configuration and product specification BO Model ..................................................................................................................................... 28
Figure 13: Configuration in ISO 10303-3001 ................................................................... 29
Figure 14: AssemblyStructure with Cartesian Transformation .................................... 30
Figure 15: Assembly Structure with Geometric Representation Relationship ............... 31
Figure 16: Breakdown Structure ................................................................................... 32
Figure 17: Instance Model ProductConceptIdentification .............................................. 36
Figure 18: Instance Model ProductConceptConfigurationIdentification ........................................ 42
Figure 19: Simplified Example for ProductDesignAssociation for the top node ............................ 47
Figure 20: Instance Model ProductClassRelationship ................................................................. 48
Figure 21: Context Mapping for Supplier Exchange .................................................................... 49
Figure 22: Instance Model Date Effectivity .................................................................................. 52
Figure 23: Instance Model Milestone Effectivity ......................................................................... 55
Figure 24: Instance Model LotEffectivity ..................................................................................... 56
Figure 25: Instance Model SerialEffectivity .................................................................................. 57
Figure 26: Instance Model TimeIntervalEffectivity ................................................................. 59
Figure 27: Instance Model ConditionalEffectivity ......................................................................... 60
Figure 28: Instance Model ConditionalConfiguration ................................................................. 61
Figure 29: Simplified Example for supported EffectivityAssignments ....................................... 64
Figure 30: Instance Model EffectivityAssignment for PartVersions ........................................... 65
Figure 31: Instance Model EffectivityAssignment for assembly links ....................................... 66
Figure 32: Instance Model EffectivityAssignment for configurations ......................................... 67
Figure 33: Specifications to define configurations ....................................................................... 73
Figure 34: Specifications defined as available for configuration given ProductClass .................... 73
Figure 35: Instance Model Specification ....................................................................................... 74
Figure 36: Example for an AND Condition for specifications ..................................................... 84
Figure 37: Instance Model SpecificationConditionAssignment .................................................... 84
Figure 38: External reference to a ProductClass within a nested XML File containing EffectivityAssignments .................................................................................................................. 98

List of Tables

Table 1: Instance Diagram Notation ............................................................................................. 14
Table 2: Configuration use cases ............................................................................................. 21
Table 3: BOM Classification for Configuration Management for internal PLM .......................... 25
Table 4: Context handling recommendations ........................................................................... 35
Table 5: “ProductClass” Attributes .......................................................................................... 39
Table 6: “Market” Attributes .................................................................................................... 40
Table 7: “ProductConfiguration” Attributes ............................................................................ 45
Table 8: “ProductDesignAssociation” Attributes ..................................................................... 46
Table 9: “ProductClassRelationship” Attributes ..................................................................... 47
Table 10: “Effectivity” Attributes ............................................................................................. 51
Table 11: “DataValidityEffectivity” Attributes .......................................................................... 53
Table 12: “Event” Attributes ..................................................................................................... 54
Table 13: “LotEffectivity” Attributes ........................................................................................ 56
Table 14: “SerialEffectivity” Attributes ...................................................................................... 58
Table 15: “TimeIntervalEffectivity” Attributes .......................................................................... 59
Table 16: “ConditionalEffectivity” Attributes ........................................................................... 61
Table 17: “ConditionalConfiguration” Attributes ...................................................................... 62
Table 18: “EffectivityAssignment” Attributes .......................................................................... 69
Table 19: “Specification” Attributes ................................................................. 77
Table 20: “SpecificationCategory” Attributes .................................................. 79
Table 21: “SpecificationAssignment” Attributes .............................................. 80
Table 22: “SpecificationCategoryAssignment” Attributes ................................ 82
Table 23: “SpecificationConditionAssignment” Attributes ............................... 91
Table 24: “Condition” Attributes ...................................................................... 94
Table 25: Supported Configuration Management approaches for PDM Systems .......... 107

Document History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Working History – to be deleted before next release

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Author</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>2016-07-15</td>
<td>L. Graßmann</td>
<td>Initial creation</td>
</tr>
</tbody>
</table>
| 0.2      | 2016-09-26 | L. Graßmann  | • Definition of the scope of the document in (1.1.2) with restriction of this release of the document
• Detailed definition of Explicit Configuration (3.3.3.1) dependent on use case
• Details for effectivities: Serial, Lot, TimeInterval, Conditional, EffectivityAssignment (4.2.1) |
| 0.3      | 2016-11-25 | L. Graßmann  | Update instance diagrams in chapter 4
(Add topics for round 3 discussions)
Add uid to LotSize for LotEffectivity (4.2.1.4)
Rename dateValidityEffectivity to dataValidityEffectivity (chapter 4.2.1.3.)
Recommendation for the default context (4.1.1) Chapter (4.3 Specification) |
| 0.4      | 2017-08-09 | L. Graßmann  | Change EffectivityAssignment from SingleOccurrence to NextAssemblyOccurrenceUsage in chapter 4.2.7 Template “EffectivityAssignment”
Add recommendation for LotEffectivity in chapter 4.2.3 Template “LotEffectivity” to avoid combination with other constraints like start or end date. Ranges of lots are not supported.
Chapter: 4.1.1 Default Context Remove the market from the default context and add some recommendations
New chapter 4.2.6 Template “ConditionalEffectivity”
New chapter 4.2.8 General Handling of Effectivities
New chapter 4.3.3 Template „SpecificationConditionAssignment“
New chapter 4.2.7.3 EffectivityAssignment for Configurations |
<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author(s)</th>
<th>Changes</th>
</tr>
</thead>
</table>
| 0.4.1   | 2017-08-10 | L. Graßmann | New chapter: 4.2.7.1 EffectivityAssignment for PartVersion  
New chapter: 4.4 Usage of Option Pool  
New chapter: 5 Supported Configuration Management for PDM Systems |
| 0.4.2   | 2017-08-10 | L. Graßmann | New chapter 4.1.6 Example Mapping an Item to AP242 Context  
Remove recommendation for ConditionalEffectivity, till the bug in the schema has been fixed  
Add chapter 4.2.6.2 ENTITY Condition |
| 0.4.3   | 2017-10-11 | L. Graßmann | Add 0 Known Issues |
| 0.5     | 2018-05-24 | G. Hirel  | Adjustments in chapter 3.5 and 4 |
| 0.6     | 2018-08-29 | G. Hirel  | 4.2.1 – Started discussion on using a Part as EffectivityContext.  
4.3.3 – Added Pre- and Post-Processor Recommendations  
4.4 – Added clarifications and recommendations |
| 0.7     | 2018-10-09 | G. Hirel  | 4.2.2 – Fixed descriptions for upper/lower bound  
4.2.7 – Fixed XML examples and definitions for effectivity indication |
| 0.8     | 2019-05-29 | J. Haenisch, F. Darré, G. Hirel, J. Boy  | Extension of in scope / out of scope statement  
General update and resolution of comments Section 3  
General update and resolution of comments Section 4  
General update and resolution of comments Section 5  
Editorial review |
| 0.9     | 2019-07-01 | J. Boy  | Editorial cleanup for publication |
1 Introduction

1.1 Document Overview

1.1.1 Goal and Objectives

The goal of this document is to describe the recommended structure and attribute population for particular instance models created from the entities and attributes defined by the STEP AP242 “Managed Model-based 3D Engineering” Business Object (BO) model and populated according to its XML Schema.

The selected instance models illustrate how to encode configuration management information that needs 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 requirements of the Aerospace & Defense and the automotive industries in the realm of configuration management.
- Prevent the emergence of “flavors”, i.e. diverging/conflicting implementations of the AP242 BO Model XML.

1.1.2 Scope

This document describes the recommended practices for the exchange of configuration management data. It is based on the STEP AP242 TC Business Object Model (ISO 10303-3001:2016) and the corresponding XML schemas, which can be found at:

  The BO Model XML Schema
  The XML Schema, which contains the structural definitions common to all STEP Business Object Models.

This document has not the intention to define all aspects of PLM and configuration management. Focus is to support the implementation of AP242 in software systems for some predefined use cases that are introduced in chapter 1.1. Only those configuration management aspects will be highlighted in chapter 3.3 that are relevant to match the domain and use case specific view to the corresponding technical view for the software developer. Because this document is only a recommendation, some document specific definitions of configuration management terms are used, which reflect industry practice. Also, the PLM system landscape and the lifecycle processes applied by industry may have a more detailed view to the handling of product structures for configuration management. This document, however, raises the claim to give a summary of the most common aspects.

The following AP242 configuration management aspects are in scope of this document:

- Explicit configuration management
- Implicit configuration management
- Effectivities of types conditional, unit/serial and date
- Configuration item
- Product concept
• Bill of Material (BOM)
• Software components are also supported as parts of kind ‘software’
• Configuration calculation results and NC programs, however, only as referenced documents.

The following aspects are out of scope for this document:
• Software code management
• Numerical control (NC) code management
• Configuration management system
• Configuration baseline
• Configuration control, that is, control of changes to configuration items and related documentation
• Configuration status accounting, that is, recording and reporting of information needed to manage configuration items effectively, including the status of proposed and approved changes
• Configuration audit and verification, that is, auditing of configuration items to verify conformance to documented requirements
• Configuration management planning, including the use of standards, guidelines, best practices
• Any other aspects of configuration management that are outside of a PDM/PLM system
• The concept of product specification, that is, the capability to describe products with a large number of variants
• Specified occurrences for configurations, due to lack of use case
• Product Structure Definition, in the sense of pre-defined filters that can be used to extract parts of larger configured product structure, e.g. a 100% structure from a 150% one.
• Assembly Structure Definition, that is, product structure and related documents with CAD models, drawings and additional descriptions, as this is already covered by [242BO-PAS]
• Revision rules for configurations, as these are already covered by [242BO-PAS]
• Views for configurations, as these are already covered by [242BO-PAS].

The following aspects may – depending on user demand – be included in future releases of this document:
• Feasibility of configurations, that is, distinction that some combinations of configuration options result in valid product variants, others in invalid ones
• Automatic completion of configurations, that is, indications of the dependance of configuration options on the presence of other options, which then may be added automatically
• Product breakdown, that is, distinction of types of breakdowns and breakdown elements into, for example, physical, functional, zonal and system.
• Other types of BoMs than engineering BOM (eBOM) and manufacturing BOM (mBOM).
1.1.3 Intended Audience
This document is intended to be an implementation guide for developers of PDM and file translation application systems that support and exchange configuration management information 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 that support configuration management definitions. It provides guidelines for the consistent preprocessor instance model creation and requirement value encoding to enable meaningful information exchange between different systems and applications using the STEP AP242 BO model, and guidelines for the consistent interpretation by a postprocessor of the STEP AP242 BO model exchange file.

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

The “template” concept is used in this document. Structures and substructures are defined in one section and then reused in other sections of the documents. These templates are represented by the blue boxes in the diagrams.

Templates for common core constructs, such as product, are defined in the “Recommended Practices for AP242 BO Model XML Product and Assembly Structure”, and reused here. References to such templates are prefixed with [242BO-PAS].

The instantiation diagrams are presented using a graphical notation intended to illustrate the instance model. Detailed notation is described in chapter 1.1.7.

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

<table>
<thead>
<tr>
<th>ENTITY &lt;Object&gt;</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Id</td>
</tr>
<tr>
<td>Description</td>
<td>OPTIONAL DescriptionSelect</td>
</tr>
<tr>
<td>AssociatedItem</td>
<td>PartVersion</td>
</tr>
</tbody>
</table>

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

- **Id**: The id attribute specifies the identifier of the `<Object>` entity.
- **associatedItem**: the PartVersion that identifies a part which is a valid implementation for the `<Object>`

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

```xml
<Object uid="obj---01">
  <Id id="Manufactural Product"/>
  <AssociatedItem uidRef="pv---02"/>
</Object>
```

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:
<table>
<thead>
<tr>
<th>Object1 #1-1</th>
<th>Object (instance of an EXPRESS ENTITY) After the #, an instance number is given</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Att1: mandatory attribute Att2: optional attribute</td>
</tr>
<tr>
<td></td>
<td>Aggregate type for the definition of the cardinality constraint: B: Bag (non-ordered and may contain duplicates) S: Set (non-ordered and may not contain duplicates) L: List (ordered) [x : y]: lower size : upper size ?: unconstrained A: Array (indexed) [x : y]: lower index : upper index</td>
</tr>
<tr>
<td></td>
<td>Additional constraint on the object: the attribute(s) depicted with '*' have to contain unique values. Currently not used in this document</td>
</tr>
<tr>
<td></td>
<td>Derived Information from another object or attribute Currently not used in this document</td>
</tr>
<tr>
<td>STRING</td>
<td>Simple data types</td>
</tr>
<tr>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>BOOLEAN</td>
<td></td>
</tr>
<tr>
<td>Type1</td>
<td>User-defined data type Currently not used in this document</td>
</tr>
<tr>
<td>EnumType1</td>
<td>Enumeration Type (consists of a limited list of possible values defined for this type)</td>
</tr>
</tbody>
</table>
### 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

### Inheritance from a Supertype to its Subtypes

An Abstract Supertype (ABS) cannot be instantiated 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 reused 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’).

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 reused through adding new XML containments to it.
1.2 Organizational Framework

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

1.2.1 Vendor Communities

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

- Coordinate the creation of the document
- Verify the approach of the recommended practices in PDM-IF Test Rounds, and publish result summaries of testing “AP242 BO Model Configuration Management”
- Ensure the consistency with other “AP242 BO Model XML Recommended Practices”

1.2.2 User Communities

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

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

1.3 Maintenance of this Document

This document describes the recommended practices to implement configuration management data using the AP242 BO Model. It is based on the joint CAx-IF / JT-IF “Recommended Practices for AP242 BO Model XML Product and Assembly Structure”. Since configuration management is an extension to the core scope that is specific to the PDM domain, it is documented in a separate document with references to the core document where needed.

It is the responsibility of the PDM-IF not only to maintain this document, but also to ensure its consistency with the core document.

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

2.1 Reference to Core Document [242BO-PAS]
This document is an extension of the joint CAx-IF / JT-IF Recommended Practices for AP242 BO Model Product and Assembly Structure. Definitions of templates and XML elements contained in the core document are not repeated here.

Where necessary, references to sections of the core document are given in this format:

[242BO-PAS] 4.6.7

This means a reference to the Recommended Practices for AP242 BO Model Product and Assembly Structure, section 4.6.7.

This Recommended Practices for AP242 BO Model XML Configuration Management are built on the following version of the core document:

Version 2.0; dated 2018-10-30

This document is publicly available from the CAx-IF and prostep ivip web pages.

2.2 Listing of Recommended Practices Version in Exchange Files
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 second string element of the Documentation attribute of the Header element in the XML file (for details see section [242BO-PAS] 4.1.5), while the first value shall be the value defined in section 3 of [242BO-PAS]. The value follows a specific pattern well established in Part 21 files:

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

The string corresponding to this version of this document is:

<Documentation>CAx-IF Rec.Pracs.---AP242 BO Model XML Configuration Management---0.9---2019-07-01</Documentation>

General Postprocessor Recommendation:
If a postprocessor encounters attribute values, or object instantiations different from the ones recommended in this version of the document, but is needed for the use case, an additional exchange agreement is supposed to be in place among the parties involved in the data exchange.
3 Fundamental Concepts and Assumptions

This chapter discusses fundamental concepts and assumptions for Configuration Management. Focus is on items that are in scope of this document (see chapter 1.1.2), but also some related aspects are elaborated upon to clarify the context of in-scope items.

3.1 The Big Picture

ISO 10303 (STEP) is an International Standard for the computer-interpretable representation of product information and for the exchange of product data. Its objective is to provide a neutral mechanism capable of describing products throughout their life cycle. The document in hand is a guide to some aspects of only ISO 10303-242: Managed model-based 3D engineering, also called AP242.

AP242 supports the definition of product concepts, which define products from market or customer-oriented viewpoints. As they are offered to the customers, these product concepts often define conceptual product models that are available for or delivered to the customer in different configurations or variants.

AP242 is suitable for the following use cases; see chapter 3.2 for details):

- Neutral file exchange
- Archiving and retrieval of product data
- Implementation and sharing of product databases within PLM processes.

AP242 has not the aspiration to support all aspects of Configuration Management and is not designed to be the internal data model of a Configuration Management System.

The scope of AP242 in the context of PLM in general is shown in Figure 1: AP242 Big Picture. The red boundaries and red text indicate the capabilities of AP242. The white circles represent sub-domains within general PLM.

The focus of this document is the subdiscipline of configuration identification of the physical features for typical or serialized Configuration Items (CIs) (see chapter 3.3.5).

In scope is product structure with explicit or implicit configuration. Configuration is restricted to product structure and related documents with CAD models, drawings and additional descriptions. Software components and NC-programs are only in scope in this release of the document if they are managed as parts; management of software or NC code is out of scope.

Product breakdown is out of scope of this release of the document.
3.2 Use Cases

With respect to Configuration Management, AP242 supports the following three main use cases, which are explained in the sub-chapters, below:

- Configuration Management for supplier exchange
- Configuration Management for archiving and retrieval
- Configuration Management for internal PLM.

3.2.1 Configuration management with AP242 for supplier exchange

Main characteristics of supplier data exchange are:

- different organizations
- different systems
- asynchronous data exchange.
In this use case, the focus for AP242 is to support explicit configurations. If partners support the same configuration management methodology, also implicit configuration could be used.

3.2.2 Configuration management with AP242 for archiving

Main characteristics for long-term archiving and retrieval are:

- same organization
- different systems (along the timeline)
- different people (all “tacet” knowledge is gone; the archival package needs to be totally self-contained).

In this use case, the focus for AP242 is to support implicit and explicit configurations with additional organisation specific information. Organisation specific information could be, for example:

- details of the status network
- information about users, groups, roles
- details of the file repository.

3.2.3 Configuration management with AP242 for internal PLM

Main characteristics of internal PLM for synchronous integration of several internal systems are:

- same organization
- different systems
- synchronous integration.
In this use case, the focus for AP242 is to support the daily business with all needed information about specifications, conditions and breakdowns. Internal exchange is needed, if different systems are in use to support the product lifecycle and to synchronize information between ERP, PLM and Production Planning and Control System (PPS). The need to exchange data is often linked to lifecycle transitions.

![Figure 4: Configuration Management for internal PLM](image)

### 3.3 Configuration Management

This chapter gives a short introduction to configuration management in general and highlights relevant aspects in AP242.

#### 3.3.1 Definition

Configuration management is a systems engineering process for establishing and maintaining consistency of a product's functional and physical features with its requirements, design, and operational information throughout its life.

Configuration management is a management discipline and is closely linked with:

- Product Structure Management
- Requirements Management
- Change Management
- Release Management
- Supply Chain Management
- Archiving and retrieval.

Configuration Management has five main functions of which only the first one is in scope of this document (see chapter 1.1.2):

- **Configuration Identification**
  Identify and document the functional and physical characteristics of the collection of Configuration Items.
• **Configuration Control**  
  Control changes to Configuration Items and related documentation.

• **Configuration Status Accounting**  
  Record and report information needed to manage Configuration Items effectively, including the status of proposed and approved changes.

• **Configuration Audit and Verification**  
  Audit Configuration Items to verify conformance to documented requirements.

• **Configuration Management Planning**  
  Application of standards, guidelines, best practices

The focus of AP242 is physical and functional configuration identification.

![Figure 5: Configuration Management functions](image)

### 3.3.2 Configuration Identification

Configuration Identification describes the identification and documentation of functional and physical features of the product, such as:

- Identification and placement of Configuration Items in the product structure
- Documentation of physical and functional features
- Methodology of numbering with unique identifiers and classifications
- Definition of baselines.

### 3.3.3 Configuration

A configuration is a description of a product at a particular time or in a defined delivery status. The description contains all documentation relevant to manufacturing, assembly, quality control and maintenance.

Documentation may include:

- Product Structure
- Documents
  - CAD Models
  - Viewing data
- Drawings
- NC programs
- Textual descriptions
- Calculation results

- Software Components

Configuration calculation results and NC programs are supported as referenced documents only. Software components are supported as parts of kind ‘software’.

Configurations may be defined explicitly or implicitly, that is, by conditions.

### 3.3.3.1 Explicit Configuration

Configuration mechanisms are used differently across various industry branches. In Aerospace, explicit representation of configurations of product concept is suitable for the management of design as-planned manufacturing configurations, the traditional BOM inputs to a manufacturing resource planning. These explicit configurations may also be suitable for management of other activities “downstream” from the design phase, such as as-built and as-maintained configurations.

From an Automotive point of view, the manufacturable object is triggered by Date Effectivities, but not for explicit configuration. Instead, the parts, the part occurrences and the conditions of the product specification have additional date effectivities (i.e. they apply from datetime xxx to datetime yyy). The explicit configuration is computed out of them all (for a given customer order at a given date).

The Explicit Configuration is mostly a 100% product structure (> 100% if for example serial/date ranges are used to define a ProductConfiguration). The content differs, dependent on the use case:

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data exchange</td>
<td>No effectivity information</td>
</tr>
<tr>
<td>Archiving</td>
<td>No effectivity information</td>
</tr>
<tr>
<td>Internal PLM</td>
<td>With effectivity information</td>
</tr>
</tbody>
</table>

*Table 2: Configuration use cases*
Configuration effectivity in AP242 allows attachment of effectivity information to part versions or to occurrences of component parts in the context of a particular Configuration Item. This enables specification of the valid use of a part version or a part occurrence in the context of a lot, serial number range or time period of a particular product configuration. These kinds of effectivities can also be used in combination with conditional effectivities. This controls the constituent parts that are planned to be used for manufacturing end items of a particular product configuration within a dated time period or for a certain lot, a serial number range of the end items and/or certain conditional Effectivities.

3.3.3.2 Implicit Configuration
It is recommended to use implicit configuration if the number of possible variations of a product concept is too large. The members of a product concept might therefore be defined implicitly by identifying the product features that characterize it or are available for it as options (so-called 150% structure). Conditions among those features may be specified to manage dependencies and define the valid product variations for a particular concept. This implicit definition of the configurations available for a certain product concept is suitable for customer option and variant specification: once all options are specified a single configuration is determined that may be represented explicitly for downstream applications. The implicit configuration is often driven by the ERP system and applied to the product structure in PLM systems for DMU purpose.

The usage of implicit configuration requires the support of the same specifications for a product at both exchange partners PLM/BoM system, or at least that the specifications are mappable to each other. If needed, the Dictionaries and SpecificationCategories also have to be mappable.

The implicit configuration will be defined in following definitions:

- Effectivities (Serial, Date, Conditional, …)
- Variants (ProductConfigurations)
- Options (Specifications, SpecificationCategories)
- Conditions (AND, OR, NOT, …)
- Dictionaries (Option Pool).

3.3.3.3 Feasibility
The use of feasibility is an example of how implicit configurations are defined by options and conditions.

The feasibility of configurations is out of scope of this document but included here for clarification.

The following table shows the possible variants that could be delivered to the customer and their corresponding options. Respecting given conditions, not all combinations of the options may result in valid product variants. The conditions themselves are not part of this matrix and could be defined as follows:

- If B then not C
- If A then not D
Configuration tables are a simple way of handling feasibility rules. More complex tables may be rule based. The list of variants may be generated out of a configuration table, but sometimes there are so many variants that they are only defined as needed and their feasibility must be checked.

### 3.3.3.4 Automatic completion

Automatic completion is another example of how implicit configurations are defined by options and conditions.

The automatic completion of configurations is out of scope of this document but included here for clarification. Depending on the presence of certain options, some further options may have to be added automatically, for example the option 'larger battery' if many electrical devices are ordered by the customer.

### 3.3.4 Configuration Baseline

A configuration baseline is the description of a frozen product configuration at a defined date and is used for company internal or external purposes.

This document focuses on the release processes and milestone reviews within the engineering phase and the manufacturing phase, as well as the transition from engineering to manufacturing planning. AP242 in general will support the product lifecycle phases from as-planned to as-built. The phases for as-deployed and as-maintained are supported by AP232 and AP239.
3.3.5 Configuration Item (CI)

A Configuration Item is an item that is identified and controlled over its entire life cycle. It can be a single part, an assembly or a software program or any combinations of those.

The following three different kinds of CIs may be distinguished for engineering and manufacturing:

- Technical Configuration Item (TCI)
- Contractual Configuration Item (CCI)
- Serialized Configuration Item (SCI).

3.3.6 Product Concept

Product concept is the understanding of the applicability of a product in order to showcase its best qualities and maximum features. Marketers spend a lot of time and research in order to target their intended audience. Marketers will look into a product concept before marketing a product towards their customers.

While the "product concept" is based upon the idea of customers who prefer products with the most quality, performance, and features, some customers prefer products that are simple and easy to use. Figure 12 shows the relationship between product concept and configuration.

3.3.7 Bill of Material (BOM)

A Bill of Material (BOM) is an annotated list of constituents of a product at a specific point in time of its life cycle.

The following table gives a rough classification of the different ways of managing BOMs and their association to possible configuration approaches. This classification is only relevant for the use case "internal PLM". The association to other use cases is industry dependent and, therefore, not included.
### Table 3: BOM Classification for Configuration Management for internal PLM

<table>
<thead>
<tr>
<th>Bill of material</th>
<th>Coverage</th>
<th>Configuration management / Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completly designed (concrete) BOM</td>
<td>100% BOM</td>
<td>• Order Engineering</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Explicit Configuration</td>
</tr>
<tr>
<td>Order neutral, configured BOM</td>
<td>150% BOM</td>
<td>• Driven by Change and Release Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Few Order Engineering</td>
</tr>
<tr>
<td>Rule based or feature based variant BOM</td>
<td>150% BOM</td>
<td>• Implicit Configuration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Driven by CPQ (Configure Price Quote) out of ERP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• No Order Engineering</td>
</tr>
</tbody>
</table>

There may be many types of BOMs. eBOM and mBOM are explicitly in scope of this document.

#### 3.3.7.1 eBOM

The engineering bill of material or eBOM is a configuration of the product to show how it is designed. It is used in the product lifecycle phase of design.

#### 3.3.7.2 mBOM

The manufacturing bill of material or mBOM is a configuration of the product to show how it will be assembled. It contains all the parts and assemblies required to build a complete and shippable product. It is used in the product lifecycle phase of manufacturing. The mBOM is normally derived from the eBOM but refactored for the requirements of manufacturing. The eBOM is usually changed as follows:

- The mBOM includes additional information, such as, machines and tools that are necessary to build the product, assembly process information and packaging material.
- The mBOM could also prune the eBOM from all component parts that are assembled by suppliers.
- The mBOM may have a completely different product structure than the eBOM, since it shall reflect the assembly process and not the top-down physical structure of the eBOM.

#### 3.3.8 Configuration Management system

A Configuration Management system consists of a set of tools to identify the information belonging to a product configuration during the entire product life cycle. Data concerning Configuration Management systems is out of scope of this document.

#### 3.3.9 Effectivity

Effectivity describes the validity of data in a certain context. The validity of data can be expressed by conditions that specify, for example, time ranges for which data are applicable. Validity may also be expressed for individual parts, which are identified by serial or unit numbers.
3.3.9.1 Unit Effectivities

A unit (or serial) effectivity could be defined for a single lot or a single serial number as well as for a range or a list of them, or a combination of single value, range and list.

The following scenarios are supported:

- For a given lot or a given serial number
- From 0 to a given serial number
- From a given serial number to a higher serial number
- From a given serial number to undefined.

![Figure 9: Unit Effectivity Example](image)

3.3.9.2 Date Effectivities

A date effectivity is mainly dependent on the initial creation, change, or release management and will mostly be defined by an ERP system. The date effectivity is also used in the PLM System in eBOMs attached to engineering change orders and in mBOMs attached to the manufacturing change orders.

The following three scenarios are supported:

- From undefined to date
- From begin date to end date
- From begin date to undefined.
3.3.9.3 Conditional Effectivity

Conditional effectivity is explained in chapter 3.3.3.2, Implicit Configuration. Like the other kinds of effectivities, conditional effectivity data are exchanged ‘for information only’.

3.4 Definitions in ISO 10303-3001

These recommended practices are based on the following document:

- TECHNICAL SPECIFICATION ISO/TS 10303-3001:2016(E)
  Industrial automation systems and integration — Product data representation and exchange
  Part 3001: Business object model: Managed model based 3d engineering

This standard is a part of AP242 and often referred to as the AP242 Business Object (BO) model.

The figure below provides an overview of the AP242 BO information model capabilities and their logical groups.
3.4.1 Configuration

AP242 supports configuration management by the four main concepts that are listed on the right-hand side of Figure 12 below. They are renamed and put in relation to one another on the left-hand side.
The figure is a simplified view of the descriptions in chapter 4.2.7. of the STEP AP242 Business Object Model.

The configuration capability of the AP242 BO model provides mechanisms to identify a single manufacturable product out of a large number of variants and versions. The main concepts that are used to handle this identification of a single manufacturable product are the following:

- Product configurations are used to identify a manufacturable product out of a possibly large number of variants of a product class by its associated specifications without having an explicit representation of the product, e.g., a customer order for a car.
- Product design association (ProductPartEquivalence) is used to associate an implicit representation of a manufacturable product (ProductConfiguration) with its explicit representation (PartVersion).
- Manufacturing configuration is used to specify the validity of occurrences of parts (PartInstance) with regard to manufacturing aspects in the context of an implicit as well as an explicit representation of a manufacturable product (ProductConfiguration, PartVersion).
- Configurations are used to configure part occurrences, alternative solutions, breakdown elements, and process plans in the context of the conditions (ClassConditionAssociation, ClassSpecificationAssociation) under which they are valid.

![Configuration in ISO 10303-3001](image-url)
3.4.2 Assembly structure
(See chapter 4.2.3. of STEP AP242 Business Object Model)

This capability allows representing relationships between objects to build up various kinds of structures. The structures are defined between part views or between part views and part occurrences.

Among the structures supported by this capability is a hierarchical assembly structure to represent the relationships between an assembly and its constituents. For representing hierarchical assembly structures two concepts are supported by this capability: the part view-based assembly structure concept and the part occurrence-based assembly structure concept.

- The part occurrence-based assembly structure concept allows building up a hierarchical assembly structure by an AssemblyDefinition representing the assembly and one or more Occurrence objects representing its constituents.
- The part view-based assembly structure concept allows building up a hierarchical assembly structure by an AssemblyDefinition object representing the assembly and one or more Part-View objects representing its constituents.

![AssemblyStructure with Cartesian Transformation](image)

*Figure 14: AssemblyStructure with Cartesian Transformation*
Figure 15: Assembly Structure with Geometric Representation Relationship

The exchange of assembly structures is already defined in the Recommended Practices for STEP AP242 Business Object Model XML “Product & Assembly Structure” and is, therefore, out of scope of the document in hand.

3.4.3 Breakdown
(See chapter 4.2.4. of STEP AP242 Business Object Model)

This capability enables the representation of parent-child structures, so-called breakdowns. A breakdown consists of breakdown elements. Different types of breakdowns and breakdown elements are distinguished, such as functional and physical. Breakdowns may be views of actual product structures with relations to only some of their constituents.

This capability also enables the representation of alternative solutions, such as final, supplier or technical solutions. These are implementations of functional and physical breakdown elements.

Finally, part occurrences, that is, occurrences of parts in certain locations, can be associated with breakdown elements by means of this capability.
The exchange of breakdowns is not covered in this document.

3.4.4 Part occurrence
(See chapter 4.2.25. of STEP AP242 Business Object Model)

Part occurrence provides the capability to represent occurrences of PartView objects. It includes master data about part occurrences as well as relationships between part occurrences. Part occurrences are used to identify an occurrence of a component in an assembly structure, e.g., to identify the elements of an AlternativeSolution.

3.4.5 Part view
(See chapter 4.2.26. of STEP AP242 Business Object Model)

This capability provides the ability to represent basic product management information about different views on part versions as well as relationships between different part views.

3.4.6 Effectivity
(See chapter 4.2.12. of STEP AP242 Business Object Model)

The Effectivity capability represents information concerning the validity of data. Implicit propagation of data specifying validity is not available. The validity of data can be expressed by effectivities that specify, for example, time ranges within which data may be used. Explicit dates or dates expressed by events can be used in order to represent the relevant points in time. Effectivities may also express rule-based conditions (so-called ConditionalEffectivity) used for implicit
configuration. Validity may also be expressed for individual parts, which are identified by serial numbers.

The usage of effectivities in AP242 is described in detail in chapter 4.2.1.

3.4.7 Product specification
(See chapter 4.2.31. of STEP AP242 Business Object Model)

The Product Specification provides the capability to describe products with a large number of variants.

Examples for aircraft products are propeller aircrafts, jet aircrafts, rocket-powered aircrafts, or engines or components of these products. Examples for automotive products are passenger cars, trucks, buses, or engines or components of these products.

Because of the large number of variants there is no unique identification for each of the variants that could be produced. The main concepts that are used to handle this large number of variants are the following:

- product classes are used to identify sets of similar products to be offered to the market
- specifications are used to describe characteristics of the products
- specification categories are used to group similar characteristics of the products
- specification conditions are used to control the usage of a part within a product and to represent conditions for product classes.

The usage of product specification is not covered in this document.
4 Configuration and Effectivity Information

This chapter will describe the templates for the usage of the AP242 Entities in the context of configuration management.

Each template will describe following content:
  o The Instance Model as AP242 BO Model XML entities and attributes (instantiation picture)
  o AP242 BO Model XML syntax
    • Table with supported attributes
    • Attribute recommendations
    • Preprocessor Recommendations
    • Postprocessor Recommendations

4.1 Configuration Identification

Configuration identification in AP242 is the identification of product concepts and their associated configurations, the composition of which is to be managed. If a configuration of a product concept is implemented by a certain design, i.e., a particular part version, this version can be associated with the configuration and managed using configuration effectivity.

The configuration identification supports two important concepts to describe the products that are sold by an organization to its customers:
  • Product Concept Identification
    o Supports the representation of an organization's products as they are conceived or offered to the customers. A product as offered to the market can directly correspond to a manufacturable object, or it can be available to the customer in different configurations where each of these configurations defines a manufacturable object.
  • Product Concept Configuration Identification
    o Supports the specification of these product configurations and their associated part designs.

4.1.1 Default Context

The default context (product concept) has to be used, if one of the involved systems does not manage the product concept.

The recommended usage of a context depends on two conditions:
  • Support of the involved PLM system
  • Complexities of the configuration

PLM systems are different in usage of a context for configurations. Following three features has to be supported for the Use Cases defined in chapter 3.2.
  • PLM systems with a mandatory context
  • Optional usage of a context
  • A context is not supported by the PLM System

There are different complexities in the scenarios:
• In case of an explicit configuration with a 100% structure and only one context, we have a simple scenario with low configuration complexity and no context is needed.

• In case of implicit configurations or if more than one context is used, we speak about a complex configuration scenario and a context has to be defined.

Most of the PLM systems require a context. To get a most common usage of the context, independent from the involved PLM systems and independent from the complexity of the scenario, it is recommended to support the context in each processor with the feature to handle a default context.

The default context has no market association and the Id should be set to “/NULL”. This is a placeholder string where the target system has to create the context itself.

The mapping of the context is different in the PDM Systems and could be as following:

• Context as a specific item (e.g. Teamcenter, 3DExperience)

• Context as a flag at the root item of the product structure (e.g. Windchill)

The context is defined in the templates:

• ProductConceptIdentification (see chapter 4.1.2)

• ProductConceptConfigurationIdentification (see chapter 4.1.3)

**The Instance Model for the default context**

```
<ProductConcept uid="PC_1" xsi:type="bom:ProductClass">
  <Id id="/NULL"/>
  ...
</ProductConcept>
```

**Preprocessor Recommendations:**

If the source PLM system does not support a context, it has to be created with default values to be compliant to the recommendations.

It is not supported to use more than one context within one AP242 file.

**Postprocessor Recommendations:**

“/NULL” is a placeholder string already used for similar purposes in other places and means the target system has to create the context itself.

A warning has to be indicated, if more than one context is specified in the configuration.

The postprocessor recommendation depends on the context support of the involved PLM system.

<table>
<thead>
<tr>
<th>Target PLM systems</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>No support of a context</td>
<td>The context has to be ignored, even if it is not the default context.</td>
</tr>
<tr>
<td>Context is optional</td>
<td>If the recommended default context is used, it has to be ignored.</td>
</tr>
<tr>
<td>Context is mandatory</td>
<td>If the recommended default context is used, it has to be ignored and the default context of the target PLM should be used.</td>
</tr>
</tbody>
</table>

*Table 4: Context handling recommendations*
4.1.2 Template “ProductConceptIdentification”

The products that are conceived or offered by an organization to its customers are often defined as variations or configurations of a common product model, referred to as product concept. This product concept is defined in a market context and can be seen as a logical container for all of its variations. In the AP242 Schema, product concept identification is the representation of these product concepts with their related market context. A product as offered to the market can directly correspond to a manufacturable object, or it can be available to the customer in different configurations where each of these configurations defines a manufacturable object.

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

![Diagram of ProductConceptIdentification]

**Figure 17: Instance Model ProductConceptIdentification**

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

```xml
<ProductConcept uid="pc--19088" xsi:type="n0:ProductClass">
  <Description>
    <CharacterString>as class description</CharacterString>
  </Description>
  <Id>
    <Identifier uid="pc-asclass--id1" id="as class" idRoleRef="rl--ii" idContextRef="o--000000178"/>
  </Id>
  <Name>
    <CharacterString>class name</CharacterString>
  </Name>
  <LevelType>
    <ClassString>platform</ClassString>
  </LevelType>
</ProductConcept>
```

© PDM Implementor Forum
4.1.2.1 ENTITY ProductClass

A product concept describes a class of similar products that an organization provides to its customers. It represents the idea of a product as identified by potential or actual customer requirements. Therefore, a product concept may exist before the parts have been defined that implement it. Depending on the kind of industry and products, a product concept might be offered to the customers in one or many different configurations. If exactly one configuration is defined, the product concept corresponds to a particular part design. If the product concept is offered in different configurations, each of these configurations is a member of the class of products defined by this product concept.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>IdentifierSelect</td>
</tr>
<tr>
<td>Name</td>
<td>OPTIONAL DescriptorSelect</td>
</tr>
<tr>
<td>Description</td>
<td>OPTIONAL DescriptorSelect</td>
</tr>
<tr>
<td>TargetMarket</td>
<td>OPTIONAL Market</td>
</tr>
<tr>
<td>ClassifiedAs</td>
<td>OPTIONAL SET[1:?] of ClassifiedAsType</td>
</tr>
<tr>
<td>MaterialIdentification</td>
<td>OPTIONAL SET[1:?] of MaterialIdentification</td>
</tr>
<tr>
<td>SameAs</td>
<td>OPTIONAL SET[1:?] of SameAsType</td>
</tr>
<tr>
<td>DateAndPersonAssignment</td>
<td>OPTIONAL SET[1:?] of DateAndPersonAssignment</td>
</tr>
<tr>
<td>DateTimeAssignment</td>
<td>OPTIONAL SET[1:?] of DateTimeAssignment</td>
</tr>
<tr>
<td>DocumentAssignment</td>
<td>OPTIONAL SET[1:?] of DocumentAssignment</td>
</tr>
<tr>
<td>EventAssignment</td>
<td>OPTIONAL SET[1:?] of EventAssignment</td>
</tr>
<tr>
<td>Property Assignment</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ModelPropertyAssignment</td>
<td>OPTIONAL SET[1:?] of ModelPropertyAssignment</td>
</tr>
<tr>
<td>OrganizationOrPersonInOrganizationAssignment</td>
<td>OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment</td>
</tr>
<tr>
<td>ProjectAssignment</td>
<td>OPTIONAL SET[1:?] of ProjectAssignment</td>
</tr>
<tr>
<td>PropertyDefinitionAssignment</td>
<td>OPTIONAL SET[1:?] of PropertyDefinitionAssignment</td>
</tr>
<tr>
<td>PropertyValueAssignment</td>
<td>OPTIONAL SET[1:?] of PropertyValueAssignment</td>
</tr>
<tr>
<td>TimeIntervalAssignment</td>
<td>OPTIONAL SET[1:?] of TimeIntervalAssignment</td>
</tr>
<tr>
<td>LevelType</td>
<td>OPTIONAL ClassSelect</td>
</tr>
<tr>
<td>ActivityAssignment</td>
<td>OPTIONAL SET[1:?] of ActivityAssignment</td>
</tr>
<tr>
<td>ActivityMethodAssignment</td>
<td>OPTIONAL SET[1:?] of ActivityMethodAssignment</td>
</tr>
<tr>
<td>ApprovalAssignment</td>
<td>OPTIONAL SET[1:?] of ApprovalAssignment</td>
</tr>
<tr>
<td>BreakdownVersionAssignment</td>
<td>OPTIONAL SET[1:?] of BreakdownVersionAssignment</td>
</tr>
<tr>
<td>EffectivityAssignment</td>
<td>OPTIONAL SET[1:?] of EffectivityAssignment</td>
</tr>
<tr>
<td>FrozenAssignment</td>
<td>OPTIONAL SET[1:?] of FrozenAssignment</td>
</tr>
<tr>
<td>InformationUsageRightAssignment</td>
<td>OPTIONAL SET[1:?] of InformationUsageRightAssignment</td>
</tr>
<tr>
<td>ProductClassRelationship</td>
<td>OPTIONAL SET[1:?] of ProductClassRelationship</td>
</tr>
<tr>
<td>SecurityClassificationAssignment</td>
<td>OPTIONAL SET[1:?] of SecurityClassificationAssignment</td>
</tr>
<tr>
<td>SpecificationAssignment</td>
<td>OPTIONAL SET[1:?] of SpecificationAssignment</td>
</tr>
<tr>
<td>SpecificationCategoryAssignment</td>
<td>OPTIONAL SET[1:?] of SpecificationCategoryAssignment</td>
</tr>
<tr>
<td>SpecificationConditionAssignment</td>
<td>OPTIONAL SET[1:?] of SpecificationConditionAssignment</td>
</tr>
<tr>
<td>SpecificationInclusionAssignment</td>
<td>OPTIONAL SET[1:?] of SpecificationInclusionAssignment</td>
</tr>
</tbody>
</table>
Table 5: "ProductClass" Attributes

Attribute recommendations:

- **Id**: The id attribute specifies the identifier of the product concept. It is usually assigned by the organization that provides the product belonging to that product concept to its customers. This identifier must be unique within the given scope. Use “Identifier” template (see [242BO-PAS]).

- **Name**: The name attribute specifies the nomenclature or common name, by which the product concept is referred. The value of this attribute need not be specified. Use “Description” template (see [242BO-PAS]).

- **Description**: The description attribute specifies additional information about the product concept. The value of this attribute need not be specified. Use “Description” template (see [242BO-PAS]).

- **TargetMarket**: The market for the products belonging to a product concept is further described in the market_context attribute. The value of this attribute need not be specified.

- **LevelType**: the level of the ProductClass in a hierarchical structure of ProductClass objects. Although the value of this attribute need not be specified, it is recommended not to leave it unset. Use ClassString type if one of the values below is used, otherwise use “Class" template (see [242BO-PAS] 4.6.4). According to prostep ivip CC8 Recommended Practices, Version 1.2, where applicable, the following values shall be used:

<table>
<thead>
<tr>
<th>RelationType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'enterprise'</td>
<td>The highest level of the ProductClass instance in a file must always represent the company. This is necessary as some systems have this class as a general starting point. The id of this instance will be a unique identifier for the relevant company. This class carries no variant control information but only characteristics (e.g. Specification, SpecificationCategory etc.) of that class.</td>
</tr>
<tr>
<td>'platform'</td>
<td>This instance of the ProductClass reflects the level of a product class in a BoM system which represents a main technical product base (e.g. project, platform, engineering series etc.). In some cases, this level carries a complete BoM (&quot;Maximum BoM&quot;) for a project, platform, engineering series etc. This level is in some cases called technical documentation. This level carries only characteristics of this class level, no variant control mechanisms are attached.</td>
</tr>
<tr>
<td>'product family'</td>
<td>This instance will be the one to which all variant control mechanisms are attached. For trucks configuration information (Specifications, Conditions etc.) may be associated on level three and four. For cars this may only be attached on level three.</td>
</tr>
<tr>
<td>‘furthest pre-configured abstract product class’</td>
<td>Furthest pre-configured abstract ProductClass: this instance represents the furthest configured class of a product, which is not yet a real product. E.g. this could be a complete vehicle, engine, gearbox etc. which has</td>
</tr>
</tbody>
</table>
not been evaluated against customer special choices or an abstract vehicle, engine, gearbox etc. which could become a real one after the associated BoM is evaluated. The purpose of this level of a product class instance is in any case to reflect that level of ProductClass of a BoM system which leads to the individual BoM for a single product. This ProductClass level will be particularly exchanged in the context of scenarios to exchange a single BoM.

- **SpecificationAssignment**: to assign (optionally) one or multiple Specifications to the ProductClass. Use the “SpecificationAssignment” template (see 4.3.3).

- **SpecificationCategoryAssignment**: to assign (optionally) one or multiple SpecificationCategories to the ProductClass. Use the “SpecificationCategoryAssignment” template (see 4.3.3).

- **SpecificationConditionAssignment**: to assign (optionally) one or multiple Conditions to the ProductClass. Use the “SpecificationConditionAssignment” template (see 4.3.4).

**Preprocessor Recommendations:**
There might be several levels of product concepts defined in a company. If product concept is not supported, the default context has to be used as product concept. (see chapter 4.1.1)

**Postprocessor Recommendations:**
If not supported by the target system, TargetMarket shall be ignored or a warning shall be returned.

Remark: a product concept represents a conceptual idea of a class of similar products that are offered to a market. No design or manufacturing related product data can be attached to the product concept directly.

### 4.1.2.2 ENTITY Market
The Market entity is a subtype of BaseRootObject. It defines the market context in which a product concept is defined and may include information characterizing the potential customers of the products of the associated product concept. The application domain is identified by the associated application_context entity.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassifiedAs</td>
<td>OPTIONAL SET[1:?] of ClassificationSelect</td>
</tr>
<tr>
<td>Id</td>
<td>OPTIONAL IdentifierSelect</td>
</tr>
<tr>
<td>MarketSegmentType</td>
<td>OPTIONAL ClassSelect</td>
</tr>
<tr>
<td>Name</td>
<td>OPTIONAL DescriptorSelect</td>
</tr>
<tr>
<td>SameAs</td>
<td>OPTIONAL SET[1:?] of ProxySelect</td>
</tr>
</tbody>
</table>

Table 6: "Market" Attributes

**Attribute recommendations:**
- **MarketSegmentType**: The market segment type is a label that identifies the kind of consumer preferences associated with a product concept. Use “Class” template (see [242BO-PAS]).

**Preprocessor Recommendations:**
There are no specific preprocessor recommendations.

**Postprocessor Recommendations:**
There are no specific postprocessor recommendations.

### 4.1.3 Template “ConfigurationItem”
This is the explicit identification of the configurations that exist for a given product concept and the representation of their composition. A Configuration Item is specified as a ProductConfiguration entity. Product configurations can be related to appropriate part versions to specify the designs that implement the product configurations through the configuration design.

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

```xml
<ProductConfiguration uid="pconf--19089">
  <Description>
    <CharacterString>as1-short description</CharacterString>
  </Description>
  <Id>
    <Identifier uid="pc-as1short--id1" id="as1 short" idRoleRef="rl--ii"
    idContextRef="o--000000178"/>
  </Id>
  <MemberOf uidRef="pc--19088"/>
  <Name>
    <CharacterString>as1-short name</CharacterString>
  </Name>
</ProductConfiguration>
```
4.1.3.1 ENTITY ProductConfiguration

The ProductConfiguration entity is a key concept to support explicit and implicit configuration management. It represents the identification of a particular Configuration Item, i.e., variation of a product concept. A ProductConfiguration is defined with respect to the product concept, i.e., the class of similar products of which it is a member.

The ProductConfiguration defines a manufacturable end item, or something that is conceived and expected as such. The association between a ProductConfiguration and a corresponding part design is established using a ProductDesignAssociation. The valid use of component parts for planned units of manufacturing of a particular ProductConfiguration may be specified using configuration effectivity (see 4.2.1).

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassifiedAs</td>
<td>OPTIONAL SET[1:?] OF Classification</td>
</tr>
<tr>
<td>DefiningSpecifications</td>
<td>OPTIONAL SET[1:?] OF Specification</td>
</tr>
<tr>
<td>Description</td>
<td>OPTIONAL DescriptorSelect</td>
</tr>
<tr>
<td>Id</td>
<td>IdentifierSelect</td>
</tr>
<tr>
<td>MaterialDefinition</td>
<td>OPTIONAL S [1:n] OF MaterialDefinition</td>
</tr>
<tr>
<td>MemberOf</td>
<td>ProductClass</td>
</tr>
<tr>
<td>Name</td>
<td>OPTIONAL DescriptorSelect</td>
</tr>
<tr>
<td>Occurrence</td>
<td>OPTIONAL SET[1:?] OF Definition-BasedOccurrence</td>
</tr>
<tr>
<td>SameAs</td>
<td>OPTIONAL S [1:n] OF ProxySelect</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>VersionId</td>
<td>OPTIONAL IdentifierSelect</td>
</tr>
<tr>
<td>ActivityAssignment</td>
<td>OPTIONAL SET[1:?] of ActivityAssignment</td>
</tr>
<tr>
<td>ActivityMethodAssignment</td>
<td>OPTIONAL SET[1:?] of ActivityMethodAssignment</td>
</tr>
<tr>
<td>ApprovalAssignment</td>
<td>OPTIONAL SET[1:?] of ApprovalAssignment</td>
</tr>
<tr>
<td>BreakdownVersionAssignment</td>
<td>OPTIONAL SET[1:?] of BreakdownVersionAssignment</td>
</tr>
<tr>
<td>CertificationAssignment</td>
<td>OPTIONAL SET[1:?] of CertificationAssignment</td>
</tr>
<tr>
<td>ContractAssignment</td>
<td>OPTIONAL SET[1:?] of ContractAssignment</td>
</tr>
<tr>
<td>DateAndPersonAssignment</td>
<td>OPTIONAL SET[1:?] of DateAndPersonAssignment</td>
</tr>
<tr>
<td>DateTimeAssignment</td>
<td>OPTIONAL SET[1:?] of DateTimeAssignment</td>
</tr>
<tr>
<td>DocumentAssignment</td>
<td>OPTIONAL SET[1:?] of DocumentAssignment</td>
</tr>
<tr>
<td>EffectivityAssignment</td>
<td>OPTIONAL SET[1:?] of EffectivityAssignment</td>
</tr>
<tr>
<td>EventAssignment</td>
<td>OPTIONAL SET[1:?] of EventAssignment</td>
</tr>
<tr>
<td>FrozenAssignment</td>
<td>OPTIONAL SET[1:?] of FrozenAssignment</td>
</tr>
<tr>
<td>InformationUsageRightAssignment</td>
<td>OPTIONAL SET[1:?] of InformationUsageRightAssign-</td>
</tr>
<tr>
<td>ModelPropertyAssignment</td>
<td>OPTIONAL SET[1:?] of ModelPropertyAssignment</td>
</tr>
<tr>
<td>OrganizationOrPersonInOrganizationAssign-</td>
<td>OPTIONAL SET[1:?] of OrganizationOrPersonInOrganiz-</td>
</tr>
<tr>
<td>ProductConfigurationRelationship</td>
<td>OPTIONAL SET[1:?] of ProductConfigurationRelation-</td>
</tr>
<tr>
<td>ProjectAssignment</td>
<td>OPTIONAL SET[1:?] of ProjectAssignment</td>
</tr>
<tr>
<td>PropertyDefinitionAssignment</td>
<td>OPTIONAL SET[1:?] of PropertyDefinitionAssignmen-</td>
</tr>
<tr>
<td>PropertyValueAssignment</td>
<td>OPTIONAL SET[1:?] of PropertyValueAssignment</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>RequirementAssignment</td>
<td>OPTIONAL SET[1:?] of RequirementAssignment</td>
</tr>
<tr>
<td>SecurityClassificationAssignment</td>
<td>OPTIONAL SET[1:?] of SecurityClassificationAssignment</td>
</tr>
<tr>
<td>SuppliedObjectRelationship</td>
<td>OPTIONAL SET[1:?] of SuppliedObjectRelationship</td>
</tr>
<tr>
<td>TimeIntervalAssignment</td>
<td>OPTIONAL SET[1:?] of TimeIntervalAssignment</td>
</tr>
<tr>
<td>WorkRequestAssignment</td>
<td>OPTIONAL SET[1:?] of WorkRequestAssignment</td>
</tr>
</tbody>
</table>

Table 7: "ProductConfiguration" Attributes

Attribute recommendations:

- **DefiningSpecifications**: in case of implicit configurations, the set of Specification objects necessary to discriminate the SpecificationAssignment within its ProductClass. Each definingSpecifications shall be associated with a ProductClass by a SpecificationAssignment with associationType not equal 'part usage'. The associated ProductClass shall be either the associated ProductClass of the ProductConfiguration or any higher level ProductClass related directly or indirectly by a ProductClassRelationship with a relation-Type 'hierarchy'. The value of this attribute need not be specified. Use “Specification” template.

- **Description**: the text or the set of texts that provide further information about the ProductConfiguration. The value of this attribute need not be specified. Use "Description" template (see [242BO-PAS]).

- **Id**: The id attribute specifies an identifier that distinguishes the ProductConfiguration. The identifier or set of identifiers for the ProductConfiguration. (must be unique in relation with a specific ProductConcept). Use “Identifier” template (see [242BO-PAS]).

- **MemberOf**: the ProductClass that a ProductConfiguration belongs to. Use “ProductConceptIdentification” template. *Comment: an issue in Bugzilla has been created under #6132 to embed ProductConfiguration into ProductClass rather than defining it as a BaseRootObject.*

- **Name**: the words or set of words by which the ProductConfiguration is known. The value of this attribute need not be specified. Use “Description” template (see [242BO-PAS])

- **VersionId**: the identification or set of identifications of a particular version of the ProductConfiguration. The value of this attribute need not be specified. Use IdentifierString type or "Identifier" template (see [242BO-PAS]).

- **EffectivityAssignment**: to assign (optionally) one or multiple Effectivities to define the Configuration. Use the “EffectivityAssignment” template (see 4.2.7.3).

Preprocessor Recommendations:

There is no standard mapping for the id attribute of ProductConfiguration; however, the value should be unique in conjunction with the id attribute of the associated ProductConcept.
Postprocessor Recommendations:
There are no specific postprocessor recommendations.

Remarks:
A ProductConfiguration can be established prior to the existence of a corresponding part design, i.e., a ProductDesignAssociation does not need to exist for a ProductConfiguration.

If the design of a ProductConfiguration is made of many distinct assemblies, more than one ProductDesignAssociation may exist to this ProductConfiguration.

4.1.3.2 ENTITY ProductDesignAssociation
A ProductDesignAssociation is a mechanism to associate a PartVersion with its corresponding ProductConfiguration. It specifies the explicit design that corresponds to the ProductConfiguration. The ProductDesignAssociation represents the statement that, in all definition contexts, the PartVersion is a valid way to implement the ProductConfiguration.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssociatedConfiguration</td>
<td>ProductConfiguration</td>
</tr>
<tr>
<td>AssociatedDesign</td>
<td>PartVersion</td>
</tr>
<tr>
<td>ClassifiedAs</td>
<td>OPTIONAL SET[1:?] of Classification</td>
</tr>
<tr>
<td>Description</td>
<td>OPTIONAL DescriptorSelect</td>
</tr>
<tr>
<td>EffectivityAssignment</td>
<td>OPTIONAL SET[1:?] of EffectivityAssignment</td>
</tr>
</tbody>
</table>

Table 8: "ProductDesignAssociation" Attributes

Attribute recommendations:
- **AssociatedConfiguration**: the ProductConfiguration that represents the requirements for which a PartVersion as solution is designated.
  
  *Comment: an issue in Bugzilla has been created under #6437 to embed ConfiguredAssemblyEffectivity into ProductDesignAssociation rather than defining it as a BaseRootObject.*

- **AssociatedDesign**: the PartVersion that identifies a part which is a valid implementation for the ProductConfiguration, i.e. meets the requirements specified by the ProductConfiguration. Use "Part" template (see [242BO-PAS]).

- **Description**: the text or the set of texts that provide further information about the ProductDesignAssociation. The value of this attribute need not be specified. Use "Description" template (see [242BO-PAS]).

Preprocessor Recommendations:
The ProductDesignAssociation should only be used for the top node of the product structure.
Postprocessor Recommendations:
There are no specific postprocessor recommendations.

4.1.4 Template “ProductClassRelationship”

A ProductClassRelationship is a relationship between two ProductClass objects. The meaning of this relationship is specified further by the attribute relationType. **Attribute Name**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassifiedAs</td>
<td>OPTIONAL SET[1:?] of Classification</td>
</tr>
<tr>
<td>Description</td>
<td>OPTIONAL DescriptorSelect</td>
</tr>
<tr>
<td>Id</td>
<td>OPTIONAL IdentifierSelect</td>
</tr>
<tr>
<td>Related</td>
<td>ProductClass</td>
</tr>
<tr>
<td>RelationType</td>
<td>ClassSelect</td>
</tr>
<tr>
<td>EffectivityAssignment</td>
<td>OPTIONAL SET[1:?] of EffectivityAssignment</td>
</tr>
</tbody>
</table>
### RelationType

<table>
<thead>
<tr>
<th>RelationType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'derivation'</td>
<td>the ProductClassRelationship defines a relationship where the related ProductClass is derived from the relating ProductClass;</td>
</tr>
<tr>
<td>'hierarchy'</td>
<td>the ProductClassRelationship defines a relationship where the relating ProductClass is on a higher level in the hierarchy of ProductClass objects than the related ProductClass;</td>
</tr>
<tr>
<td>'substitution'</td>
<td>the ProductClassRelationship defines a relationship where the related ProductClass replaces the relating ProductClass</td>
</tr>
<tr>
<td>'version sequence'</td>
<td>the ProductClassRelationship defines a relationship where the relating ProductClass is the preceding version and the related ProductClass is the following version.</td>
</tr>
</tbody>
</table>

- **Related**: the other object of ProductClass that is part of the relationship

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

![Diagram](attachment:image)

**Figure 20: Instance Model ProductClassRelationship**

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

```
<ProductConcept uid="pc--19087" xsi:type="n0:ProductClass">
  <Id>
    <Identifier uid="pc-ent--id1" id="Smith.com" idRoleRef="rl--ii" idContextRef="0--000000178"/>
  </Id>
</ProductConcept>
```
4.1.5 General Handling of Product Configurations
It is possible to have one ProductConcept with more than one ProductConfigurations.

4.1.6 Example Mapping an Item to AP242 Context
Some systems manage the context in an Item and not in a separate context object. In most cases, the item is normally the top node of the product structure or an additional node on top of the product structure. The following example demonstrates the mapping of a 3Dexperience item to the AP242 context and the mapping of the AP242 context to a Teamcenter EndItem for the use case “supplier exchange” (in this example the EndItem could be another Part or the top assembly Part used as a context).

![Figure 21: Context Mapping for Supplier Exchange](image)

4.2 Configuration Composition Management
Configuration composition management is concerned with the specification of the different product configurations that exist for a given product concept, and the association with product data that is necessary to build those configurations. This includes the identification of the actual constituents that are to be included in a planned unit of production of a product configuration.

Effectivity is a key concept to control the valid use of product data. The AP242 Schema supports the association of effectivity information to different types of product data. Three different effectivity concepts are available:
• Configuration effectivity (serial, lot, time interval), describing the planned use of part versions (so-called revision effectivities) or part occurrences (i.e., occurrences of parts as sub-assemblies or component parts in some higher-level assembly, so-called occurrence effectivities) in the context of a Configuration Item defined for a ProductConcept.

• A more generic effectivity (DataValidityEffectivity), describing the validity period of part versions or part occurrences in which the associated product data may be used independent from any additional context (e.g., date, lifecycle or organization related) that further restricts the applicability of that effectivity.

• A conditional effectivity, describing the use of part versions or part occurrences based on specification expressions.

4.2.1 Common supertype “Effectivity”
Supertype of all effectivity kinds, it allows attachment of effectivity information to part versions or to occurrences of component parts in the context of a particular Configuration Item or Product Concept. This enables specification of the valid use of a part version or part occurrence in the context of a lot, serial number range or time period of a particular product configuration. This controls the constituent parts that are planned to be used for manufacturing end items of a particular product configuration within a dated time period, for a certain lot or serial number range of the end items or according to specification expressions.

A configured assembly effectivity is an identification of the valid use of an aspect of product data tracked by date, event, serial number, lot size or specifications.

Effectivity is supertype of:
• DataValidityEffectivity (for Date Effectivities and Milestone Effectivities)
• LotEffectivity
• SerialEffectivity
• TimeIntervalEffectivity
• ConditionalEffectivity

Effectivities are related by:
• EffectivityAssignment
• Condition

<table>
<thead>
<tr>
<th>ENTITY</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassifiedAs</td>
<td>OPTIONAL SET[1:?] of Classification</td>
</tr>
<tr>
<td>ConcernedOrganizations</td>
<td>OPTIONAL SET[1:?] of Organization</td>
</tr>
<tr>
<td>Description</td>
<td>OPTIONAL DescriptorSelect</td>
</tr>
<tr>
<td>EffectivityContext</td>
<td>OPTIONAL EffectivityContext</td>
</tr>
<tr>
<td>EndDefinition</td>
<td>OPTIONAL SET[1:?] of EventOrDateSelect</td>
</tr>
<tr>
<td>Id</td>
<td>OPTIONAL IdentifierSelect</td>
</tr>
<tr>
<td>SameAs</td>
<td>OPTIONAL SET[1:?] of ProxySelect</td>
</tr>
<tr>
<td>StartDefinition</td>
<td>OPTIONAL SET[1:?] of EventOrDateSelect</td>
</tr>
</tbody>
</table>
### VersionId

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActivityAssignment</td>
<td>OPTIONAL SET[1:?] of ActivityAssignment</td>
</tr>
<tr>
<td>ApprovalAssignment</td>
<td>OPTIONAL SET[1:?] of ApprovalAssignment</td>
</tr>
<tr>
<td>ConditionAssignment</td>
<td>OPTIONAL SET[1:?] of ConditionAssignment</td>
</tr>
<tr>
<td>DateAndPersonAssignment</td>
<td>OPTIONAL SET[1:?] of DateAndPersonAssignment</td>
</tr>
<tr>
<td>DateTimeAssignment</td>
<td>OPTIONAL SET[1:?] of DateTimeAssignment</td>
</tr>
<tr>
<td>DocumentAssignment</td>
<td>OPTIONAL SET[1:?] of DocumentAssignment</td>
</tr>
<tr>
<td>EffectivityRelationship</td>
<td>OPTIONAL SET[1:?] of EffectivityRelationship</td>
</tr>
<tr>
<td>EventAssignment</td>
<td>OPTIONAL SET[1:?] of EventAssignment</td>
</tr>
<tr>
<td>FrozenAssignment</td>
<td>OPTIONAL SET[1:?] of FrozenAssignment</td>
</tr>
<tr>
<td>OrganizationOrPersonInOrganizationAssignment</td>
<td>OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment</td>
</tr>
<tr>
<td>SuppliedObjectRelationship</td>
<td>OPTIONAL SET[1:?] of SuppliedObjectRelationship</td>
</tr>
<tr>
<td>TimeIntervalAssignment</td>
<td>OPTIONAL SET[1:?] of TimeIntervalAssignment</td>
</tr>
</tbody>
</table>

#### Table 10: "Effectivity" Attributes

NOTE — The assignment of effectivities is often done during the approval process, i.e., when releasing product data for the next stage of the development process, it gets effectivity information assigned to define when and in which context these product data may be used.

**Attribute recommendations:**

- **Id**: The id attribute specifies the identifier that distinguishes the effectivity. Use “Identifier” template (see [242BO-PAS]). The value of this attribute need not be specified.

- **EffectivityContext**: the ProductClass for which the effectivity is valid (see “ProductConceptIdentification” template) or the ProductConfiguration for which this Effectivity applies (see “ConfigurationItem” template). The value of this attribute need not be specified.

**Preprocessor Recommendations:**

Even if defined as a BaseRootObject, it is recommended to not instantiate the effectivity instance as standalone instance, i.e., an effectivity should always be assigned to some product data by using EffectivityAssignment. Furthermore, it is always recommended to instantiate an effectivity as one of its subtypes, unless the definition of the effectivity is based on some other effectivity using effectivity_relationship.

DataValidityEffectivities of kind Date are the only Effectivities that do not require a Context in most PDM systems.
An identifier need not to be specified for a particular effectivity, except for shared effectivities (i.e. that are referenced multiple times within a product structure). If used, the id value should be unique in conjunction with the product data the effectivity is assigned to, and if present, the context restricting the applicability of the effectivity to a certain usage.

If the source system does not support a Product Context, it is recommended to create an EffectivityAssignment with a Role “context” and to attach it to a PartVersion representing the context (see “EffectivityAssignment template” 4.2.7.4).

**Postprocessor Recommendations:**

If the target system requires an EffectivityContext (like in 3DExperience), but none is provided (if optional for example like in Teamcenter for DataValidityEffectivities or if none is supported for example like in Aras), the postprocessor has to decide which (general) context to use.

### 4.2.2 Template “DataValidityEffectivity”

The DataValidityEffectivity is a subtype of Effectivity that applies onwards from a point in time, or between two points in time which define the start and end of the DataValidityEffectivity. These points in time may be specified by:

- Date
- Date and time
- Event occurrence (currently recommended for the mapping of Milestone effectivities)

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

```
ProductConceptIdentification

ProductClass #1-1

EffectivityContext

DateValidityEffectivity #1-1

Id

[AP242-PAS] Identifier

StartDefinition

DateTimeString>2016-12-10T12:14:23</DateTimeString>

EndDefinition

DateTimeString>2016-12-31T23:11:05</DateTimeString>
```

**Figure 22: Instance Model Date Effectivity**

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

```
<Effectivity uid="dveff-20161231" xsi:type="n0:DataValidityEffectivity">
  <EffectivityContext uidRef="pc--19088"/>
  <EndDefinition>
    <DateTimeString>2016-12-10T12:14:23</DateTimeString>
  </EndDefinition>
  <StartDefinition>
    <DateTimeString>2016-12-31T23:11:05</DateTimeString>
  </StartDefinition>
</Effectivity>
```
4.2.2.1 ENTITY DataValidityEffectivity
Additional attributes, derived from entity effectivity, will not be repeated in this table.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>EndDefinition</td>
<td>OPTIONAL EventOrDateSelect</td>
</tr>
<tr>
<td>StartDefinition</td>
<td>OPTIONAL EventOrDateSelect</td>
</tr>
</tbody>
</table>

Table 11: "DataValidityEffectivity" Attributes

Attribute recommendations:
- **EndDefinition**: The EndDefinition specifies the date and time (or Milestone) when the effectivity of the associated product data ends. It defines the upper bound of the interval of applicability. The value of this attribute need not be specified. If a value for this attribute is not defined, the interval of applicability has no upper limit. Use “DateTime” template for DateTimeString without DateTimeAssignment (see [242BO-PAS]).

- **StartDefinition**: The StartDefinition specifies the Date and Time (or Milestone) when the effectivity of the associated product data begins to apply. It defines the lower bound of the interval of applicability. The value of this attribute need not be specified. If a value for this attribute is not defined, the interval of applicability has no lower limit. Use “DateTime” template for DateTimeString without DateTimeAssignment (see [242BO-PAS]) in case of a Date Effectivity. Use “Event” in case of a Milestone Effectivity (see chapter 4.2.2.2).

Preprocessor Recommendations:
The StartDefinition shall be prior to the EndDefinition.
If a PDM system does not support an empty from-date, the dummy value 1970-01-01T00:00:00 may be used instead.

Postprocessor Recommendations:
The from-date value 1970-01-01T00:00:00 shall be interpreted as if from-date was empty.

4.2.2.2 ENTITY Event
An Event is the fact of the existence of a state at some point in time. It is recommended to use Event to map Milestone Effectivities

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActualStartDate</td>
<td>OPTIONAL DateTime</td>
</tr>
<tr>
<td>ClassifiedAs</td>
<td>OPTIONAL SET[1:?] of Classification</td>
</tr>
<tr>
<td>Description</td>
<td>OPTIONAL DescriptorSelect</td>
</tr>
<tr>
<td>EventType</td>
<td>ClassSelect</td>
</tr>
<tr>
<td>Id</td>
<td>OPTIONAL IdentifierSelect</td>
</tr>
</tbody>
</table>
### Table 12: "Event" Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset</td>
<td>xsd:duration</td>
</tr>
<tr>
<td>PlannedStartDate</td>
<td>OPTIONAL DateTime</td>
</tr>
<tr>
<td>ActivityAssignment</td>
<td>OPTIONAL SET[1:?] of ActivityAssignment</td>
</tr>
<tr>
<td>DateAndPersonAssignment</td>
<td>OPTIONAL SET[1:?] of DateAndPersonAssignment</td>
</tr>
<tr>
<td>DateTimeAssignment</td>
<td>OPTIONAL SET[1:?] of DateTimeAssignment</td>
</tr>
<tr>
<td>EventAssignment</td>
<td>OPTIONAL SET[1:?] of EventAssignment</td>
</tr>
<tr>
<td>FrozenAssignment</td>
<td>OPTIONAL SET[1:?] of FrozenAssignment</td>
</tr>
<tr>
<td>OrganizationOrPersonInOrganizationAssignment</td>
<td>OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment</td>
</tr>
<tr>
<td>SuppliedObjectRelationship</td>
<td>OPTIONAL SET[1:?] of SuppliedObjectRelationship</td>
</tr>
</tbody>
</table>

**Attribute recommendations:**

- **EventType:** specifies the kind of event. Use ClassString if the value is recommended within this document, otherwise use “Class” template (see [242BO-PAS]). Where applicable, the following values shall be used:
  - 'milestone': the event is a milestone.
  - **Id:** The id attribute specifies the identifier that distinguishes the event. Use “Identifier” template (see [242BO-PAS]). Although the value of this attribute need not be specified, it is recommended to set it in case of a Milestone.

**Preprocessor Recommendations:**

There are no specific Preprocessor recommendations

**Postprocessor Recommendations:**

If not supported by the target system, Milestone Effectivities shall be ignored, or a warning shall be returned.

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

```xml
<Effectivity uid="dveff-M1" xsi:type="n0:DataValidityEffectivity">
  <EffectivityContext uidRef="pc--19088"/>
  <EndDefinition>
    <Event uidRef="ev--2"/>
  </EndDefinition>
  <StartDefinition>
    <Event uidRef="ev--1"/>
  </StartDefinition>
</Effectivity>

<Event uid="ev--1">
  <EventType>
    <ClassString>milestone</ClassString>
  </EventType>
  <Id>
    <Identifier uid="ev--1-id1" id="M1" idRoleRef="rl--ii" idContextRef="o--000000178"/>
  </Id>
</Event>

<Event uid="ev--2">
  <EventType>
    <ClassString>milestone</ClassString>
  </EventType>
  <Id>
    <Identifier uid="ev--2-id1" id="M2" idRoleRef="rl--ii" idContextRef="o--000000178"/>
  </Id>
</Event>
```

Figure 23: Instance Model Milestone Effectivity
4.2.3 Template “LotEffectivity”

The LotEffectivity is a subtype of Effectivity with an additional attribute LotSize. It defines the domain of applicability as a given batch of end items.

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

```
<Effectivity uid="loteff-x-1" xsi:type="n0:LotEffectivity">
  <EffectivityContext uidRef="pc--19088"/>
  <LotSize uid="nv--l2" xsi:type="n0:NumericalValue">
    <Definition>
      <PropertyDefinitionString>lot size property</PropertyDefinitionString>
    </Definition>
    <Unit uidRef="i--1231"/>
    <ValueComponent>35</ValueComponent>
  </LotSize>
</Effectivity>
```

**4.2.3.1 ENTITY LotEffectivity**

Additional attributes, derived from entity effectivity will not be repeated in this table

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>LotSize</td>
<td>ValueWithUnit</td>
</tr>
</tbody>
</table>

*Table 13: "LotEffectivity" Attributes*

**Attribute recommendations:**

- **LotSize:** The LotSize attribute specifies the size of the batch of end items for the manufacturing of which the effectivity applies. The lotsize definition (LotSize.Definition.PropertyDefinitionString) should be set to “lot size property”. A range of lots is not supported. Use NumericalValue template (see [242BO-PAS]).
Preprocessor Recommendations:

It is not recommended to use other attributes of the super object to add effectivity information like StartDefinition or EndDefinition together with LotEffectivity, to avoid rule out each other.

Detailed informations about the lot itself are not supported. The lot will only be represented by the defined Id.

A range of lots is not supported in AP242. If a range is supported by the exporting system, the preprocessor has to multiply the LotEffectivity corresponding to the range.

Postprocessor Recommendations:

If not supported by the target system, LotEffectivities shall be ignored or a warning shall be returned.

4.2.4 Template “SerialEffectivity”

A SerialEffectivity is a type of Effectivity for which the domain of applicability is defined as a possibly open-ended interval of serial numbers.

The Instance Model: AP242 BO Model XML entities and attributes

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

```
<Effectivity uid="sneff-33-2" xsi:type="n0:SerialEffectivity">
  <EffectivityContext uidRef="pc--19088"/>
  <EndNumber>
    <IdentifierString>3</IdentifierString>
  </EndNumber>
  <StartNumber>
    <IdentifierString>3</IdentifierString>
  </StartNumber>
</Effectivity>
```
4.2.4.1 ENTITY SerialEffectivity

Additional attributes, derived from entity effectivity will not be repeated in this table.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>EndNumber</td>
<td>OPTIONAL IdentifierSelect</td>
</tr>
<tr>
<td>StartNumber</td>
<td>IdentifierSelect</td>
</tr>
</tbody>
</table>

Table 14: "SerialEffectivity" Attributes

Attribute recommendations:

- **EndNumber**: The EndNumber attribute identifies the last valid serial number for which the effectivity applies (may be equal to the StartNumber). The value of this attribute need not be specified. If a value for this attribute is not defined, the interval of applicability has no serial number defined as upper bound, i.e., the effectivity is expected to apply to all serial numbers greater than the StartNumber. Use IdentifierString type or "Identifier” template (see [242BO-PAS]).

- **StartNumber**: The StartNumber attribute identifies the first valid serial number for which the effectivity applies. Use IdentifierString type or "Identifier” template (see [242BO-PAS]).

Preprocessor Recommendations:

It is not recommended to use other attributes of the super object to add effectivity information like StartDefinition or EndDefinition together with SerialEffectivity, to avoid rule out each other.

Postprocessor Recommendations:

There are no specific postprocessor recommendations.

4.2.5 Template “TimeIntervalEffectivity”

A TimeIntervalEffectivity is a type of Effectivity for which the domain of applicability is defined as duration.

The Instance Model: AP242 BO Model XML entities and attributes
Figure 26: Instance Model TimeIntervalEffectivity

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

```xml
<Effectivity uid="tieff-20160301P3D" xsi:type="n0:TimeIntervalEffectivity">
  <EffectivityContext uidRef="pc--19088"/>
  <StartDefinition>
    <DateTimeString>2016-12-31T00:00:00</DateTimeString>
  </StartDefinition>
  <EffectivityPeriod>P3D</EffectivityPeriod>
</Effectivity>
```

### 4.2.5.1 ENTITY TimeIntervalEffectivity

Additional attributes, derived from entity effectivity, will not be repeated in this table.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>EffectivityPeriod</td>
<td>xsd:duration</td>
</tr>
<tr>
<td>EndDefinition</td>
<td>OPTIONAL SET[1:?] of EventOrDateSelect</td>
</tr>
<tr>
<td>StartDefinition</td>
<td>OPTIONAL SET[1:?] of EventOrDateSelect</td>
</tr>
</tbody>
</table>

**Table 15: "TimeIntervalEffectivity" Attributes**

**Attribute recommendations:**

- **EffectivityPeriod**: The EffectivityPeriod attribute specifies the time intervall that defines the domain of validity. The possibilities to define values for duration are explained in xsd:duration.
A complex example for a negative duration:
-P3Y6M4DT12H30M17.12345678901234567890S

- **EndDefinition:** Use a negative duration together with EndDefinition (in this case, StartDefinition shall not be set). Use “DateTime” template for DateTimeString without DateTimeAssignment (see [242BO-PAS]).

- **StartDefinition:** Use a positive duration together with StartDefinition (in this case, EndDefinition shall not be set). Use “DateTime” template for DateTimeString without DateTimeAssignment (see [242BO-PAS]).

**Preprocessor Recommendations:**
The usage of TimeIntervalEffectivity is not recommended since not supported by most of the PDM systems.

**Postprocessor Recommendations:**
TimeIntervalEffectivities shall be ignored or a warning shall be returned.

### 4.2.6 Template “ConditionalEffectivity”

The ConditionalEffectivity is a specialization of Effectivity for which the domain of applicability is defined by an explicit condition.

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

![Figure 27: Instance Model ConditionalEffectivity](image)

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

```
<Effectivity uid="condeff-Spec001" xsi:type="n0:ConditionalEffectivity">
  <EffectivityContext uidRef="pc--19088"/>
  <Condition uidRef="spece--19133"/>
</Effectivity>
```

#### 4.2.6.1 ENTITY ConditionalEffectivity

Additional attributes, derived from entity effectivity, will not be repeated in this table.
Table 16: "ConditionalEffectivity" Attributes

**Attribute recommendations:**
- **Condition**: Reference to a condition. See “SpecificationConditionAssignment” template in chapter 0.

**Preprocessor Recommendations:**
The use of OneOfCondition is not recommended, since not supported by most of the PDM systems.

It is not recommended to use other attributes of the super object to add effectivity information like StartDefinition or EndDefinition together with ConditionalEffectivity, to avoid rule out each other.

It should not be used to parametrize a product configuration by EffectivityAssignment. In this case, rather use ProductConfiguration.definingSpecifications.

**Postprocessor Recommendations:**
There are no specific postprocessor recommendations.

4.2.6.2 ENTITY ConditionalConfiguration
A ConditionalConfiguration adds usage information to a ConditionalEffectivity so that it may be evaluated in combination with further ConditionalEffectivities.

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

![Figure 28: Instance Model ConditionalConfiguration](image-url)
The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```xml
<Effectivity uid="condconf-Spec001" xsi:type="n0:ConditionalConfiguration">
    <EffectivityContext uidRef="pc--19088"/>
    <Condition uidRef="spece--19133"/>
    <ConfigurationType>
        <ClassString>usage</ClassString>
    </ConfigurationType>
    <InheritanceType>
        <ClassString>local</ClassString>
    </InheritanceType>
</Effectivity>
```

Additional attributes, derived from entity ConditionalEffectivity, will not be repeated in this table.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConfigurationType</td>
<td>ClassSelect</td>
</tr>
<tr>
<td>InheritanceType</td>
<td>ClassSelect</td>
</tr>
</tbody>
</table>

Table 17: "ConditionalConfiguration" Attributes

**Attribute recommendations:**

- **configurationType:** the valid usage of a ConditionalConfiguration object that is applied to the business object as configuredElement. Use ClassString if the value is recommended within this document, otherwise use “Class” template (see [242BO-PAS]). According to the ISO AP242 Specification, where applicable, the following values shall be used:
  - 'design': the object referenced as configuredElement has to be designed and verified before it can actually be used in a given context. This context is specified by the SpecificationConditionAssignment and SpecificationAssignment objects referenced as the resolvedConfiguration. This type of ConditionalConfiguration is applicable for AlternativeSolution or BreakdownElement objects
  - 'usage': the object referenced as the configuredElement is controlled by a ConditionalConfiguration. The SpecificationConditionAssignment and SpecificationAssignment objects specify the usage cases and are referenced as the resolvedConfiguration. This type of ConditionalConfiguration is applicable for AlternativeSolution, Occurrence or PhysicalBreakdown objects

- **Examples:**
  - The presence of an optional third rear axle of a truck is controlled by specifications on the level of PhysicalBreakdown.
  - A 'sunroof' has to be provided, i.e., designed for a class of vehicles that has an optional 'sunroof' as part of its specification.

- **inheritanceType:** specifies whether or not an inheritance scheme for the configuration information in a hierarchical structure is applied to the business object referenced as the configuredElement. The levels within such a hierarchy are defined through BreakdownElementRealization objects or the attribute baseElement of AlternativeSolution. Use ClassString if the value is recommended within this document, otherwise use “Class” template (see [242BO-PAS]). According to the ISO AP242 Specification, where applicable, the following values shall be used:
• 'exception': No inheritance scheme is applicable and all required configuration information must be attached locally at the business object. The value indicates that the configuration information may be inconsistent to the structural levels above it or that it is, on purpose, contradictory to it. Such a condition implies that an inheritance scheme shall not continue beyond this point in the product structure tree.

• 'inherited': A scheme for inheritance of configuration information applies. The complete configuration information shall be collected from the different levels in the structure by evaluation of results. The results shall be evaluated using the logical AND to combine configuration information starting at the referenced configuredElement and using the logical OR to combine alternatives. In addition, this evaluation shall consider related effectivity information. 'inherited' only applies for objects for which the same value of ConditionalConfigurationType is defined.

• Examples:
  
  • A situation where the inheritanceType 'exception' is applicable is a technical solution released for one particular customer without support in higher structure levels.
  
  • The following figure shows how inheritance is applied along the tree of a product structure.

• Picture 16 Inheritance along product structure tree

The complete configuration information of a Occurrence can be obtained by adding any such information to AlternativeSolution objects which are linked through BreakdownElementRealization objects and baseElement attributes respectively. Whenever more than one higher level instance is present, the current information available is branched in as many branches as instances are present. For example, the total configuration information for the Occurrence given in the figure above could be expressed as follows: (C0.0 AND C1.0 AND C2.0) OR (C0.0 AND C1.1 AND C2.1) OR (C0.0 AND C1.1 AND C2.2).

• 'local': No inheritance scheme is applicable and all required configuration information must be attached locally at the business object. Nevertheless, any potentially inherited configuration information of a higher level shall be consistent, i.e., it shall be a subset of the locally defined configuration information.
Preprocessor Recommendations:
The same Preprocessor Recommendations than Effectivity are valid (see “Effectivity template” 4.2.1).

Postprocessor Recommendations:
The same Postprocessor Recommendations than Effectivity are valid (see “Effectivity template” 4.2.1).

4.2.7 Template “EffectivityAssignment”
An EffectivityAssignment is an object that associates effectivity with the product or activity data whose effectivity is controlled by the associated Effectivity.

It is recommended to use EffectivityAssignments for following Entities:

- PartVersion (so-called revision effectivities)
- NextAssemblyOccurrenceUsage (so-called occurrence effectivities)
- ProductConfiguration (to define the configuration)

Rules:

- In case of serial/timeinterval/date range effectivities, the PartVersion or NextAssemblyOccurrenceUsage is valid for the configuration if the effectivity assigned to the configuration (single value of range) is within the given range.
- In case of ConditionalEffectivities, the boolsche expression shall be evaluated for all of:
  - the serial/timeinterval/date/lot effectivities assigned to the Configuration
  - the Specifications assigned to the Configuration via its attribute definingSpecifications
- Special cases:
  - The PartVersion is valid for the Configuration, if both have an assignment to the same effectivity

Figure 29: Simplified Example for supported EffectivityAssignments
The NextAssemblyOccurrenceUsage is valid for the Configuration, if both have an assignment to the same effectivity.

4.2.7.1 EffectivityAssignment for PartVersion
If the effectivity of a part is always the same for all usages of the part, Effectivities may be assigned to the PartVersions of the part structure definition, to define the validity for assembly structure links.

The Instance Model: AP242 BO Model XML entities and attributes

![Instance Model EffectivityAssignment for PartVersions](image)

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

```xml
<Part uid="p--0000000017D374A0">
  ...
  <Versions>
    <PartVersion uid="pv--0000000017D374A0--id1">
      ...
      <EffectivityAssignment uid="ea--19215v">
        <AssignedEffectivity uidRef="condeff-Spec001"/>
        <EffectivityIndication>true</EffectivityIndication>
        <Role>
          <ClassString>actual</ClassString>
        </Role>
      </EffectivityAssignment>
      ...
    </PartVersion>
  </Versions>
</Part>
```
4.2.7.2 EffectivityAssignment for Assembly Links

If the effectivity of a part may be different, depending on each usage of the part, it is recommended to assign Effectivities from the Entity NextAssemblyOccurrenceUsage of the part structure definition, to define the validity for assembly structure links.

The Instance Model: AP242 BO Model XML entities and attributes

![Instance Model EffectivityAssignment for assembly links]

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

```xml
<Part uid="p--0000000017D36260">
  <Id>
    ...
  </Id>
  <Versions>
    <PartVersion uid="pv--0000000017D36260--id8">
      ...
    </PartVersion>
    <View>
      <PartView xsi:type="n0:AssemblyDefinition" uid="pvv--0000000017D36260--id8">
        ...
        <ViewOccurrenceRelationship uid="pvvid--000000001EA37D20--20" xsi:type="n0:NextAssemblyOccurrenceUsage">
          <Related uidRef="pi--000000001EA37D20--20"/>
          <RelationType>
            <ClassString>next assembly occurrence</ClassString>
          </RelationType>
          <EffectivityAssignment uid="ea--19218">
            <AssignedEffectivity uidRef="dveff-20161231"/>
            <EffectivityIndication>true</EffectivityIndication>
            <Role>
              <ClassString>actual</ClassString>
            </Role>
          </EffectivityAssignment>
          <Placement>
            <CartesianTransformation uid="cto--000000001EA37D20--20"/>
          </Placement>
        </ViewOccurrenceRelationship>
      </PartView>
    </View>
  </Versions>
</Part>
```

© PDM Implementor Forum
4.2.7.3 EffectivityAssignment for Configurations

It is recommended to assign effectivities to the configurations to define the lot/serial/date validity of the links dependent on the selected configuration.

The specifications that shall apply to the configuration shall be mapped in DefiningSpecifications.

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

![Instance Model EffectivityAssignment for configurations](image)

*Figure 32: Instance Model EffectivityAssignment for configurations*

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

```
<ProductConfiguration uid="pconf--19089">
  <Description>
    <CharacterString>as1 short description</CharacterString>
  </Description>
  <Id>
    <Identifier uid="pc-as1short--id1" id="as1 short" idRoleRef="rl--ii" idContextRef="o--000000178"/>
  </Id>
</ProductConfiguration>
```
4.2.7.4 ENTITY EffectivityAssignment

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssignedEffectivity</td>
<td>Effectivity</td>
</tr>
<tr>
<td>ClassifiedAs</td>
<td>OPTIONAL SET[1:?] of Classification</td>
</tr>
<tr>
<td>Description</td>
<td>OPTIONAL DescriptorSelect</td>
</tr>
<tr>
<td>EffectivityIndication</td>
<td>BOOLEAN</td>
</tr>
<tr>
<td>Id</td>
<td>OPTIONAL Identifier</td>
</tr>
<tr>
<td>Role</td>
<td>ClassSelect</td>
</tr>
<tr>
<td>ActivityAssignment</td>
<td>OPTIONAL SET[1:?] of ActivityAssignment</td>
</tr>
<tr>
<td>ActivityMethodAssignment</td>
<td>OPTIONAL SET[1:?] of ActivityMethodAssignment</td>
</tr>
<tr>
<td>ApprovalAssignment</td>
<td>OPTIONAL SET[1:?] of ApprovalAssignment</td>
</tr>
<tr>
<td>AssignmentObjectRelationship</td>
<td>OPTIONAL SET[1:?] of AssignmentObjectRelationship</td>
</tr>
<tr>
<td>ConditionAssignment</td>
<td>OPTIONAL SET[1:?] of ConditionAssignment</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>DateAndPersonAssignment</td>
<td>OPTIONAL SET[1:?] of DateAndPersonAssignment</td>
</tr>
<tr>
<td>DateTimeAssignment</td>
<td>OPTIONAL SET[1:?] of DateTimeAssignment</td>
</tr>
<tr>
<td>DocumentAssignment</td>
<td>OPTIONAL SET[1:?] of DocumentAssignment</td>
</tr>
<tr>
<td>EffectivityAssignment</td>
<td>OPTIONAL SET[1:?] of EffectivityAssignment</td>
</tr>
<tr>
<td>EventAssignment</td>
<td>OPTIONAL SET[1:?] of EventAssignment</td>
</tr>
<tr>
<td>FrozenAssignment</td>
<td>OPTIONAL SET[1:?] of FrozenAssignment</td>
</tr>
<tr>
<td>ModelPropertyAssignment</td>
<td>OPTIONAL SET[1:?] of ModelPropertyAssignmen t</td>
</tr>
<tr>
<td>OrganizationOrPersonInOrganizationAssig nment</td>
<td>OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment</td>
</tr>
<tr>
<td>PropertyDefinitionAssignment</td>
<td>OPTIONAL SET[1:?] of PropertyDefinitionAssignment</td>
</tr>
<tr>
<td>PropertyValueAssignment</td>
<td>OPTIONAL SET[1:?] of PropertyValueAssignment</td>
</tr>
<tr>
<td>TimeIntervalAssignment</td>
<td>OPTIONAL SET[1:?] of TimeIntervalAssignmen t</td>
</tr>
</tbody>
</table>

Table 18: "EffectivityAssignment" Attributes

Attribute recommendations:

- **AssignedEffectivity**: the instance of Effectivity that is assigned. Use “Effectivity” template in chapters 4.2.2 to 4.2.6.

- **Description**: the text or the set of texts that provide further information about the EffectivityAssignment. The value of this attribute need not be specified. Use "Description" template (see [242BO-PAS]).

- **EffectivityIndication**: the indication whether the assignedEffectivity defines a period of effectivity (value 'TRUE') or a period of ineffectivity (value 'FALSE') for the assignedTo objects. In the first case, the use of assignedTo objects is or was valid during the considered period. In the second case, the use of assignedTo objects is or was not valid during the considered period. The use of 'TRUE' is recommended.

- **Role**: the meaning of the assignment. Use ClassString if the value is recommended within this document, otherwise use “Class” template (see [242BO-PAS]). According to the ISO AP242 Specification, where applicable, the following values shall be use:
  - 'actual': The actual period during which the assignedEffectivity lasted: EffectivityAssignments or PartVersions (see chapter 4.2.7.1) and Assembly Links (see chapter 4.2.7.2)
• 'configuration': The effectivity (mostly one single serial number, date or a list of specifications) used to define a product variant: EffectivityAssignments for ProductConfiguration (see chapter 4.2.7.3)
• 'planned': The period associated with the assignedEffectivity defines a planned period of time during which the assignedTo objects are or were supposed to be effective;
• 'required': The AssignedTo objects shall be kept effective for this period.
• 'context': This value means that the AssignedTo object represent the context of the AssignedEffectivity (for systems not supporting Product Context elements). The AssignedTo object shall be a PartVersion.

**Preprocessor Recommendations:**

- If more than one configuration is used within the product concept, the configurations has to be assigned with different effectivities to define the filters for the configuration
- It is recommended to assign at most:
  - One LotEffectivity xor
    - One SerialEffectivity with StartNumber=EndNumber and/or
    - One DataValidityEffectivity with StartDefinition=EndDefinition
  - And/or one or many Specifications (via the attribute definingSpecifications)
- to a ProductConfiguration, since this is the way most of the PDM systems support ProductConfigurations
- Some PDM systems support the definition of serial/date ranges for ProductConfigurations: doing so, a >100% product structure gets selected (for overview purpose).
- If a PDM system (like VPM V4) interpretes a PartVersion or PartOccurrence without any effectivity as never valid, a ConditionalEffectivity containing an Specification with id="/DUMMY" shall be mapped during export => in the target systems, not selecting the "/DUMMY" specification will cause these PartVersions or PartOccurrences to be never selected.
- It is not recommended to reuse an Effectivity for several PartVersions, for several Assembly Links or for several ProductConfigurations
- If the source system does not support EffectivityAssignment.Role, it should be set to ‘actual’.
- Usually, the BOM consists either of EffectivityAssignment for Occurrences xor EffectivityAssignment for PartVersions

**Postprocessor Recommendations:**

- If no effectivities are assigned to the assembly structure or only one configuration is defined without any effectivity assignments to the configuration, all links with all effectivities within the data structure have to be considered as valid.
- If the receiving system does not support the EffectivityAssignment for PartVersions, it has to be ignored or a warning shall be returned.
- If the receiving system does not support the EffectivityAssignment for Occurrences, it has to be ignored or a warning shall be returned.
- It is not recommended to map EffectivityAssignment for PartVersions as EffectivityAssignment for Occurrences or vice versa.
• If the receiving system does not support effectivityIndication=FALSE, it shall negate the effectivity as if it came with effectivityIndication=TRUE and being embedded into a Non-EqualsCondition effectivity.

• If the target system does not support EffectivityAssignment.Role, it should be ignored or a warning shall be returned.

• Since not recommended, reuse of an Effectivity for several PartVersions, for several Assembly Links or for several ProductConfigurations, has to be mapped to a distinct Effectivity in the target system.

4.2.8 General Handling of Effectivities

There are different possibilities to assign effectivity information to the product. This chapter will give an overview of the possibilities with the example to define the validity of an assembly structure link. The diagrams are not compliant to the instantiation diagrams defined in chapter 1.1.7 and are strongly reduced to the minimum information needed to show the dependencies between the units.

<table>
<thead>
<tr>
<th>Simplified Diagrams</th>
<th>Recommended interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="" alt="Diagram 1" /></td>
<td>The assembly structure link or PartVersion is always valid. See special handling of PDM systems that assume the contrary. There is no effectivity assigned</td>
</tr>
<tr>
<td><img src="" alt="Diagram 2" /></td>
<td>The assembly structure link or PartVersion is valid if the effectivity #1 is valid. The effectivity could be one of the introduced effectivities except for the ConditionalEffectivity</td>
</tr>
<tr>
<td><img src="" alt="Diagram 3" /></td>
<td>The assembly structure link or PartVersion is valid if the effectivity #1 or effectivity #2 is valid. If more than one effectivity is assigned, it has to be handled as an implicit OR operation between all of them.</td>
</tr>
</tbody>
</table>
It is not recommended to use EffectivityAssignment and EffectivityRelationship together for assigning the additional effectivity information of Effectivity#2.

Combined Effectivities

Combined Effectivities with ConditionalEffectivities are supported in the TC patch schema (see Bugzilla issue #5086).
4.3 **Product Specification**

4.3.1 Template “Specification”

A Specification shall exist only within the context of a ProductConceptIdentification (via SpecificationAssignment) to define its ‘availability’ within one (or several) ProductClass.

Additionally, they are assigned to ProductConfigurations in which they influence if the Effectivities on PartVersion/Occurrence are valid or not for this ProductConfiguration.

The specifications define the characteristics of the product and discriminate one product from other members of the same ProductClass.

A specification belongs to a SpecificationCategory that completes the semantics of the specification.

---

**Figure 33:** Specifications to define configurations

---

**Figure 34:** Specifications defined as available for configuration given ProductClass
The Instance Model: AP242 BO Model XML entities and attributes

Figure 35: Instance Model Specification

The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)
<ProductConcept uid="pc--19088" xsi:type="n0:ProductClass">
  <Description>
    <CharacterString>as class description</CharacterString>
  </Description>
  <Id>
    <Identifier uid="pc-asclass--id1" id="as class" idRoleRef="rl--ii" idContextRef="o--000000178"/>
  </Id>
  <Name>
    <CharacterString>as class name</CharacterString>
  </Name>
  <LevelType>
    <ClassString>platform</ClassString>
  </LevelType>
  <VersionId id="A.1"/>
  <SpecificationAssignment uid="csa--19126">
    <AssignedSpecification uidRef="spec--19191"/>
    <AssociationType>
      <ClassString>availability</ClassString>
    </AssociationType>
  </SpecificationAssignment>
  <SpecificationAssignment uid="csa--19127">
    <AssignedSpecification uidRef="spec--19192"/>
    <AssociationType>
      <ClassString>availability</ClassString>
    </AssociationType>
  </SpecificationAssignment>
  <SpecificationCategoryAssignment uid="cca--19124">
    <AssignedCategory uidRef="specc--19123"/>
    <Mandatory>true</Mandatory>
  </SpecificationCategoryAssignment>
</ProductConcept>

<Specification uid="spec--19191">
  <Category uidRef="specc--19123"/>
  <Description>
    <CharacterString>Spec001 description</CharacterString>
  </Description>
  <Id>
    <Identifier uid="proj--INM8C--id1" id="Spec001" idRoleRef="rl--ii" idContextRef="o--000000178"/>
  </Id>
  <Name>
    <CharacterString>Spec001 name</CharacterString>
  </Name>
  <Package>false</Package>
  <VersionId id="A.1"/>
</Specification>

<Specification uid="spec--19192">
  <Category uidRef="specc--19123"/>
  <Description>
    <CharacterString>Spec002 description</CharacterString>
  </Description>
  <Id>
    <Identifier uid="proj--INM9C--id1" id="Spec002" idRoleRef="rl--ii" idContextRef="o--000000178"/>
  </Id>
  <Name>
    <CharacterString>Spec002 name</CharacterString>
  </Name>
  <Package>false</Package>
  <VersionId id="A.1"/>
4.3.1.1 ENTITY Specification

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassifiedAs</td>
<td>OPTIONAL SET[1:?] of Classification</td>
</tr>
<tr>
<td>Description</td>
<td>OPTIONAL DescriptorSelect</td>
</tr>
<tr>
<td>Id</td>
<td>IdentifierSelect</td>
</tr>
<tr>
<td>Name</td>
<td>OPTIONAL DescriptorSelect</td>
</tr>
<tr>
<td>Category</td>
<td>Reference to SpecificationCategory</td>
</tr>
<tr>
<td>Package</td>
<td>BOOLEAN</td>
</tr>
<tr>
<td>VersionId</td>
<td>OPTIONAL IdentifierSelect</td>
</tr>
<tr>
<td>ActivityAssignment</td>
<td>OPTIONAL SET[1:?] of ActivityAssignment</td>
</tr>
<tr>
<td>ApprovalAssignment</td>
<td>OPTIONAL SET[1:?] of ApprovalAssignment</td>
</tr>
<tr>
<td>DateAndPersonAssignment</td>
<td>OPTIONAL SET[1:?] of DateAndPersonAssignment</td>
</tr>
<tr>
<td>DateTimeAssignment</td>
<td>OPTIONAL SET[1:?] of DateTimeAssignment</td>
</tr>
<tr>
<td>DocumentAssignment</td>
<td>OPTIONAL SET[1:?] of DocumentAssignment</td>
</tr>
<tr>
<td>EffectivityAssignment</td>
<td>OPTIONAL SET[1:?] of EffectivityAssignment</td>
</tr>
<tr>
<td>EventAssignment</td>
<td>OPTIONAL SET[1:?] of EventAssignment</td>
</tr>
<tr>
<td>FrozenAssignment</td>
<td>OPTIONAL SET[1:?] of FrozenAssignment</td>
</tr>
</tbody>
</table>
Table 19: "Specification" Attributes

Attribute recommendations:

- **Description**: the text by which the specification is described. The value of this attribute need not be specified. Use “Description” template (see [242BO-PAS]).

- **Id**: the identifier that distinguishes the specification. Use “Identifier” template (see [242BO-PAS]).

- **Name**: the text by which the specification is known. The value of this attribute need not be specified. Use “Description” template (see [242BO-PAS]).

- **Category**: the SpecificationCategory that completes the semantics of the specification (see template “Category” 4.3.1.2).
  
  Comment: an issue in Bugzilla has been created under #6288 to embed Specification into SpecificationCategory rather than defining it as a BaseRootObject. Then the Specification.Id may also be mapped to an IdentifierString (since the context is given by the SpecificationCategory).

- **Package**: specifies whether this specification represents a package of specification objects or not. Such a specification combines those specification objects that shall be offered to the market as a set. In the case where package’s t’ue, there shall be exactly one SpecificationInclusion per ProductClass considered, that refers to this Specification by its attribute ifCondition. The Specification objects that are members of the package, shall be specified in the attribute includedSpecification of the SpecificationInclusion.

- **VersionId**: the identification or set of identifications of a particular version of the Specification. The value of this attribute need not be specified. Use IdentifierString type or “Identifier” template (see [242BO-PAS]).
Preprocessor Recommendations:
Since SpecificationInclusion is not yet in scope of this document, at the moment it is recommended to set Package to ‘false’ and to mention all specifications in the effectivities (no specification will be implicitly included).

Postprocessor Recommendations:
Ignore at the moment the value of Package and any related SpecificationInclusion.

4.3.1.2 ENTITY SpecificationCategory
A SpecificationCategory is a definition of a set of specification objects serving the same purpose (for example all engines that may be alternatively chosen (one and only one) radio specifications that may be alternatively chosen (zero or one), or all confort specifications that may be ordered (zero to many).

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassifiedAs</td>
<td>OPTIONAL SET[1:?] of Classification</td>
</tr>
<tr>
<td>Description</td>
<td>OPTIONAL DescriptorSelect</td>
</tr>
<tr>
<td>Id</td>
<td>IdentifierSelect</td>
</tr>
<tr>
<td>ImplicitExclusiveCondition</td>
<td>BOOLEAN</td>
</tr>
<tr>
<td>SpecificationCategoryHierarchy</td>
<td>OPTIONAL SET[1:?] of SpecificationCategoryHierarchy</td>
</tr>
<tr>
<td>ActivityAssignment</td>
<td>OPTIONAL SET[1:?] of ActivityAssignment</td>
</tr>
<tr>
<td>ApprovalAssignment</td>
<td>OPTIONAL SET[1:?] of ApprovalAssignment</td>
</tr>
<tr>
<td>DateAndPersonAssignment</td>
<td>OPTIONAL SET[1:?] of DateAndPersonAssignment</td>
</tr>
<tr>
<td>DateTimeAssignment</td>
<td>OPTIONAL SET[1:?] of DateTimeAssignment</td>
</tr>
<tr>
<td>DocumentAssignment</td>
<td>OPTIONAL SET[1:?] of DocumentAssignment</td>
</tr>
<tr>
<td>EffectivityAssignment</td>
<td>OPTIONAL SET[1:?] of EffectivityAssignment</td>
</tr>
<tr>
<td>EventAssignment</td>
<td>OPTIONAL SET[1:?] of EventAssignment</td>
</tr>
<tr>
<td>FrozenAssignment</td>
<td>OPTIONAL SET[1:?] of FrozenAssignment</td>
</tr>
<tr>
<td>InformationUsageRightAssignment</td>
<td>OPTIONAL SET[1:?] of InformationUsageRightAssignment</td>
</tr>
</tbody>
</table>
Table 20: "SpecificationCategory" Attributes

**Attribute recommendations:**

- **Description:** the text by which the SpecificationCategory is described. The value of this attribute need not be specified. Use “Description” template (see [242BO-PAS]).
- **Id:** the identifier that distinguishes the SpecificationCategory. Use “Identifier” template (see [242BO-PAS]).
- **ImplicitExclusiveCondition:** specifies whether the specification objects within the SpecificationCategory are mutually exclusive for the production of one particular product. A value ‘true’ indicates that the referenced objects are mutually exclusive for the production of the particular product.

**Preprocessor Recommendations:**
If the source system does not support ImplicitExclusiveCondition, it should be set to ‘false’.

**Postprocessor Recommendations:**
If the target system does not support ImplicitExclusiveCondition, it should be ignored, or a warning shall be returned.

### 4.3.2 Template “SpecificationAssignment”
A SpecificationAssignment exists only within the context of one ProductConcept to define the meaning of a specification within a ProductClass.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssignedSpecification</td>
<td>Specification</td>
</tr>
<tr>
<td>AssociationType</td>
<td>ClassSelect</td>
</tr>
<tr>
<td>ClassifiedAs</td>
<td>OPTIONAL SET[1:?] of Classification</td>
</tr>
<tr>
<td>Description</td>
<td>OPTIONAL DescriptorSelect</td>
</tr>
<tr>
<td>ActivityAssignment</td>
<td>OPTIONAL SET[1:?] of ActivityAssignment</td>
</tr>
<tr>
<td>ActivityMethodAssignment</td>
<td>OPTIONAL SET[1:?] of ActivityMethodAssignment</td>
</tr>
<tr>
<td>ApprovalAssignment</td>
<td>OPTIONAL SET[1:?] of ApprovalAssignment</td>
</tr>
<tr>
<td>AssignmentObjectRelationship</td>
<td>OPTIONAL SET[1:?] of AssignmentObjectRelationship</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>ConditionAssignment</td>
<td>OPTIONAL SET[1:?] of ConditionAssignment</td>
</tr>
<tr>
<td>DateAndPersonAssignment</td>
<td>OPTIONAL SET[1:?] of DateAndPersonAssignment</td>
</tr>
<tr>
<td>DateTimeAssignment</td>
<td>OPTIONAL SET[1:?] of DateTimeAssignment</td>
</tr>
<tr>
<td>DocumentAssignment</td>
<td>OPTIONAL SET[1:?] of DocumentAssignment</td>
</tr>
<tr>
<td>EffectivityAssignment</td>
<td>OPTIONAL SET[1:?] of EffectivityAssignment</td>
</tr>
<tr>
<td>EventAssignment</td>
<td>OPTIONAL SET[1:?] of EventAssignment</td>
</tr>
<tr>
<td>FrozenAssignment</td>
<td>OPTIONAL SET[1:?] of FrozenAssignment</td>
</tr>
<tr>
<td>InformationUsageRightAssignment</td>
<td>OPTIONAL SET[1:?] of InformationUsageRightAssignment</td>
</tr>
<tr>
<td>ModelPropertyAssignment</td>
<td>OPTIONAL SET[1:?] of ModelPropertyAssignment</td>
</tr>
<tr>
<td>OrganizationOrPersonInOrganizatiAssignment</td>
<td>OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment</td>
</tr>
<tr>
<td>PropertyDefinitionAssignment</td>
<td>OPTIONAL SET[1:?] of PropertyDefinitionAssignment</td>
</tr>
<tr>
<td>PropertyValueAssignment</td>
<td>OPTIONAL SET[1:?] of PropertyValueAssignment</td>
</tr>
<tr>
<td>TimeIntervalAssignment</td>
<td>OPTIONAL SET[1:?] of TimeIntervalAssignment</td>
</tr>
</tbody>
</table>

**Table 21: "SpecificationAssignment" Attributes**

**Attribute recommendations:**

- **AssignedSpecification**: the Specification that is associated with the ProductClass (see Template “Specification”).
- **AssociationType**: the kind of availability of a particular specification in a ProductClass. Use ClassString if the value is recommended within this document, otherwise use “Class” template (see [242BO-PAS]). According to the ISO AP242 Specification, where applicable, the following values shall be used:
  - ‘availability’: The specification is a potential characteristic of any product belonging to a high level ProductClass. It is not specified if this is an option or a standard. This builds the option pool as defined in chapter 4.4.
EXAMPLE: ‘front wheel drive’ or ‘four-wheel drive’ are available for a product class ‘Voyager’

- **identification**: The specification is a characteristic that enables to distinguish the associated ProductClass from other ProductClass objects. This is a kind of ‘non replaceable standard’. This value is not applicable for a top level node in a hierarchy of ProductClass objects. This identification is part of the identification of all sub classes of this ProductClass;
  
  EXAMPLE: The car body type ‘limousine’ is identifying product class ‘S40’.

- **non replaceable standard**: The specification is a characteristic of all products belonging to the ProductClass;
  
  EXAMPLE: ‘South East Asia climate zone’ is a ‘non-replaceable standard’ for a car that is produced for use in that particular geographic area.

- **option**: The specification is a characteristic of a product if explicitly chosen. The Specification replaces another Specification of the same SpecificationCategory if the replaced specification is associated with the ProductClass as ‘replaceable standard’;
  
  EXAMPLE: ‘sunroof’ or ‘radio with CD’ are common options proposed to the customer.

- **part usage**: The specification is a characteristic for the usage of the components of an AlternativeSolution, the usage of an Occurrence, or for the application of a ProcessPlan or a ProcessOperationOccurrence in the products of the associated ProductClass;

- **replaceable standard**: The specification is a default characteristic of the products belonging to the ProductClass as long as no other Specification of the same SpecificationCategory is chosen.

**Description**: the text by which the SpecificationAssignment is described. The value of this attribute need not be specified. Use “Description” template (see [242BO-PAS]).

*Preprocessor Recommendations:*

Until further use cases are in scope of this document, use SpecificationAssignment only with AssociationType=‘availability’.

*Postprocessor Recommendations:*

Ignore at the moment any SpecificationAssociations not having AssociationType= ‘availability’.

### 4.3.3 Template “SpecificationCategoryAssignment”

A SpecificationCategoryAssignment exists only within the context of one ProductConcept to define the meaning of a SpecificationCategory within a ProductClass.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssignedCategory</td>
<td>Specification</td>
</tr>
<tr>
<td>ClassifiedAs</td>
<td>OPTIONAL SET[1:?] of Classification</td>
</tr>
<tr>
<td>Description</td>
<td>OPTIONAL DescriptorSelect</td>
</tr>
<tr>
<td>Mandatory</td>
<td>BOOLEAN</td>
</tr>
<tr>
<td>ActivityAssignment</td>
<td>OPTIONAL SET[1:?] of ActivityAssignment</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>ActivityMethodAssignment</td>
<td>OPTIONAL SET[1:?] of ActivityMethodAssignment</td>
</tr>
<tr>
<td>ApprovalAssignment</td>
<td>OPTIONAL SET[1:?] of ApprovalAssignment</td>
</tr>
<tr>
<td>AssignmentObjectRelationship</td>
<td>OPTIONAL SET[1:?] of AssignmentObjectRelationship</td>
</tr>
<tr>
<td>ConditionAssignment</td>
<td>OPTIONAL SET[1:?] of ConditionAssignment</td>
</tr>
<tr>
<td>DateAndPersonAssignment</td>
<td>OPTIONAL SET[1:?] of DateAndPersonAssignment</td>
</tr>
<tr>
<td>DateTimeAssignment</td>
<td>OPTIONAL SET[1:?] of DateTimeAssignment</td>
</tr>
<tr>
<td>DocumentAssignment</td>
<td>OPTIONAL SET[1:?] of DocumentAssignment</td>
</tr>
<tr>
<td>EventAssignment</td>
<td>OPTIONAL SET[1:?] of EventAssignment</td>
</tr>
<tr>
<td>FrozenAssignment</td>
<td>OPTIONAL SET[1:?] of FrozenAssignment</td>
</tr>
<tr>
<td>InformationUsageRightAssignment</td>
<td>OPTIONAL SET[1:?] of InformationUsageRightAssignment</td>
</tr>
<tr>
<td>ModelPropertyAssignment</td>
<td>OPTIONAL SET[1:?] of ModelPropertyAssignment</td>
</tr>
<tr>
<td>OrganizationOrPersonInOrganiza- tionAssignment</td>
<td>OPTIONAL SET[1:?] of OrganizationOrPersonInOrganizationAssignment</td>
</tr>
<tr>
<td>PropertyDefinitionAssignment</td>
<td>OPTIONAL SET[1:?] of PropertyDefinitionAssignment</td>
</tr>
<tr>
<td>PropertyValueAssignment</td>
<td>OPTIONAL SET[1:?] of PropertyValueAssignment</td>
</tr>
<tr>
<td>TimeIntervalAssignment</td>
<td>OPTIONAL SET[1:?] of TimeIntervalAssignment</td>
</tr>
</tbody>
</table>

**Table 22: "SpecificationCategoryAssignment" Attributes**

**Attribute recommendations:**

- **AssignedCategory**: the SpecificationCategory that is associated with the ProductClass (see Template “Specification”).
- **Description**: the text by which the SpecificationAssignment is described. The value of this attribute need not be specified. Use “Description” template (see [242BO-PAS]).
• **Mandatory**: specifies whether the Specification objects referring to the associated SpecificationCategory have to be used or may be used (optional) for products within the referenced ProductClass. A value of ‘true’ indicates that the usage is mandatory.

• Example: the SpecificationCategory ‘radio’ may be associated 'optional' to the ProductClass of a car; the SpecificationCategory ‘engine’ is an example for a ‘mandatory’ association.

**Preprocessor Recommendations:**
If the source system does not support Mandatory, it should be set to ‘false’.

Consistency between the SpecificationCategoryAssignments and the SpecificationAssignments: the SpecificationCategory and its Specifications should be associated to the same ProductClass(es).

If the source system allows to associate a SpecificationCategory to more than one ProductClass, then multiple SpecificationCategoryAssignments may use the same SpecificationCategory.

**Postprocessor Recommendations:**
If the target system does not support Mandatory, it should be ignored, or a warning shall be returned.

If the target system does not support the reuse of the same SpecificationCategory in multiple ProductClasses, then they shall be duplicated in each ProductClass.

**4.3.4 Template “SpecificationConditionAssignment”**
A SpecificationConditionAssignment exists only within the context of one ProductConcept.

It is used to restrict the scope of ConditionalEffectivities to a given ProductClass.

*An issue in Bugzilla has been created under #5086 to enable to combine multiple effectivities using ConditionalEffectivities (rather than EffectivityRelationship).*

If the scope shall apply to many ProductClasses, any higher level ProductClass related directly or indirectly by a ProductClassRelationship with a relationType ‘hierarchy’ shall used as scope.

Supported (and recommended) conditions are:

• AndCondition
• OrCondition
• NotCondition
• EqualsCondition
  The usage of these conditions is not recommended:
• OneOfCondition
• NotEqualsCondition

**4.3.4.1 Example for an AND Condition**
The following diagram will show an example how to add an AND-Condition for two effectivities or specifications.
The **Instance Model**: AP242 BO Model XML entities and attributes

**Figure 36**: Example for an AND Condition for specifications

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

```xml
<ProductConcept uid="pc--19088" xsi:type="n0:ProductClass">

...  
```
More complex example to map the following condition:

(Shimano AND (NOT Recourbe))

OR (Selle_en_mousse AND Bequille AND (NOT MAgura) AND (NOT Recourbe))

OR (Droit AND (NOT Magura)
</Name>

...<SpecificationConditionAssignment uid="SCA__27">
  <AssignedCondition uid="AC__28" xsi:type="n0:OrCondition">
    <ConditionType>
      <ClassString>or</ClassString>
    </ConditionType>
    <Parameters>
      <Condition uidRef="AC__30"/>
      <Condition uidRef="AC__50"/>
      <Condition uidRef="AC__64"/>
    </Parameters>
  </AssignedCondition>
  <ConditionType>
    <ClassString>or</ClassString>
  </ConditionType>
</SpecificationConditionAssignment>

<SpecificationConditionAssignment uid="SCA__29">
  <AssignedCondition uid="AC__30" xsi:type="n0:AndCondition">
    <ConditionType>
      <ClassString>and</ClassString>
    </ConditionType>
    <Parameters>
      <Condition uidRef="AC__38"/>
      <Condition uidRef="AC__40"/>
    </Parameters>
  </AssignedCondition>
  <ConditionType>
    <ClassString>and</ClassString>
  </ConditionType>
</SpecificationConditionAssignment>

<SpecificationConditionAssignment uid="SCA__49">
  <AssignedCondition uid="AC__50" xsi:type="n0:AndCondition">
    <ConditionType>
      <ClassString>and</ClassString>
    </ConditionType>
    <Parameters>
      <Condition uidRef="AC__55"/>
      <Condition uidRef="AC__57"/>
    </Parameters>
  </AssignedCondition>
  <ConditionType>
    <ClassString>and</ClassString>
  </ConditionType>
</SpecificationConditionAssignment>

<SpecificationConditionAssignment uid="SCA__63">
  <AssignedCondition uid="AC__64" xsi:type="n0:AndCondition">
    <ConditionType>
      <ClassString>and</ClassString>
    </ConditionType>
    <Parameters>
      <Condition uidRef="AC__72"/>
      <Condition uidRef="AC__77"/>
      <Condition uidRef="AC__79"/>
      <Condition uidRef="AC__83"/>
    </Parameters>
  </AssignedCondition>
  <ConditionType>
    <ClassString>and</ClassString>
  </ConditionType>
</SpecificationConditionAssignment>
<SpecificationConditionAssignment uid="SCA__37">
  <AssignedCondition uid="AC__38" xsi:type="n0:EqualsCondition">
    <ConditionType>
      <ClassString>equals</ClassString>
    </ConditionType>
    <Parameters>
      <Specification uidRef="SP__34"/>
    </Parameters>
  </AssignedCondition>
  <ConditionType>
    <ClassString>part usage</ClassString>
  </ConditionType>
</SpecificationConditionAssignment>

<SpecificationConditionAssignment uid="SCA__39">
  <AssignedCondition uid="AC__40" xsi:type="n0:NotCondition">
    <ConditionType>
      <ClassString>not</ClassString>
    </ConditionType>
    <Parameters>
      <Condition uidRef="AC__48"/>
    </Parameters>
  </AssignedCondition>
  <ConditionType>
    <ClassString>not</ClassString>
  </ConditionType>
</SpecificationConditionAssignment>

<SpecificationConditionAssignment uid="SCA__54">
  <AssignedCondition uid="AC__55" xsi:type="n0:EqualsCondition">
    <ConditionType>
      <ClassString>equals</ClassString>
    </ConditionType>
    <Parameters>
      <Specification uidRef="SP__52"/>
    </Parameters>
  </AssignedCondition>
  <ConditionType>
    <ClassString>part usage</ClassString>
  </ConditionType>
</SpecificationConditionAssignment>

<SpecificationConditionAssignment uid="SCA__56">
  <AssignedCondition uid="AC__57" xsi:type="n0:NotCondition">
    <ConditionType>
      <ClassString>not</ClassString>
    </ConditionType>
    <Parameters>
      <Condition uidRef="AC__62"/>
    </Parameters>
  </AssignedCondition>
  <ConditionType>
    <ClassString>not</ClassString>
  </ConditionType>
</SpecificationConditionAssignment>

<SpecificationConditionAssignment uid="SCA__71">
  <AssignedCondition uid="AC__72" xsi:type="n0:EqualsCondition">
    <ConditionType>
      <ClassString>equals</ClassString>
    </ConditionType>
    <Parameters>
      <Specification uidRef="SP__68"/>
    </Parameters>
  </AssignedCondition>
</SpecificationConditionAssignment>
<ConditionType>
  <ClassString>part usage</ClassString>
</ConditionType>
</SpecificationConditionAssignment>

<SpecificationConditionAssignment uid="SCA__76">
  <AssignedCondition uid="AC__77" xsi:type="n0:EqualsCondition">
    <ConditionType>
      <ClassString>equals</ClassString>
    </ConditionType>
    <Parameters>
      <Specification uidRef="SP__74"/>
    </Parameters>
  </AssignedCondition>
</SpecificationConditionAssignment>

<SpecificationConditionAssignment uid="SCA__77">
  <AssignedCondition uid="AC__78" xsi:type="n0:NotCondition">
    <ConditionType>
      <ClassString>not</ClassString>
    </ConditionType>
    <Parameters>
      <Condition uidRef="AC__81"/>
    </Parameters>
  </AssignedCondition>
</SpecificationConditionAssignment>

<SpecificationConditionAssignment uid="SCA__48">
  <AssignedCondition uid="AC__49" xsi:type="n0:EqualsCondition">
    <ConditionType>
      <ClassString>equals</ClassString>
    </ConditionType>
    <Parameters>
      <Specification uidRef="SP__44"/>
    </Parameters>
  </AssignedCondition>
</SpecificationConditionAssignment>

<SpecificationConditionAssignment uid="SCA__60">
  <AssignedCondition uid="AC__61" xsi:type="n0:EqualsCondition">
    <ConditionType>
      <ClassString>equals</ClassString>
    </ConditionType>
    <Parameters>
      <Specification uidRef="SP__59"/>
    </Parameters>
  </AssignedCondition>
</SpecificationConditionAssignment>

<SpecificationConditionAssignment uid="SCA__80">
  <AssignedCondition uid="AC__81" xsi:type="n0:EqualsCondition">
    <ConditionType>
      <ClassString>equals</ClassString>
    </ConditionType>
    <Parameters>
      <Specification uidRef="SP__80"/>
    </Parameters>
  </AssignedCondition>
</SpecificationConditionAssignment>
<Parameters>
  <Specification uidRef="SP__44"/>
</Parameters>

</AssignedCondition>

<ConditionType>
  <ClassString>part usage</ClassString>
</ConditionType>

</SpecificationConditionAssignment>
...

</ProductConcept>
...

<SpecificationCategory uid="SC__32">
  <Description>
    <CharacterString>Freinage description</CharacterString>
  </Description>
  <Id>
    <Identifier uid="I__31" id="Freinage" idRoleRef="C__5" idContextRef="O__3"/>
  </Id>
  <ImplicitExclusiveCondition>true</ImplicitExclusiveCondition>
</SpecificationCategory>

<Specification uid="SP__34">
  <Category uidRef="SC__32"/>
  <Id>
    <Identifier uid="I__33" id="Shimano" idRoleRef="C__5" idContextRef="O__3"/>
  </Id>
  <Name>
    <CharacterString>Shimano</CharacterString>
  </Name>
  <Package>false</Package>
</Specification>

<SpecificationCategory uid="SC__42">
  <Description>
    <CharacterString>Cintre description</CharacterString>
  </Description>
  <Id>
    <Identifier uid="I__41" id="Cintre" idRoleRef="C__5" idContextRef="O__3"/>
  </Id>
  <ImplicitExclusiveCondition>true</ImplicitExclusiveCondition>
</SpecificationCategory>

<Specification uid="SP__44">
  <Category uidRef="SC__42"/>
  <Id>
    <Identifier uid="I__43" id="Recourbe" idRoleRef="C__5" idContextRef="O__3"/>
  </Id>
  <Name>
    <CharacterString>Recourbe</CharacterString>
  </Name>
  <Package>false</Package>
</Specification>

<Specification uid="SP__52">
  <Category uidRef="SC__42"/>
  <Id>
    <Identifier uid="I__51" id="Droit" idRoleRef="C__5" idContextRef="O__3"/>
  </Id>
  <Name>
    <CharacterString>Droit</CharacterString>
  </Name>
  <Package>false</Package>
</Specification>
<CharacterString>Droit</CharacterString>
</Name>
</Specification>
<Specification uid="SP__59">
<Category uidRef="SC__32"/>
<Id>
  <Identifier uid="I__58" id="Magura" idRoleRef="C__5" idContextRef="O__3"/>
</Id>
</Specification>
<SpecificationCategory uid="SC__66">
<Description>
  <CharacterString>Confort description</CharacterString>
</Description>
<Id>
  <Identifier uid="I__65" id="Confort" idRoleRef="C__5" idContextRef="O__3"/>
</Id>
  <ImplicitExclusiveCondition>true</ImplicitExclusiveCondition>
</SpecificationCategory>
<Specification uid="SP__68">
<Category uidRef="SC__66"/>
<Id>
  <Identifier uid="I__67" id="Selle en mousse" idRoleRef="C__5" idContextRef="O__3"/>
</Id>
</Specification>
<Specification uid="SP__74">
<Category uidRef="SC__66"/>
<Id>
  <Identifier uid="I__73" id="Bequille" idRoleRef="C__5" idContextRef="O__3"/>
</Id>
</Specification>

### 4.3.4.2 ENTITY SpecificationConditionAssignment

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssignedCondition</td>
<td>Condition</td>
</tr>
<tr>
<td>ClassifiedAs</td>
<td>OPTIONAL SET[1:?] of Classification</td>
</tr>
<tr>
<td>ConditionType</td>
<td>ClassSelect</td>
</tr>
<tr>
<td>Description</td>
<td>OPTIONAL DescriptorSelect</td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ActivityAssignment</td>
<td>OPTIONAL SET[1:?]</td>
</tr>
<tr>
<td>ActivityMethodAssignment</td>
<td>OPTIONAL SET[1:?]</td>
</tr>
<tr>
<td>ApprovalAssignment</td>
<td>OPTIONAL SET[1:?]</td>
</tr>
<tr>
<td>AssignmentObjectRelationship</td>
<td>OPTIONAL SET[1:?]</td>
</tr>
<tr>
<td>ConditionAssignment</td>
<td>OPTIONAL SET[1:?]</td>
</tr>
<tr>
<td>DateAndPersonAssignment</td>
<td>OPTIONAL SET[1:?]</td>
</tr>
<tr>
<td>DateTimeAssignment</td>
<td>OPTIONAL SET[1:?]</td>
</tr>
<tr>
<td>DocumentAssignment</td>
<td>OPTIONAL SET[1:?]</td>
</tr>
<tr>
<td>EffectivityAssignment</td>
<td>OPTIONAL SET[1:?]</td>
</tr>
<tr>
<td>EventAssignment</td>
<td>OPTIONAL SET[1:?]</td>
</tr>
<tr>
<td>FrozenAssignment</td>
<td>OPTIONAL SET[1:?]</td>
</tr>
<tr>
<td>InformationUsageRightAssignment</td>
<td>OPTIONAL SET[1:?]</td>
</tr>
<tr>
<td>ModelPropertyAssignment</td>
<td>OPTIONAL SET[1:?]</td>
</tr>
<tr>
<td>OrganizationOrPersonInOrganizationAssignment</td>
<td>OPTIONAL SET[1:?]</td>
</tr>
<tr>
<td>PropertyDefinitionAssignment</td>
<td>OPTIONAL SET[1:?]</td>
</tr>
<tr>
<td>PropertyValueAssignment</td>
<td>OPTIONAL SET[1:?]</td>
</tr>
<tr>
<td>TimeIntervalAssignment</td>
<td>OPTIONAL SET[1:?]</td>
</tr>
</tbody>
</table>

Table 23: "SpecificationConditionAssignment" Attributes

Attribute recommendations:
- **AssignedCondition**: Specifies the Condition (see chapter 4.3.4.3) that is assigned to the ProductClass.

- **ConditionType**: specifies the meaning of the association. Use ClassString if the value is recommended within this document, otherwise use “Class” template (see [242BO-PAS]). According to the ISO AP242 Specification, where applicable, the following values shall be used:
  
  o 'design case': The Condition specifies a condition when a given object has to be designed and verified. This value of the conditionType is for information only and shall not be interpreted when querying design cases or usage cases. For such a query, the value of the attribute configurationType of ConditionalConfiguration shall be evaluated. This value may be used to precise when a given BreakdownElement has to be studied by the design department so that it provides solutions appropriate for the case specified by the attributes assignedCondition and assignedTo.
  
  o 'identification': The Condition specifies a condition that enables to distinguish the assigned ProductClass from other ProductClass objects. This value is not applicable for a top-level node in a hierarchy of ProductClass objects. This identification is part of the identification of all sub classes of this ProductClass;

  o 'part usage': The Condition specifies a condition for the usage of the components of an AlternativeSolution, the usage of an Occurrence or for the application of a ProcessPlan or a ProcessOperationOccurrence in the products of the assignedTo ProductClass. In this case, the SpecificationConditionAssignment shall be referenced by at least one ConditionalConfiguration object;

  o 'validity': The Condition specifies a condition that is used to verify a SpecificationAssignment for the assignedTo ProductClass. That means that the Condition evaluates to 'true' if the set of Specification objects is valid; otherwise it evaluates to 'false' with the meaning that the specified object is invalid for the ProductClass. The assignedCondition is valid for all products belonging to the assignedTo ProductClass if conditionType is 'identification' or 'validity'.

- **Description**: the text by which the SpecificationConditionAssignment is described. The value of this attribute need not be specified. Use “Description” template (see [242BO-PAS]).

**Preprocessor Recommendations**:

- Until further use cases are in scope of this document, use SpecificationConditionAssignment only with ConditionType='validity'.

- If the Context is set it should be the same for all the SpecificationConditionAssignment and the ConditionalConfiguration

**Postprocessor Recommendations**:

- Ignore at the moment any SpecificationConditionAssociations not having ConditionType=‘validity’.

**4.3.4.3 ENTITY Condition**

A condition is a definition of the precedent that must be fulfilled before a statement or relationship becomes valid.

Following subtypes of condition are supported: EqualsCondition, OneOfCondition (not recommended), OrCondition, AndCondition, NotCondition, NotEqualsCondition (not recommended)
The parameters against which the condition is to be evaluated are identified by `ConditionParameter`. The target or consequence of a condition is represented by `ConditionAssignment`.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassifiedAs</td>
<td>OPTIONAL SET[1:?] of Classification</td>
</tr>
<tr>
<td>ConditionType</td>
<td>ClassSelect</td>
</tr>
<tr>
<td>Description</td>
<td>OPTIONAL DescriptorSelect</td>
</tr>
<tr>
<td>Id</td>
<td>OPTIONAL IdentifierSelect</td>
</tr>
<tr>
<td>Parameters</td>
<td>SET[1:?] OF ParameterSelect</td>
</tr>
<tr>
<td>SameAs</td>
<td>OPTIONAL SET[1:?] of ProxySelect</td>
</tr>
<tr>
<td>ActivityAssignment</td>
<td>OPTIONAL SET[1:?] of ActivityAssignment</td>
</tr>
<tr>
<td>ApprovalAssignment</td>
<td>OPTIONAL SET[1:?] of ApprovalAssignment</td>
</tr>
<tr>
<td>ConditionAssignment</td>
<td>OPTIONAL SET[1:?] of ConditionAssignment</td>
</tr>
<tr>
<td>DateAndPersonAssignment</td>
<td>OPTIONAL SET[1:?] of DateAndPersonAssignment</td>
</tr>
<tr>
<td>DateTimeAssignment</td>
<td>OPTIONAL SET[1:?] of DateTimeAssignment</td>
</tr>
<tr>
<td>DocumentAssignment</td>
<td>OPTIONAL SET[1:?] of DocumentAssignment</td>
</tr>
<tr>
<td>EffectivityAssignment</td>
<td>OPTIONAL SET[1:?] of EffectivityAssignment</td>
</tr>
<tr>
<td>EventAssignment</td>
<td>OPTIONAL SET[1:?] of EventAssignment</td>
</tr>
<tr>
<td>FrozenAssignment</td>
<td>OPTIONAL SET[1:?] of FrozenAssignment</td>
</tr>
<tr>
<td>InformationUsageRightAssignment</td>
<td>OPTIONAL SET[1:?] of InformationUsageRightAssignment</td>
</tr>
<tr>
<td>ModelPropertyAssignment</td>
<td>OPTIONAL SET[1:?] of ModelPropertyAssignment</td>
</tr>
<tr>
<td>OrganizationOrPersonInOrganizationAssignment</td>
<td>OPTIONAL SET[1:?] of OrganizationOr-PersonInOrganizationAssignment</td>
</tr>
<tr>
<td>PropertyDefinitionAssignment</td>
<td>OPTIONAL SET[1:?] of PropertyDefinitionAssignment</td>
</tr>
</tbody>
</table>
Table 24: "Condition" Attributes

**Attribute recommendations:**

- **ConditionType:** the meaning of the condition. When using the subtypes of Condition, the value of ConditionType shall comply to the name of the subtype, i.e. 'equals', 'and', 'or', 'not'. 'one of' and 'not equals' are not recommended.

- **Description:** the text by which the Condition is described. The value of this attribute need not be specified. Use "Description" template (see [242BO-PAS]).

- **Parameters:** specifies the set of specifications, SerialEffectivities, DataValidityEffectivities, LotEffectivities or embedded conditions that represent the parameters against which the condition is to be evaluated.

**Comments:**

- An issue in Bugzilla has been created under #6289 to embed Condition into Condition rather than defining it as a Reference. Advantage: much easier to 'read' in plain XML and only one instance of SpecificationConditionAssignment for the whole 'tree' of Conditions that make one effectivity. Only inconvenient: Conditions within the trees cannot be reused somewhere else in another ConditionalEffectivity. Remark: doing so, Conditions have to be embedded either in SpecificationConditionAssignment.AssignedCondition or in Condition.Parameters

- An issue in Bugzilla has been created under #5086 to support combined Effectivities with ConditionalEffectivities. This is supported in the TC patch schema. Problem: if no Specification is involved in the Condition, even if the effectivities have no context, the Condition has to be defined within a SpecificationConditionAssignment (again within a ProductClass) => in such a case, a ProductClass with Id='/NULL' has to be created.

**Example of combined effectivities within a condition:**

Spec001 AND #3-3

```xml
<ProductConcept uid="pc--19088" xsi:type="n0:ProductClass">
...
<SpecificationConditionAssignment uid="cda--19136">
  <AssignedCondition uid="spece--19136" xsi:type="n0:AndCondition">
    <ConditionType>
      <ClassString>and</ClassString>
    </ConditionType>
    <Parameters>
      <Specification uidRef="spec--19192"/>
      <Effectivity uidRef="sneff-33-2"/>
    </Parameters>
  </AssignedCondition>
  <ConditionType>
    <ClassString>part usage</ClassString>
  </ConditionType>
</SpecificationConditionAssignment>
```
Preprocessor Recommendations:

Although not defined as an abstract supertype, it is recommended to instantiate only the subtypes of Condition.

The usage in ‘Parameters’ of further objects defined in ParameterSelect (other than Specification, SerialEffectivity, DataValidityEffectivity, LotEffectivity and the subtypes of Condition) is not recommended. Especially the usage of ConditionalEffectivity is not recommended.

The usage of EqualConditions is recommended only for the top-level Condition referenced directly by a ConditionalEffectivity.Condition, but not for ‘Parameters’.

EqualConditions and NotConditions should have only one value in ‘Parameters’.

NotConditions and AndConditions should have at least two values in ‘Parameters’.

Postprocessor Recommendations:

Instances of Condition that are not one of its subtypes shall be ignored or a warning shall be returned.

Further objects defined in ParameterSelect (other than Specification, SerialEffectivity, DataValidityEffectivity, LotEffectivity and the subtypes of Condition) in ‘Parameters’ shall be ignored, or a warning shall be returned.

4.4 Usage of Option Pool (Dictionary)

An option pool defines all options with the possible values that could be used to configure a product within a product context.

The configuration options used in the effectivities along a Product structure need to have a context (i.e. a ProductClass in which they are available). This ProductClass is called Option Pool or Dictionary.

Some PDM systems support only a global dictionary (applies to all ProductClasses), others only local dictionaries (apply to one single ProductClass), others both.

The complete option pool itself must not be part of the AP242 file containing the configured product structure but must be part of a contract between the exchange partners. The contract should also contain a rule to match the relation between the context and the option set.

For the use case “archiving” the option pool should also be defined in the AP242 file containing the configured product structure.

For the use case “internal PLM” the context should be common within all involved systems. Therefore, the Id attribute can be used as foreign key to get all information for the context as well as option pool.

Common Recommendations:

Create contract with the partner to define the supported option pool

Send options inside of each AP242 File together with the context

Preprocessor Recommendations:

Reduce the used options to the options that are part of the contract for the data exchange.

For the use case “Internal PLM” the Id attribute of the ProductClass could be used to match the relation between the context and the option set.
For the use case “supplier exchange”, the Name attribute has to used to match the relation to the option pool.

For the use case “archiving”, the option pool itself should be part of the AP242 file.

A global dictionary shall be mapped to a ProductClass.Id ‘global dictionary’.

**Postprocessor Recommendations:**

If the target system does not support local dictionaries, an appropriate logic has to be implemented like:

- all incoming options are imported without context,
- prefix the option names with the name of the incoming dictionary
- ...

If the target system does not support a global dictionary, an appropriate logic has to be implemented like:

Duplicate all incoming options in each involved local dictionary

5 Managing nested file structure

The recommendations how to split large product structure in a so-called nested structure are defined in the Core Document [242BO-PAS].

Regarding Configuration Management information, the following additional recommendations are made:

- ProductConfigurations, their AssignedEffectivites and ProductDesignAssociations shall be stored in the XML File where the PartVersion referenced by ProductDesignAssociation via AssociatedDesign is defined.
- EffectivityAssignments/Conditions and Effectivities on PartVersion/NextAssemblyOccurrenceUsage and all their attributes shall be defined in the superordinate assembly node file, rather than in the subordinate component file. Doing so, the component file is independent from where and how often it is built into product structures.

In both cases, the ProductClass(es), Specification(s) and SpecificationCategory(ies) referenced:

- as ConfigurationContext of the Effectivities (defined on the PartVersion/NextAssemblyOccurrenceUsages of the part described by the nested file)
- as SpecificationAssignment / SpecificationCategoryAssignment / SpecificationConditionAssignment involved in the above Effectivities
- as ProductConfiguration.AssignedSpecification (on the PartVersion of the part described by the nested file)
- as ProductClassRelationship.Related (in case the nested file describes a ProductClass having subordinated ProductClasses, and not a Part)

shall be mapped as a reference by:

1. mapping a minimum set of entities and attributes (subset of those mapped in the referenced XML file or in the monolithic mapping):
   - ProductClass.id with idRoleRef and idContextRef,
   - ProductClass.VersionId (if relevant for the data exchange),
   - Specification.id with idRoleRef and idContextRef,
   - Specification.VersionId (if relevant for the data exchange),
2. mapping the reference to the ProductClass XML file as DocumentAssignment between the ProductClass and a DigitalFile

3. mapping a dedicated Classification 'specified reference' on the ProductClass
The Instance Model: STEP exchange file format (ISO10303 AP242 BO Model XML syntax)

```xml
<Part uid="ID_208">
...  
  <Versions>
    <PartVersion uid="ID_129">
      ...
      <Views>
```

Figure 38: External reference to a ProductClass within a nested XML File containing EffectivityAssignments
<PartView uid="ID_130" xsi:type="bom:AssemblyDefinition">
  ...
  <ViewOccurrenceRelationship uid="ID_250" xsi:type="bom:NextAssemblyOccurrenceUsage">
    <Related uidRef="ID_252"/>
    <RelationType>
      <ClassString>next assembly occurrence</ClassString>
    </RelationType>
    <EffectivityAssignment uid="EA__603591">
      <AssignedEffectivity uidRef="CC__603588"/>
      <EffectivityIndication>true</EffectivityIndication>
      <Role>
        <ClassString>actual</ClassString>
      </Role>
      </EffectivityAssignment>
  ...
  </ViewOccurrenceRelationship>
  ...
  </PartView>
</Views>

</Part>

<Effectivity uid="DVE__603585" xsi:type="bom:DataValidityEffectivity">
  <EffectivityContext uidRef="pc--19088"/>
  <EndDefinition>
    <DateTimeString>2017-12-31T23:59:59.0Z</DateTimeString>
  </EndDefinition>
</Effectivity>

<Effectivity uid="CC__603588" xsi:type="bom:ConditionalConfiguration">
  <EffectivityContext uidRef="pc--19088"/>
  <Condition uidRef="AC__603586"/>
  <ConfigurationType>
    <ClassString>usage</ClassString>
  </ConfigurationType>
  <InheritanceType>
    <ClassString>local</ClassString>
  </InheritanceType>
</Effectivity>

<SpecificationCategory uid="SC__603581">
  <Description>
    <CharacterString>car body description</CharacterString>
  </Description>
  <Id>
    <Identifier id="FreerideMTB" idContextRef="ID_64" idRoleRef="ID_66" uid="SC__603581--id"/>
  </Id>
  <ImplicitExclusiveCondition>true</ImplicitExclusiveCondition>
</SpecificationCategory>

<Specification uid="S__603583">
  <Category uidRef="SC__603581"/>
  <Id>
    <Identifier id="FreerideMTB_Black" idContextRef="ID_64" idRoleRef="ID_66" uid="S__603583--id"/>
  </Id>
  <Package>false</Package>
</Specification>

<Specification uid="S__603592">
  ...
</Specification>
<Category uidRef="SC__603581"/>
  <Id>
    <Identifier id="FreerideMTB_Red" idContextRef="ID_64" idRoleRef="ID_66" uid="S__603592--id"/>
  </Id>
  <Package>false</Package>
</Specification>
-Classification uid="nested_pv">
  <Class>
    <ClassString>specified reference</ClassString>
  </Class>
</Classification>
<File uid="ID_90_df" xsi:type="bom:DigitalFile">
  <FileContent uid="ID_90_cp" xsi:type="bom:ContentProperty">
    <GeometryTypes>
      <Classification uidRef="nested_pc"/>
    </GeometryTypes>
    <FileFormat uidRef="nested_ff"/>
    <FileType>
      <ClassString>structured product data</ClassString>
    </FileType>
    <Id>
      <Identifier uid="ID_90_df_id" id="dictionary.stpx" idRoleRef="ID_66" idContextRef="ID_64"/>
    </Identifier>
  </FileContent>
  <Locations>
    <ExternalItem uid="ID_90_ei">
      <Id id="dictionary.stpx"/>
      <Source>
        <IdentifierString>./</IdentifierString>
      </Source>
    </ExternalItem>
  </Locations>
</File>
<ProductConcept uid="pc--19088" xsi:type="bom:ProductClass">
  <ClassifiedAs>
    <Classification uidRef="nested_pv"/>
  </ClassifiedAs>
  <Id>
    <Identifier uid="pc-asclass--id1" id="Configuration test" idRoleRef="ID_66" idContextRef="ID_64"/>
  </Id>
  <DocumentAssignment uid="ID_91_da">
    <AssignedDocument uidRef="ID_90_df"/>
    <Role>
      <ClassString>mandatory</ClassString>
    </Role>
  </DocumentAssignment>
  <VersionId id="A.1"/>
  <SpecificationAssignment uid="SA__603584">
    <AssignedSpecification uidRef="S__603583"/>
    <AssociationType>
      <ClassString>availability</ClassString>
    </AssociationType>
  </SpecificationAssignment>
  <SpecificationAssignment uid="SA__603593">
    <AssignedSpecification uidRef="S__603592"/>
    <AssociationType>
      <ClassString>availability</ClassString>
    </AssociationType>
  </SpecificationAssignment>
</ProductConcept>
The referenced XML file containing the ProductClass shall contain its full definition and the full definition of all available Specifications and SpecificationCategories (attached to the ProductClass via SpecificationAssignment and SpecificationCategoryAssignment), but no Effectivities, no SpecificationConditionAssignments and no Conditions.

Doing so, the uids used in the referenced ProductClass XML file are independent from the uids used in the assembly node nested file.
Limitations of the current recommendation:

- Effectivities and Conditions shall not be reused in multiple assembly nodes.
- ProductConfigurations may be associated to only one top level PartVersion via ProductDesignAssociation.
- Events (used for Milestone Effectivities) would be repeated in the XML files and not treated as referenced object.
6 Supported Configuration Management for PDM Systems

This chapter contains an overview of the various aspects of configuration management supported by each PLM System for the matching constraints to support the use cases of this document with AP242 standard. Not all capabilities of the PDM systems are mentioned here.

Since Aras currently does not support configuration management (announced for future release), the topics depicted below for Aras consider the productive data model extension of T-Systems.

<table>
<thead>
<tr>
<th>Concept</th>
<th>STEP AP242 BO Model</th>
<th>Teamcenter</th>
<th>Windchill</th>
<th>Enovia VPM</th>
<th>3DExperience</th>
<th>Aras (with T-Syst extensions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effectivities at PartVersions</td>
<td>PartVersion.EffectivityAssignment</td>
<td>X</td>
<td>X</td>
<td>?</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>Effectivities at Occurrences</td>
<td>NextAssemblyOccurrenceUsage.EffectivityAssignment</td>
<td>X</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Date Range</td>
<td>DataValidityEffectivity</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Time interval</td>
<td>TimeIntervalEffectivity</td>
<td>--</td>
<td>--</td>
<td>?</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Serial number Range</td>
<td>SerialEffectivity</td>
<td>X (so-called Unit Effectivity)</td>
<td>x</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Multiple intervals (combined implicitly with 'OR')</td>
<td>Several EffectivityAssignments</td>
<td>X (for Serial)</td>
<td>X (For Date, Lot and Serial)</td>
<td>?</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>---</td>
<td>----</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>----</td>
</tr>
<tr>
<td>Specification Category attributes/versioning</td>
<td>X</td>
<td>--</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>X (but no versioning)</td>
</tr>
<tr>
<td>ConditionalEffectivity</td>
<td>ConditionalEffectivity</td>
<td>X (so-called ‘classic variants’. The other methods are not detailed here (Open-PDM: OK?))</td>
<td>?</td>
<td>?</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Logical operators</td>
<td>Logical operators</td>
<td>Equals, Or, And, Not, NotEquals, one of AND, OR, NOT</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>AND, OR, NOT</td>
</tr>
<tr>
<td>Parenthesis</td>
<td>Parenthesis</td>
<td>X, but NOT can only be used on single options (not for further condition) =&gt; similar to NotEquals</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>X</td>
</tr>
<tr>
<td>Logical combination of different effectivities</td>
<td>Logical combination of different effectivities</td>
<td>X* BugZilla #5086 to allow any non-ConditionalEffectivities to be referenced by any Conditions</td>
<td>With the same type it's implicitly OR operand</td>
<td>With different types it's an AND operand</td>
<td>Deep logical combination, i.e. opt1 or (dateEff and serialEff), is not supported</td>
<td>?</td>
</tr>
<tr>
<td>Effectivity context</td>
<td>X (optional)</td>
<td>Required for Serial and Lot</td>
<td>Optional for Date (support of contextual date)</td>
<td>Required for ConditionedEffectivity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------</td>
<td>-----------------------------</td>
<td>-----------------------------------------------</td>
<td>-------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Effectivity Context as EndItem</td>
<td>--</td>
<td>The assembly structure link that has a conditional effectivity has to be related up or down to the EndItem where the Specification(Categories) are defined</td>
<td>?</td>
<td>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type of EffectivityContext</td>
<td>ProductClass or ProductConfiguration</td>
<td>End Item (the ProductClass is managed as an Item)</td>
<td>Model (equivalent to ProductClass)</td>
<td>--</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ProductClass attributes/versioning</td>
<td>ProductClass</td>
<td>EndItem</td>
<td>End Item (the ProductClass is managed as an Item)</td>
<td>--</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EffectivityContext structure</td>
<td>ProductClassRelationship, ProductClass</td>
<td>EndItem assembly structure link.</td>
<td>?</td>
<td>--</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### PDM-IF Recommended Practices
#### AP242 BO Model XML Configuration Management
Version 0.9; July 1, 2019

<table>
<thead>
<tr>
<th>Open intervals</th>
<th>SerialEffectivity (right-hand) DateEffectivity (right-hand and left-hand)</th>
<th>Not structure of configuration filter</th>
<th>left-hand</th>
<th>right-hand</th>
<th>Both supported (right-hand and left-hand)</th>
<th>Both supported (right-hand and left-hand)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effectivity sharing</td>
<td>X</td>
<td>X</td>
<td>Not shared</td>
<td>Not shared</td>
<td>Not shared</td>
<td>Not shared</td>
</tr>
<tr>
<td>ProductConfiguration definition</td>
<td>0-n ProductConfiguration. Defining Specifications 0-n+ EffectivityAssignments to Serial/Date/Milestone/Lot/Time Interval Effectivities (possibly with intervals)</td>
<td>0-n Specifications + 0-1 Effective Date + 0-1 Effective Unit Number with EndItem</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>0-n Specifications + 0-1 Effective Date(interval) + 0-1 Effective Unit Number(interval) + 0-1 Effective Lot + 0-1 Effective Milestone(interval)</td>
</tr>
<tr>
<td>Explicit link of a ProductConfiguration to 1-n assembly top nodes</td>
<td>ProductDesignAssociation</td>
<td>--</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>--</td>
</tr>
<tr>
<td>Explicit link of a ProductConfiguration to the assembly links that are valid for the given configuration</td>
<td>ConfiguredAssemblyEffectivity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--</td>
<td>--</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table 25: Supported Configuration Management approaches for PDM Systems*
7 References

<table>
<thead>
<tr>
<th>Standard</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>EIA-649-A</td>
<td>National Consenses Standard for Configuration Management</td>
</tr>
<tr>
<td>CMII</td>
<td>Configuration Management II</td>
</tr>
<tr>
<td>ISO 10303:3001</td>
<td>Industrial automation systems and integration — Product data representation and exchange</td>
</tr>
<tr>
<td></td>
<td>Part 3001</td>
</tr>
<tr>
<td></td>
<td>Business object model: Managed model based 3d engineering</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDM Schema Usage Guide v4.3.</td>
</tr>
<tr>
<td>AP214 CC8 Recommended Practices</td>
</tr>
<tr>
<td>Recommended Practices for STEP AP242 Business Object Model XML Product &amp; Assembly Structure, Release 2.0</td>
</tr>
</tbody>
</table>
Annex A  Known Issues

This section lists known issues with the AP242 Business Object Model, both related to the assembly structure and to other domains within the BO 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.

<table>
<thead>
<tr>
<th>Issue# (Link)</th>
<th>Short Description</th>
<th>Target Milestone</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>#5086</td>
<td>Combination of effectivities</td>
<td>AP242 ed1 TC patch</td>
<td>Must be further discussed to avoid to instantiate SpecificationConditionAssignment when no Specification is involved…</td>
</tr>
<tr>
<td>#6132</td>
<td>ProductConfiguration must not be a BaseRootObject</td>
<td>AP242ed2</td>
<td>Confirmed</td>
</tr>
<tr>
<td>#6288</td>
<td>Embed the Specifications into SpecificationCategory</td>
<td>AP242ed2</td>
<td>Confirmed</td>
</tr>
<tr>
<td>#6289</td>
<td>Embedding Condition along Condition.Pa</td>
<td>AP242ed2</td>
<td>Confirmed</td>
</tr>
<tr>
<td>#6437</td>
<td>Embedding ProductDesignAssociation into ProductConfiguration</td>
<td>AP242ed2</td>
<td>?</td>
</tr>
<tr>
<td>#6438</td>
<td>Embedding ConfiguredAssemblyEffectivity into ProductDesignAssociation</td>
<td>AP242ed2</td>
<td>?</td>
</tr>
</tbody>
</table>