

---

---

**Software and systems engineering —  
Methods and tools for variability  
modelling in software and systems  
product line**

*Ingénierie des systèmes et du logiciel — Méthodes et outils pour  
modéliser la variabilité dans les gammes de produits des logiciels et  
systèmes*





**COPYRIGHT PROTECTED DOCUMENT**

© ISO/IEC 2017, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
Ch. de Blandonnet 8 • CP 401  
CH-1214 Vernier, Geneva, Switzerland  
Tel. +41 22 749 01 11  
Fax +41 22 749 09 47  
copyright@iso.org  
www.iso.org

# Contents

Page

<b>Foreword</b>	<b>v</b>
<b>Introduction</b>	<b>vi</b>
<b>1 Scope</b>	<b>1</b>
<b>2 Normative references</b>	<b>1</b>
<b>3 Terms and definitions</b>	<b>1</b>
<b>4 Variability modelling in software and systems product line</b>	<b>2</b>
4.1 Overview	2
4.2 Reference model for variability modelling in software and systems product line	4
<b>5 Variability model management</b>	<b>6</b>
5.1 General	6
5.2 Variability model planning	7
5.2.1 Purpose of variability model planning	7
5.2.2 Design variability modelling strategy	7
5.2.3 Define quality assurance measures for variability modelling	8
5.2.4 Assign responsibility for variability modelling	8
5.2.5 Record variability model plan	8
5.3 Variability model enabling	9
5.3.1 Purpose of variability model enabling	9
5.3.2 Provide guidance for variability modelling	10
5.3.3 Mobilize roles and responsibilities for variability modelling	10
5.3.4 Enable variability model-centric variability management	10
5.3.5 Enable variability modelling operations	11
5.3.6 Enable quality assurance measurement for variability modelling	11
5.4 Variability model managing	11
5.4.1 Purpose of variability model managing	11
5.4.2 Review the plan versus actual of variability modelling	12
5.4.3 Control issues on domain/application variability modelling	13
5.4.4 Control issues on variability model-centred variability management	13
5.4.5 Control issues on variability model support	13
5.4.6 Support corrective actions for variability modelling	14
5.4.7 Make improvement actions for variability modelling	14
<b>6 Variability modelling</b>	<b>14</b>
6.1 General	14
6.2 Domain variability modelling	15
6.2.1 Purpose of domain variability modelling	15
6.2.2 Construct domain variability model	15
6.2.3 Annotate domain variability model	16
6.2.4 Verify domain variability model	16
6.2.5 Optimize domain variability model	17
6.3 Application variability modelling	17
6.3.1 Purpose of application variability modelling	17
6.3.2 Construct application variability model	18
6.3.3 Annotate application variability model	18
6.3.4 Verify application variability model	18
6.3.5 Optimize application variability model	19
6.4 Relating variability model to artefacts	19
6.4.1 Purpose of relating variability model to artefacts	19
6.4.2 Retrieve variation points and variants in relevant artefacts	20
6.4.3 Relate domain variability model to domain artefacts	20
6.4.4 Relate application variability model to application artefacts	20
6.5 Relating domain variability model to application variability model	21
6.5.1 Purpose of domain variability model to application variability model	21

6.5.2	Trace binding decisions made in an application .....	22
6.5.3	Establish relations between domain and application variability models .....	22
6.5.4	Add decision-related annotations to relations .....	22
6.5.5	Verify relations between domain and application variability models .....	23
<b>7</b>	<b>Variability model support .....</b>	<b>23</b>
7.1	General .....	23
7.2	Relating variability model to variability mechanism .....	23
7.2.1	Purpose of relating variability model to variability mechanism .....	23
7.2.2	Identify variability including variability mechanism constraints .....	24
7.2.3	Establish relations from variability model to variability mechanism .....	24
7.2.4	Add variability mechanism constraint annotations into variability model .....	25
7.3	Quality assurance for variability model .....	25
7.3.1	Purpose of quality assurance for variability model .....	25
7.3.2	Objectively evaluate variability modelling activities .....	26
7.3.3	Objectively evaluate variability model work products .....	26
7.3.4	Communicate and resolve noncompliance issues .....	27
7.3.5	Establish records of variability modelling quality assurance activities .....	27
7.4	Binding decision support .....	28
7.4.1	Purpose of binding decision support .....	28
7.4.2	Establish full of references to binding decision tables .....	28
7.4.3	Verify binding decisions from variability models view .....	29
7.5	Application configuration support .....	29
7.5.1	Purpose of application configuration support .....	29
7.5.2	Relate variability models to binding decision tables .....	30
7.5.3	Provide different views of variability models by binding stages .....	30
7.5.4	Support full of traces from variability model to artefacts .....	30
	<b>Annex A (informative) Variability meta model .....</b>	<b>32</b>
	<b>Annex B (informative) Orthogonal variability model .....</b>	<b>33</b>
	<b>Annex C (informative) Formal descriptions for variability relationships .....</b>	<b>34</b>
	<b>Annex D (informative) Orthogonal variability decision table .....</b>	<b>35</b>
	<b>Annex E (informative) Orthogonal variability model validation .....</b>	<b>36</b>
	<b>Bibliography .....</b>	<b>38</b>

## Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see [www.iso.org/directives](http://www.iso.org/directives)).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see [www.iso.org/patents](http://www.iso.org/patents)).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: [www.iso.org/iso/foreword.html](http://www.iso.org/iso/foreword.html).

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

## Introduction

Software and Systems Product Line (SSPL) engineering and management creates, exploits and manages a common platform to develop a family of products (e.g. software products, systems architectures) at lower cost, reduced time to market and with better quality. As a result, it has gained increasing global attention since the 1990s.

Variability, which differentiates a member product from other products within a product line, plays an important role in SSPL; and hundreds of variabilities are introduced throughout the whole SSPL domain engineering stages. Those variabilities are defined, refined, newly added as domain engineering stages go forward. Variabilities thus are modelled carefully so as to manage and control them in a systematic way. This document deals with methods and tools capability for supporting variability modelling using consistent notations and for managing and/or utilizing variability models in domain and application engineering lifecycle processes.

This document can be used in the following modes:

- by the users of this document: to benefit people who want to adopt SSPL for producing their products by guiding how to model variabilities among member products;
- by a product line organization: to provide guidance in the evaluation and selection for methods and tools for variability modelling;
- by providers of tools and methods: to provide guidance in implementing or developing methods and tools by providing a comprehensive set of methods and tools capabilities for supporting variability modelling.

The ISO/IEC 26550 family of standards addresses both engineering and management processes and capabilities of methods and tools in terms of the key characteristics of product line development. This document provides processes and capabilities of methods and tools for variability modelling in product lines. Other ISO/IEC 26550 family of standards are as follows:

- processes and capabilities of methods and tools for domain requirements engineering and application requirements engineering are provided in ISO/IEC 26551;
- processes and capabilities of methods and tools for domain design and application design are provided in ISO/IEC 26552;
- processes and capabilities of methods and tools for domain realization and application realization are provided in ISO/IEC 26553 (International Standard under development);
- processes and capabilities of methods and tools for domain testing and application testing are provided in ISO/IEC 26554;
- processes and capabilities of methods and tools for technical management are provided in ISO/IEC 26555;
- processes and capabilities of methods and tools for organizational management are provided in ISO/IEC 26556;
- processes and capabilities of methods and tools for variability mechanisms are provided in ISO/IEC 26557;
- processes and capabilities of methods and tools for variability traceability are provided in ISO/IEC 26559;
- processes and capabilities of methods and tools for product management are provided in ISO/IEC 26560;
- processes and capabilities of methods and tools for technical probe are provided in ISO/IEC 26561;

- processes and capabilities of methods and tools for transition management are provided in ISO/IEC 26562;
- processes and capabilities of methods and tools for configuration management of asset are provided in ISO/IEC 26563;
- others (ISO/IEC 26564 to ISO/IEC 26599): to be developed.

ISO/IEC 26550, ISO/IEC 26551 and ISO/IEC 26555 are published. ISO/IEC 26557 and ISO/IEC 26559 are to be published. ISO/IEC 26552, ISO/IEC 26553, ISO/IEC 26554, ISO/IEC 26556, ISO/IEC 26560, ISO/IEC 26561, ISO/IEC 26562, ISO/IEC 26563 are planned International Standards.





# Software and systems engineering — Methods and tools for variability modelling in software and systems product line

## 1 Scope

This document, within the context of methods and tools for supporting explicit and/or separate variability modelling, variability model management and variability model support in software and systems product lines:

- provides the terms and definitions specific to variability modelling for software and systems product line;
- defines processes for variability modelling, variability model management and variability model support throughout the product line lifecycle. Those processes are described in terms of purpose, inputs, tasks and outcomes;
- defines method capabilities to support the defined tasks of each process;
- defines tool capabilities that automate or semi-automate tasks and methods.

This document does not concern processes and capabilities of tools and methods for a single system but rather deals with those for a family of products.

## 2 Normative references

There are no normative references in this document.

## 3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

### 3.1

#### **application configuration**

composition results of an application by both binding variability and adding application specific variability

### 3.2

#### **application variability model**

variability model for a particular application including variability binding results, application specifically modified variability and application specifically added variability

### 3.3

#### **aspect**

special consideration within product line engineering process groups and tasks to which we can associate specialized methods and tools

### 3.4

#### **domain variability model**

explicit definition of product line variability

### 3.5

#### **constraints dependency**

relationship between *variation points* (3.12), between *variants* (3.11) and between a variation point and a variant

Note 1 to entry: Two types of constraints are possible: “excludes” which means a variant or a variation point forbids another variant or variation point and “requires” which means a variant or a variation point demands (an-)other variant or variation point.

### 3.6

#### **texture**

##### **architectural texture**

collection of common development rules and constraints for realizing the applications of a product line

### 3.7

#### **variability dependency**

association from a *variation point* (3.12) to a *variant* (3.11) or variants

### 3.8

#### **variability modelling**

explicit definition for product line variability

### 3.9

#### **variability modelling plan**

documentation that includes schedules, defined roles and responsibilities, and defined quality assurance measures that will be applied to *variability modelling* (3.8)

### 3.10

#### **variability modelling strategy**

*variability modelling* (3.8) methodology, strictness degree of variability model validation, rules, constraints, other details for supporting the role of variability model in the whole variability management

### 3.11

#### **variant**

instance or a value of a *variation point* (3.12)

### 3.12

#### **variation point**

indication of product differentiation based on particular variable characteristics of products, domain assets, and application assets in the context of a product line

## **4 Variability modelling in software and systems product line**

### **4.1 Overview**

Variability is a key differentiator between single-system engineering and management and product line engineering and management. Product line engineering and management has to take explicitly into account the variations within and between multiple products. The product line variabilities are introduced and defined during product management, domain engineering and application engineering processes defined in ISO/IEC 26550. Their abstraction levels at each lifecycle stage can differ and much variability are refined or newly added as the development progresses. Variability should be defined, modelled, implemented, versioned, verified and validated. Variability model supports abstractions and explicit expressions of the defined variabilities. Variability modelling means the operation for creating, maintaining and supporting variability models using variability together with variability-relevant information defined from product management, domain engineering to application engineering of ISO/IEC 26550. This document supports variability modelling using consistent notations and provides management and required supports for managing and/or utilizing variability models in domain and application engineering.

There are two types of variability models: domain variability models and application variability models. Domain engineering typically provides most of the variability information necessary for structuring the domain variability model. The model is refined and managed throughout the domain engineering lifecycle. On the other hand, application-specific variabilities are introduced during application engineering because each member product of a product line may offer plenty of variability for differentiating itself from other products. Application-specific variabilities including the bound variability of domain variability model are documented as an application variability model. This model is also refined throughout the application engineering lifecycle. The levels of detail of variability information differ depending on the process (e.g. application requirements engineering) where the information is produced.

The orthogonal variability model (OVM) defines the variability of a product line separately unlike feature model that defines the whole domain including both commonality and variability or integrated modelling approach that represents variability by integrating within the development artefacts. A typical product line has hundreds of variability, so it is difficult to manage (i.e. tracing, changing and so on) variability in the forms of feature model or integrated modelling approach. For defining the variability orthogonally, some types of variability dependency, variability constraints and the elements consisting model may be used as it is, or some of them should be defined differently or newly added. This clause describes the elements of the OVM.

The variability consists of the following elements, so the OVM that models the variability should include the following elements:

- variation point;
- variant;
- variability dependency.

A variation point should have relations with more than one variant. The basic variability dependencies include the following:

- mandatory: a variant should be selected, namely it should be part of a member product if the corresponding variation point is selected;
- optional: a variant can be selected or not, namely it can be part of a member product or not.

Optional variability dependency has the following special types of variability dependency:

- exclusive-or, alternative: only one variant should be selected among variants that have optional variability dependency with the same variation point;
- inclusive-or: numbers of variants can be selected among variants that have optional variability dependency with the same variation point, and the numbers of variants that can be selected are defined by range.

Variability can have relations with other variability. The selection of a variation point or a variant can constrain other selections of a variation point or a variant. Such restrictions are called constraint dependency. Constraint dependency includes the following types:

- requires: a variation point or a variant requires another selection of a variation point or a variant;
- excludes: a variation point or a variant should not be selected when a variation point or a variant is selected.

NOTE 1 The principles of variability in SSPL and the orthogonal variability model are depicted in [Annex A](#) and [Annex B](#).

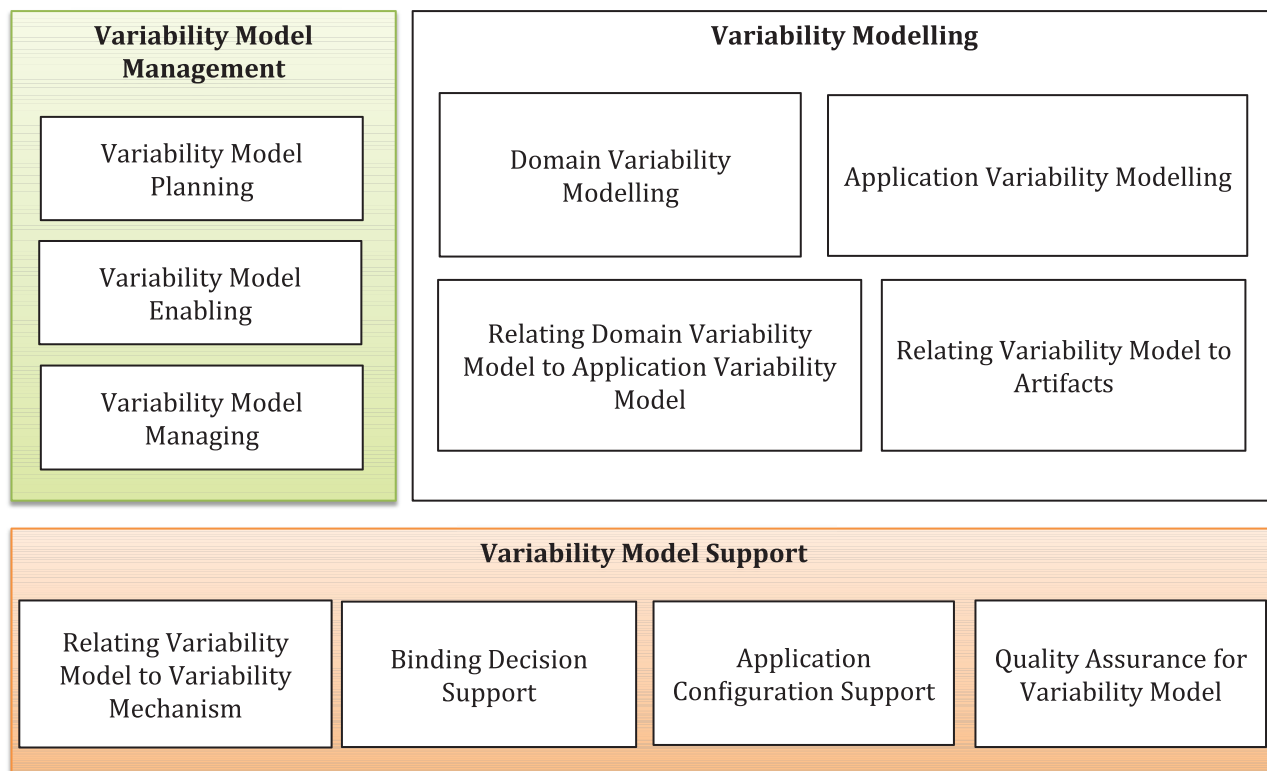
NOTE 2 Formal descriptions for the variability dependency and constraints dependency are depicted in [Annex C](#).

## 4.2 Reference model for variability modelling in software and systems product line

The reference model specifies the structure of supporting processes and subprocesses for variability modelling in product line. As shown in [Figure 1](#), variability modelling in product line can be structured into three processes: variability model management, variability modelling and variability model support. In the rest of this document, tasks, methods and tools are described in terms of processes and subprocesses defined in the reference model.

Each process is divided into subprocesses and each subprocess is described in terms of the following attributes:

- the title of the subprocess;
- the purpose of the subprocess;
- the inputs to produce the outcomes;
- the tasks to achieve the outcomes;
- the outcomes of the subprocess;
- the capabilities of methods and tools required for performing the tasks effectively and efficiently.



**Figure 1 — Variability modelling in SSPL**

The variability modelling management process provides managerial supports for planning variability modelling (e.g. variability model elements, variability model notation, resource estimation, responsibility allocation, quality assurance measures), supports for providing necessary resources, tools and infrastructures for realizing variability modelling plans and supports for analysing the plan versus actual status of variability modelling realization status. Variability modelling management shall do the following:

- variability model planning;
- variability model enabling;

- variability model managing.

The variability modelling process supports variability modelling for identifying and visualizing all variabilities and their relationships, for visualizing detailed relations from variability models to domain/application assets and for visualizing relations between two variability models, namely domain and application variability models. Variability modelling shall do the following:

- domain variability modelling;
- application variability modelling;
- relating domain variability model to application variability model;
- relating variability model to artefacts.

The variability model support process provides supports required for establishing the right variability models and for establishing the right roles of variability models, i.e. establishing and maintaining the detailed relations from variability model to variability mechanisms, binding supports and application configuration supports. Variability model support shall do the following:

- relating variability model to variability mechanism;
- binding decision support;
- application configuration support;
- quality assurance for variability model.

The identification and analysis of the key differentiators between single-system engineering and management and product line engineering and management can help organizations to understand the product line and to formulate a strategy for successful implementation of product line engineering and management. The key aspects have been defined in ISO/IEC 26550 and [Table 1](#) shows the category of the key aspects.

**Table 1 — Key aspects for identifying product line-specific variability modelling tasks**

Category	Aspects
Reuse management	Application engineering, domain assets, domain engineering, product management, platform, reusability
Variability management	Binding, variability
Complexity management	Collaboration, configuration, enabling technology support, reference architecture, texture, traceability
Quality management	Measurement and tracking, cross functional verification and validation

The following are the descriptions for each aspect concerning variability modelling for product lines. The variability modelling relevant processes and tasks shall be identified on the basis of these aspects. The concerns specific to variability modelling for product lines will enable an organization to understand the variability modelling relevant processes, subprocesses, tasks, methods and tools' capabilities.

- Application engineering: Application engineering uses variability model in order to determine the values [variant(s)] of variability and adds application-specific variabilities. Application engineering produces application variability model as the results of those activities.
- Binding: Variability model should devise for providing detailed information required for the right binding. Decision table or annotation can be ways to resolve this.
- Collaboration: Variability model provides the integrated view of the whole variabilities defined and managed in a product line. Domain engineering and application engineering collaborate with each other revolving around variability model for defining, binding and managing variabilities.

- Configuration: Configuration is produced through binding variabilities defined in variability model. Configuration is produced from architecture stage through runtime, so variability model should be defined having different level of abstraction in accordance with the binding information that variability model should contain.
- Domain asset: Domain assets include variability-relevant implementations. Variabilities within domain assets shall be uniquely identified because there are hundreds of variabilities and it is essential to visualize them for their proper management and efficient application of a product line.
- Domain engineering: During domain engineering, domain variability model is produced.
- Enabling technology support: Technologies for managing domain variability model, application variability model and essential information for the right binding should be supported.
- Measurement and tracking: The optimality and sufficiency of variability model for supporting binding decision and product line maintenance and evolution should be measured and tracked.
- Platform: Platform should include the proper implementation of variation points that enable variability binding. Variation points of a variability model should have links with these parts of a platform.
- Product management: Variabilities defined in a variability model are continuously changed and evolved. Variability model is thus changed and evolved so the major focus of product management should be on the defined variability of a variability model.
- Reference architecture: Reference architecture in a product line should be the structure of handling variability well and evolutionary. Thus, reference architecture should be harmonized with variability model and its evolution.
- Reusability: Variability model should be reusable in order to ensure consistency and extensibility of products within the product line.
- Texture: Texture provides rules and constraints that support correct and consistent implementation of variability defined in variability model.
- Traceability: Trace links between domain variability model and domain assets, between application variability model and application assets and between domain variability model and application variability model should be established and managed.
- Cross functional validation and verification: Correctness and consistency of domain and application variability models are validated and verified.
- Variability: The whole variabilities of a product line should be managed through the variability model, i.e. domain variability model and application variability model.

## 5 Variability model management

### 5.1 General

Variability models should be modelled in complete and consistent ways so that the whole variability of a product line is modelled and managed through the explicitly defined variability models. Variability model management should generate plans, provide enabling environments, monitor and control their status.

The variability model management process includes the following subprocesses:

- variability model planning;
- variability model enabling;
- variability model management.

## 5.2 Variability model planning

### 5.2.1 Purpose of variability model planning

#### 5.2.1.1 General

The purpose of this subprocess is to establish and maintain plans for defining and maintaining domain variability models, application variability models and required relations centred on variability models.

#### 5.2.1.2 Inputs

The following inputs should be available to perform the variability model planning process:

- product line adoption plan;
- variability management plan.

#### 5.2.1.3 Outcomes

The following outcomes shall be available as a result of the successful implementation of the variability model planning process:

- variability modelling plan including strategy for variability modelling, cooperation plan between domain and application variability models, responsibility and quality measures is agreed and recorded.

#### 5.2.1.4 Tasks

The organization shall implement the following tasks with respect to the variability model planning process.

- Design variability modelling strategy: to make a strategy for designing variability models as the centre of variability management.
- Define quality assurance measures for variability modelling: to define quality assurance measures to monitor and control quality assurance activities in variability modelling so as to achieve quality assurance throughout the SSPL stages.
- Assign responsibility for variability modelling: to clarify roles and responsibilities with respect to variability modelling.
- Record variability model plan: to document and get agreement and commitment on variability model plan from the appropriate staffs and managers.

### 5.2.2 Design variability modelling strategy

The goal of this task is to establish a strategy for determining the overall structure of variability model and guidance for variability model management.

The strategy includes variability modelling methodology, strictness degree of model validation, rules, constraints, other details for supporting the role of variability model in the whole variability management.

The method should support designing variability modelling strategy with the following capabilities:

- providing selection guides for variability modelling relevant standards, methods and validation;
- providing templates for informing what is major contents of strategy;
- providing examples for major documentation contents.



A tool should support designing variability modelling strategy by allowing the user to do the following:

- supporting the access of selection guides;
- providing electronic documentation template for strategy;
- allowing immediate reference for the examples of major documentation contents during documentation.

### **5.2.3 Define quality assurance measures for variability modelling**

The goal of this task is to define measures used for assuring qualities of variability modelling activities. Quality assurance activity should assure that the defined variability model adheres to the structures, rules and constraints defined in the variability modelling strategy.

The method should support defining quality assurance measures for variability modelling with the following capabilities:

- providing characteristic functions for expressing the overall quality level of variability modelling;
- supporting quality measure and metric definition;
- supporting quality measurement activities related to variability modelling and variability models.

A tool should support defining quality assurance measures for variability modelling by allowing the user to do the following:

- supporting the calculation of characteristic functions;
- supporting data collection related to measures;
- allowing integration or expansion of quality measures for quality level analysis.

### **5.2.4 Assign responsibility for variability modelling**

The goal of this task is to assign roles and responsibilities for proceeding variability modelling. Roles include identifying variability with its dependencies and constraints, defining variability model with tools, validating and verifying variability model and quality assurance for variability modelling.

The method should support assigning responsibility for variability modelling with the following capabilities:

- extracting organization units that are proper to the defined variability modelling roles;
- assuring their capabilities with the roles and responsibilities to be assigned;
- providing a way to define roles and responsibilities concretely, correctly and consistently for the right communication.

A tool should support assigning responsibility for variability modelling by allowing the user to do the following:

- allowing access for the structure of organization units;
- providing documentation for the assignment results;
- sharing the defined roles and responsibilities with relevant participants.

### **5.2.5 Record variability model plan**

The goal of this task is to document plans for variability modelling. Plan documentation includes schedules, the defined roles and responsibilities, defined quality assurance measures.



The method should support recording variability model plan with the following capabilities:

- supporting variability modelling scheduling;
- supporting integration of the pre-defined roles and responsibilities to variability model plan;
- providing documentation template for variability model plan.

A tool should support recording variability model plan by allowing the user to do the following:

- providing automatic consistency check for scheduling;
- allowing electronic plan documentation.

### **5.3 Variability model enabling**

#### **5.3.1 Purpose of variability model enabling**

##### **5.3.1.1 General**

The purpose of this subprocess is to provide enabling environments for defining and maintaining domain variability models, application variability models and required relations centred on variability models.

##### **5.3.1.2 Inputs**

The following inputs should be available to perform the variability model enabling process:

- variability modelling plan.

##### **5.3.1.3 Outcomes**

The following outcomes shall be available as a result of the successful implementation of the variability model enabling process:

- guidance for variability modelling is defined;
- roles and responsibilities for variability modelling are mobilized;
- variability model-centred variability management enablers are established;
- quality assurance measurement enablers are established.

##### **5.3.1.4 Tasks**

The organization shall implement the following tasks with respect to the variability model enabling process.

- Provide guidance for variability modelling: to define detailed ways to make variability models.
- Mobilize roles and responsibilities for variability modelling: to provide appropriate roles and responsibilities necessary to variability modelling.
- Enable variability model-centric variability management: to provide managerial environments necessary to enabling variability model-centric variability management throughout product line engineering activities.
- Enable variability modelling operations: to provide enablers necessary to variability model design and utilization.
- Enable quality assurance measurement for variability modelling: to provide enablers necessary to measuring quality assurance activities.

### 5.3.2 Provide guidance for variability modelling

The goal of this task is to define detailed guides used for variability modelling. The guidance includes detailed procedures, techniques used at the specific steps, detailed rules and constraints adhered during procedures and required supports for the successful processing.

The method should support providing guidance for variability modelling with the following capabilities:

- defining the contents of the guidance;
- providing examples for each content.

A tool should support providing guidance for variability modelling by allowing the user to do the following:

- providing documentation environment for variability modelling guidance;
- allowing easy reference to each contents of guidance.

### 5.3.3 Mobilize roles and responsibilities for variability modelling

The goal of this task is to make all assigned roles and responsibilities ready for variability modelling.

The method should support mobilizing roles and responsibilities for variability modelling with the following capabilities:

- providing supports required for completing each role and responsibility's assignment;
- allowing early discovering of uncompleted roles and responsibilities.

A tool should support mobilizing roles and responsibilities for variability modelling by allowing the user to do the following:

- allowing traces for each role's current status;
- supporting share of the same views for the roles and responsibilities between managers and persons in charge.

### 5.3.4 Enable variability model-centric variability management

The goal of this task is to provide all necessities required for managing variability by using variability model.

The method should support enabling variability model-centric variability management with the following capabilities:

- providing ways to provide managerial support (traceability management, change management) based on variability model;
- establishing variability model that is proper to support variability model-centred variability management;
- devising traceability and/or change management method that are proper to support variability-centred variability management.

A tool should support enabling variability model-centric variability management by allowing the user to do the following:

- supporting variability modelling functions enabling variability model-centric variability management;
- supporting tracing function enabling variability model-centric variability management.

### 5.3.5 Enable variability modelling operations

The goal of this task is to make variability modelling roles ready for implementing the established variability modelling strategy and plans.

The method should support enabling variability modelling operations with the following capabilities:

- providing variability modelling notations proper to the defined strategy and plans;
- providing ways to scalable variability modelling;
- supporting ways to managing complicated variability models.

A tool should support enabling variability modelling operations by allowing the user to do the following:

- providing graphical notations to model variability;
- implementing ways to scalable variability modelling;
- implementing ways to managing complicated variability models.

### 5.3.6 Enable quality assurance measurement for variability modelling

The goal of this task is to make quality assurance roles ready for measuring the quality of variability model.

The method should support enabling quality assurance measurement for variability modelling with the following capabilities:

- providing enablers (i.e. enabling quality assurance roles, tools, and methods) that make it possible to quality assurance measurement;
- providing enablers that make it possible to improve variability model quality based on quality assurance measurement results.

A tool should support enabling quality assurance measurement for variability modelling by allowing the user to support enablers that interact with each other.

## 5.4 Variability model managing

### 5.4.1 Purpose of variability model managing

#### 5.4.1.1 General

The purpose of this subprocess is to provide managerial support for domain variability models and application variability models.

#### 5.4.1.2 Inputs

The following inputs should be available to perform the variability model managing process:

- variability modelling plans;
- variability modelling status data;
- variability management-centric variability management data.

#### 5.4.1.3 Outcomes

The following outcomes shall be available as a result of the successful implementation of the variability model managing process:

- plans versus actuals are reviewed;
- quality measures for variability modelling are en-valued;
- corrective actions are performed and the list including their status is maintained;
- inputs for improvement are generated.

#### 5.4.1.4 Tasks

The organization shall implement the following tasks with respect to the variability model managing process.

- Review the plan versus actual result of variability modelling: to assign the quality objectives achieved through variability modelling with their pre-defined measures and metrics and monitor whether they are operated in line with the pre-defined plans.
- Control issues on domain/application variability modelling: to examine issues raised during variability modelling in domain engineering and during using variability model in application engineering so as to find obstacles for achieving quality objectives and ways to remove the obstacles.
- Control issues on variability model-centred variability management: to examine issues raised during variability management centred on variability model.
- Control issues on variability model support: to examine issues raised during variability model support activities.
- Support corrective actions for variability modelling: to help repair variability models.
- Make improvement actions for variability modelling: to improve variability models so as to achieve the assigned quality objectives.

#### 5.4.2 Review the plan versus actual of variability modelling

The goal of this task is to check the progress of variability modelling.

The method should support reviewing the plan versus actual result of variability modelling with the following capabilities:

- defining monitoring procedures for variability modelling status compared to plans;
- providing ways to collect data for comparing plan versus actual variability model relevant operations;
- defining ways to integrate data for judging the status.

A tool should support reviewing the plan versus actual result of variability modelling by allowing the user to do the following:

- sharing consensus on the defined roles and responsibilities;
- providing (semi-) automated measurement environment for data collection and integration;
- supporting visualization for the plan versus actual status.

### 5.4.3 Control issues on domain/application variability modelling

The goal of this task is to collect and review issues raised during domain and application variability modelling.

The method should support controlling issues on domain/application variability modelling with the following capabilities:

- providing ways to find and review obstacles to the successful operation of variability modelling;
- defining decision criteria for classifying issues required for corrective actions;
- providing ways to defining corrective actions based on the type of obstacles reviewed in 'review issues of variability modelling task';
- providing documentation templates for recoding assessment results and corrective action plans.

A tool should support controlling issues on domain/application variability modelling by allowing the user to do the following:

- supporting issues collection raised by different roles and responsibility;
- allowing documentation for assessment results and corrective action plans.

### 5.4.4 Control issues on variability model-centred variability management

The goal of this task is to collect and review issues raised during managing variability based on orthogonal variability model.

The method should support controlling issues on variability model-centred variability management with the following capabilities:

- providing ways to find and review obstacles to the successful operation of variability model-centred variability management;
- defining decision criteria for classifying issues required for corrective actions;
- providing ways to defining corrective actions based on the type of obstacles reviewed in review issues of variability model-centred variability management task;
- providing documentation templates for recoding assessment results and corrective action plans.

A tool should support controlling issues on variability model-centred variability management by allowing the user to do the following:

- supporting issues collection raised by different roles and responsibility;
- allowing documentation for assessment results and corrective action plans.

### 5.4.5 Control issues on variability model support

The goal of this task is to collect and review issues raised on managerial supports and enablers for variability modelling.

The method should support controlling issues on variability model support with the following capabilities:

- providing ways to find and review obstacles to the successful operation of variability model support;
- defining decision criteria for classifying issues required for corrective actions;
- providing ways to defining corrective actions based on the type of obstacles reviewed in review issues of variability model supporting task;

- providing documentation templates for recoding assessment results and corrective action plans.

A tool should support controlling issues on variability model support by allowing the user to do the following:

- supporting issues collection raised by different roles and responsibility;
- allowing documentation for assessment results and corrective action plans.

#### **5.4.6 Support corrective actions for variability modelling**

The goal of this task is to implement corrective actions for resolving issues raised on variability modelling.

The method should support corrective actions for variability modelling with the following capabilities:

- providing ways to monitor and control the status of corrective actions;
- communicating corrective action results with the relevant participants.

A tool should support corrective actions for variability modelling by allowing the user to do the following:

- supporting simulation for confirming the achievement of quality level related to the variability modelling operation;
- allowing traces for the status of corrective actions.

#### **5.4.7 Make improvement actions for variability modelling**

The goal of this task is to define and implement improvement actions for variability modelling, management and supports.

The method should support making improvement actions for variability modelling with the following capabilities:

- collecting the improvement items of variability modelling operation;
- implementing the improvement items of variability modelling operation.

A tool should support making improvement actions for variability modelling by allowing the user to do the following:

- supporting improvement input collection;
- providing communication environment among the relevant participants.

## **6 Variability modelling**

### **6.1 General**

Domain variability models are defined and refined during domain engineering and application variability models are defined and refined by binding variability and adding new application specific variability to domain variability models. Because variability in a product line is orthogonally modelled from artefacts relations from variability models to corresponding artefacts should be established.

The variability modelling process includes the following subprocesses:

- domain variability modelling;
- application variability modelling;

- relating variability model to artefacts;
- relating domain variability model to application variability model.

## 6.2 Domain variability modelling

### 6.2.1 Purpose of domain variability modelling

#### 6.2.1.1 General

The purpose of this subprocess is to make domain variability modelling for a product line. There exist two distinct approaches for variability modelling: ad-hoc (conventional) variability modelling that includes variability information in some software modelling notation (e.g. UML, feature, etc.) and orthogonal variability modelling that integrates all the variability information in a single model. The variability models alone cannot represent the full contents of variability information in both approaches for variability modelling. The development artefacts of domain/application engineering and the variability model need to be traced.

#### 6.2.1.2 Inputs

The following inputs should be available to perform the domain variability modelling process:

- variability modelling plan;
- defined domain variability during domain engineering;
- decision tables.

#### 6.2.1.3 Outcomes

The following outcomes shall be available as a result of the successful implementation of the domain variability modelling process:

- domain variability model is established.

#### 6.2.1.4 Tasks

The organization shall implement the following tasks with respect to the domain variability modelling process.

- Construct domain variability model: to model variability of a domain with an orthogonal approach.
- Annotate domain variability model: to add annotations to domain variability model for supporting the right bindings.
- Verify domain variability model: to help ensure that constructed domain variability models are correct, complete and consistent.
- Optimize domain variability model: to make domain variability models optimal by fine-tuning variability and its relationship.

### 6.2.2 Construct domain variability model

The goal of this task is to model domain variability in accordance with the defined variability model structure and its elements.

The method should support constructing domain variability model with the following capabilities:

- defining variability model units (e.g. variation point, variants, variability dependency, constraints);

- defining other variability model units required for configuration and management (e.g. required trace links with development artefacts for deriving a member product);
- allowing modularization for modelling large scale of systems;
- allowing variability visibility;
- deciding variability representation way (e.g. orthogonal way, notations, use of data types);
- discriminating the relevant development stage of variability (i.e. variability in requirements, variability in design).

A tool should support constructing domain variability model by allowing the user to do the following:

- allowing variability modelling by using notations for the defined variability model units;
- allowing variability visibility setting;
- supporting variability grouping for the handling complexity of variability models;
- providing a way to discriminating the relevant development stage of variability.

### 6.2.3 Annotate domain variability model

The goal of this task is to add information (e.g. description for variability, type of a variant if necessary, or value constraints) for the right use of variability.

The method should support annotating domain variability model with the following capabilities:

- providing a standardized way for describing binding times;
- defining other annotations required for supporting the right binding;
- defining notations for defining annotations.

A tool should support annotating domain variability model by allowing the user to do the following:

- allowing adding annotations in variability model;
- supporting extending/shrinking operation for reducing complexity due to annotations.

### 6.2.4 Verify domain variability model

The goal of this task is to analyse variability model whether it includes void, dead, false option, or redundancy variants.

The method should support verifying domain variability model with the following capabilities:

- providing algorithms to support analysis operation (e.g. void variability, valid products) on domain variability model;
- supporting analysis operation for detecting anomalies (e.g. dead and false variability, wrong cardinality) in domain variability model;
- providing descriptions about the reasons related to the analysis operation;
- providing corrective explanations.

A tool should support verifying domain variability model by allowing the user to do the following:

- automating the algorithms to support analysis operation on domain variability model;
- automating the analysis operation for detecting anomalies in domain variability model;



- supporting (semi-) automated description generation;
- supporting (semi-) automated corrective explanations generation.

NOTE Example errors that variability modelling tool should validate automatically are depicted in [Annex E](#).

### 6.2.5 Optimize domain variability model

The goal of this task is to optimize domain variability model by resolving void, dead, false option, or redundancy variants.

The method should support optimizing domain variability model with the following capabilities:

- revising domain variability model in accordance with the results of analysis operations;
- verifying optimized domain variability model.

A tool should support optimizing domain variability model by allowing the user to do the following:

- providing environment for revising domain variability model with the recommendation;
- supporting instant verification for the optimized domain variability model.

## 6.3 Application variability modelling

### 6.3.1 Purpose of application variability modelling

#### 6.3.1.1 General

The purpose of this subprocess is to define and maintain application variability models including binding results and application-specific variabilities.

#### 6.3.1.2 Inputs

The following inputs should be available to perform the application variability modelling process:

- variability modelling plan;
- defined application specific variability during application engineering;
- decision tables.

#### 6.3.1.3 Outcomes

The following outcomes shall be available as a result of the successful implementation of the application variability modelling process:

- application variability model is established.

#### 6.3.1.4 Tasks

The organization shall implement the following tasks with respect to the application variability modelling process:

- Construct application variability model: to model a member product specific variability.
- Annotate application variability model: to add annotations to application variability model for the right use of application variability model.
- Verify application variability model: to help ensure that constructed application variability models are correct, complete and consistent.

- Optimize application variability model: to make application variability models optimal by fine-tuning variability and its relationship.

### 6.3.2 Construct application variability model

The goal of this task is to model application variability by binding and adding application specific variability.

The method should support constructing application variability model with the following capabilities:

- allowing modularization for modelling large scale of systems;
- providing a way for application specific variability representation;
- allowing the visibility of application specific variability;
- defining a way for discriminating the resolution results of variability delta.

A tool should support constructing application variability model by allowing the user to do the following:

- allowing application variability modelling by using notations for the defined application specific variability;
- supporting variability grouping for the handling complexity of application variability model in line with domain variability model;
- allowing variability visibility setting;
- providing a way to discriminating application specific variability.

### 6.3.3 Annotate application variability model

The goal of this task is to add information (e.g. rationale for binding, types or value constraints for application specific variants) for capturing information for users taking decisions (e.g. a realizing engineer realizes the right member product).

The method should support annotating application variability model with the following capabilities:

- providing a standardized way for describing rationale for binding decision;
- defining other annotations required for explaining application specific variability including resolution for variability delta;
- defining notations for defining application specific annotations.

A tool should support annotating application variability model by allowing the user to do the following:

- allowing adding annotations in application variability model;
- supporting extending/shrinking operation for application specific annotations.

### 6.3.4 Verify application variability model

The goal of this task is to analyse variability model whether it includes inconsistent application specific variability.

The method should support verifying application variability model with the following capabilities:

- providing algorithms to support analysis operation on application variability model, whether it is consistent with domain variability model and adheres to the rules and constraints defined in domain variability model;

- supporting analysis operation for detecting anomalies in application variability model;
- providing descriptions about the reasons related to the analysis operation;
- providing corrective explanations.

A tool should support verifying application variability model by allowing the user to do the following:

- automating the algorithms to support analysis operation;
- automating the analysis operation for detecting anomalies in application variability model;
- supporting (semi-) automated description generation;
- supporting (semi-)automated corrective explanations generation.

### **6.3.5 Optimize application variability model**

The goal of this task is to optimize application variability model by resolving inconsistent application specific variability.

The method should support optimizing application variability model with the following capabilities:

- revising application variability model in accordance with the results of analysis operations;
- verifying optimized application variability model.

A tool should support optimizing application variability model by allowing the user to do the following:

- providing environment for revising application variability model with the recommendation;
- supporting instant verification for the optimized application variability model.

## **6.4 Relating variability model to artefacts**

### **6.4.1 Purpose of relating variability model to artefacts**

#### **6.4.1.1 General**

The purpose of this subprocess is to establish and maintain relations from variability models to development artefacts for supporting traceability management.

#### **6.4.1.2 Inputs**

The following inputs should be available to perform the relating variability model to artefacts process:

- variability modelling plan;
- domain variability model;
- application variability model;
- decision tables.

#### **6.4.1.3 Outcomes**

The following outcomes shall be available as a result of the successful implementation of the relating variability model to artefacts process:

- relations between domain variability model and domain artefacts are established;
- relations between application variability model and application artefacts are established.

#### 6.4.1.4 Tasks

The organization shall implement the following tasks with respect to the relating variability model to artefacts process.

- Retrieve variation points and variants in relevant artefacts: to identify artefacts that include variation points or related to variants among domain and application artefacts.
- Relate domain variability model to domain artefacts: to establish relation from the element of domain variability model to relevant variation point of a domain artefact or to relevant variable artefact.
- Relate application variability model to application artefacts: to establish relation from the element of application variability model to relevant application specific artefact.

#### 6.4.2 Retrieve variation points and variants in relevant artefacts

The goal of this task is to locate a variation point and its variants in relevant development artefacts.

The method should support retrieving variation points and variants in relevant artefacts with the following capabilities:

- providing a way to separate a variation point included in artefacts (e.g. trace link, naming);
- providing a way to separate artefacts related to the specific variant (e.g. trace link, naming).

A tool should support retrieving variation points and variants in relevant artefacts by allowing the user to do the following:

- allowing artefact retrieval for the selected variation points;
- allowing automatic artefact retrieval for the selected variation point;
- allowing automatic artefacts retrieval for the selected variants.

#### 6.4.3 Relate domain variability model to domain artefacts

The goal of this task is to establish relations from variability model elements to the identified domain artefacts.

The method should support relating domain variability model to domain artefacts with the following capabilities:

- defining necessary relations from domain variability model elements to the relevant domain artefacts;
- providing a way to trace the defined relations.

A tool should support relating domain variability model to domain artefacts by allowing the user to do the following:

- storing relations from domain variability model elements to domain artefacts;
- allowing automatic traces to the defined relations.

#### 6.4.4 Relate application variability model to application artefacts

The goal of this task is to establish relations from application specific variability model elements and the identified application artefacts.

The method should support relating application variability model to application artefacts with the following capabilities:

- defining necessary relations from application variability model elements to the relevant application artefacts;
- providing a way to trace the defined relations.

A tool should support relating application variability model to application artefacts by allowing the user to do the following:

- storing relations from application variability model elements to domain artefacts;
- allowing automatic traces to the defined relations.

## **6.5 Relating domain variability model to application variability model**

### **6.5.1 Purpose of domain variability model to application variability model**

#### **6.5.1.1 General**

The purpose of this subprocess is to establish and maintain relations from domain variability model to application variability model so as to support traceability management.

#### **6.5.1.2 Inputs**

The following inputs should be available to perform the domain variability model to application variability model process:

- variability modelling plan;
- domain variability model;
- binding decisions;
- application specific variability;
- application variability model.

#### **6.5.1.3 Outcomes**

The following outcomes shall be available as a result of the successful implementation of the domain variability model to application variability model process:

- relations between domain and application variability models are established.

#### **6.5.1.4 Tasks**

The organization shall implement the following tasks with respect to the domain variability model to application variability model process.

- Trace binding decisions made in an application: to identify selected variants during a member product development.
- Establish relations between domain and application variability models: to establish relations between the elements of domain variability model and application variability model.
- Add decision-related annotations to relations: to amplify on the relations between domain and application variability model so as to support application engineers' correct understanding with regard to variability.

- Verify relations between domain and application variability models: to help ensure that relations between domain and application variability models are correct, complete and consistent.

### **6.5.2 Trace binding decisions made in an application**

The goal of this task is to allow tracing all binding decisions made during application engineering.

The method should support tracing binding decisions made in an application with the following capabilities:

- recoding binding decisions in the form of easily being traced;
- allowing referring binding decision relevant information (e.g. rationale).

A tool should support tracing binding decisions made in an application by allowing the user to do the following:

- allowing tracing binding decisions for each variability;
- allowing referring binding decision relevant information with tracing.

### **6.5.3 Establish relations between domain and application variability models**

The goal of this task is to make relations that make it possible to trace from domain variability model to application variability model.

The method should support establishing relations between domain and application variability models with the following capabilities:

- providing a way to make relation between variation points of domain variability model and application variability model;
- providing a way to make relation between variants of domain variability model and those selected in application variability model;
- allowing tracing from domain variability model to application variability model through the established relations.

A tool should support establishing relations between domain and application variability models by allowing the user to do the following:

- implementing the way to make relations between domain variability model and application variability model;
- allowing automatic traces from domain variability model to application variability model.

### **6.5.4 Add decision-related annotations to relations**

The goal of this task is to annotate additional information to the established relations. Decision-related annotations include weak constraints with respect to binding, adaptation tips, short explanations for variation point and its variants and so on.

The method should support adding decision-related annotations to relations with the following capabilities:

- providing annotation types to be added;
- supporting a way to add annotations as a part of relations.

A tool should support adding decision-related annotations by allowing the user to do the following:

- supporting addition of the defined annotation types;

- allowing addition of annotations to relations.

### 6.5.5 Verify relations between domain and application variability models

The goal of this task is to help ensure the correctness, completeness and consistency of the established relations between domain and application variability models.

The method should support verifying relations between domain and application variability models with the following capabilities:

- providing checklists to be confirmed to help ensure correctness, completeness and consistency;
- providing a way to automatic extract for the suspicious relations.

A tool should support verifying relations between domain and application variability models by allowing the user to do the following:

- providing instant reference for the defined checklists;
- visualizing suspicious relations;
- supporting documentation for negative decisions.

## 7 Variability model support

### 7.1 General

For constructing correct, complete and consistent variability model and relations from variability model to other artefacts of a product line necessary to configuring the right applications, proper supports including methods and tools should be provided.

The variability model support process includes the following subprocesses:

- relating variability model to variability mechanism;
- quality assurance for variability model;
- binding decision support;
- application configuration support.

### 7.2 Relating variability model to variability mechanism

#### 7.2.1 Purpose of relating variability model to variability mechanism

##### 7.2.1.1 General

The purpose of this subprocess is to establish and maintain relations from variability model viewpoint with relevant variability mechanisms.

##### 7.2.1.2 Inputs

The following inputs should be available to perform the relating variability model to variability mechanism process:

- domain variability models;
- variability mechanism constraints (e.g. binding times, specific implementation constraint) related to variability;

- variability mechanisms used.

#### **7.2.1.3 Outcomes**

The following outcome shall be available as a result of the successful implementation of the relating variability model to variability mechanism process:

- relations including variability mechanism constraints from variability models to variability mechanisms are established.

#### **7.2.1.4 Tasks**

The organization shall implement the following tasks with respect to the relating variability model to variability mechanism process.

- Identify variability including variability mechanism constraints: to grasp variability realized by using mechanism that has specific constraints such as binding time restriction.
- Establish relations from variability model to variability mechanism: to refine trace links by reflecting the variability model perspective with respect to variability mechanisms.
- Add variability mechanism constraint annotations into variability model: to annotate variability mechanism specific restriction to the trace links from variability model to variability mechanism.

### **7.2.2 Identify variability including variability mechanism constraints**

The goal of this task is to identify the relationships between the variability model and variability mechanisms with significant or complicated constraints. Such constraints include safety critical requirements and complicate composition rules.

The method should support identifying variability including variability mechanism constraints with the following capabilities:

- providing guidance (including example constraints) to identify variability mechanisms that require establishing of relations;
- providing a way to make decisions for selecting mechanisms.

A tool should support identifying variability including variability mechanism constraints by allowing the user to do the following:

- allowing instant reference to the guidance;
- supporting documentation for decisions.

### **7.2.3 Establish relations from variability model to variability mechanism**

The goal of this task is to make relations from variability model to mechanisms identified.

The method should support establishing relations from variability model to variability mechanism with the following capabilities:

- providing a way to establish relations from variability model to mechanisms;
- providing a way to trace the established relations.

A tool should support establishing relations from variability model to variability mechanism by allowing the user to do the following:

- providing tool supported relation management for variability model and mechanism;
- allowing automatic tracing from variability model to variability mechanism.



#### 7.2.4 Add variability mechanism constraint annotations into variability model

The goal of this task is to annotate the relations among the identified important/complicated variability mechanism constraints.

The method should support adding variability mechanism constraint annotations into variability model with the following capabilities:

- providing annotation types to be added;
- supporting a way to add annotations as the part of relations.

A tool should support adding variability mechanism constraint annotations into variability model by allowing the user to do the following:

- supporting addition of the defined annotation types;
- allowing addition of annotations to relations.

### 7.3 Quality assurance for variability model

#### 7.3.1 Purpose of quality assurance for variability model

##### 7.3.1.1 General

The purpose of this subprocess is to help ensure that variability models and their relevant processes comply with predefined provisions and plans.

##### 7.3.1.2 Inputs

The following inputs should be available to perform the quality assurance for variability model process:

- domain variability model;
- application variability model;
- relations established with variability models;
- processes used for variability modelling;
- quality assurance measures defined in variability modelling planning.

##### 7.3.1.3 Outcomes

The following outcomes shall be available as a result of the successful implementation of the quality assurance for variability model process:

- variability model-specific quality assurance strategy is developed;
- evidences of variability model quality assurance are produced and maintained;
- non-conformance issues in variability models are identified and recorded;
- adherence of variability models and variability modelling processes to the PL organizationally adopted standards, rules and constraints are assured.

#### 7.3.1.4 Tasks

The organization shall implement the following tasks with respect to the quality assurance for variability model process:

- Objectively evaluate variability modelling activities: to help ensure that employed activities for variability modelling comply with the provisions and plans.
- Objectively evaluate variability model work products: to help ensure that produced planned artefacts satisfy the predefined quality criteria.
- Communicate and resolve noncompliance issues: to help ensure that the noncompliance issues of variability modelling are objectively tracked, communicated and resolved with appropriate staffs and managers.
- Establish records of variability modelling quality assurance activities: to record and revise quality assurance activities performed with respect to variability modelling.

#### 7.3.2 Objectively evaluate variability modelling activities

The goal of this task is to assure whether the conducted variability model activities adhere to the defined processes.

The method should support objectively evaluating variability modelling activities with the following capabilities:

- assuring employed variability model management and variability modelling activities;
- assuring performed variability management and variability modelling practices;
- assuring measurement processes related to variability modelling;
- assuring activities related to variability modelling supports (e.g. binding decision, application configuration).

A tool should support objectively evaluating variability modelling activities by allowing the user to do the following:

- supporting the access of PL organization's process definitions related to variability modelling (maintaining organization's process repository);
- accessing performed variability modelling practices for quality assurance;
- allowing the access of performed measurement activities performed for variability model management and variability modelling.

#### 7.3.3 Objectively evaluate variability model work products

The goal of this task is to assure the produced planned work products during variability modelling and variability model management.

Variability model work products include domain and application variability models including rules and constraints among variability, variability model validation results, traceability related to variability model and so on.

The method should support objectively evaluating variability model work products with the following capabilities:

- assuring variability modelling work products;
- assuring relations that they are correct, complete and consistent with other relevant relations established from variability modelling perspective;

- assuring the work products of variability modelling supports;
- assuring provided variability modelling supports (e.g. tools, methods, resources).

A tool should support objectively evaluating variability model work products by allowing the user to do the following:

- allowing the access of variability modelling work products (e.g. establishing and maintaining repository for variability modelling work products);
- allowing relation tracking for established trace links (e.g. traceability table or explicit link to trace);
- supporting the access of variability modelling support work products;
- providing a way to measure the satisfaction degree of the provided supports for variability model management and operation.

#### **7.3.4 Communicate and resolve noncompliance issues**

The goal of this task is to define action items to resolve noncompliance issues found with relevant participants.

The method should support communicating and resolving noncompliance issues with the following capabilities:

- establishing escalation lines for resolving noncompliance issues when they cannot be resolved with the appropriate variability modelling staffs (the escalation line might include variability modelling staffs, appropriate domain engineers, or application engineers);
- tracking noncompliance issues throughout the established escalation lines;
- analysing noncompliance issues if there are any variability relevant quality trends (i.e. right implementation, right binding or reuse with low cost in application engineering).

A tool should support communicating and resolving noncompliance issues by allowing the user to do the following:

- supporting documentation for noncompliance issues to be escalated;
- providing communication space among staffs or managers within the established escalation lines;
- sharing the status of noncompliance issues among staffs and managers within the established escalation lines
- supporting statistical analysis for variability relevant quality trends.

#### **7.3.5 Establish records of variability modelling quality assurance activities**

The goal of this task is to document quality assurance activities with their outcomes.

The method should support establishing records of variability modelling quality assurance activities with the following capabilities:

- recording and revising variability modelling quality assurance activities.

A tool should support establishing records of variability modelling quality assurance activities by allowing the user to do the following:

- supporting documentation for variability modelling quality assurance activities.

## 7.4 Binding decision support

### 7.4.1 Purpose of binding decision support

#### 7.4.1.1 General

The purpose of this subprocess is to support correct binding decisions from variability modelling viewpoint.

#### 7.4.1.2 Inputs

The following inputs should be available to perform the binding decision support process:

- binding decision tables;
- domain variability models.

#### 7.4.1.3 Outcomes

The following outcomes shall be available as a result of the successful implementation of the binding decision support process:

- binding decision tables are referred and conformed;
- binding decisions are verified from variability model viewpoint.

#### 7.4.1.4 Tasks

The organization shall implement the following tasks with respect to the binding decision support process.

- Establish full of references to binding decision tables: to enable to access all kinds of information of binding decision tables.
- Verify binding decisions from variability models view: to help ensure that variability model annotation(s) with regard to binding decisions is correct.

NOTE An example variability decision table format is depicted in [Annex D](#).

### 7.4.2 Establish full of references to binding decision tables

The goal of this task is to make relevant participants possible to refer binding decision tables so that they can make the right binding relevant decisions. Binding decision table includes possible binding times, variants selection cardinality, binding rules and constraints.

The method should support establishing full of references to binding decision tables with the following capabilities:

- supporting bringing up of relevance decision tables for the selected variability;
- extracting relevant decision support information from binding decision tables.

A tool should support establishing full of references to binding decision tables by allowing the user to do the following:

- providing a connector for referring binding decision tables;
- supporting instant reference for the working currently variability binding decisions.

### 7.4.3 Verify binding decisions from variability models view

The goal of this task is to help ensure the correctness of binding decisions from variability model.

The method should support verifying binding decisions from variability models view with enabling binding results verification in accordance with variant selection.

A tool should support verifying binding decisions from variability models view by allowing the user to do the following:

- supporting immediate verification in accordance with variant selection;
- supporting immediate display for the verification result;
- providing recommendation for the right decision.

## 7.5 Application configuration support

### 7.5.1 Purpose of application configuration support

#### 7.5.1.1 General

The purpose of this subprocess is to support the derivation of correct, consistent and complete applications by applying bindings.

#### 7.5.1.2 Inputs

The following inputs should be available to perform the application configuration support process:

- variability modelling plan;
- domain variability model;
- binding decision tables;
- domain artefacts;
- application variability models.

#### 7.5.1.3 Outcomes

The following outcomes shall be available as a result of the successful implementation of the application configuration support process:

- relations between variability models and binding decision tables are established;
- different variability model views by the binding times of variabilities are defined;
- trace links from variability model to domain artefacts are established and verified.

#### 7.5.1.4 Tasks

The organization shall implement the following tasks with respect to the application configuration support process.

- Relate variability models to binding decision tables: to establish relations between variability models and binding decision tables so that application engineers can make right decisions with regard to bindings.
- Provide different views of variability models by binding stages: to enable that variability models can be examined from different angles according to binding times of variability.

- Support full of traces from variability model to artefacts: to realize complete trace links between variability models and artefacts so that right artefacts can be extracted in accordance with the binding results of variability models.

### 7.5.2 Relate variability models to binding decision tables

The goal of this task is to establish relations from variability model to decision table.

The method should support relating variability models to binding decision tables with the following capabilities:

- providing a way to establish and maintain links between variability model and decision table;
- providing a way to display binding decision table in accordance with user selection or assigned significance level of variability.

A tool should support relating variability models to binding decision tables by allowing the user to do the following:

- supporting explicit traces between variability model and decision table;
- enabling the selective display of binding decision tables in accordance with user selection or significance level of variability.

### 7.5.3 Provide different views of variability models by binding stages

The goal of this task is to enable to capture variability models from different viewpoints for providing insight into the variability.

Variability viewpoints include stakeholder specific viewpoints such as business user and personal user. Concerns such as secure usage and easy usage can be used as viewpoints.

The method should support providing different views of variability models by binding stages with the following capabilities:

- separating variabilities related to the specific viewpoint;
- constructing variability model by using separated variabilities;
- validating consistency among variability models for different viewpoints;
- providing a way to combine variability models in different viewpoints.

A tool should support providing different views of variability models by binding stages by allowing the user to do the following:

- providing explicit display that will be separated in accordance with the selected viewpoint;
- implementing variability model construction function for the selected viewpoint;
- supporting automatic consistency validation among variability models for different viewpoints;
- enabling combination of variability models in different viewpoints.

### 7.5.4 Support full of traces from variability model to artefacts

The goal of this task is to establish and maintain trace links from variability model to artefacts. These trace links deal with traces only from variability model viewpoint.

The method should support supporting full of traces from variability model to artefacts with the following capabilities:

- supporting a way to find trace links omitted;
- providing a way to combine with other trace links.

A tool should support supporting full of traces from variability model to artefacts by allowing the user to do the following:

- supporting search for omitted trace links;
- providing the combined management of newly searched trace links with other trace links.

## Annex A (informative)

### Variability meta model

The variability meta model shown in [Figure A.1](#) describes the principles of variability in SSPL and the orthogonal variability model. The basic elements of the OVM are variation points and variants. Variation point is specialized into the two types: internal variation point and external variation point. A “variability dependency” association exists between a variation point and its variants. “Alternative choice” is a special association of “optional” dependency. And there are two types of constraint dependency: requires and excludes, of which possible types are “variant to variant”, “variant to variation point” and “variation point to variation point”. Lastly, artefact dependency from variability model to development artefacts’ exists. It has two types of dependency: “represented by” and “realized by”.

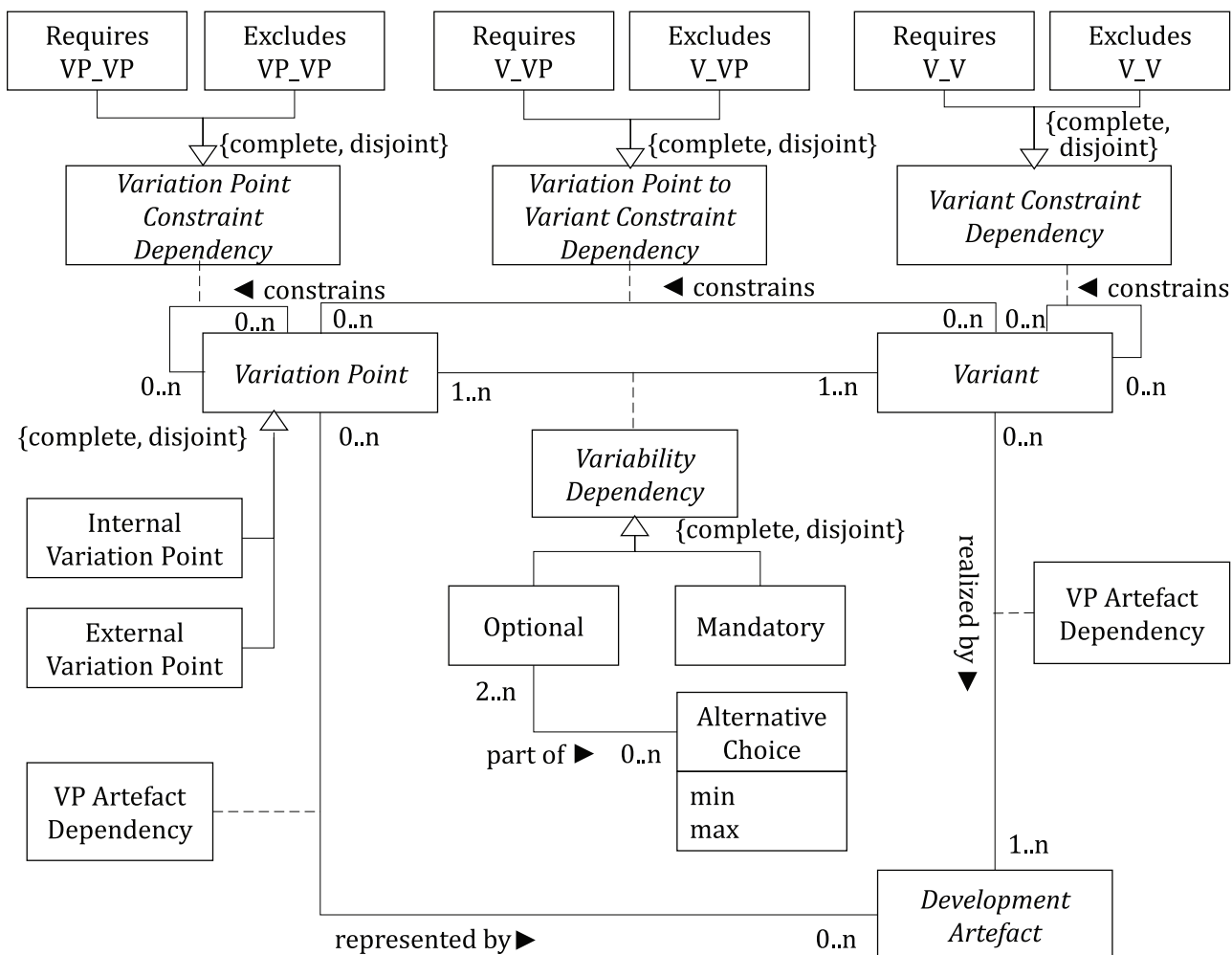


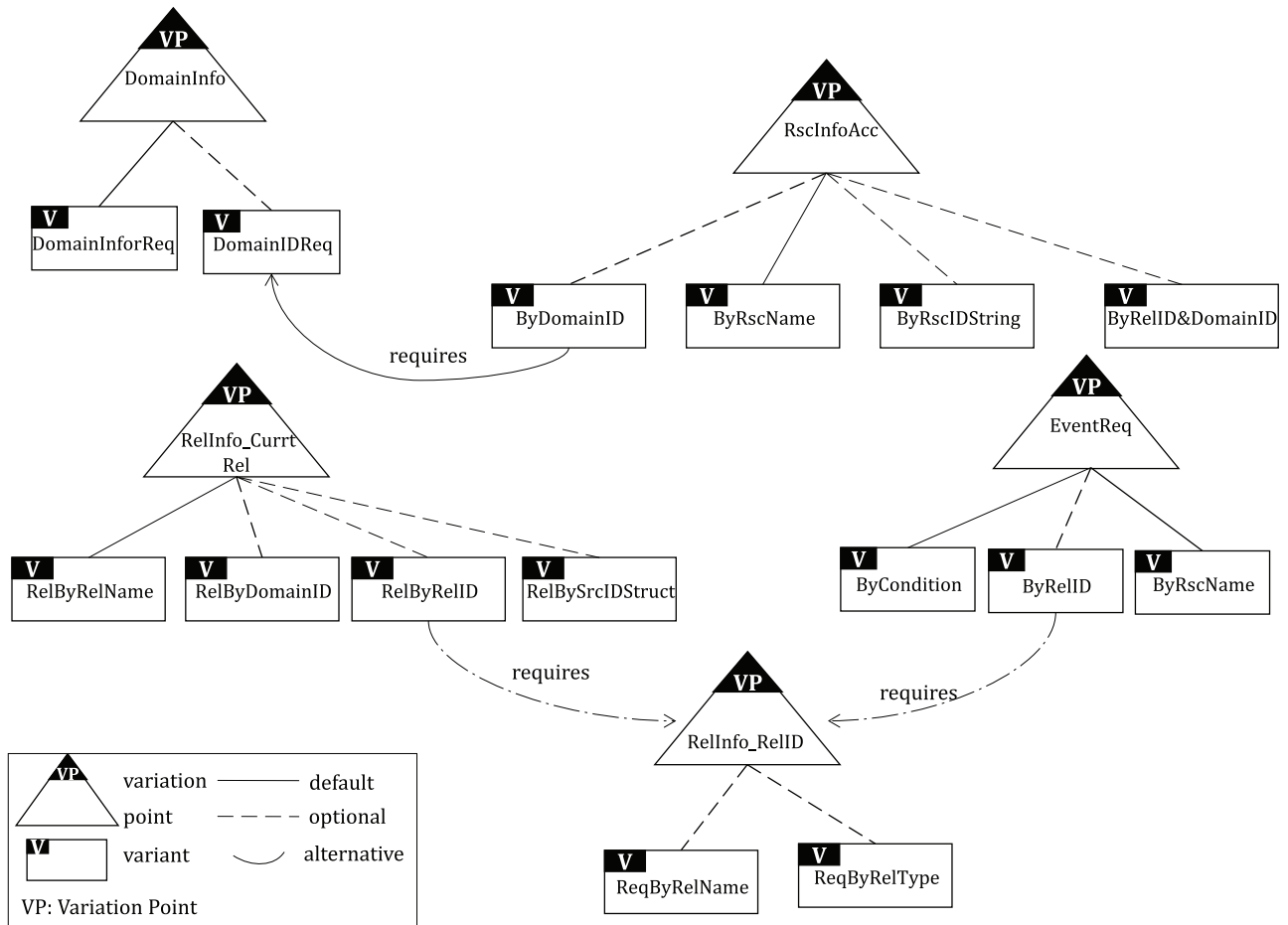
Figure A.1 — Orthogonal variability meta model[5]



## Annex B (informative)

### Orthogonal variability model

The variability model in [Figure B.1](#) is an example of an orthogonal variability model. Notations used are those proposed by Reference [5].



**Figure B.1 — Example orthogonal variability model**

## Annex C (informative)

### Formal descriptions for variability relationships

[Table C.1](#) provides formal descriptions for the variability dependency and variability constraints that should be modelled through the OVM.

**Table C.1 — Formal description for the defined OVM variability relationships**

OVM variability relationship	Description
Exclusive-or/Alternative	$vp: e_1, \dots, e_k$
Inclusive-or	$vp: v = v' \subseteq \{e_1, \dots, e_k\}$ $\min \leq  v'  \leq \max, \min \geq 1$
Optional	(1) Used without <i>inclusive-or</i> $vp: e, null$ (2) <i>Inclusive-or optional</i> Used with <i>inclusive-or</i> and its range is [min..max], $vp: v \subseteq v' \cup \{e\}$ $v' \subseteq \{e_1, \dots, e_k\}, \min \leq  v'  \leq \max$ (3) <i>Exclusive-or optional</i> $vp: e_1, \dots, e_k$ (same meaning with <i>exclusive-or</i> )
Mandatory	(1) Used without <i>inclusive-or</i> $vp: v /*$ special case of <i>exclusive-or */ (2) Used with <i>exclusive-or</i> and range is [min..max], <math>vp: \{e\} \subseteq v \subseteq \{e, e_1, \dots, e_k\}</math> <math>\min+1 \leq  v  \leq \max+1</math> (3) In the case of being used with <i>exclusive-or</i> its meaning is same with case (1).</i>
Requires	$(vp1) \text{ req } (vp2, v), (vp1) \text{ req } (vp2), (vp1, v1) \text{ req } (vp2, v2), (vp1, v) \text{ req } (vp2)$
Excludes	$(vp1) \text{ excl } (vp2, v), (vp1) \text{ excl } (vp2), (vp1, v1) \text{ excl } (vp2, v2), (vp1, v) \text{ excl } (vp2)$
$vp$ Variation point. $v$ Variant. $\text{req}$ Requires. $\text{excl}$ Excludes.	

The defined elements of the OVM can be modelled by various notations, but the notations should adhere to the semantic and syntactic meanings. In a product line, there are hundreds of variabilities, so the modeller that supports the OVM method should be scalable. Moreover, the OVM has the key role for establishing and maintaining end-to-end variability traceability in a product line the appropriate modeller to support the OVM based variability should be developed.

## Annex D (informative)

### Orthogonal variability decision table

The orthogonal variability modelling approach provides a simple way for consistent representation and traceability support in variability modelling, but it suffers from the lack of capability to describe variability with relevant information such as binding time, constraints adhered for the right binding and rationales for decisions. To tackle these difficulties, a tabular format called the variability decision table which is similar to the decision model approach can be used to describe variation points, variants, binding information and dependencies and other constraints.

[Figure D.1](#) shows an example of a variability decision table format in the requirements stage. Variability decision table can be applied throughout all of the SSPL life cycle stages together with the orthogonal variability model. In the first column of the table, which is called “No.”, the ID for each requirement is represented. Then, the name of each requirement is followed in the column called “Var in Req.”. The dependencies and the number of selections denoting variability dependency relations such as exclusive-or, inclusive-or, or optional follow in the third and fourth column. The next columns Product 1 to Product N, planned member products, define the selection of each requirement per product. The constraints among the requirements are listed and in the last column, the binding information is described. In the binding information column, N/A means binding is not conducted at this stage.

A variability decision table is completed easily by using spreadsheets. This brings the following advantages: firstly, it describes clearly a variation point and its variants. Secondly, it is easier to describe binding information than the graphical variability model (the graphical variability model should use much more spaces than tabular formats). Thirdly, it might provide better usability due to its familiar tabular formats.

No.	Var in Reqs.		Dependen cies	# of Selection	Product1	...	ProductN	Constraints	Binding Info.
DFR_A1	VP1	v1	man	1..*	V	V	V		N/A
DFR_A2		v2	opt		V		V		N/A
DFR_A3	VP2	v3	opt	1			V	Requires VP2_v3 VP1_E1	Req. phase
DFR_A4		v4	opt			V	V		N/A

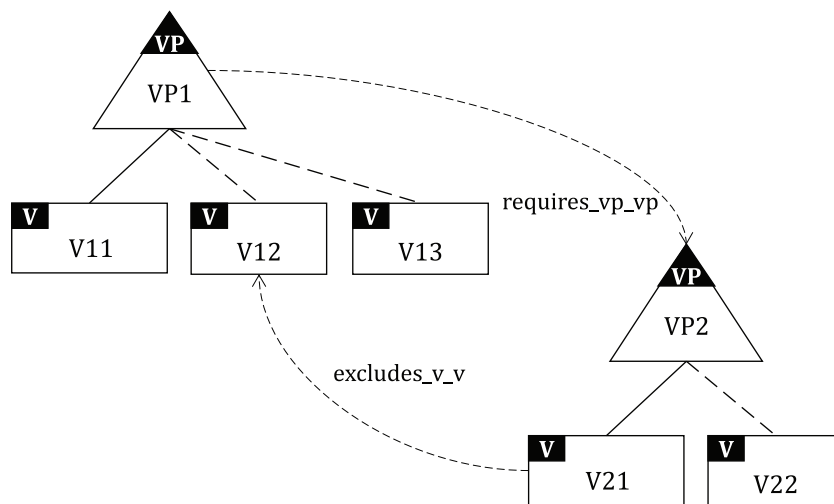
**Figure D.1 — Variability decision table**

## Annex E (informative)

### Orthogonal variability model validation

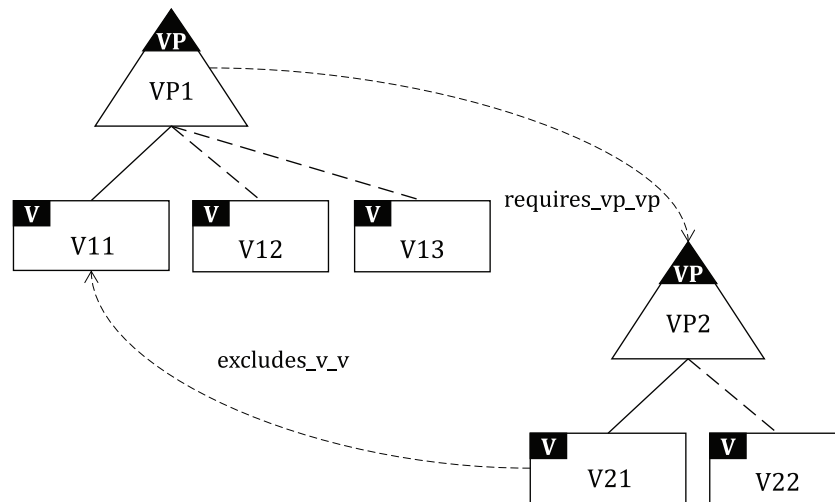
For the correct, consistent and efficient variability modelling, tool support for automatic validation of variability model is essential. Moreover, hundreds number of variability exist in SSPL, so it is difficult to validate the model manually. Therefore, the variability modelling tool should have functions for validating variability model. For providing functions for variability model, validation types of errors should be defined and formally defined. The following are example errors that the variability modelling tool should validate automatically.

- Dead variant: Variants never appear in any member products. For example, in [Figure E.1](#), V12 never appears in any member products because variation point VP1 requires VP2 and mandatory variant V21 excludes V12.



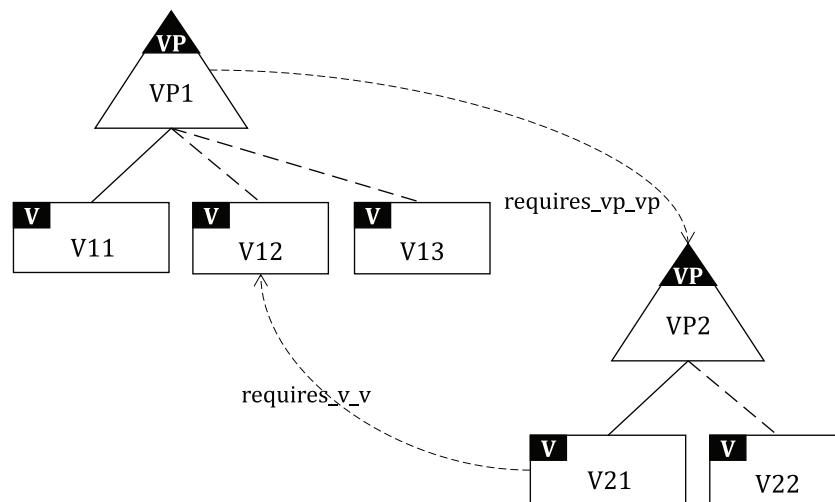
**Figure E.1 — Example of dead variant**

- False mandatory variant: A variant cannot be selected even if it is a mandatory variant. Variant V11 is a mandatory variant of VP1, but it cannot be selected because variant V21 of VP2 excludes V11 (see [Figure E.2](#)).



**Figure E.2 — Example of false mandatory variant**

- False optional variant: A variant is always selected although it is an optional variant. Though V12 is an optional variant, it is always selected because variation point VP1 requires VP2 and mandatory variant V21 requires V12. An example is shown in [Figure E.3](#).



**Figure E.3 — Example of false optional variant**

## Bibliography

- [1] ISO/IEC 14102, *Information technology — Guideline for the evaluation and selection of CASE tools*
- [2] ISO/IEC 15940, *Systems and software engineering — Software Engineering Environment Services*
- [3] ISO/IEC/TR 19759, *Software Engineering — Guide to the software engineering body of knowledge (SWEBOK)*
- [4] ISO/IEC 25000, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE*
- [5] POHL K., BÖCKLE G., VAN DER LINDEN F.J. *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer, 2005
- [6] PAUL C. *A Framework for Software Product Line Practice*, Version 5.0. Software Engineering Institute, Carnegie Mellon University, July 2007
- [7] CAPILLA R., BOSCH J., KANG K.C. *Systems and Software Variability Management Concepts, Tools and Experiences*. Springer, 2013



