

# Abordagem SyMPLES v1.0

## I. Visão Geral da Abordagem SyMPLES v1.0

A abordagem *SysML-based Product Line Approach for Embedded Systems (SyMPLES)* tem por objetivo apoiar o desenvolvimento de Linhas de Produto de Software (LP) baseadas na linguagem SysML para o domínio de Sistemas Embarcados (SE).

A abordagem SyMPLES contém dois perfis, que são extensões para a linguagem SysML, criados por um mecanismo de *profiling*: (i) o *SyMPLES Profile for Functional Blocks (SyMPLES-ProfileFB)*; e (ii) o *SyMPLES Profile for Representation of Variability (SyMPLES-ProfileVar)*. O objetivo do *SyMPLES-ProfileFB* é fornecer uma semântica adicional aos blocos SysML, enquanto que o *SyMPLES-ProfileVar* visa a possibilitar a representação de artefatos da LP para SE baseada em SysML.

## II. A abordagem SMarty

A abordagem *SMarty* foi inicialmente aplicada a modelos UML e posteriormente estendida para SysML para apoiar o método SyMPLES.

## III. Estereótipos e Diretrizes

Nesta seção, são apresentados os estereótipos do perfil *SyMPLES-ProfileVar* para aplicação em diagramas SysML, por meio da Tabela II. Em seguida, é apresentado um exemplo do uso destes, seguidos das diretrizes para cada tipo de modelo.

Tabela II – Estereótipos do perfil SyMPLES-ProfileVar

Estereótipos do perfil SyMPLES-ProfileVar	
Estereótipo	Utilização
<<variationPoint>>	Representa o local em que ocorre uma variabilidade. Um ponto de variação está sempre associado a uma ou mais variantes.
<<mandatory>>	A variante estará obrigatoriamente presente na configuração de qualquer produto da linha de produto.
<<optional>>	A variante pode ou não estar presente na configuração de um produto da linha de produto. Variantes opcionais também podem ou não estar associadas a um ponto de variação.
<<alternative_OR>>	Estão sempre associadas aos pontos de variação. Pelo menos uma das variantes deverá ser escolhida para resolver o ponto de variação, ou seja, para estar presente na configuração de um produto da linha de produto.
<<alternative_XOR>>	Estão sempre associadas aos pontos de variação. Somente uma das variantes deverá ser escolhida para resolver o ponto de variação.
<<variability>>	Indica uma variabilidade existente em um modelo UML.
<<requires>>	Indica um relacionamento de dependência entre variantes no qual a variante dependente só existirá em uma configuração se a variante relacionada existir.
<<mutex>>	Indica um relacionamento de dependência entre variantes no qual a variante dependente só existirá em uma configuração se a variante relacionada obrigatoriamente não existir. São conhecidas como variantes mutuamente exclusivas.
<<variable>>	Representa que internamente, tal componente possui algum tipo de variabilidade.

### III.1 Diretrizes para identificação de variabilidades

Esta atividade tem como objetivo identificar as variabilidades associadas aos modelos da LP. O perfil *SyMPLES-ProfileVar* auxilia a realização desta atividade por meio da aplicação dos seus estereótipos aos modelos SysML e da atribuição dos valores dos meta-atributos.

As diretrizes fornecidas para apoio desta atividade são as seguintes:

- D.1.1** Os elementos de modelos de casos de uso relacionados aos mecanismos de extensão e de pontos de extensão indicam pontos de variação com variantes associadas, as quais podem ser inclusivas ou exclusivas. Assim, os casos de uso estendidos representam pontos de variação, enquanto os casos de uso que se estendem, representam variantes;
- D.1.2** Elementos de modelos de casos de uso relacionados com a associação de inclusão (`<<include>>`) ou associados a atores indicam variantes obrigatórias ou opcionais;
- D.1.3** Em modelos de requisitos, pontos de variação e suas variantes podem ser identificadas nos seguintes relacionamentos:
  - a. contenção: (*relação containment*), os requisitos mais gerais são os pontos de variação, enquanto os requisitos contidos são as suas variantes, as quais podem ser inclusivas ou exclusivas;
  - b. derivação: (*relação deriveReq*), os requisitos derivados indicam variantes obrigatórias ou opcionais;
  - c. dependência: (*relação dependency*), os requisitos dependentes indicam variantes obrigatórias ou opcionais;
- D.1.4** Em modelos de blocos (Diagrama de Definição de Blocos), pontos de variação e suas variantes são identificadas nos seguintes relacionamentos:
  - a. generalização: os classificadores mais gerais são os pontos de variação, enquanto os mais específicos são as variantes;
  - b. realização de interface: os “*suppliers*” (especificações) são os pontos de variação e as implementações (clientes) são as variantes;
  - c. agregação: as instâncias tipadas com losangos não preenchidos são os pontos de variação e as instâncias associadas são as variantes; e
  - d. composição: as instâncias tipadas com losangos preenchidos são os pontos de variação e as instâncias associadas são as variantes;
- D.1.5** Os elementos de modelos de blocos (Diagrama de Definição de Blocos), relacionados às associações nas quais os seus atributos *aggregationKind* possuem valor none, ou seja, não representam nem agregação nem composição, indicam variantes obrigatórias ou opcionais;
- D.1.6** Variantes que, ao serem selecionadas para fazer parte de um produto, exigem a presença de outra(s) determinada(s) variante(s) devem ter seus relacionamentos de dependência marcados com o estereótipo `<<requires>>`;
- D.1.7** Variantes mutuamente exclusivas para um determinado produto devem ter seus relacionamentos de dependência marcados com o estereótipo `<<mutex>>`;
- D.1.8** Blocos ou partes que possuem variabilidade em sua estrutura interna (Diagrama Interno do Blocos) devem ser marcados com o estereótipo `<<variable>>`;

- D.1.9** Se um bloco ou parte marcado com o estereótipo `<<Subsystem>>` possuir uma porta de entrada marcada com o estereótipo `<<enabler>>`, tal bloco será identificado como uma variante opcional. Isso acontece, pois um bloco que possui uma porta de entrada do tipo *enabler* tem a sua funcionalidade desabilitada quando o valor recebido em tal porta é igual a zero;
- D.1.10** Partes ou blocos que possuem uma porta de entrada do tipo *enabler*, quando ligados a um bloco marcado com o estereótipo `<<mux>>` sugerem variantes alternativas inclusivas, pois um bloco do tipo *mux* combina o valor de suas entradas em um único valor de saída; e
- D.1.11** Partes ou blocos ligados a um bloco marcado com o estereótipo `<<switch>>` sugerem variantes alternativas exclusivas, pois um bloco do tipo *switch* seleciona como saída apenas uma de suas entradas, de acordo com o valor de uma constante;

## III.2 Diretrizes para Delimitação de Variabilidades

Esta atividade tem como objetivo definir os valores dos seguintes meta-atributos de uma variabilidade: *minSelection*, *maxSelection*, *bindingTime*, *allowsAddingVar* e *variants*. Para tanto, devem ser seguidas as seguintes diretrizes:

- D.2.1.** variabilidades com variantes opcionais possuem multiplicidade *minSelection* = 0 e *maxSelection* = 1;
- D.2.2.** variabilidades com variantes alternativas exclusivas possuem multiplicidade *minSelection* = 1 e *maxSelection* = 1;
- D.2.3.** variabilidades com variantes alternativas inclusivas possuem multiplicidade *minSelection* = 1 e *maxSelection* = *size(variants)* em que *size(x)* é uma função que retorna a quantidade de elementos da coleção *x*;
- D.2.4.** o valor de *bindingTime* deve ser definido escolhendo-se um dos valores da classe de enumeração *BindingTime*;
- D.2.5.** o valor booleano do atributo *allowsAddingVar* deve ser analisado de acordo com a possibilidade de manter o ponto de variação aberto (true) ou fechado (false);
- D.2.6.** o valor da coleção *variants* é o conjunto formado pelas instâncias das variantes associadas ao ponto de variação ou variabilidade;

## III.3 Exemplo

A Figura 1 apresenta um exemplo de representação de variabilidades na LP “Printer” em um Diagrama de Definição de Blocos da SysML.

- Todos os elementos identificados como obrigatórios foram anotados com o estereótipo `<<mandatory>>`.
- A LP apresenta 3 interfaces de conexão: um conector USB 2.0, conector USB 3.0 e um conector WI-FI. Assim, no mínimo um produto deve conter uma interface de conexão.
- De acordo com a diretriz D.1.4.a, os blocos *Usb2Connector*, *Usb3Connector* e *WifiConnector* são considerados variantes alternativas inclusivas, portanto receberam o estereótipo `<<alternative_OR>>`.
- De acordo com a diretriz D.1.4.a, os blocos *BWTONner* e *ColorTonner* foram considerados variantes alternativas exclusivas e foram anotados com o estereótipo `<<alternative_XOR>>`.
- O bloco *EmgPower* foi identificado como opcional, assim, foi anotado com o estereótipo `<<optional>>`.
- Todos os pontos de variação e elementos opcionais possuem uma nota UML/SysML com o estereótipo `<<variability>>` detalhando as propriedades de tal variabilidade de acordo com a abordagem em questão.
- Neste modelo existem 2 restrições: (a) a dependência com o estereótipo `<<requires>>` indica que se o bloco *ColorTonner* for selecionado, o bloco *Usb3Connector* obrigatoriamente deve ser selecionado e, (b) a dependência com o estereótipo `<<mutex>>` indica que se o bloco *EmgPower* for selecionado, então o bloco *WifiConnector* não poderá estar presente em um mesmo produto.

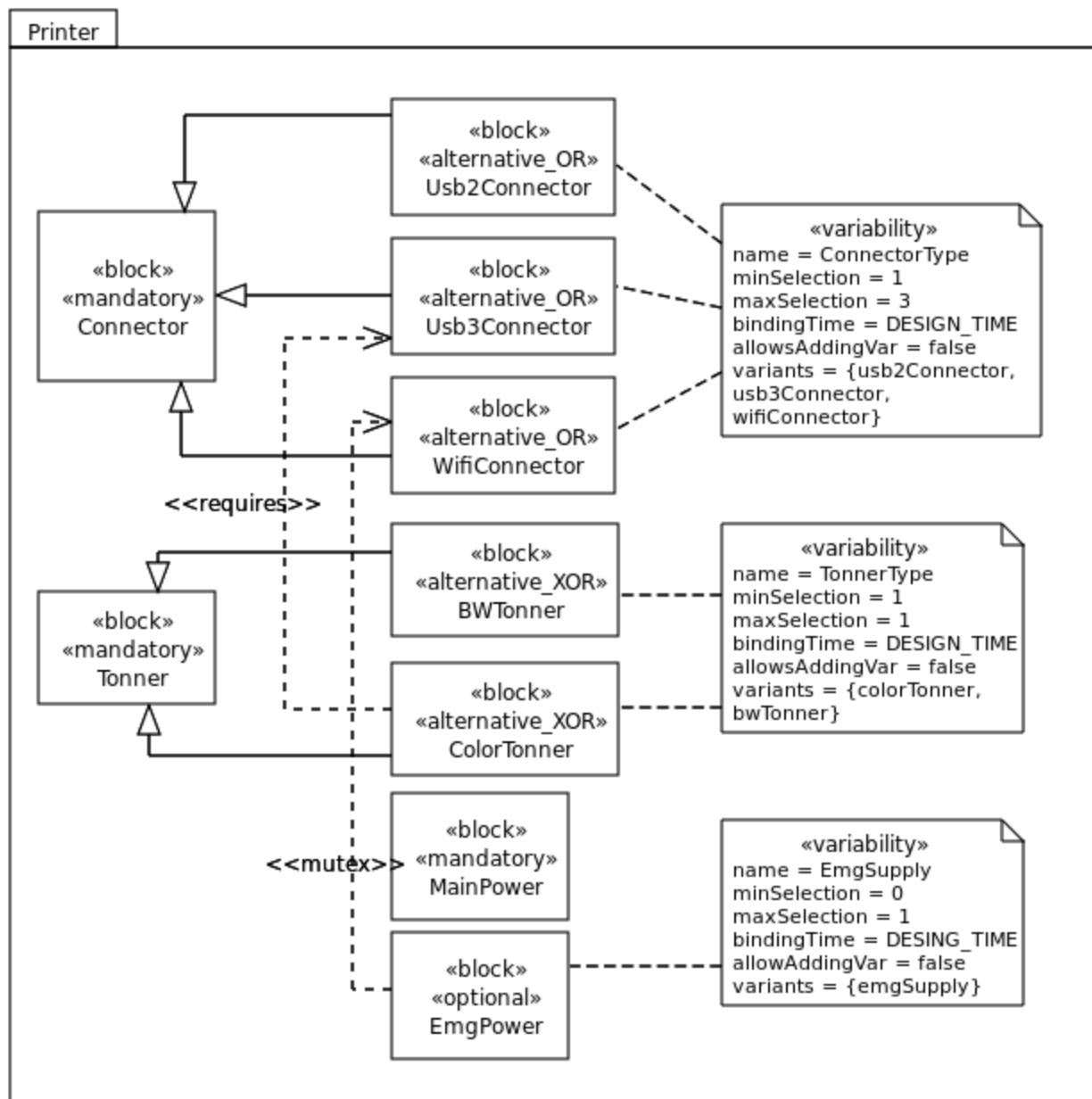


Figura 1 – Exemplo de Diagrama de Definição de Blocos do SysML com variabilidades representadas com uso da abordagem SyMPLES v1.0.