

UNIVERSIDADE ESTADUAL DE MARINGÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

THAIS SANTOS NEPOMUCENO

**Avaliação empírica da abordagem SMarty na perspectiva de
configuração de produtos específicos e rastreabilidade de modelos**

Maringá

2020

THAIS SANTOS NEPOMUCENO

**Avaliação empírica da abordagem SMarty na perspectiva de
configuração de produtos específicos e rastreabilidade de modelos**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Informática, Centro de Tecnologia da Universidade Estadual de Maringá, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Edson A. Oliveira
Junior

Maringá
2020

Dados Internacionais de Catalogação na Publicação (CIP)
(Biblioteca Central - UEM, Maringá, PR, Brasil)

N441a Nepomuceno, Thais Santos
Avaliação empírica da abordagem SMarty na perspectiva de configuração de produtos específicos e rastreabilidade de modelos / Thais Santos Nepomuceno. -- Maringá, 2020.
116, [76] f. : il. color., figs., tabs.

Orientador: Prof. Dr. Edson A. Oliveira Junior.
Dissertação (Mestrado) - Universidade Estadual de Maringá, Centro de Tecnologia, Departamento de Informática, Programa de Pós-Graduação em Ciência da Computação, 2020.

1. Linha de produto de software. 2. SMarty - Gerenciamento de variabilidades. 3. UML (Linguagem de Modelagem Unificada) 4. Experimentação (Engenharia de Software). I. Oliveira Junior, Edson A., orient. II. Universidade Estadual de Maringá. Centro de Tecnologia. Departamento de Informática. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDD 23.ed. 005.12

Síntique Raquel Eleuterio - CRB 9/1641


FOLHA DE APROVAÇÃO

THAIS SANTOS NEPOMUCENO


Avaliação empírica da abordagem SMarty na perspectiva de configuração de produtos específicos e rastreabilidade de modelos

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Informática, Centro de Tecnologia da Universidade Estadual de Maringá, como requisito parcial para obtenção do título de Mestre em Ciência da Computação pela Banca Examinadora composta pelos membros:

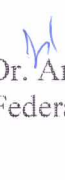
BANCA EXAMINADORA



Prof. Dr. Edson A. Oliveira Junior
Universidade Estadual de Maringá – DIN/UEM



Profa. Dra. Aline Maria Malachini Miotto Amaral
Universidade Estadual de Maringá – DIN/UEM



Prof. Dr. Anderson da Silva Marcolino
Universidade Federal do Paraná – DEE/UFPR-Palotina

Aprovada em: 03 de novembro de 2020.

Local da defesa: Sala virtual

<https://meet.google.com/npe-xytj-kvy>

DEDICATÓRIA

Dedico este trabalho à minha família: minha mãe **Cleonice**, meu pai **José** e meu irmão **Thiago** pelo apoio e incentivo constante na busca dos meus objetivos.

AGRADECIMENTOS

À DEUS, por ter me abençoado em todos os momentos até aqui.

Agradeço à minha mãe **Cleonice**, minha fonte de inspiração, minha melhor amiga, por todo gesto de amor e carinho, que sempre esteve ao meu lado e se alegrou com minhas conquistas.

Ao meu pai **José**, meu melhor amigo, pelo carinho, incentivo e apoio incondicional de toda a vida.

Ao meu irmão **Thiago** pelo companheirismo, amizade e carinho constante.

À minha avó **Elisete**, por todas as orações, palavras de incentivo e amor.

Ao meu namorado **Felipe**, pelo amor, carinho, paciência e incentivo.

Ao **Marcos** e à **Ieda**, por todo o incentivo.

Ao meu orientador **Prof. Dr. Edson A. Oliveira Junior**, a quem admiro, pelo ensinamento, apoio, paciência, dicas e contribuições durante todo esse período.

Aos professores das disciplinas cursadas no mestrado.

Aos amigos do mestrado, de forma especial à **Giovanna**, amiga desde a graduação, por sempre estar presente, me ajudar a todo momento, me auxiliar na aplicação dos experimentos e tornar esta caminhada mais leve.

À todos os participantes dos estudos experimentais, e professores que contribuíram sedendo as turmas.

À secretária do departamento, **Maria Inês**, que sempre pronta a nos auxiliar.

E a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro concedido a este trabalho.

Avaliação empírica da abordagem SMarty na perspectiva de configuração de produtos específicos e rastreabilidade de modelos

RESUMO

Uma forma das organizações se manterem em uma posição competitiva no mercado é por meio do reúso de software, que tem como vantagens a redução de custos na produção de bens em larga escala, além de menor tempo para construção e facilidade para manutenção. Para tanto, existem diversas maneiras de praticar o reúso, uma delas é por meio da abordagem de Linha de Produto de Software (LPS). Na engenharia de LPS, para construir os artefatos a serem reusados, deve-se levar em conta tudo o que é comum (similaridades) e o que varia (variabilidades). Dessa maneira, o gerenciamento de variabilidades é uma das atividades técnicas mais importantes na engenharia de LPS. Existem muitos métodos e abordagens para o gerenciamento das variabilidades em uma LPS. As baseadas em UML têm se destacado por permitirem gerenciar as variabilidades em modelos com diferentes níveis de abstração e permitir o rastreamento entre tais níveis e modelos. A *Stereotype-based Management of Variability* (SMarty) é uma delas que, por meio de seu perfil UML 2.0 e de um processo sistemático, permite a identificação, a representação e o rastreamento de variabilidades em diversos diagramas da UML. No entanto, ainda não existem estudos empíricos suficientes na literatura que sustentem a hipótese de que a abordagem SMarty é mais efetiva que as demais baseadas em UML em relação à configuração de produtos específicos e rastreabilidade entre modelos. Neste trabalho são propostos estudos experimentais para avaliar a efetividade da abordagem SMarty em relação a diagramas de classes, casos de uso, sequência e componentes fornecendo subsídios para o desenvolvimento de uma ferramenta para apoiar SMarty. Ao decorrer dos experimentos, novas diretrizes foram criadas, com o intuito de auxiliar os usuários a gerenciar variabilidades e rastrear elementos entre modelos de uma LPS. A ordem desses experimentos foi pré-estabelecida, do diagrama mais abstrato para o menos abstrato, de acordo com a visão que eles fornecem do sistema, na intenção de gerar e reusar conhecimento de um experimento para o outro de forma incremental por meio dos dados obtidos nesses experimentos. Como resultados, existem indícios da melhora da efetividade de SMarty a cada estudo experimental, se sobressaindo também em relação à rastreabilidade de elementos entre os diagramas UML testados.

Palavras-chave: Avaliação Experimental. Gerenciamento de Variabilidade. Linha de Produto de Software. SMarty. UML. Configuração de Produtos. Rastreabilidade .

Empirical evaluation of the SMarty approach in the perspective of specific product configuration and models traceability

ABSTRACT

One way for organizations to remain in a competitive position in the market is through the reuse of software, which has the advantages of reducing costs in the production of goods on a large scale, in addition to shorter construction time and easier maintenance. For this, there are several ways to practice reuse, one of which is through the Software Product Line (LPS) approach. In LPS engineering, to build the artifacts to be reused, one must take into account everything that is common (similarities) and what varies (variability). In this way, the management of variability is one of the most important technical activities in LPS engineering. There are many methods and approaches for managing variability in an LPS. UML-based ones have stood out for allowing to manage the variability in models with different levels of abstraction and to allow the tracking between such levels and models. Textit Stereotype-based Management of Variability (SMarty) is one of them that, through its UML 2.0 profile and a systematic process, allows the identification, representation and tracking of variability in various UML diagrams. However, there are not yet enough empirical studies in the literature to support the hypothesis that the SMarty approach is more effective than the others based on UML in relation to the configuration of specific products and traceability between models. In this work, experimental studies are proposed to evaluate the effectiveness of the SMarty approach in relation to class diagrams, use cases, sequence and components providing subsidies for the development of a tool to support SMarty. During the experiments, new guidelines were created, in order to help users manage variability and track elements between models of an LPS. The order of these experiments was pre-established, from the most abstract to the least abstract diagram, according to the vision they provide of the system, with the intention of incrementally generating and reusing knowledge from one experiment to the other through the data obtained in these experiments. As a result, there are signs of improvement in the effectiveness of SMarty in each experimental study, also standing out in relation to the traceability of elements among the tested UML diagrams.

Keywords: *Experimental Evaluation. Variability Management. Software Product Line. SMarty. UML. Product Configuration. Traceability .*

LISTA DE FIGURAS

Figura 1.1	Metodologia de desenvolvimento do trabalho	18
Figura 2.1	Framework Engenharia de LPS	22
Figura 2.2	Modelagem de um Diagrama de Casos de Uso com a abordagem SMarty	26
Figura 2.3	Modelagem de um Diagrama de Casos de Uso com a abordagem PLUS	31
Figura 2.4	Modelagem de um Diagrama de Componentes com a abordagem Razavian e Khosravi	33
Figura 2.5	Modelagem de um Diagrama de Classes com a abordagem Ziadi et al	34
Figura 2.6	Modelagem de um Diagrama de Sequência com a abordagem Ziadi et al	35
Figura 2.7	Processo de Experimentação em Engenharia de Software	40
Figura 3.1	Variáveis dependentes e independentes do experimento #1.	52
Figura 3.2	Variáveis dependentes e independentes do experimento #2.	52
Figura 3.3	Variáveis dependentes e independentes do experimento #3.	53
Figura 3.4	Artefatos MM - Parte do Diagrama de Classes	55
Figura 3.5	Artefatos AGM - Diagrama de Casos de Uso	56
Figura 3.6	Artefatos AGM - Diagrama de Classes	57
Figura 4.1	Escala de Correlação de Spearman	65
Figura 5.1	Visão Geral da Abordagem SMarty com as novas diretrizes	86
Figura 5.2	Exemplo da diretriz C.1	87
Figura 5.3	Exemplo da diretriz C.1	87
Figura 5.4	Exemplo da diretriz C.2	88
Figura 5.5	Exemplo da diretriz C.2.1	89
Figura 5.6	Exemplo da diretriz C.3	90
Figura 5.7	Exemplo da diretriz C.4	90
Figura 5.8	Exemplo da diretriz R.1	91
Figura 5.9	Exemplo da diretriz R.2	91
Figura 5.10	Exemplo da diretriz R.3	92
Figura 5.11	Exemplo da diretriz R.4	92

Figura 1.1	Distribuição de Estudos por Base	114
------------	--	-----

LISTA DE TABELAS

Tabela 3.1	Descrição do nível de conhecimento dos participantes do Experimento #1	49
Tabela 3.2	Descrição do nível de conhecimento dos participantes do Experimento #2	50
Tabela 3.3	Descrição do nível de conhecimento dos participantes do Experimento #3	51
Tabela 4.1	Respostas sobre Rastreabilidade de SMarty	67
Tabela 4.2	Respostas sobre Rastreabilidade de SMarty	70
Tabela 5.1	Valores de Média, Mediana e Desvio Padrão Observados com SMarty	80
Tabela 5.2	Valores de Média, Mediana e Desvio Padrão Observados com o método PLUS	80
Tabela 5.3	Valores de Média, Mediana e Desvio Padrão Observados com o método Razavian e Khosravi	80
Tabela 5.4	Valores de Média, Mediana e Desvio Padrão Observados com o método Ziadi et al	81
Tabela 5.5	Valores Observados com o teste Mann-Whitney-Wilcoxon	81
Tabela 5.6	Nível de formação dos participantes	82
Tabela 5.7	Conhecimento dos participantes em UML	83
Tabela 5.8	Conhecimento dos participantes em LPS	83
Tabela 1.1	Bases de busca	110
Tabela 1.2	Strings de Busca	111
Tabela 1.3	Quantidade de Resultados por Base de Busca	113
Tabela 1.4	Estudos Selecionados	114
Tabela 1.5	Estudos Adicionados	115

SUMÁRIO

1	Introdução	15
1.1	Contextualização	15
1.2	Motivação e Justificativa	16
1.3	Objetivos	17
1.4	Metodologia de Desenvolvimento de Pesquisa	18
1.5	Organização do Trabalho	19
2	Fundamentação Teórica	20
2.1	Considerações Iniciais	20
2.2	Linha de Produto de Software	20
2.3	Gerenciamento de Variabilidades e Abordagens Consideradas	23
2.3.1	SMarty	24
2.3.2	PLUS	30
2.3.3	Razavian e Khosravi	31
2.3.4	Ziadi et al	32
2.4	Configuração de Produtos	36
2.5	Rastreabilidade em LPS	37
2.6	Experimentação em Engenharia de Software	38
2.7	Trabalhos Relacionados	40
2.8	Considerações Finais	41
3	Protocolo dos Experimentos	42
3.1	Considerações Iniciais	42
3.2	Metodologia e Planejamento Experimental	42
3.2.1	Objetivos (D.1)	43
3.2.2	Formulação das Hipóteses (D.2)	43
3.2.3	Definição das Variáveis (D.3)	47
3.2.4	Tamanho da Amostra (D.4)	48
3.2.5	Definição e Seleção dos Participantes (D.5)	49
3.2.6	Identificação do Tópico de Pesquisa do Experimento (D.6)	51
3.2.7	Descrição do <i>Design</i> Experimental (D.7)	51
3.2.8	Definição e Seleção dos Materiais Experimentais (D.8)	51
3.2.8.1	Descrição da LPS Utilizada (D.8.1)	54
3.2.8.2	Artefatos da LPS (D.8.2)	55
3.2.9	Validação dos Materiais Experimentais (D.9)	55

3.2.10	Descrição das Tarefas do Experimento (D.10)	56
3.2.11	Descrição dos Requisitos de Treinamento (D.11)	57
3.2.12	Condução do Projeto Piloto (D.12)	58
3.2.13	Ambiente da Condução do Experimento (D.13)	58
3.2.14	Data de Execução dos Experimentos (D.14)	58
3.2.15	Condução dos Experimentos (D.15)	59
3.3	Considerações Finais	59
4	Execução, Análise e Interpretação dos Dados	60
4.1	Considerações Iniciais	60
4.2	Execução dos experimentos	60
4.2.1	Descrição de como a coleta de dados foi realizada: (D.16)	61
4.3	Procedimentos de análise dos dados coletados: (D.17)	62
4.3.1	Experimento #1	62
4.3.2	Experimento #2:	65
4.3.3	Experimento #3:	68
4.4	Taxa de Mortalidade (D.18)	70
4.5	Ferramenta estatística para análise dos dados (D.19)	71
4.6	Tamanho do Efeito (D.20)	71
4.7	Resultados Obtidos do Ponto de Vista de Pesquisadores e de Profissionais (D.21)	72
4.8	Implicações dos Tratamentos Desenvolvidos (D.22)	72
4.9	Ameaças à Validade Identificadas no Experimento (D.23)	73
4.10	Fonte do Pacote Experimental (D.24)	73
4.11	Template Experimental Utilizado para Conduzir, Planejar ou Documentar o Experimento (D.25)	73
4.12	Considerações Finais	73
5	Discussão dos Resultados, Meta Análise e Novas Diretrizes	75
5.1	Considerações Iniciais	75
5.2	Discutindo os Resultados dos Experimentos	76
5.2.1	Estudo Experimental #1 (Diagramas de Casos de Uso e Diagramas de Classes)	76
5.2.2	Estudo Experimental #2 (Diagramas de Classes e Diagramas de Componentes)	76
5.2.3	Estudo Experimental #3 (Diagramas de Classes e Sequência)	77

5.3	Meta Análise	78
5.3.1	Estatística Descritiva	79
5.3.2	Testes de Hipótese	81
5.3.3	Formação dos Participantes	82
5.3.4	Nível de Conhecimento dos Participantes	82
5.4	Ameaças à Validade	83
5.4.1	Ameaças à Validade de Conclusão	84
5.4.2	Ameaças à Validade de <i>Constructo</i>	84
5.4.3	Ameaças à Validade Interna	84
5.4.4	Ameaças à Validade Externa	85
5.5	Apresentação e Empacotamento dos Resultados	85
5.6	Novas Diretrizes para SMarty	85
5.6.1	Diretrizes para Configuração de Produtos	85
5.6.2	Diretrizes para Rastreabilidade	91
5.7	Considerações Finais	93
6	Conclusão	94
6.1	Contribuições	95
6.2	Limitações	97
6.3	Trabalhos Futuros	98
	REFERÊNCIAS	100
A	Apêndice A: Avaliação Experimental das abordagens de Gerenciamento de Variabilidades: um Mapeamento Sistemático	106
A.1	Introdução	107
A.2	Especificação do Mapeamento Sistemático	107
A.2.1	Questões de Pesquisa	107
A.2.2	Refinamento das Questões de Pesquisa	108
A.2.3	Critérios de Inclusão e Exclusão	108
A.2.4	Esquema de Classificação	109
A.3	Seleção dos Estudos	110
A.3.1	Processo de Seleção Preliminar	110
A.3.2	Processo de Seleção Final	111
A.3.3	Extração dos Resultados	111
A.4	Condução da Revisão	112
A.4.1	Definição das Strings de Busca	113

A.4.2	Aplicação das Strings de Busca e Obtenção dos Estudos nas Fontes Especificadas	113
A.4.3	Seleção Preliminar dos Estudos	113
A.5	Estudos Seleccionados	113
A.5.1	Síntese dos Estudos Seleccionados	115
A.6	Conclusão	116
B	Apêndice B: Instrumentação dos Experimentos	117
B.1	Materiais do Experimento #1	117
B.2	Materiais do Experimento #2	137
B.3	Materiais do Experimento #3	163

Introdução

1.1 Contextualização

O poder de compra das pessoas tem melhorado pela maneira como os bens são produzidos. Sabe-se que, quando alguma coisa é produzida e vendida em larga escala, o preço é reduzido, em comparação a bens que são produzidos individualmente, de forma artesanal. Para que algo seja produzido em massa é necessária a generalização de algumas características pertencentes ao produto. Esse conceito chegou à forma de produzir softwares. As empresas querem aumentar o reúso diminuindo, assim, os custos e o tempo na construção de um sistema (Pohl *et al.*, 2005)

A ideia de reusar partes constituintes de um sistema deve-se, principalmente, ao panorama competitivo das empresas de software, que está mudando e crescendo rapidamente. Além disso, os sistemas estão maiores e mais complexos, acompanhando a velocidade que a inovação acelera (Capilla *et al.*, 2013).

Dessa forma, a indústria de software tem adotado Linha de Produto de Software (LPS), um paradigma para desenvolver e reutilizar aplicações de software usando plataformas e customização em massa, que permite uma produção ampla muito mais barata do que produzir individualmente, entretanto, reduz a possibilidade de diversificação (Almeida, 2019a; Pohl *et al.*, 2005). O uso de uma LPS normalmente é apoiado por questões econômicas, já que adotá-la significa que os custos da construção de um sistema são reduzidos, juntamente com o tempo que leva para a sua criação. Além disso, a qualidade do produto resultante de uma LPS é superior a sistemas feitos da maneira tradicional (van der Linden *et al.*, 2007).

Segundo Pohl *et al.* (2005) e van der Linden *et al.* (2007), a flexibilidade de uma LPS é a base da customização em massa, chamada de variabilidade. A variabilidade deve ser gerenciada em toda a Engenharia de LPS. Construir aplicações para utilização em massa significa empregar esse conceito de variabilidade gerenciada.

O gerenciamento das variabilidades tem um grande impacto na maneira em que um software é desenvolvido, estendido e mantido (Capilla *et al.*, 2013). Entre as abordagens para gerenciamento de variabilidades, estão as baseadas na Linguagem de Modelagem Unificada (UML). Algumas delas são: *Product Line UML-based Software Engineering* (PLUS) (Gomaa, 2005), *Ziadi et al* (Ziadi *et al.*, 2003) e a *Stereotype-based Management of Variability* (SMarty). Tais abordagens são amplamente conhecidas na literatura de LPS, cada uma com as suas particularidades.

1.2 Motivação e Justificativa

Vários experimentos sobre a efetividade de identificação e representação de variabilidades já foram conduzidos com SMarty em comparação às abordagens citadas anteriormente em relação a diagramas de classes, sequência, componentes, atividades e casos de uso da UML. Além desses, um experimento que comparava duas abordagens em relação à configuração de produtos específicos foi realizado com diagramas de classes (Marcolino e OliveiraJr, 2015, 2017; Marcolino *et al.*, 2014a,b).

Em alguns desses experimentos houveram evidências de que a abordagem SMarty foi mais efetiva para a resolução e identificação de variabilidades. Obter evidências da efetividade de uma determinada abordagem é fundamental para que a indústria possa adotá-la para gerenciar variabilidades em LPS, já que segundo Mafra e Travassos (2005), a indústria apresenta imaturidade quanto à escolha de processos e tecnologias, gerando incertezas e falta de credibilidade nos engenheiros de software, quando relacionadas a novas propostas e à adoção de tecnologias emergentes.

Segundo Travassos (2002), a experimentação pode levar a novas descobertas que não estavam previstas, fazendo com que a pesquisa tome novas perspectivas. Sabendo dessa importância, como ainda não se tem dados experimentais que comparam a efetividade da abordagem SMarty com as demais no que se refere à configuração de produtos e rastreabilidade entre modelos, entende-se a necessidade de tais experimentos em diagramas UML de casos de uso, classes, sequência e componentes.

Neste trabalho foi estabelecida uma ordem de abstração dos diagramas SMarty para tais experimentos, permitindo que os participantes sejam partes ativas da experimentação sistemática fornecendo *feedback* (Munezero *et al.*, 2017). Esse *feedback* guia os experimen-

tos em seus respectivos planejamentos (Lindgren e Münch, 2016). Os dados são analisados e servem como conhecimento adquirido para os experimentos futuros.

Sabendo da importância do gerenciamento de variabilidades no desenvolvimento de uma LPS, foi realizada uma série de experimentos para averiguar e comparar a efetividade entre diversas abordagens na configuração de produtos e na rastreabilidade entre modelos.

Uma das abordagens consideradas é a Ziadi et al (Ziadi *et al.*, 2003) (Ziadi e Jezequel, 2006) que foi escolhida por propor uma das mais representativas abordagens de gerenciamento de variabilidades para diagramas de sequência.

Razavian e Khosravi (Razavian e Khosravi, 2008), também foi uma abordagem escolhida para comparação com a abordagem SMarty já que propõe gerenciamento de variabilidades em diagramas de componentes.

Outro método a ser comparado é o PLUS (Gomaa, 2005), referenciado em estudos como os de Ryu *et al.* (2012), Nguyen (2009), Catal (2009) e Tawhid e Petriu (2008) sendo um dos mais difundidos na literatura para a representação de variabilidades em diagramas de casos de uso e classes.

Vários experimentos já foram conduzidos pelo grupo de pesquisa em Reúso Sistemático e Experimentação (GReater) da UEM.

Os experimentos realizados por Marcolino (Marcolino *et al.*, 2013), (Marcolino *et al.*, 2014a), (Marcolino *et al.*, 2014b), (Marcolino *et al.*, 2017), (Marcolino e Oliveira Jr, 2017), não seguiam uma ordem pré-estabelecida entre os diagramas, já que os resultados e *feedbacks* recebidos não serviam como subsídios para melhorar o próximo experimento. Eles eram aplicados aleatoriamente, não importando qual diagrama era avaliado por vez. Dessa maneira, os resultados não possuem nenhum estudo de correlação entre eles, diferente do atual trabalho, em que uma meta análise é realizada entre os experimentos conduzidos.

Assim, apesar de já existirem alguns estudos, ainda não se tem dados experimentais que comparam a efetividade da abordagem SMarty com as demais abordagens no que se refere à configuração de produtos em diagramas de casos de uso, sequência, classes e componentes e que possam ser usados para meta análise de tais experimentos, o que também motivou, portanto, a realização deste trabalho.

1.3 Objetivos

O objetivo deste trabalho é planejar e conduzir experimentos com o objetivo de avaliar a efetividade da abordagem SMarty em comparação com abordagens similares existentes na literatura do ponto de vista da resolução de variabilidades e configuração de produtos

específicos de LPS em diagramas de casos de uso, sequência, classes e componentes além da possibilidade de rastreabilidade entre elementos de tais diagramas.

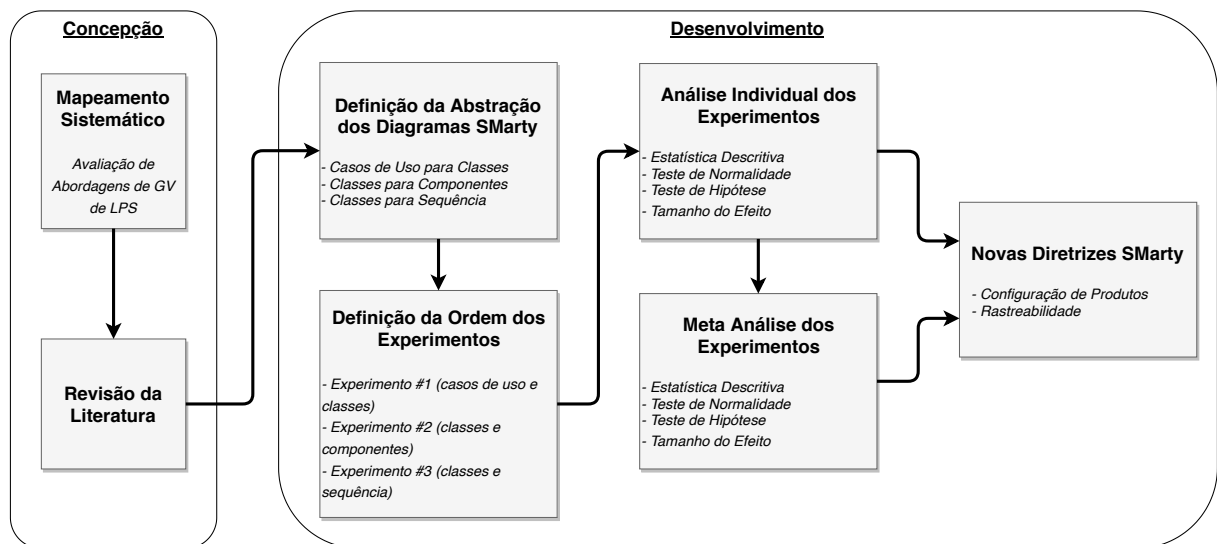
Como objetivos específicos têm-se:

- buscar estudos que reportam a avaliação empírica de abordagens de gerência de variabilidades para LPS baseadas em UML;
- planejar, conduzir e analisar experimentos para caracterizar a efetividade da abordagem SMarty em relação às abordagens similares reportadas na literatura;
- analisar o conjunto de experimentos e os seus resultados, produzindo uma síntese da evidência obtida;
- evoluir a abordagem SMarty com base nos resultados gerais obtidos.

1.4 Metodologia de Desenvolvimento de Pesquisa

A Figura 1.1 ilustra a metodologia seguida para o desenvolvimento deste trabalho.

Figura 1.1: Metodologia de desenvolvimento do trabalho



Na fase de **Concepção** foi conduzido uma atualização de um **Mapeamento Sistemático da Literatura** (MSL) realizado por (Marcolino, 2014) para identificar estudos empíricos de avaliação de abordagens para gerenciamento de variabilidades. Com base nos resultados do MSL, a atividade de **Revisão da Literatura** foi realizada considerando os principais tópicos para entendimento do contexto deste trabalho.

A seguir, na fase de **Desenvolvimento** foi realizada a atividade de **Definição da Abstração dos Diagramas SMarty**, para que se pudesse passar à atividade de **Definição da Ordem dos Experimentos**, uma vez que a ordem ajuda no ganho de conhecimento de um experimento para outro, além de permitir a realização dos experimentos em uma ordem de abstração.

Uma vez definida a ordem de condução dos experimentos, cada um deles foi planejado e conduzido aos pares de abordagens, conforme ilustrado na Figura 1.1. Assim, a atividade de **Análise Individual dos Experimentos** permitiu entender a estatística descritiva, teste de normalidade, teste de hipótese e tamanho de efeito de cada amostra coletada dos experimentos.

Tendo todos os experimentos sido devidamente conduzidos e analisados individualmente, foi realizada a atividade de **Meta Análise** buscando entender o conjunto de evidências obtidas por todos os experimentos. Assim, cada experimento, juntamente com a Meta Análise, contribuiu para a criação de **Novas Diretrizes SMarty** para auxiliar os usuários a configurar produtos e rastrear elementos entre os diagramas experimentados com a abordagem SMarty, contribuindo dessa maneira com a evolução da abordagem.

1.5 Organização do Trabalho

O Capítulo 2 apresenta os conceitos básicos sobre LPS, gerenciamento de variabilidades e as abordagens de gerenciamento de variabilidades em LPS baseadas em UML, como PLUS, Ziadi et al, SMarty, Razavian e Khosravi os conceitos de experimentação em Engenharia de Software. No Capítulo 3 é apresentado o protocolo dos experimentos realizados, enquanto que no Capítulo 4 são apresentados a execução, análise e interpretação dos resultados de tais experimentos. O capítulo 5 discute os resultados obtidos nos experimentos e as respectivas ameaças à validade, bem como apresenta a meta análise de tais resultados e novas diretrizes para SMarty. O Capítulo 6 apresenta as contribuições e limitações deste trabalho e elenca os principais trabalhos futuros.

Fundamentação Teórica

2.1 Considerações Iniciais

O fato de existir tantas abordagens para gerenciamento de variabilidades faz com que diferentes metodologias para representação dessas abordagens sejam usadas. Existem aquelas baseadas em modelos de características, as que definem uma linguagem de domínio específica (DSL), e as que utilizam notações conhecidas, como a UML (Galster *et al.*, 2014) (Chen e Ali Babar, 2011): como Ziadi *et al.*, PLUS, Razavian e Khosravi e SMarty.

Neste trabalho foram consideradas apenas as abordagens de gerenciamento de variabilidades baseadas em UML. Dessa maneira, neste capítulo é apresentada a fundamentação teórica elementar para a compreensão dos conceitos utilizados para a avaliação experimental da abordagem SMarty, bem como os fundamentos sobre LPS, variabilidade e as abordagens de gerenciamento de variabilidade utilizadas nos experimentos e os conceitos sobre experimentação em engenharia de software.

2.2 Linha de Produto de Software

Um crescente número de organizações tem adotado o reúso para desenvolver produtos de alta qualidade, mais rápido e mais barato. Uma dessas formas de reúso é a abordagem de LPS (McGregor *et al.*, 2002) (van der Linden *et al.*, 2007) (Almeida, 2019b). A adoção bem sucedida de uma LPS fornece oportunidade para que as empresas melhorem suas posições competitivas no mercado (Capilla *et al.*, 2013). Dessa maneira, uma LPS é um conjunto de produtos que compartilham características comuns para satisfazer necessidades específicas

de um segmento de mercado particular, e que são desenvolvidos a partir de um conjunto de artefatos básicos, de maneira bem definida (Clements e Northrop, 2001). Em LPS é a variabilidade que fornece a flexibilidade necessária para diferenciação e diversificação de produtos (Chen *et al.*, 2009), por isso gerenciá-la é tão importante na engenharia de LPS.

Vários princípios são fundamentais para uma engenharia de linha de produtos de software bem-sucedida e são descritos a seguir conforme van der Linden *et al.* (2007) e McGregor *et al.* (2002):

- Gerenciamento de variabilidade: sistemas individuais são considerados variações de um sistema comum. Essa variabilidade é explícita e deve ser sistematicamente gerenciada;
- Centrada nos negócios: a engenharia de linha de produtos de software visa conectar a engenharia da linha de produtos com a estratégia de longo prazo do negócio;
- Centrada na arquitetura: o lado técnico do software deve ser desenvolvido de uma forma que permite tirar proveito das semelhanças entre os sistemas individuais;
- Abordagem de ciclo de vida duplo: os sistemas individuais são desenvolvidos com base em uma plataforma de software. Esses produtos devem ser projetados e têm seus ciclos de vida individuais.

Produzir algo em larga escala, traz alguns benefícios comparados a produzir algo individualmente. Como afirmam Pohl *et al.* (2005) e van der Linden *et al.* (2007), algumas das vantagens são:

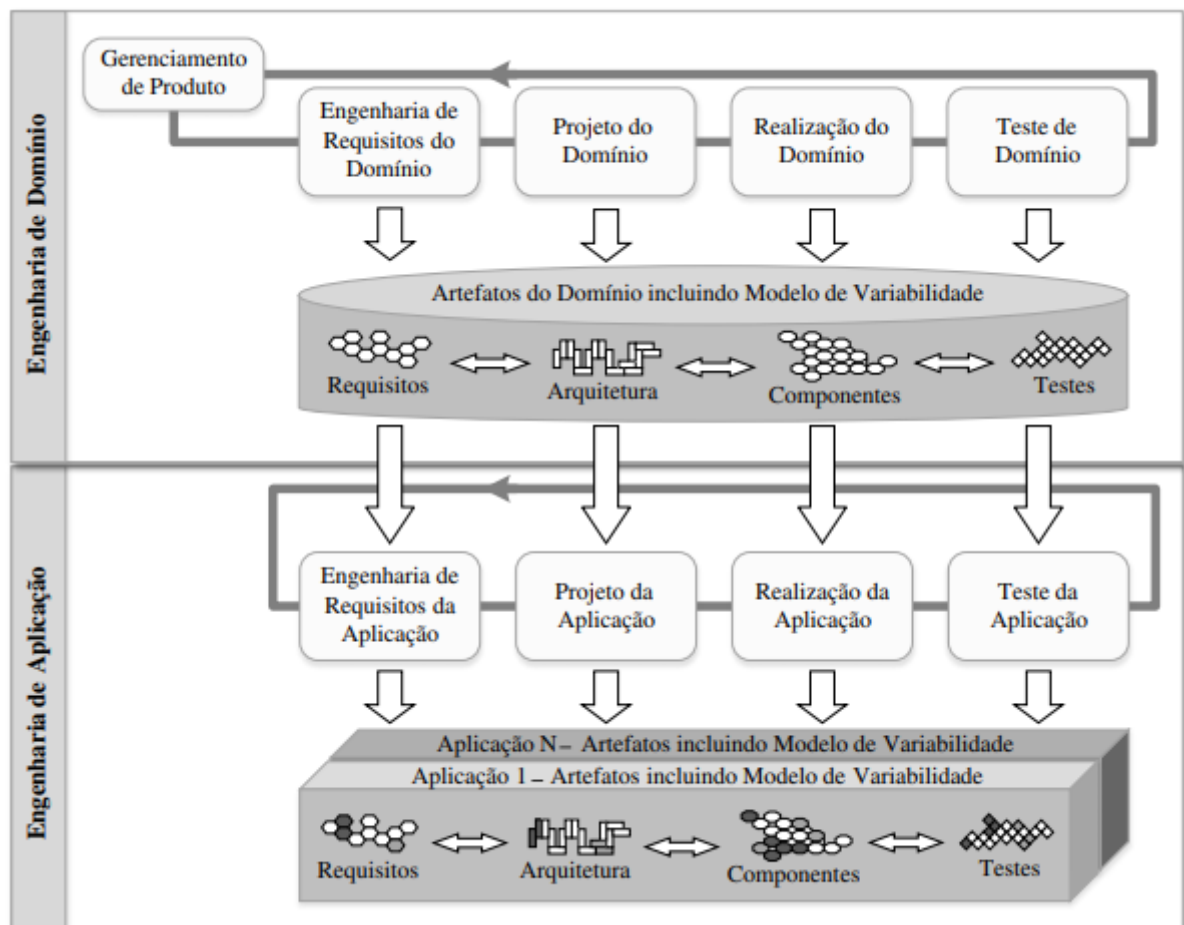
- Redução de custos para desenvolvimento: quando um artefato é reutilizado em diferentes tipos de sistemas, isso implica em uma redução de custos de cada sistema. Para que isso seja possível, inicialmente são necessários investimentos para produzir este artefato a ser reusável em diversos produtos, para aí sim, poder reduzir tais custos;
- Melhoria da qualidade: para produzir um artefato que possa ser reutilizado em diversos produtos, ele deve ser testado e revisado em todos eles. Estes artefatos devem ter um bom funcionamento em todos os produtos que o utilizarão. Essa extensa avaliação faz com que a chance de encontrar erros e corrigi-los seja maior, aumentando, desta forma, a qualidade dos produtos;

- Diminuição do tempo para construção: inicialmente na LPS, o maior tempo demandado é para a construção do artefato padrão. Depois de elaborado, o tempo para construção dos produtos torna-se bem reduzido, já que o artefato é empregado em cada novo produto.

A partir das vantagens listadas, outros benefícios são alcançados, como: redução de esforço para manutenção, facilidade em lidar com a evolução e melhora das estimativas de custos.

A Engenharia de LPS compreende três atividades principais: Engenharia de Domínio, Engenharia de Aplicação e o Gerenciamento de LPS (Pohl *et al.*, 2005). A relação do que é realizado nessas atividades é ilustrada na Figura 2.1.

Figura 2.1: Framework Engenharia de LPS



Fonte: traduzido de (Pohl *et al.*, 2005)

A Engenharia de Domínio é a atividade que desenvolve o núcleo de artefatos, esse núcleo constitui recursos comuns à LPS (van der Linden *et al.*, 2007). Esta atividade

possui 5 fases principais: elicitaco, documentaco, negociao, validao e verificao, gerenciamento (Pohl *et al.*, 2005).

A Engenharia de Aplicao desenvolve o produto na LPS. Corresponde à gerao de produtos especficos e tem como propsito explorar os requisitos e construir os produtos finais. A variabilidade explicitamente modelada nesta atividade fornece a base para derivar os produtos individuais (van der Linden *et al.*, 2007; Pohl *et al.*, 2005).

O Gerenciamento de LPS corresponde à manuteno e evoluo da LPS. O controle das atividades j desenvolvidas, planejamento de novas funcionalidades e as melhorias dos artefatos j criados acontecem nessa atividade. As Engenharias de Domnio e de Aplicao influenciam fortemente uma à outra, e os processos devem ser bem alinhados para que sejam eficazes (van der Linden *et al.*, 2007).

2.3 Gerenciamento de Variabilidades e Abordagens Consideradas

A Engenharia de LPS é uma abordagem que desenvolve e mantm famlias de produtos tomando conhecimento dos seus aspectos comuns e das variaoes previstas (Weiss e Lai, 1999). Ao longo de dcadas, a variabilidade em ativos de software tornou-se cada vez mais importante na Engenharia de Software (van Gurp *et al.*, 2002) (Capilla *et al.*, 2013).

Os artefatos que so utilizados em diversos produtos devem ser adaptveis a diversos produtos de uma LPS. Isso significa que por meio do processo de desenvolvimento deve-se identificar e descrever onde os produtos de uma LPS podem se diferenciar em termos de caractersticas, arquitetura e requisitos. Para tal, é necessrio prover flexibilidade para que haja customizao em massa (Pohl *et al.*, 2005).

A flexibilidade descrita é chamada de variabilidade. A definio explcita e o gerenciamento dessas variabilidades é o que distingue uma LPS de sistemas nicos e do reso de software (Pohl *et al.*, 2005). Variabilidade é o conceito central no desenvolvimento de uma LPS. É ela quem possibilita a derivao de produtos de software diferentes, e surgem do adiamento de decisoes de projeto (Halmans e Pohl, 2003).

Segundo Capilla, Bosch, Kang (2013), variabilidade de software é a habilidade de um sistema de software ou artefato ser eficientemente estendido, mudado, customizado ou configurado para ser usado em um determinado contexto. Variabilidades devem ser exploradas e gerenciadas em cada fase do desenvolvimento de uma LPS (Halmans e Pohl, 2003).

Com base no pressuposto em como os membros de uma LPS podem se diferenciar uns dos outros, a variabilidade é definida durante a engenharia de domínio por meio da introdução de pontos de variação e variantes (Halmans e Pohl, 2003). Um ponto de variação, segundo Jacobson et al. (1997), indica o local onde a variação ocorrerá. Ou pode ser definido como uma decisão de projeto atrasada, como colocado por Bosch et al. (2002). As possíveis escolhas para esse ponto de variação são chamadas de variantes (Bachmann e Clements, 2005).

Definir variabilidades por meio de diferentes estágios do ciclo de vida de uma LPS é apoiado pelo conceito de gerenciamento de variabilidades e engloba as seguintes questões segundo Pohl, Bockle e Linden (2005):

- apóia as atividades relacionadas à definição de variabilidade;
- gerencia os artefatos variáveis;
- suporta atividades relacionadas à resolução de variabilidades;
- coleta, armazena e gerencia informações rastreadas para cumprir essas tarefas.

O gerenciamento de variabilidades abrange as atividades de representar explicitamente a variabilidade em artefatos de software por meio de ciclos de vida, gerenciando as dependências entre diferentes atividades e apoiando as instanciações dessas variabilidades (Schmid e John, 2004). Envolve tarefas extremamente complexas e desafiadoras, que precisam ser apoiadas por abordagens, técnicas e ferramentas (Bosch *et al.*, 2002; Schmid e John, 2004).

Pela importância do gerenciamento de variabilidades, na literatura há destaque para as abordagens baseadas em UML, uma notação padronizada, que expressa em seus modelos representações que auxiliam no processo de desenvolvimento de software. Exemplos dessas abordagens são: PLUS para diagramas de casos de uso e classes; Ziadi et al para diagramas de sequência, Razavian e Khosravi para diagramas de componentes e SMarty para diagramas de casos de uso, classes, sequência, atividades e componentes. Tais abordagens são apresentadas a seguir.

2.3.1 SMarty

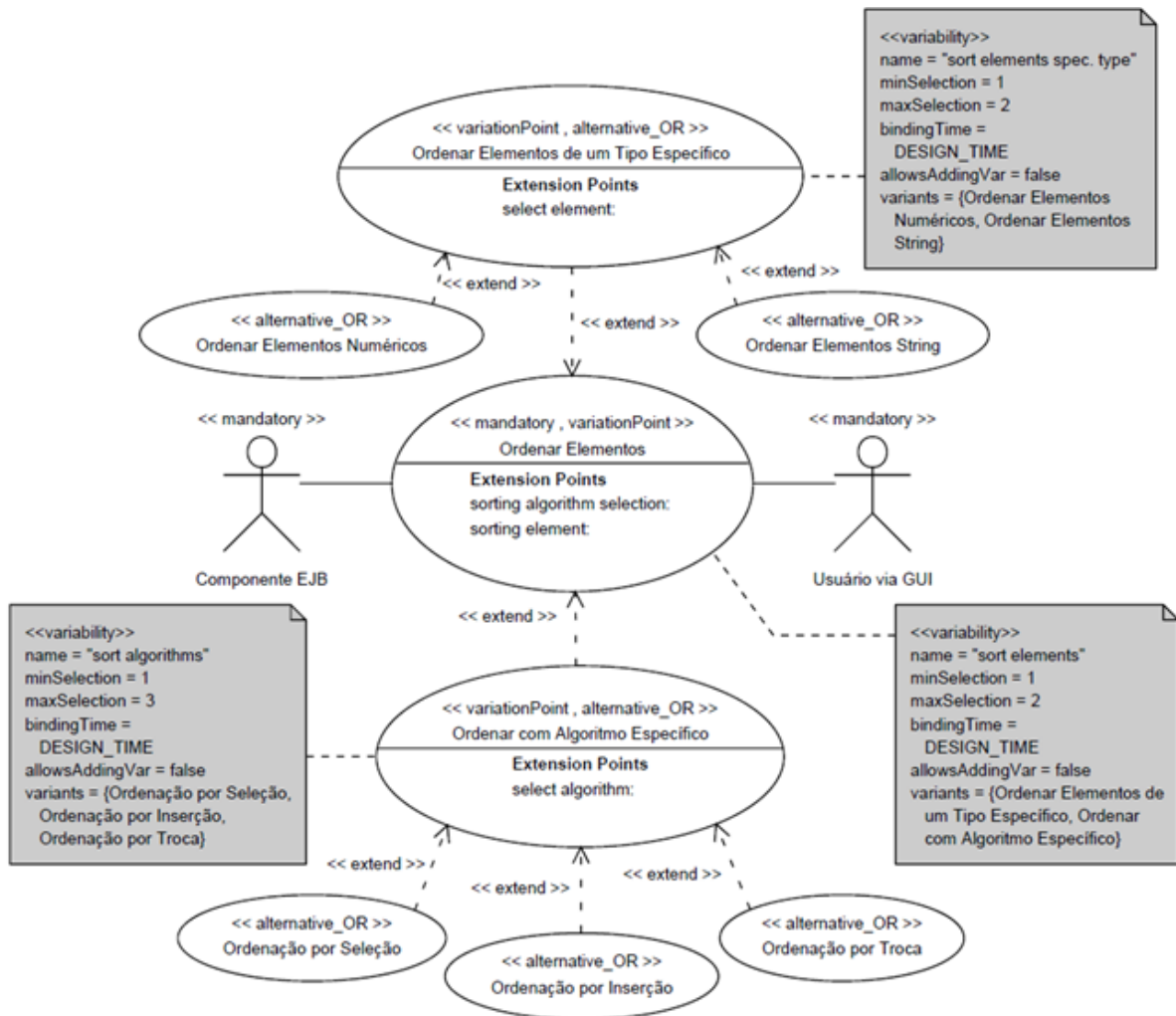
SMarty é uma abordagem de LPS baseada em UML, composta de um perfil UML 2, o SMartyProfile e o processo SMartyProcess. SMartyProfile é baseado no inter-relacionamento dos principais conceitos de gerenciamento de variabilidades em LPS. Contém um conjunto

de estereótipos e *tagget-values* para representar variabilidade em modelos de LPS (Oliveira Jr *et al.*, 2010). Os estereótipos são:

- «variability» representa o conceito de variabilidade;
- «variationPoint» representa o ponto de variação. Este estereótipo aplica-se somente a nós de decisão, atores, casos de uso, interface, classes e pacotes;
- «variant» refere-se ao conceito de variante;
- «mandatory» é usado para representar uma variante obrigatória presente em todos os produtos da LPS;
- «optional» representa uma variante que pode ser escolhida para resolver uma variabilidade ou um ponto de variação;
- «alternative_OR» representa uma variante que faz parte de um grupo de variantes inclusivas e diferentes combinações delas podem resolver pontos de variação, gerando produtos diferentes;
- «alternative_XOR» representa uma variante que participa de um grupo de variantes exclusivas, na qual apenas uma variante desse grupo pode ser escolhida para resolver o ponto de variação;
- «mutex» denota o conceito de exclusão mútua, significa que para uma variante ser escolhida, a variante relacionada não pode ser selecionada;
- «requires» é um relacionamento entre variantes, onde a variante escolhida requer outra variante relacionada;
- «variable» indica que um *component* é formado por um conjunto de classes com variabilidades explícitas.

Na Figura 2.2 é apresentado um exemplo da modelagem de um diagrama de casos de uso usando a abordagem SMarty. Nela é possível ver os estereótipos nos atores e nos casos de uso. Por exemplo, os atores “Componente EJB” e “Usuário via GUI” possuem os estereótipos «mandatory», ou seja, significa que são obrigatórios em qualquer produto dessa LPS. Assim como o caso de uso “Ordenar elementos”, que também é obrigatório e representa um ponto de variação. Os demais casos de uso possuem estereótipos «alternative OR», por isso podem ser escolhidos em diferentes combinações para resolver o ponto de variação ao qual pertencem, gerando produtos distintos.

Figura 2.2: Modelagem de um Diagrama de Casos de Uso com a abordagem SMarty



Fonte: (OliveiraJr *et al.*, 2010)

O SMartyProcess é um processo sistemático que guia o usuário na identificação, delimitação, representação e rastreamento de variabilidades em modelos de LPS (OliveiraJr *et al.*, 2010). Este processo está alinhado com as atividades da LPS e é realizado pelo engenheiro da LPS, sendo iterativo e incremental (OliveiraJr *et al.*, 2010).

Para que as atividades anteriores possam ser realizadas e concretizadas com sucesso o SMartyProcess fornece um conjunto de diretrizes (Fiori *et al.*, 2012)) (OliveiraJr *et al.*, 2010) (Marcolino e OliveiraJr, 2015). São elas:

Diretrizes para Identificação e Representação de Variabilidade (RV)

- RV.1 Variabilidades com variantes opcionais (*optional*) possuem multiplicidade $minSelection = 0$ e $maxSelection = 1$;

- RV.2 Variabilidades com variantes exclusivas (*alternative_XOR*) possuem multiplicidade $minSelection = maxSelection = 1$;
- RV.3 Variabilidades com variantes inclusivas (*alternative_OR*) possuem multiplicidade $minSelection = 1$ e $maxSelection = size(variants)$ em que $size(x)$ é uma função que retorna a quantidade de elementos da coleção x ;
- RV.4 O valor *bindingTime* deve ser definido escolhendo-se um dos valores da classe de enumeração *BindingTime*, que são: DESIGN_TIME, LINK_TIME, COMPLETE_TIME, RUNTIME;
- RV.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*); e
- RV.6 O valor da coleção de variantes é o conjunto formado pelas instâncias das variantes associadas ao ponto de variação ou variabilidade.

Diretrizes para Casos de Uso (UC)

- UC.1 Elementos de modelos de casos de uso relacionados aos mecanismos de extensão e de pontos de extensão sugerem pontos de variação com variantes associadas, que podem ser inclusivas ou exclusivas;
- UC.2 Modelos de casos de uso com o relacionamento de inclusão («include») ou associados a atores sugerem variantes obrigatórias ou opcionais;
- UC.3 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»;
- UC.4 Variantes mutuamente exclusivas para um determinado produto devem ter seus relacionamentos de dependência marcados com o estereótipo «mutex».

Diretrizes para Diagrama de Classes (CL)

- CL.1 Em modelos de classes, pontos de variação e as 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 os clientes (implementações) 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.
- CL.2 Elementos de modelos de classes, relacionados a associações nas quais os seus atributos *aggregationKind* possuem valor *none*, ou seja, não representam nem agregação, nem composição, sugerem variantes obrigatórias ou opcionais.
 - CL.3 Variantes em modelos de classes, 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»;
 - CL.4 Variantes em modelos de classes, mutuamente exclusivas para um determinado produto devem ter seus relacionamentos de dependência marcados com o estereótipo «mutex».

Diretrizes para Componentes (CP)

- CP.1 Componentes formados por classes com variabilidades são marcados com o estereótipo «variable».

Diretrizes para Diagrama de Atividades (AT)

- AT.1 Elementos de modelos de diagramas de atividades como *DecisionNode* sugerem pontos de variação marcados com «variationPoint», pois é um local formado explicitamente por possíveis caminhos para grupos de ações distintas;
- AT.2 Elementos Action dos diagramas de atividades podem ser definidos como variantes obrigatórias ou opcionais;
- AT.3 Elementos Action que representam fluxos alternativos de saída de um *DecisionNode* sugerem variantes alternativas inclusivas ou exclusivas;
- AT.4 Elementos *ActivityPartition* que possuem elementos variáveis, *DecisionNode* como ponto de variação ou *Action* como variantes, devem ser marcados como «variable», pois são compostos por elementos que sofrem algum tipo de variação.

Diretrizes para Diagrama de Sequência

- SQ.1 Elementos de diagramas de sequência como *CombinedFragment* que possuem do *interactionOperator* do tipo "alt" (*alternative*), indicam que apenas um fluxo do *CombinedFragment* será realizado, ou seja, sugerem variantes mutuamente exclusivas onde os pontos de variação serão anotados como «variationPoint» e serão relacionados a um comentário da UML especificando a variabilidade («variability»). As variantes correspondentes às mensagens devem ser estereotipadas como «alternative_XOR»;
- SQ.2 Em diagramas de sequência, as duas possíveis ocorrências a seguir, sugerem variantes opcionais: a) Elementos de diagramas de sequência como o *CombinedFragment* que possuem *interactionOperator* do tipo "opt" (optional) sugerem variantes opcionais, sendo estereotipados como «optional», e são relacionados a um comentário da UML especificando a variabilidade («variability»). Os *lifelines* contidos nesse *CombinedFragment* e que fazem parte da variabilidade deverão ser estereotipados também como «optional»; b) Troca de mensagens entre dois objetos não obrigatórios, ou entre um objeto obrigatório e outro não, sugerem uma variante opcional, estereotipadas como «optional» e estarão relacionados a um comentário da UML especificando a variabilidade («variability»). A(s) *lifeline(s)* correspondente(s) a essa variante será(ao) estereotipada(s) também como «optional».
- SQ.3 O elemento *interactionUse* "ref" sugere ponto de variação para variantes alternativas inclusivas, sendo estereotipado como «variationPoint» e relacionado a um comentário da UML, que identifica os elementos da variabilidade («variability»). Os diagramas de sequência referenciados pelo *interactionUse* "ref" correspondem às variantes do ponto de variação, são considerados portanto, alternativos inclusivos, podendo um ou mais serem selecionados, sendo estereotipadas como «alternative_OR».
- SQ.4 as mensagens (*messages*) que são independentes dos fluxos contidos no *CombinedFragment* "alt", "opt", *interactionUse* "ref", ou não estejam relacionadas diretamente a uma variabilidade e seus elementos, são mantidas sem estereótipos e consideradas assim, obrigatórias;
- SQ.5 Variantes em diagramas de sequência que, ao serem selecionadas para fazer parte de um produto específico, exigirem a presença de outra(s) determinada(s) variante(s) devem ter seus relacionamentos de dependência marcados com o estereótipo «requires»;

- SQ.6 Variantes mutuamente exclusivas de um diagrama de sequência, para um determinado produto devem ter seus relacionamentos de dependência marcados com o estereótipo «mutex».

2.3.2 PLUS

No PLUS há uma divisão das atividades de requisitos, análise e projeto para LPS. Este método permite o desenvolvimento de LPS por meio da sua integração a outros modelos de processo de software, como o processo de desenvolvimento unificado. Este método objetiva a modelagem explícita das características comuns e variáveis de uma LPS, por meio de atividades apoiadas por extensões UML, para modelos de casos de uso e classes (Gomaa, 2005).

O método PLUS permite a sua integração com outros modelos de processo de software, como o processo unificado de desenvolvimento de software (PU). Os nomes das fases do PLUS também coincidem com o nome das fases do PU e são descritas por Gomaa (2005) da seguinte maneira:

- 1- *Inception*: estudo para determinar se a LPS é viável ou não, o seu contexto, funcionalidades, grau de semelhança e variabilidades;
- 2- *Elaboration*: o modelo de casos de uso e *features* são revisados e elaborados em maiores detalhes, identificando seus pontos de variação. A arquitetura de LPS é expandida incluindo componentes opcionais e variantes;
- 3- *Construction*: Os componentes são desenvolvidos e testados;
- 4- *Transition*: Os componentes são integrados e ficam disponíveis para os usuários poderem testar.

O método PLUS permite a identificação de componentes de variantes por meio do uso de estereótipos para representar a variabilidade. Estes são descritos a seguir (Gomaa, 2005):

- «kernel» usado para representar elementos obrigatórios;
- «optional» usado para representar elementos opcionais que podem ser selecionados ou não para um produto específico;
- «alternative» usado para representar elementos alternativos, mutuamente exclusivos ou inclusivos. Este estereótipo não é suportado por diagramas de classes.

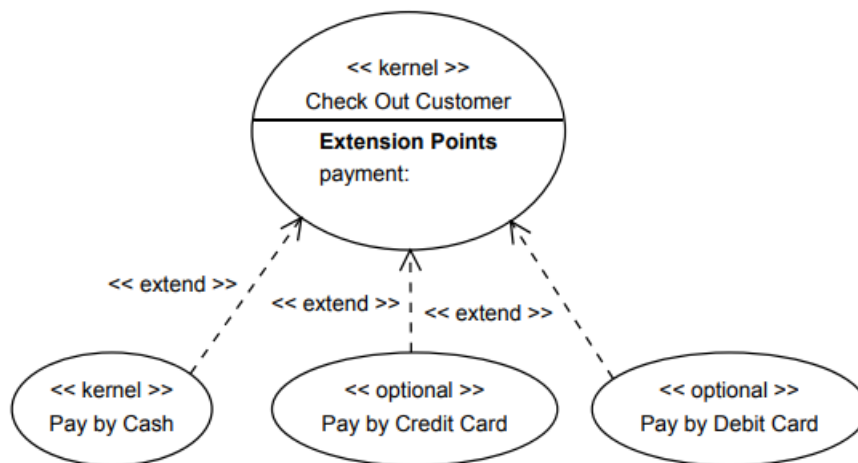
O método PLUS não oferece nenhum tipo de apoio no rastreamento de elementos entre os digramas UML que esse método suporta.

Na Figura 2.3 podemos observar o uso dos estereótipos. Para os casos de uso obrigatórios, *Check Out Customer* e *Pay by Cash*, há a aplicação do estereótipo «kernel», logo, esta variabilidade, e ponto de variação, são obrigatórios e estarão presente em todos os produtos delas derivadas.

Já o caso de uso *Pay by Credit Card* e *Pay by Debit Card* são opcionais («optional»), podendo ou não ser inseridos como funcionalidade.

As regras aplicadas à esta LPS são meramente demonstrativas, pode ser necessário formar um grupo de variantes assim estereotipadas, atendendo às restrições e regras de negócio aplicadas na LPS.

Figura 2.3: Modelagem de um Diagrama de Casos de Uso com a abordagem PLUS



Fonte: (Gomaa, 2005)

2.3.3 Razavian e Khosravi

Razavian e Khosravi Razavian e Khosravi (2008) propõem uma maneira de modelar rastreabilidade em componentes e conectores. Para isso, fazem uso dos seguintes estereótipos:

- «altvp» usado para representar os pontos de variação do tipo alternativos, onde suas variantes são alternativas e representadas pelo estereótipo «variant»;
- «optvp» usado para representar os pontos de variação do tipo opcional, onde suas variante são opcionais representadas pelo estereótipo «variant»;
- «variant» usado para representar as variantes associadas a um determinado ponte de variação;

- «optional» usado para representar elementos que podem ou não ser selecionados;
- «altv_vp» utilizado para representar pontos de variação existentes em variantes, e que são do tipo alternativos;
- «optv_vp» usado para representar pontos de variação existentes em variantes do tipo opcionais.

A abordagem Razavian e Khosravi é baseada em UML 2, o benefício é que o arquiteto usa uma linguagem padrão, amplamente utilizada entre os profissionais. Essa abordagem não conta com mecanismos para auxiliar no rastreamento entre variabilidades.

Para o exemplo de linha de produto "AGM", o componente identificado como ponto de variação, contendo variantes alternativas, está anotado com o estereótipo «altvp» (GameCtrl), e suas variantes, com o estereótipo «variant», indicando que ao menos uma variante deve ser selecionada. Os elementos considerados opcionais, neste caso as interfaces ISaveScore e ICheckScore, foram anotadas com o estereótipo «optional», indicando que elas podem ou não existirem no produto configurado.

2.3.4 Ziadi et al

A abordagem Ziadi et al (Ziadi *et al.*, 2003) (Ziadi e Jezequel, 2006), é uma das mais representativas abordagens de gerenciamento de variabilidades para modelos de sequência, suportando também modelos de classes.

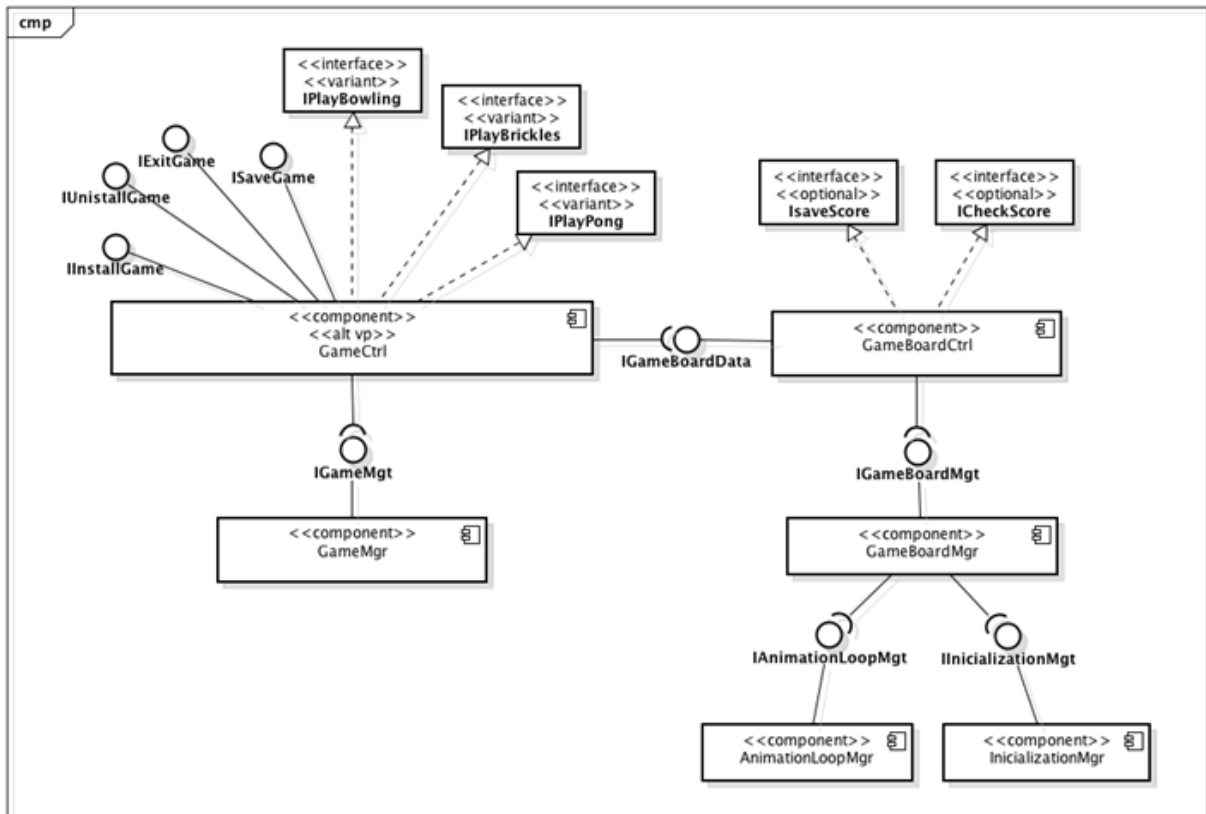
A abordagem Ziadi et al é composta de um perfil UML 2.0 que permite a integração com ferramentas UML, possibilitando a representação de variabilidades.

Há um conjunto de meta-atributos e metaclasses para permitir a anotação dos diversos elementos na representação de variabilidades.

Os estereótipos propostos por Ziadi et al (Ziadi *et al.*, 2003) para diagramas de sequência e classes estão descritos a seguir:

- «optionalLifeline» usado para representar lifelines alternativas ou opcionais, este estereótipo é suportado apenas por modelos de sequência;
- «optionalInteraction» usado para representar interações opcionais. Indica os elementos que podem ser selecionados ou não, para um produto específico, é suportado por modelos de sequência;
- «optional» usado para representar elementos opcionais, como classes e pacotes, que podem ser incluídos ou não, em produtos específicos, é um estereótipo suportado apenas por modelos de classes;

Figura 2.4: Modelagem de um Diagrama de Componentes com a abordagem Razavian e Khosravi



Fonte: (Razavian e Khosravi, 2008)

- «variation» usado para representar pontos de variação, que por sua vez estão relacionados a variantes inclusivas ou exclusivas, usado em modelos de classes e sequencia;
- «variant» usado para representar variantes e também é suportado para modelos de classes e sequencia;
- «virtual» usado para indicar quando uma interação representa uma situação específica em que o comportamento de um diagrama de sequência pode ser representado por outro diagrama de sequência, usado apenas em modelos de sequencia.

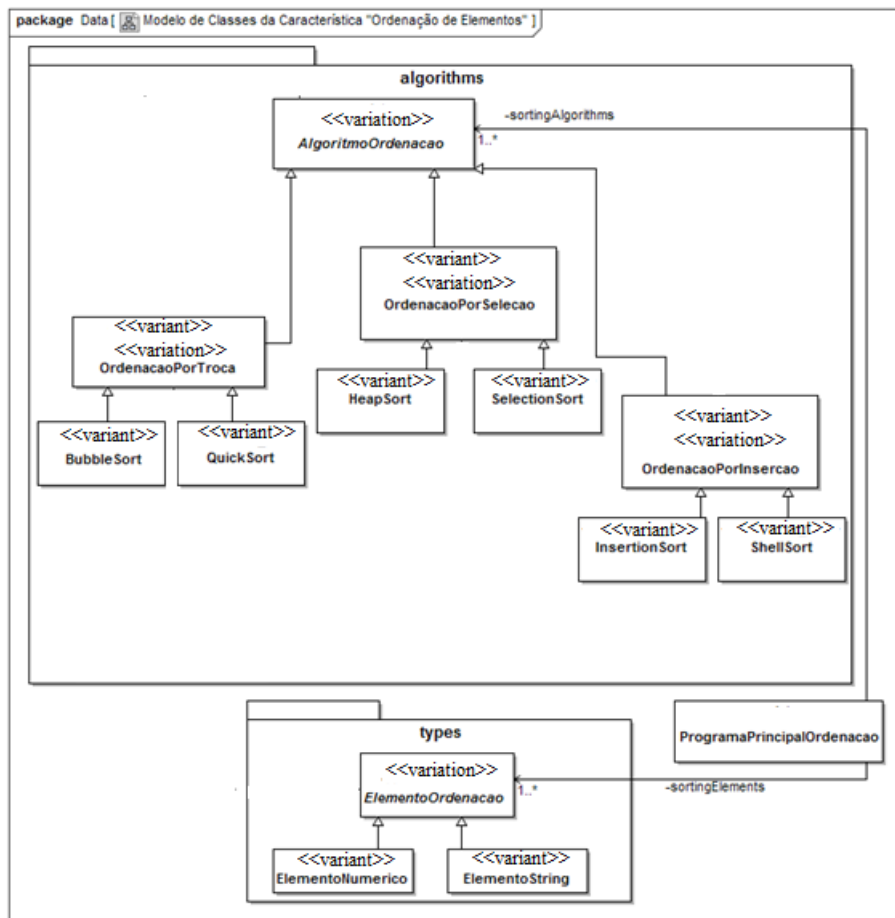
Na Figura 2.5 observamos a aplicação da abordagem Ziadi et al, e seus elementos. Passamos a analisar cada um deles, bem como as diretrizes presentes no processo da abordagem Ziadi et al, que auxiliam sua utilização em outras LPS:

A classe AlgoritmoOrdenacao identifica uma classe obrigatória e representa também um ponto de variação («variation»), com três variantes. Estas variantes estão descritas

no elemento comentário, relacionado a classe, por meio do *TaggedValue* (variants). As três variantes desta classe são OrdenacaoPorTroca, OrdenacaoPorSelecao e OrdenacaoPorInsercao. Todas estas são estereotipadas como «variant», o que indica o tipo de restrição para tais variantes, neste caso, significa que ao menos uma ou todas elas podem solucionar o ponto de variação.

OrdenacaoPorTroca, OrdenacaoPorSelecao e OrdenacaoPorInsercao, além de variantes («variant»), são, por sua vez, pontos de variação («variation»), e assim uma delas, ao menos, deve ser selecionada ou todas. A classe ProgramaPrincipalOrdenacao, representa uma classe obrigatória, portanto estará presente em todos os produtos desta LPS. A classe ElementoOrdenacao, também é obrigatória e representa um ponto de variação («variation») e possui duas classes variantes: ElementoNumerico e ElementoString, marcadas como variantes «variant», onde, ambas podem ser selecionadas, ou ao menos uma.

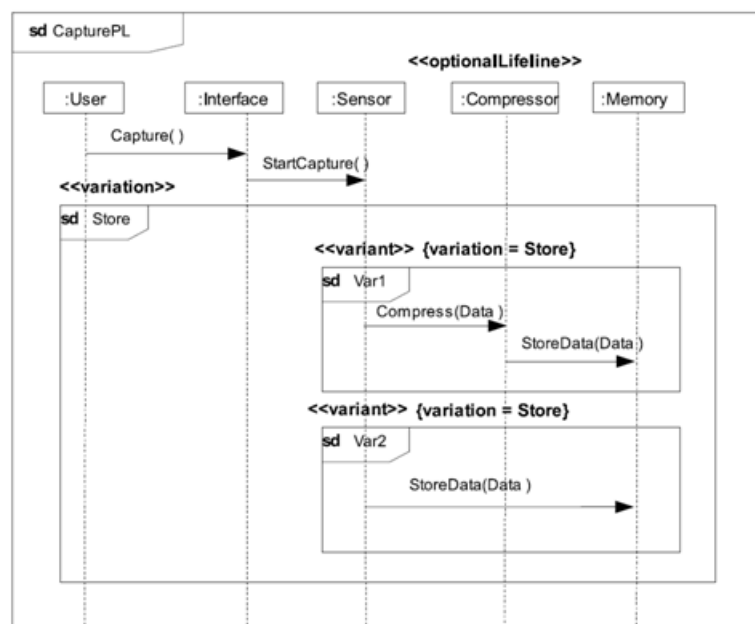
Figura 2.5: Modelagem de um Diagrama de Classes com a abordagem Ziadi et al



fonte: adaptado de (OliveiraJr *et al.*, 2010)

O trecho de LPS apresentado na Figura 2.6 corresponde a uma câmera fotográfica, que possui uma interface com usuário (*interface*), o sensor de captura (*sensor*), o compressor das imagens capturadas (*compressor*) e a memória (*memory*) onde tais imagens são armazenadas (Ziadi *et al.*, 2003). Para o exemplo de linha de produto de "Câmera Digital", os elementos representados pelas linhas de vida (*lifelines*) ou demais elementos sem a aplicação de estereótipos são considerados obrigatórios, ou seja, estarão presentes em todos os produtos.

Figura 2.6: Modelagem de um Diagrama de Sequência com a abordagem Ziadi et al



Fonte (Ziadi *et al.*, 2003)

O Elemento Compressor recebe o estereótipo «optionalLifeline», designando que o objeto apresentado por esta linha de vida é opcional, e será inserido ou não, segundo o ponto de variação e variantes a qual ele faz parte. Neste exemplo, somente se a variante de nome Var1 for selecionada no produto, esta linha de vida deverá fazer parte do produto a ser criado.

A variabilidade *Store* é representada por meio do conjunto de *frame*, com os *interactionsOperators* (operadores de interação) de valores: *sd Store*, *sd Var1* e *sd Var2*. O *frame* com o *interactionOperator sd Store*, corresponde ao ponto de variação, recebendo, desta forma, o estereótipo «variation», por sua vez, os *frames* inseridos no *frame sd Store*, com o *interactionOperator sd Var1* e *sd Var2*, correspondem as variantes, logo recebem o estereótipo «variant» com o meta-atributo *variation = Store*, onde, *Store* corresponde ao nome do ponto de variação a qual aquela variante pertence.

Ziadi et al não oferece nenhum tipo de apoio no rastreamento de elementos entre os digramas UML que esse método suporta.

2.4 Configuração de Produtos

A abordagem de LPS contribui para a obtenção de produtos a partir de uma plataforma formada por características comuns e variáveis, o núcleo de artefatos. Na Engenharia de LPS, é a Engenharia de Aplicação que usa o que foi definido como ativos comuns na Engenharia de Domínio para a criação dos produtos finais. São as variabilidades que tornam possível a criação de produtos distintos em um mesmo domínio (van der Linden *et al.*, 2007; Pohl *et al.*, 2005).

Existem várias abordagens de configuração de produtos, como *Modeling Scenario Variability as Crosscutting Mechanisms* (MSVCM), *Variability Modeling Language* (VML) e a *Common Variability Language* (CVL). Todas elas partem do espaço do problema usando *features*, uma característica do sistema visível ao usuário final, definida como uma unidade lógica de comportamento que é especificada por um conjunto de requisitos funcionais e de qualidade (Bosch, 2000). Neste trabalho estamos interessados no espaço da solução usando UML.

Segundo Bonifácio e Borba Bonifácio e Borba (2009) e Bonifácio, Teixeira e Borba Bonifácio *et al.* (2009), em princípio a MSVCM foi proposta para lidar com variabilidades de requisitos, mas foi ampliada para lidar com variabilidades no código-fonte. Nessa abordagem o conhecimento de configuração é especificado em um modelo separado, relacionando *features* e suas combinações (expressões de *features*) às transformações que traduzem artefatos e LPS em artefatos específicos do produto. Para lidar com variabilidades de requisitos, três transformações foram definidas: seleção de cenário (*selectScenario*), seleção de adendos (*selectAdvice*) e a criação de vínculo de parâmetros com o cenário selecionado (*bindParameter*).

O VML pode ser usado para especificar derivações de produtos de modelos de arquitetura. É uma linguagem textual que facilita a composição de arquiteturas reusáveis de software e se baseia em abstrações típicas de arquitetura projetos, como componentes ou interfaces. Essa linguagem fornece mecanismos de composição para especificar explicitamente variabilidades relativas às preocupações dos arquitetos de LPS. Uma especificação VML é composta de interesses. Cada interesse pode conter vários pontos de variação, que podem ser alternativos ou opcionais. VML também fornece construções para especificar a configuração de produtos de LPS, ou seja, seleção de variantes que devem ser incluídas em um produto específico (Sánchez *et al.*, 2009).

A CVL foi proposta pela OMG como uma padronização de modelagem de variabilidade. Ela expressa a variabilidade em termos de fragmentos do modelo, como fragmento de posicionamento (pontos de variação) e fragmento de substituição (variantes). A materialização dos modelos de produtos é realizada por meio de fragmentos de substituição entre o modelo base (posicionamento) e uma biblioteca de modelos (substituições) (Echeverría *et al.*, 2015). Porém, CVL possui problemas em suas ferramenta de apoio. Nenhuma dessas abordagens citadas, para configuração de produtos pode ser utilizada neste trabalho.

2.5 Rastreabilidade em LPS

Na maioria das organizações e projetos com processo de Engenharia de Software bem desenvolvido, softwares e artefatos criados acabam se desconectando entre si (Cavalcanti *et al.*, 2012).

Existem diversos fatores que levam à falta de rastreabilidade entre os artefatos, alguns deles são relacionado a seguir (Rilling *et al.*, 2007):

- os artefatos podem estar escritos em linguagens diferentes;
- o sistema é descrito considerando vários níveis de abstração;
- processos de software não consideram a manutenção dos links de rastreabilidade como uma prática obrigatória;
- falta de suporte de uma ferramenta para criar e manter a rastreabilidade entre os artefatos de software.

A falta de rastreabilidade entre os artefatos de software pode ser considerado o maior desafio para a atividade de manutenção de software (Rilling *et al.*, 2007). Como resultado, durante a compreensão dos sistemas, os engenheiros de software tem muito esforço em sintetizar e integrar informações de várias fontes para estabelecer links entre esses artefatos (Cavalcanti *et al.*, 2012).

Ao estabelecer a rastreabilidade, os engenheiros tem capacidade de entender como os artefatos de software interagem entre si, em termos de relações e dependências (Cavalcanti *et al.*, 2012).

Os links de rastreabilidade são úteis quando se considera a característica evolutiva do software com artefatos que provavelmente ainda serão alterados no seu ciclo de vida (Cavalcanti *et al.*, 2012).

De acordo com Anquetil *et al.* (2010), estabelecer a rastreabilidade gera uma série de benefícios, como:

- relacionar artefatos de software e decisões de design correspondentes;
- fornecer *feedback* aos arquitetos e *designers* sobre o estado atual de desenvolvimento, permitindo que eles considerem decisões de projetos e entendam erros;
- facilitar a comunicação entre partes desejadas.

Em LPS, onde os ativos podem ser usados por muitos produtos, a rastreabilidade é ainda mais importante. Uma mudança em um artefato em uma LPS pode levar a alterações resultantes em todos os produtos que foram desenvolvidos reutilizando esse artefato. Portanto, é necessário definir relações de dependência entre os artefatos para apoiar a integração consistente de mudanças (Moon *et al.*, 2007).

2.6 Experimentação em Engenharia de Software

A experimentação está no centro do processo científico (Travassos, 2002) como uma das ferramentas para verificação e validação das teorias na ciência. O método científico clássico depende da formulação de uma teoria, da sua experimentação e observação provendo um *loop* para validar, modificar e melhorar uma teoria (Zelkowitz e Wallace, 1997). Algumas atividades são parte essencial do método científico segundo Juristo e Moreno (2010), sendo elas:

- interação com a realidade: pode ser realizada através da observação e experimentação. Por meio da observação, os pesquisadores percebem os fatos, mas não interagem com ele. Por meio da experimentação, os pesquisadores entram em contato com o objeto de estudo, submetendo-o a novas condições para observar as reações;
- especulação: os pesquisadores especulam sobre a percepção do mundo lá fora;
- confronto com a realidade: a experimentação é de novo usada para comparar especulações teóricas e a realidade. Desta vez, as novas condições as quais os objetos são submetidos são especialmente planejadas para confirmar ou refutar a hipótese.

Segundo Tichy (1998), a experimentação é usada principalmente onde a teoria e análise dedutiva não alcançam. Os experimentos provam as suposições, eliminam alternativas de explicações de um fenômeno e expõem novos fenômenos que precisam de explicações. Desta forma, experimentos ajudam com a indução ao derivarem teorias da observação. A experimentação cria um método “qualitativo e/ou quantitativo, aplica um experimento, mede e analisa, avalia o modelo e repete o processo”.

A Engenharia de Software é uma ciência laboratorial que não pode confiar apenas na observação e no pensamento lógico (Basili, 1996). Necessita do processo de experimentação para as novas abordagens, técnicas e processos que são desenvolvidos por ela, assim como toda ciência e/ou engenharia (Zelkowitz e Wallace, 1997).

A informação é um dos principais assuntos da computação, já que ela visa estudar as estruturas e processos da informação (Tichy, 1998). Um cientista da computação para entender o processo da informação deve observar um fenômeno, formular explicações e testá-las. O processo deve ser repetido para garantir que o resultado possa ser checado independentemente, aumentando assim a sua confiança, permitindo testar ou refutar um produto ou processo para entender como funciona, seus limites e como pode ser melhorado (Basili, 1996).

Em outras palavras a experimentação em engenharia de software tem como objetivo a “caracterização, avaliação, previsão, controle e melhoria a respeito de produtos, processos, recursos, modelos, teorias, entre outros” (Travassos, 2002).

Segundo Juristo e Moreno (2010), o processo de experimentação pode ser dividido com as seguintes atividades:

1. Definição dos objetivos do experimento
2. Projeto do experimento
3. Execução do experimento
4. Análise dos resultados obtidos

Essas atividades são descritas conforme Juristo e Moreno (2001) e Wohlin et al (2012) a seguir.

Durante a definição dos objetivos do experimento, uma hipótese geral é transformada em uma hipótese definida em termos das variáveis do fenômeno que vai ser examinado. Nesta fase o experimento é definido em termos do problema, objetivos e metas.

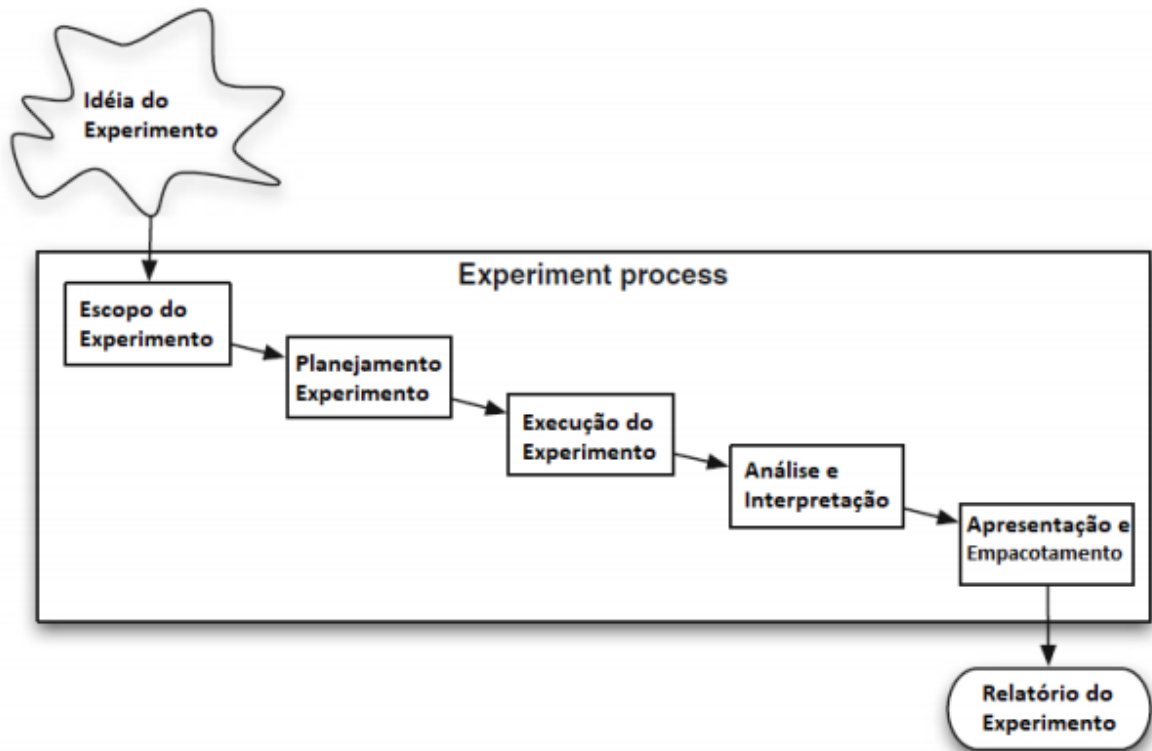
No projeto, a atividade a ser realizada envolve um plano de como o experimento deverá ser conduzido. Ele será elaborado determinando as condições exatas as quais o experimento será submetido. Isso envolve determinar quais variáveis podem afetar o experimento, ou seja, as ameaças, além de considerar a instrumentação.

No estágio de execução, os primeiros experimentos são conduzidos, seguindo o projeto realizado. Esses dados são analisados e interpretados na fase seguinte, testando a hipótese proposta no início.

Após estas fases, Wohlin et al (2012) ainda consideram outra atividade pertencente ao processo de experimentação, em que os dados são apresentados e empacotados.

A Figura 2.7 ilustra a sequência das atividades descritas do processo de experimentação.

Figura 2.7: Processo de Experimentação em Engenharia de Software



Fonte: traduzido de (Wohlin *et al.*, 2012)

2.7 Trabalhos Relacionados

Vários experimentos já foram realizados por pelo Grupo de Pesquisa para comparar a efetividade da abordagem SMarty com outras abordagens de gerenciamento de variabilidades em LPS.

Marcolino Marcolino e Oliveira Jr (2015) apresentou os resultados de diversas avaliações realizadas para obter evidência de que a abordagem SMarty fosse mais efetiva que as demais.

Em 2013, Marcolino *et al.* (2013) conduziu uma avaliação para comparar a abordagem SMarty e o método PLUS com diagramas de casos de uso. Neste experimento, houve evidências de que a abordagem SMarty era mais efetiva que a PLUS.

Em outro experimento aplicado para diagrama de sequencia, realizado em 2014 por Marcolino *et al.* (2014a), comparando a abordagem SMarty com a Ziadi et al, também houve evidências de que a abordagem SMarty era mais efetiva.

Em 2014, Marcolino *et al.* (2014b), realizou dois experimentos com diagramas de classes. O primeiro indicou que o método PLUS era mais efetivo, o segundo, mesmo

depois de algumas melhorias na abordagem SMarty, o método PLUS se mostrou mais efetivo.

Outro experimento aplicado por Marcolino *et al.* (2017), comparava a abordagem SMarty com o método PLUS em relação à resolução de variabilidade e configuração de produtos, em diagramas de classes. Nesta avaliação, a hipótese nula foi aceita, ou seja, não havia diferença estatística entre as medias de efetividade na configuração de produtos com a abordagem SMarty ou o método PLUS.

Por fim, um experimento realizado por Nepomuceno e OliveiraJr (2018), comparava a abordagem SMarty com o método PLUS em diagramas de casos de uso. Neste estudo, os resultados não evidenciaram vantagem de uma abordagem em relação à outra.

2.8 Considerações Finais

Neste capítulo foram abordados os principais referenciais teóricos a fim de auxiliar no entendimento do trabalho proposto: como a criação de diretrizes e a condução dos estudo experimentais.

As principais abordagens de Gerenciamento de Variabilidades foram descritas para possibilitar o entendimento dos diversos cenários de experimentos propostos: com diagramas de casos de uso, classes, componentes e sequência da UML. Além disso, foi discutido sobre rastreabilidade entre modelos, experimentação em Engenharia de Software e configuração de produtos, conceitos também utilizado para a execução dos experimentos.

Tendo como base os fundamentos apresentados, nos próximos capítulos são relatados os experimentos, bem como análises e discussão dos resultados.

Protocolo dos Experimentos

3.1 Considerações Iniciais

Segundo Wohlin *et al.* (2012) o projeto de um experimento refere-se ao plano de como o experimento deverá ser conduzido, prevendo as condições as quais o experimento será submetido. Neste capítulo será explicada a metodologia aplicada em cada experimento realizado, os objetivos com a aplicação de tal experimento, bem como a instrumentação e as hipóteses levantadas. As diretrizes definidas por Furtado (2018) foram utilizadas para documentar tais informações.

3.2 Metodologia e Planejamento Experimental

As opções selecionadas para serem comparadas com a abordagem SMarty, foram o método PLUS, Razavian e Khosravi e Ziadi *et al.*

O método PLUS foi escolhido devido ao suporte a diagramas de casos de uso e classes da UML, usados nos primeiros experimentos dessa pesquisa. A seleção da abordagem Razavian e Khosravi se deu devido a ausência de suporte do método PLUS para diagramas de componentes da UML. E da mesma maneira, pela falta de suporte das abordagens já escolhidas para diagramas de sequência, se deu a seleção da abordagem Ziadi *et al.*

Ao todo, foram realizados três experimentos e a partir deles todo o trabalho esta estruturado, são eles:

- 1 - Estudo Experimental em Diagramas de Casos de Uso e Classes
- 2 - Estudo Experimental em Diagramas de Classes e Componentes

3 - Estudo Experimental em Diagramas de Classes e Sequência

Os itens a seguir apresentam a definição dos elementos experimentais essenciais para o planejamento dos experimentos realizados. Para tanto, seguimos as diretrizes e recomendações para documentação de experimentos de LPS de Furtado (2018). Cada diretriz possui a seguinte nomenclatura "D.x", sendo x o número da diretriz.

3.2.1 Objetivos (D.1)

Experimento #1:

O objetivo da avaliação experimental foi comparar a efetividade da abordagem SMarty e do método PLUS com relação à configuração de produtos específicos a partir de diagramas de casos de uso e classes utilizando a LPS Mobile Media, além de averiguar a efetividade do suporte que cada abordagem fornece na rastreabilidade de elementos entre os diagramas citados. Este estudo é caracterizado como um quasi-experimento, pois a seleção dos participantes não foi randomizada com base no fato que os participantes foram escolhidos por conveniência.

Experimento #2:

O objetivo da avaliação experimental foi comparar a efetividade da abordagem SMarty em relação ao método PLUS e a abordagem Razaviam na configuração de produtos específicos a partir de diagramas de classes e componentes utilizando a LPS Mobile Media, além de averiguar o suporte que a abordagem SMarty fornece na rastreabilidade de elementos entre os diagramas citados. Este estudo também é caracterizado como um quasi-experimento, pois a seleção dos participantes não foi randomizada com base no fato que os participantes foram escolhidos por conveniência.

Experimento #3:

O objetivo da avaliação experimental foi comparar a efetividade da abordagem SMarty em relação à abordagem Ziadi et al na configuração de produtos específicos a partir de diagramas de classes e sequência utilizando a LPS AGM, além de averiguar o suporte que a abordagem SMarty fornece na rastreabilidade de elementos entre os diagramas citados. Este estudo também é caracterizado como um quasi-experimento, pois a seleção dos participantes não foi randomizada com base no fato que os participantes foram escolhidos por conveniência.

3.2.2 Formulação das Hipóteses (D.2)

Experimento #1:

Para o diagrama de caso de uso:

- Hipótese Nula ($H0_{efet}$): não há diferença significativa de efetividade entre as abordagens SMarty e PLUS na geração de produtos específicos de LPS a partir de diagramas de casos de uso.

$$H0_{efet}: \mu(\text{Efetividade}(\text{SMarty})) = \mu(\text{Efetividade}(\text{PLUS}))$$

- Hipótese Alternativa ($H1_{efet}$): existe diferença significativa de efetividade entre as abordagens SMarty e PLUS na geração de produtos específicos de LPS a partir de diagramas de casos de uso.

$$H1_{efet}: \mu(\text{Efetividade}(\text{SMarty})) \neq \mu(\text{Efetividade}(\text{PLUS}))$$

Para o diagrama de classes:

- Hipótese Nula ($H0_{efet}$): não há diferença significativa de efetividade entre as abordagens SMarty e PLUS na geração de produtos específicos de LPS a partir de diagramas de classes.

$$H0_{efet}: \mu(\text{Efetividade}(\text{SMarty})) = \mu(\text{Efetividade}(\text{PLUS}))$$

- Hipótese Alternativa ($H1_{efet}$): existe diferença significativa de efetividade entre as abordagens SMarty e PLUS na geração de produtos específicos de LPS a partir de diagramas de classes.

$$H1_{efet}: \mu(\text{Efetividade}(\text{SMarty})) \neq \mu(\text{Efetividade}(\text{PLUS}))$$

Para a quantidade de consultas aos materiais de apoio:

- Hipótese Nula ($H0_{cons}$): não há diferença significativa entre as abordagens SMarty e PLUS no número de consultas aos materiais instrucionais ao gerar produtos específicos.

$$H0_{cons}: \mu(\text{Consultas}(\text{SMarty})) = \mu(\text{Consultas}(\text{PLUS}))$$

- Hipótese Alternativa ($H1_{cons}$): existe diferença significativa entre as abordagens SMarty e PLUS no número de consultas aos materiais instrucionais ao gerar produtos específicos.

$$H1_{cons}: \mu(\text{Consultas}(\text{SMarty})) \neq \mu(\text{Consultas}(\text{PLUS}))$$

Para a efetividade do rastreamento de elementos entre diagramas com cada abordagem:

- Hipótese Nula ($H0_{efet}$): não há diferença significativa de efetividade entre a rastreabilidade de elementos entre os diagramas de caso de uso e classe usando as abordagens SMarty e PLUS.

$$\mathbf{H0}_{efet}: \mu(\mathbf{Efetividade(SMarty)}) = \mu(\mathbf{Efetividade(PLUS)})$$

- Hipótese Alternativa ($H1_{efet}$): existe diferença significativa de efetividade entre a rastreabilidade de elementos entre os diagramas de caso de uso e classe usando as abordagens SMarty e PLUS.

$$\mathbf{H1}_{efet}: \mu(\mathbf{Efetividade(SMarty)}) \neq \mu(\mathbf{Efetividade(PLUS)})$$

Experimento #2:

Para o diagrama de classes:

- Hipótese Nula ($H0_{efet}$): não há diferença significativa de efetividade entre as abordagens SMarty e PLUS na geração de produtos específicos de LPS a partir de diagramas de classes.

$$\mathbf{H0}_{efet}: \mu(\mathbf{Efetividade(SMarty)}) = \mu(\mathbf{Efetividade(PLUS)})$$

- Hipótese Alternativa ($H1_{efet}$): existe diferença significativa de efetividade entre as abordagens SMarty e PLUS na geração de produtos específicos de LPS a partir de diagramas de classes.

$$\mathbf{H1}_{efet}: \mu(\mathbf{Efetividade(SMarty)}) \neq \mu(\mathbf{Efetividade(PLUS)})$$

Para a quantidade de consultas ao material de apoio:

- Hipótese Nula ($H0_{cons}$): não há diferença significativa entre as abordagens SMarty e PLUS no número de consultas aos materiais instrucionais ao gerar produtos específicos.

$$\mathbf{H0}_{cons}: \mu(\mathbf{Consultas (SMarty)}) = \mu(\mathbf{Consultas (PLUS)})$$

- Hipótese Alternativa ($H1_{cons}$): existe diferença significativa entre as abordagens SMarty e PLUS no número de consultas aos materiais instrucionais ao gerar produtos específicos.

$$\mathbf{H1}_{cons}: \mu(\mathbf{Consultas (SMarty)}) \neq \mu(\mathbf{Consultas (PLUS)})$$

Para o diagrama de Componentes:

- Hipótese Nula ($H0_{efet}$): não há diferença significativa de efetividade entre as abordagens SMarty e Razavian e Khosravi na geração de produtos específicos de LPS a partir de diagramas de componentes.

$$\mathbf{H0}_{efet}: \mu(\mathbf{Efetividade(SMarty)}) = \mu(\mathbf{Efetividade(Razavian)})$$

- Hipótese Alternativa ($H1_{efet}$): existe diferença significativa de efetividade entre as abordagens SMarty e Razavian na geração de produtos específicos de LPS a partir de diagramas de componentes.

$$H1_{efet}: \mu(\text{Efetividade}(\text{SMarty})) \neq \mu(\text{Efetividade}(\text{Razavian}))$$

Para a quantidade de consultas ao material de apoio:

- Hipótese Nula ($H0_{cons}$): não há diferença significativa entre as abordagens SMarty e Razavian no número de consultas aos materiais instrucionais ao gerar produtos específicos.

$$H0_{cons}: \mu(\text{Consultas}(\text{SMarty})) = \mu(\text{Consultas}(\text{Razavian}))$$

- Hipótese Alternativa ($H1_{cons}$): existe diferença significativa entre as abordagens SMarty e Razavian no número de consultas aos materiais instrucionais ao gerar produtos específicos.

$$H1_{cons}: \mu(\text{Consultas}(\text{SMarty})) \neq \mu(\text{Consultas}(\text{Razavian}))$$

Experimento #3:

Para o diagrama de Sequência:

- Hipótese Nula ($H0_{efet}$): não há diferença significativa de efetividade entre as abordagens SMarty e Ziadi et al na geração de produtos específicos de LPS a partir de diagramas de sequencia.

$$H0_{efet}: \mu(\text{Efetividade}(\text{SMarty})) = \mu(\text{Efetividade}(\text{Ziadi et al}))$$

- Hipótese Alternativa ($H1_{efet}$): existe diferença significativa de efetividade entre as abordagens SMarty e Ziadi et al na geração de produtos específicos de LPS a partir de diagramas de sequencia.

$$H1_{efet}: \mu(\text{Efetividade}(\text{SMarty})) \neq \mu(\text{Efetividade}(\text{Ziadi et al}))$$

Para o diagrama de classes:

- Hipótese Nula ($H0_{efet}$): não há diferença significativa de efetividade entre as abordagens SMarty e PLUS na geração de produtos específicos de LPS a partir de diagramas de classes.

$$H0_{efet}: \mu(\text{Efetividade}(\text{SMarty})) = \mu(\text{Efetividade}(\text{Ziadi et al}))$$

- Hipótese Alternativa ($H1_{efet}$): existe diferença significativa de efetividade entre as abordagens SMarty e Ziadi et al na geração de produtos específicos de LPS a partir de diagramas de classes.

$$H1_{efet}: \mu(\text{Efetividade}(\text{SMarty})) \neq \mu(\text{Efetividade}(\text{Ziadi et al}))$$

Para a efetividade do rastreamento de elementos entre diagramas com cada abordagem:

- Hipótese Nula ($H0_{efet}$): não há diferença significativa de efetividade entre a rastreabilidade de elementos entre os diagramas de sequencia e classe usando as abordagens SMarty e Ziadi et al.

$$H0_{efet}: \mu(\text{Efetividade}(\text{SMarty})) = \mu(\text{Efetividade}(\text{Ziadi et al}))$$

- Hipótese Alternativa ($H1_{efet}$): existe diferença significativa de efetividade entre a rastreabilidade de elementos entre os diagramas de sequencia e classes usando as abordagens SMarty e Ziadi et al.

$$H1_{efet}: \mu(\text{Efetividade}(\text{SMarty})) \neq \mu(\text{Efetividade}(\text{Ziadi et al}))$$

Para calcular a efetividade consideramos a seguinte equação:

$$\text{Efetividade (z)} = \text{nVarC} / \text{Total}$$

Onde:

z = a abordagem de gerenciamento de variabilidades;

nVarC = o número de elementos com variabilidade corretamente resolvidas; e

Total = o número total de elementos do diagrama.

3.2.3 Definição das Variáveis (D.3)

Experimento #1:

As variáveis deste experimento são definidas como segue:

- Variáveis Independentes: há duas variáveis de entrada para o experimento. A primeira, abordagem de gerenciamento de variabilidades, que é um fator com dois tratamentos: a abordagem SMarty e o método PLUS. A segunda variável é pré-fixada, sendo a LPS MM.
- Variáveis Dependentes: as variáveis de saída do experimento são a efetividade, o número de consultas ao material de cada abordagem, a influência do conhecimento dos participantes no valor observado de efetividade e a efetividade de cada abordagem no rastreamento de elementos entre os diagramas.

Experimento #2:

- Variáveis Independentes: há duas variáveis de entrada para o experimento. A primeira, abordagem de gerenciamento de variabilidades, que é um fator com três tratamentos: a abordagem SMarty, o método PLUS e a abordagem Razavian e Khosravi. A segunda variável é pré-fixada, sendo a LPS MM.
- Variáveis Dependentes: as variáveis de saída do experimento são a efetividade, o número de consultas ao material de cada abordagem e a influência do conhecimento dos participantes no valor observado de efetividade.

Experimento #3:

As variáveis deste experimento foram definidas como segue:

- Variáveis Independentes: há duas variáveis de entrada para o experimento. A primeira, abordagem de gerenciamento de variabilidades, que é um fator com dois tratamentos: a abordagem SMarty e o método Ziadi et al. A segunda variável é pré-fixada, sendo a LPS AGM.
- Variáveis Dependentes: as variáveis de saída do experimento são a efetividade, o número de consultas ao material de cada abordagem, a influência do conhecimento dos participantes no valor observado de efetividade e a efetividade de cada abordagem no rastreamento de elementos entre os diagramas.

3.2.4 Tamanho da Amostra (D.4)**Experimento #1:**

Trinta e seis estudantes participaram do estudo, sendo eles da graduação do curso de Ciência da Computação.

Experimento #2:

Cinquenta e um estudantes participaram do estudo, sendo eles graduandos e pós graduandos do curso de Ciência da Computação e graduandos do curso de Engenharia da Computação.

Experimento #3:

Trinta pessoas participaram da avaliação experimental os participantes eram graduandos, mestrandos ou doutorandos em Ciência da Computação e alunos de graduandos do curso de Engenharia da Computação.

3.2.5 Definição e Seleção dos Participantes (D.5)

Experimento #1:

Os participantes eram da graduação do curso de Ciência da Computação da Universidade Estadual de Maringá e Universidade Tecnológica Federal do Paraná - campus Campo Mourão, com conhecimento variando entre básico e avançado em UML e entre nenhum e Moderado em LPS. Esses dados foram obtidos por meio de um questionário de caracterização (Apêndice B) e estão representados na Tabela 3.1.

Tabela 3.1: Descrição do nível de conhecimento dos participantes do Experimento #1

Partic.	Conhecimento		Partic.	Conhecimento	
	UML	LPS		UML	LPS
1	Básico	Básico	24	Avançado	Leu sobre
2	Moderado	Nenhum	25	Básico	Nenhum
3	Básico	Nenhum	26	Moderado	Nenhum
4	Básico	Nenhum	27	Moderado	Leu sobre
5	Moderado	Leu sobre	28	Avançado	Leu sobre
6	Básico	Nenhum	29	Moderado	Nenhum
7	Básico	Nenhum	30	Moderado	Nenhum
8	Básico	Nenhum	31	Básico	Nenhum
9	Básico	Leu sobre	32	Moderado	Nenhum
10	Moderado	Nenhum	33	Avançado	Nenhum
11	Básico	Nenhum	34	Avançado	Leu sobre
12	Básico	Nenhum	35	Moderado	Nenhum
13	Básico	Nenhum	36	Moderado	Nenhum
14	Básico	Nenhum	37	Moderado	Nenhum
15	Básico	Nenhum	38	Moderado	Nenhum
16	Moderado	Nenhum	39	Avançado	Leu sobre
17	Avançado	Leu sobre	40	Básico	Leu sobre
18	Básico	Leu sobre	41	Básico	Leu sobre
19	Básico	Nenhum	42	Básico	Nenhum
20	Básico	Leu sobre	43	Moderado	Leu sobre
21	Básico	Nenhum	44	Básico	Nenhum
22	Moderado	Nenhum	45	Moderado	Nenhum
23	Moderado	Nenhum	46	Moderado	Nenhum

Experimento #2:

Os participantes do experimento eram alunos graduandos e pós graduandos do curso de Ciência da Computação e graduandos do curso de Engenharia da Computação da Universidade Estadual de Maringá, Uningá, Universidade Tecnológica Federal do Paraná

- Campus Cornélio Procópio e Pontifícia Universidade Católica - Curitiba. Com conhecimento variado em UML e LPS. Os níveis de conhecimento dos participantes foram obtidos a partir da aplicação de uma Questionário de Caracterização (Apêndice A) e estão na Tabela 3.2.

Tabela 3.2: Descrição do nível de conhecimento dos participantes do Experimento #2

Partic.	Conhecimento		Partic.	Conhecimento	
	UML	LPS		UML	LPS
1	Básico	Leu sobre	27	Moderado	Nenhum
2	Avançado	Moderado	28	Básico	Leu sobre
3	Moderado	Leu sobre	29	Básico	Básico
4	Básico	Básico	30	Básico	Nenhum
5	Moderado	Leu sobre	31	Básico	Leu sobre
6	Moderado	Leu sobre	32	Moderado	Básico
7	Básico	Leu sobre	33	Moderado	Básico
8	Moderado	Leu sobre	34	Moderado	Leu sobre
9	Básico	Leu sobre	35	Moderado	Leu sobre
10	Moderado	Leu sobre	36	Básico	Leu sobre
11	Básico	Nenhum	37	Básico	Leu sobre
12	Básico	Nenhum	38	Básico	Leu sobre
13	Básico	Leu sobre	39	Moderado	Leu sobre
14	Básico	Nenhum	40	Moderado	Nenhum
15	Moderado	Leu sobre	41	Nenhum	Leu sobre
16	Básico	Leu sobre	42	Básico	Básico
17	Nunca	Leu sobre	43	Básico	Nenhum
18	Avançado	Moderado	44	Básico	Nenhum
19	Nunca	Leu sobre	45	Moderado	Nenhum
20	Nenhum	Nenhum	46	Moderado	Nenhum
21	Básico	Leu sobre	47	Básico	Nenhum
22	Básico	Nenhum	48	Moderado	Básico
23	Básico	Leu sobre	49	Básico	Leu sobre
24	Moderado	Nenhum	50	Básico	Nenhum
25	Básico	Nenhum	51	Moderado	Nenhum
26	Básico	Leu sobre	-	-	-

Experimento #3:

Os participantes eram graduandos, mestrandos ou doutorandos em Ciência da Computação e alunos de graduados do curso de Engenharia da Computação da Universidade Estadual de Maringá e Unicesumar, com diferentes níveis de conhecimento em UML e LPS. Os dados foram obtidos por meio de uma Questionário de Caracterização (Apêndice A) e estão resumidos na Tabela 3.3.

Tabela 3.3: Descrição do nível de conhecimento dos participantes do Experimento #3

Partic.	Conhecimento		Partic.	Conhecimento	
	UML	LPS		UML	LPS
1	Básico	Básico	16	Básico	Nenhum
2	Moderado	Leu sobre	17	Moderado	Leu sobre
3	Moderado	Básico	18	Moderado	Leu sobre
4	Moderado	Básico	19	Moderado	Nenhum
5	Básico	Leu sobre	20	Moderado	Leu sobre
6	Básico	Nenhum	21	Básico	Leu sobre
7	Básico	Nenhum	22	Básico	Nenhum
8	Moderado	Leu sobre	23	Básico	Nenhum
9	Básico	Leu sobre	24	Moderado	Nenhum
10	Moderado	Leu sobre	25	Moderado	Nenhum
11	Básico	Nenhum	26	Básico	Nenhum
12	Básico	Leu sobre	27	Moderado	Nenhum
13	Básico	Leu sobre	28	Básico	Nenhum
14	Básico	Leu sobre	29	Básico	Nenhum
15	Moderado	Nenhum	30	Básico	Nenhum

3.2.6 Identificação do Tópico de Pesquisa do Experimento (D.6)

Experimento #1, Experimento #2 e Experimento #3

O estudo experimental foram conduzidos no tópico de pesquisa em Engenharia de Software e LPS.

3.2.7 Descrição do *Design* Experimental (D.7)

Experimento #1:

A relação entre as variáveis é mostrada na Figura 3.1.

Experimento #2:

Na Figura 3.2 é ilustrada a relação entre as variáveis:

Experimento #3:

A relação entre as variáveis está ilustrada na Figura 3.3:

3.2.8 Definição e Seleção dos Materiais Experimentais (D.8)

No treinamento, os participantes dos três experimentos receberam um conjunto com três documentos, sendo eles:

Figura 3.1: Variáveis dependentes e independentes do experimento #1.

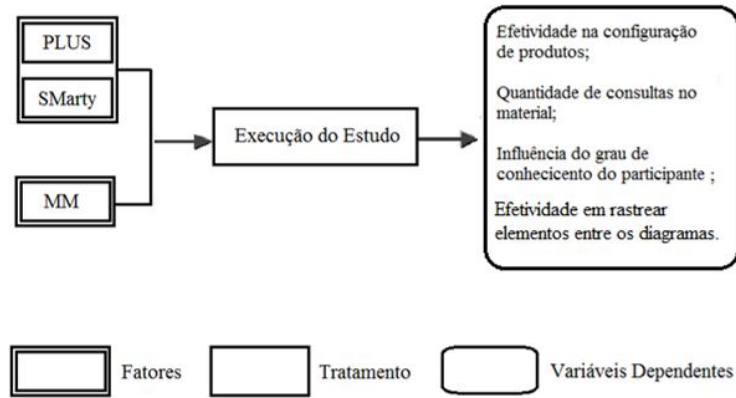
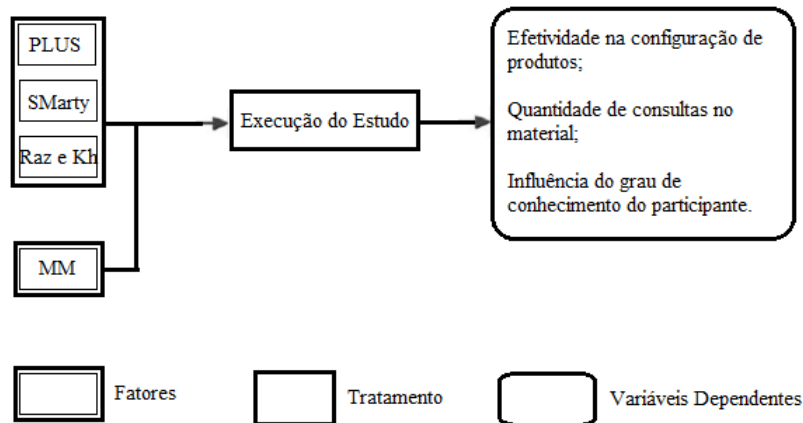
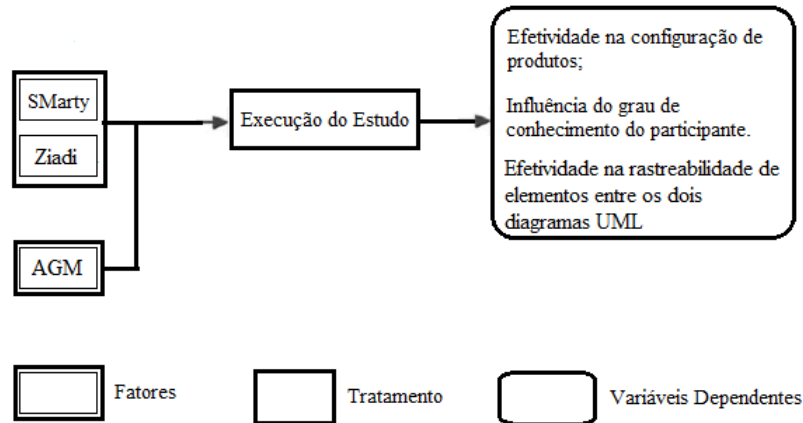


Figura 3.2: Variáveis dependentes e independentes do experimento #2.



- Termo de Consentimento Livre e Esclarecido (TCLE): contendo as principais informações sobre o experimento a ser aplicado, como por exemplo: confidencialidade, procedimentos e benefícios. Tal documento permitiu que o participante tomasse sua decisão sobre a sua participação na pesquisa de forma justa;
- Questionário de caracterização: aplicado aos participantes para analisar o nível de conhecimento e experiência sobre UML, LPS e variabilidade em LPS;
- Documento com a síntese teórica: conteúdos vistos durante o treinamento sobre cada abordagem. Para facilitar o participante a encontrar as informações, o documento foi dividido em três seções. A primeira com os principais conceitos de Linha de Produto de Software e a segunda com a descrição geral da LPS MM. Como os

Figura 3.3: Variáveis dependentes e independentes do experimento #3.



participantes foram divididos em dois blocos (um bloco para cada abordagem), a terceira seção, que compreendia as informações sobre as abordagens, foi diferente para cada grupo.

Os documentos que se diferenciaram entre os experimentos estão expostos a seguir:

Experimento #1:

- Bloco com a abordagem PLUS: compreendia um resumo sobre os conceitos da abordagem PLUS, bem como seus estereótipos e exemplos. A abordagem PLUS na documentação foi representada por X;
- Bloco com a abordagem SMarty: conceitos sobre a abordagem SMarty, seus estereótipos e exemplos. Essa abordagem foi identificada como Y.

Experimento #2:

- Bloco com a abordagem PLUS: compreendia um resumo sobre os conceitos da abordagem PLUS, bem como seus estereótipos e exemplos. A abordagem PLUS na documentação foi representada por X;
- Bloco com a abordagem SMarty: conceitos sobre a abordagem SMarty, seus estereótipos e exemplos. Essa abordagem foi identificada como Y.
- Bloco com a abordagem Razavian e Khosravi: sobre a abordagem Razavian, seus estereótipos e exemplos. Essa abordagem foi identificada como Z.

Experimento #3:

- Bloco com a abordagem SMarty: conceitos sobre a abordagem SMarty, seus estereótipos e exemplos. Essa abordagem foi identificada como Y.
- Bloco com a abordagem Ziadi et al: conceitos sobre a abordagem Ziadi et al seus estereótipos e exemplos. Essa abordagem foi identificada como X.

Alguns participantes realizaram o experimento de forma online. Um link foi disponibilizado com os documentos realizados anteriormente, além de vídeos de treinamento sobre LPS e a abordagem de gerenciamento de variabilidades que seria utilizada.

Todos os documentos aqui citados estão relacionados no Apêndice B.

3.2.8.1 Descrição da LPS Utilizada (D.8.1)

As LPS foram escolhidas de acordo com a disponibilização da sua descrição e os diagramas existentes de cada LPS.

Experimento #1 e Experimento #2:

Para os experimentos 1 e 2, a LPS MM foi utilizada. MM (Young, 2005), é uma LPS composta de aplicações (produtos) que manipulam músicas, vídeos e fotos para dispositivos móveis, como celulares e palm tops. Ela provê suporte para gerenciar (criar, excluir, visualizar, executar, enviar) diferentes tipos de mídia. A MM surgiu da extensão de uma LPS já existente denominada Mobile Photo (Young, 2005), por meio da inserção de novas propriedades multimídia, como manipulação de vídeos e músicas, que somente podem ser realizados em alguns tipos de aparelhos. De certa forma, pode-se dizer que a inserção das características opcionais e alternativas a determinados aparelhos caracterizou o surgimento da Mobile Media.

Experimento #3:

A AGM é uma LPS pedagógica criada para auxiliar no aprendizado e experimentação de LPS. Ela segue a abordagem de LPS descrita por Clements e Northrop (2002). A LPS engloba três jogos de arcade (Brickles, Bowling e Pong).

Fonte da LPS (D.8.1.1)**Experimento #1 e Experimento #2:**

(Young, 2005)

Experimento #3:

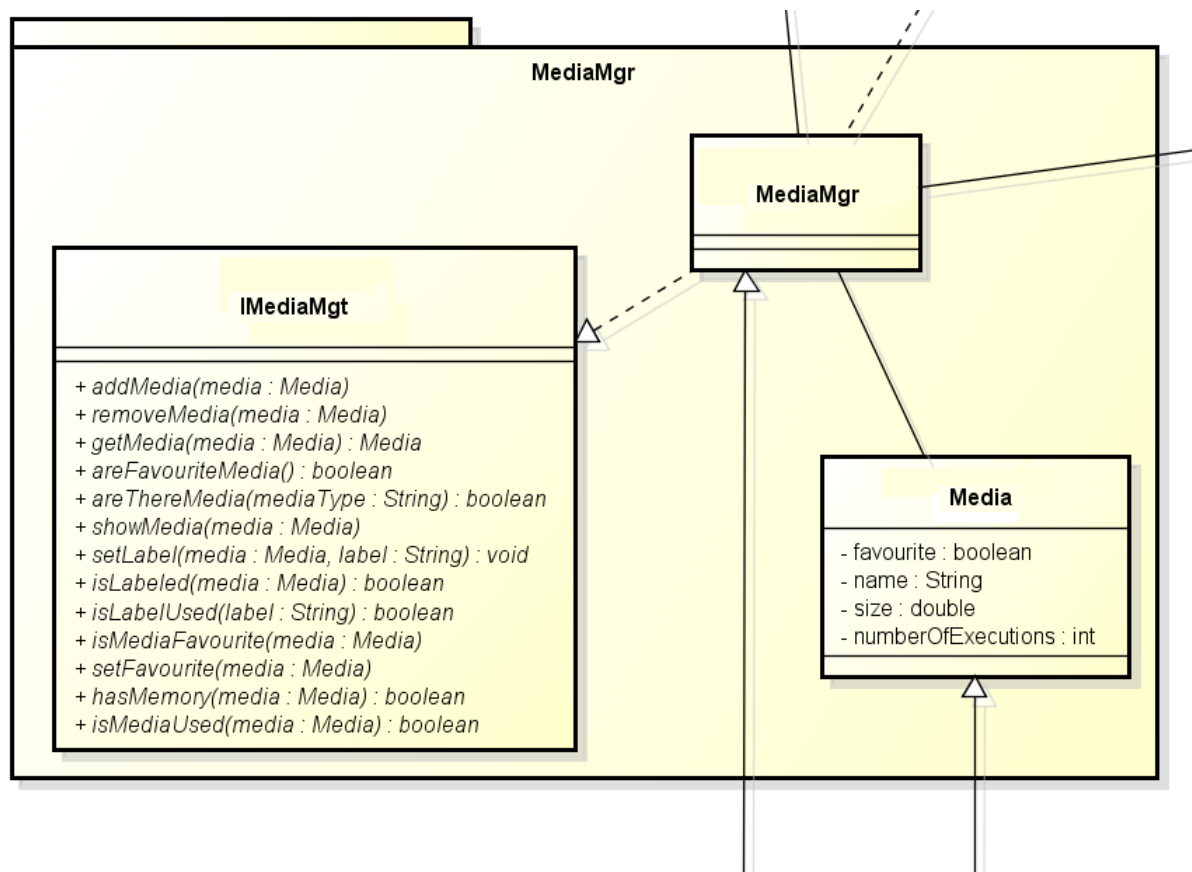
https://resources.sei.cmu.edu/asset_files/WhitePaper/2009_019_001_485943.pdf

3.2.8.2 Artefatos da LPS (D.8.2)

Experimento #1 e Experimento #2:

Na Figura 3.4 é mostrado parte de um diagrama de classes da LPS MM.

Figura 3.4: Artefatos MM - Parte do Diagrama de Classes



Fonte: (Young, 2005)

Experimento #3:

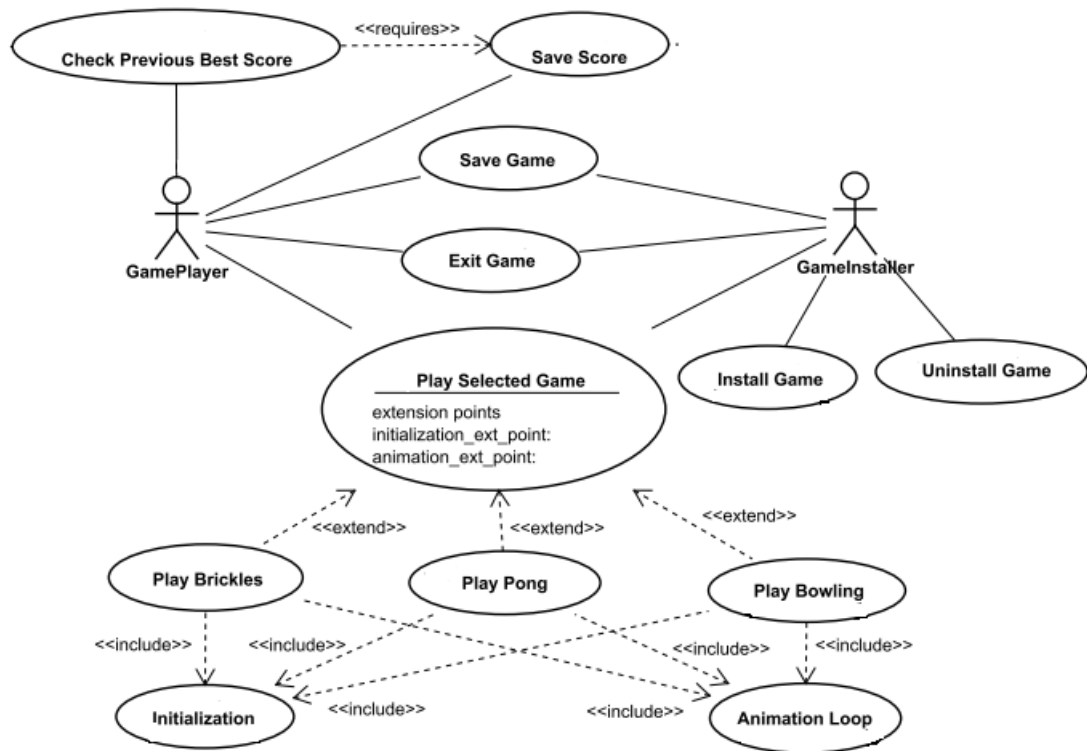
Nas Figura 3.5 e Figura 3.6, são ilustrados dois diagramas da LPS AGM, um de casos de uso e outro de classes.

3.2.9 Validação dos Materiais Experimentais (D.9)

Experimento #1, Experimento #2 e Experimento #3:

Os materiais utilizados nos três experimentos passaram por validação por professores e mestrandos da área de Engenharia de Software. Um reunião foi marcada para a apresentação do material experimental para os mesmos. Sugestões puderam ser ouvidas,

Figura 3.5: Artefatos AGM - Diagrama de Casos de Uso



Fonte: https://resources.sei.cmu.edu/asset_files/WhitePaper/2009_019_014_85943.pdf

como maneira de conduzir o experimento e alguns pontos dos documentos passaram por melhoria depois de tal validação, como estrutura textual.

3.2.10 Descrição das Tarefas do Experimento (D.10)

Experimento #1:

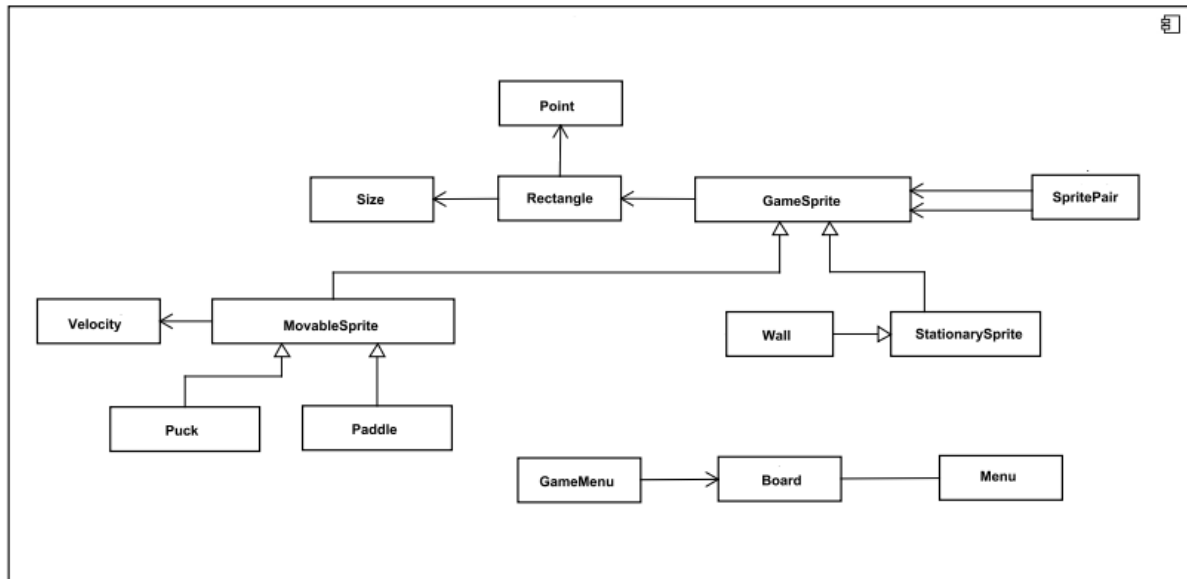
No primeiro experimento, os participantes receberam a tarefa de configurar dois produtos a partir da LPS MM, um com diagramas de casos de uso e outro com diagramas de classe. Além disso, tanto os participantes que utilizaram a abordagem SMarty, quanto os que utilizaram o método PLUS tiveram que rastrear elementos entre os diagramas.

Experimento #2:

Os participantes que receberam a abordagem SMarty configuraram dois produtos, um a partir de diagramas de classes e um a partir de diagramas de componentes. Além disso, receberam a tarefa de rastrear elementos entre os modelos.

Os participantes que utilizaram a abordagem PLUS e Razavian e Khosravi, se limitaram a rastrear apenas um produto, com diagramas de classes e componentes respectivamente.

Figura 3.6: Artefatos AGM - Diagrama de Classes



Fonte: https://resources.sei.cmu.edu/asset_files/WhitePaper/2009_019_01485943.pdf

Experimento #3:

Os participantes do experimento, configuraram dois produtos, um a partir do diagrama de classes e um a partir do diagrama de sequência da LPS AGM. Os participantes que utilizaram a abordagem SMarty ainda tiveram que analisar a restreabilidade de elementos entre os modelos UML, por ser a única abordagem dentre as comparadas que apoia esta tarefa.

3.2.11 Descrição dos Requisitos de Treinamento (D.11)

Experimento #1 e Experimento #2:

Como os experimentos 1 e 2 foram realizados inteiramente de forma presencial, os participantes receberam o treinamento de LPS e da abordagem que utilizariam no mesmo dia do estudo experimental.

Experimento #3:

Para parte dos alunos que fizeram o experimento presencialmente, o treinamento de LPS e sobre a abordagem a ser utilizada também aconteceu de maneira presencial. Para os participantes que realizaram o experimento de maneira online, foram disponibilizados vídeos com os mesmos conteúdos de treinamento.

3.2.12 Condução do Projeto Piloto (D.12)

Experimento #1, Experimento #2 e Experimento #3:

Um projeto piloto foi aplicado para um conjunto de mestrandos da área de engenharia de software da Universidade Estadual de Maringá com conhecimento em gerenciamento de variabilidades em LPS. A partir da aplicação deste projeto, eventuais erros no material puderam ser corrigidos e algumas alterações para melhorá-lo puderam ser feitas. O grupo que participou do projeto piloto foi diferente do que participou da validação dos materiais descrito na diretriz D.9.

As modificações apontadas foram:

- alteração na síntese teórica sobre LPS;
- alteração nos slides de treinamento;
- melhoria dos exemplos no material de apoio.

3.2.13 Ambiente da Condução do Experimento (D.13)

Experimento #1:

O experimento 1 foi executado em dois ambientes, salas de aula e laboratórios da Universidade Estadual de Maringá e Universidade Tecnológica Federal do Paraná - Campus Campo Mourão.

Experimento #2:

O experimento foi realizado em quatro dias, em salas de aulas das Pontifícia Universidade Católica de Curitiba, Universidade Tecnológica Federal do Paraná - Campus Cornélio Procopio, Uningá e Universidade Estadual de Maringá.

Experimento #3:

O experimento foi realiza de maneira online e de forma presencial na residência da condutora.

3.2.14 Data de Execução dos Experimentos (D.14)

Experimento #1:

O estudo foi conduzido entre os meses Abril e Maio de 2019

Experimento #2:

Os experimentos foram realizados em Outubro e Novembro de 2019.

Experimento #3:

A data de execução do experimento foi entre Junho e Setembro de 2020.

3.2.15 Condução dos Experimentos (D.15)

Experimento #1 e Experimento #2:

Os experimentos 1 e 2 seguiram uma mesma dinâmica, já que foram realizados inteiramente de forma presencial. Em cada dia em que foi conduzido o estudo, um treinamento sobre LPS foi dado a turma como um todo.

Em seguida, a turma era dividida, para que cada parte recebesse o treinamento apenas da abordagem que utilizaria.

Os alunos, recebiam a documentação de todo o treinamento e por fim, as atividades já descritas poderia ser executada.

Experimento #3:

Para os alunos do experimento 3 que realizaram o estudo de forma presencial, o processo foi o mesmo do descrito para o experimento 1 e 2.

Para os participantes que realizaram de forma remota, foi disponibilizado um link no *Google Drive* contendo toda documentação e treinamento necessário para a execução do experimento.

3.3 Considerações Finais

Neste capítulo foi apresentada as circunstâncias que ocorreram cada experimento, com a explicação e ilustração das variáveis envolvidas para melhor entendimento do próximo capítulo, que consiste na apresentação da execução e análise dos experimentos aqui planejados.

Execução, Análise e Interpretação dos Dados

4.1 Considerações Iniciais

Neste capítulo é apresentada a execução e análise dos três experimentos conduzidos. A partir das questões de pesquisa definidas, os testes estatísticos foram realizados, com os dados obtidos por meio dos experimentos, para determinar a efetividade das abordagens, a correlação do nível de conhecimento do participante com a efetividade de produto configurado e a efetividade em rastrear elementos entre os diagramas.

Além disso, outras questões foram esclarecidas, como taxa de mortalidade dos dados, ferramentas utilizadas na análise, fonte do pacote experimental e template experimental utilizado.

4.2 Execução dos experimentos

Os itens a seguir apresentam os resultados obtidos com a execução dos experimentos a partir do protocolo definido no capítulo anterior. Para isso, seguimos as diretrizes e recomendações para documentação de experimentos de LPS de Furtado (2018). Cada diretriz possui a seguinte nomenclatura "D.x", sendo x o número da diretriz.

4.2.1 Descrição de como a coleta de dados foi realizada: (D.16)

Experimento #1:

Após responder o questionário de caracterização, o grupo de alunos foi dividido de acordo com a abordagem (SMarty ou PLUS) recebida por cada participante. Essa divisão foi feita de maneira que em cada grupo o nível de conhecimento dos participantes fosse semelhante para não haver influência nos resultados obtidos. Os participantes com PLUS ficaram na sala para receber o treinamento, enquanto os demais aguardavam do lado de fora. Ao término do treinamento com o método PLUS, os participantes eram encaminhados para outra sala, supervisionados por uma pessoa e já iniciavam a configuração dos produtos e o rastreo de elementos entre os diagramas. Em sequência os participantes da abordagem SMarty eram treinados e também iniciavam a configuração dos produtos e o rastreo de elementos entre os diagramas.

Cada grupo recebeu um formulário com a identificação da atividade a ser realizada. Metade do grupo recebeu o formulário para especificar duas configurações de produtos usando o diagrama de caso de uso da LPS MM com a abordagem SMarty, enquanto a outra metade recebeu o formulário para especificar dois produtos a partir do diagrama de casos de uso e classes da MM modelado com a abordagem PLUS. Ao mesmo tempo em que os participantes iam gerando as respectivas configurações, eles apontavam o número de consultas ao material instrucional.

Experimento #2:

O grupo de alunos foi dividido de acordo com a abordagem recebida (SMarty, PLUS ou Razavian e Khosravi) por cada participante, essa divisão foi feita de maneira que em cada grupo o nível de conhecimento dos participantes fosse semelhante para não haver influência nos resultados obtidos. Essa informação foi obtida com a aplicação de um questionário de caracterização. Os participantes que receberam o método PLUS ficaram na sala para receber o treinamento, enquanto os demais aguardavam do lado de fora. Ao término do treinamento com o método PLUS, os participantes eram encaminhados para outra sala, supervisionados por uma pessoa e já iniciavam a configuração dos produtos. Em sequência os participantes da abordagem SMarty eram treinados e também iniciavam a configuração dos produtos. E por fim os alunos da abordagem Razavian recebiam o treinamento e executavam a configuração na sequência.

Cada grupo recebeu um formulário com a identificação da atividade a ser realizada. Um terço do grupo recebeu o formulário para especificar duas configurações de produtos usando o diagrama de classes e outro de componentes da LPS MM com a abordagem SMarty, enquanto outra parte recebeu o formulário para especificar um produto a partir

do diagrama de classes com o método PLUS e a terceira parte a partir do diagrama de componentes com a abordagem Razavian e Khosravi. Os participantes configuraram os produtos e os que utilizaram a abordagem SMarty também analisaram a rastreabilidade entre modelos usando o formulário de respostas.

Experimento #3:

Da mesma maneira, os participantes foram divididos de acordo com a abordagem (SMarty ou Ziadi et al) recebida por cada um. A execução do experimento aconteceu de forma individual.

Cada pessoa recebeu um formulário com a identificação da atividade a ser realizada. Metade do grupo recebeu o formulário para especificar duas configurações de produtos usando o diagrama de classes e outro de sequência da LPS AGM e restrear elementos entre os modelos com a abordagem SMarty, enquanto outra parte recebeu o formulário para especificar um produtos a partir do diagrama de classes e de sequência com a abordagem Ziadi et al.

4.3 Procedimentos de análise dos dados coletados: (D.17)

4.3.1 Experimento #1

As seguintes questões de pesquisa foram definidas para este estudo:

Q.P.1: Qual abordagem é mais efetiva para derivar configurações de produtos específicos a partir de diagramas de casos de uso e classes?

Q.P.2: Qual abordagem requer menos consultas no respectivo material instrucional para obter melhor conhecimento da LPS e realizar a configuração de produtos específicos?

Q.P.3: Qual a influência do grau de conhecimento do participante na aplicação da respectiva abordagem para derivar produtos específicos a partir de diagramas de casos de uso e de classes?

Q.P.4: Qual abordagem é mais efetiva na rastreabilidade de elementos entre os dois diagramas UML?

Efetividade das Abordagens

Teste de Normalidade

Para o teste de normalidade das amostras de efetividade foi aplicado o teste de Shapiro-Wilk, tanto para as amostras da efetividade com diagramas de casos de uso

como de classes das duas abordagens. As amostras obtiveram $p > 0,05$, portanto não sendo normalmente distribuídas.

Assim, o teste não paramétrico de Mann-Whitney-Wilcoxon foi conduzido para as amostras para indicar qual abordagem obteve melhor efetividade na configuração de produtos.

Teste de Mann-Whitney-Wilcoxon

Diagrama de Casos de Uso

O valor calculado para p foi 0,1898 e em comparação $p = 0,1898 > \alpha = 0,05$ confirmando que a hipótese nula ($H_{0_{efet}}$) não pode ser refutada. Assim, não há diferença estatística significativa entre as amostras de efetividade na configuração de produtos a partir de diagramas de casos de uso.

Diagrama de Classes

O valor identificado no teste foi de $p = 0,02797$ e em comparação $p = 0,02797 < \alpha = 0,05$ confirmando que não existem indícios de que a hipótese nula ($H_{0_{efet}}$) deve ser aceita e assim, método PLUS fornece melhor apoio na configuração de produtos efetivos a partir de diagramas de classes.

Número de Consultas ao Material

O número de consultas aos materiais instrucionais é um valor discreto anotado pelo participante ao longo do experimento. A influência do conhecimento de cada participante foi calculada usando correlação entre os cinco níveis de conhecimento do Questionário de Caracterização e o valor obtido de efetividade para cada participante.

A quantidade de consultas no material disponibilizado sobre a LPS MM, o método PLUS e a abordagem SMarty foi contabilizada com o objetivo de se entender quais das duas abordagens oferece um entendimento mais claro e facilitado para interpretação da modelagem de variabilidade em casos de uso. Assim, quanto menor o número de consultas, mais se entende que a abordagem oferece maior compreensibilidade.

Teste de Normalidade

O teste de Shapiro-Wilk foi aplicado para as amostras do número de consultas aos materiais.

Consultas do método PLUS ($N = 23$)

O teste de normalidade indicou, para uma amostra de tamanho de 23 com 95% de significância ($\alpha = 0,05$), $p = 0,00000003296 < 0,05$. A amostra é, portanto, não normalmente distribuída.

Consultas da Abordagem SMarty ($N = 18$)

O teste de normalidade indicou, assim como para a amostra de PLUS, para uma amostra de tamanho de 18 com 95% de significância ($\alpha = 0,05$), $p = 0,00000004344 < 0,05$. A amostra é, portanto, não normalmente distribuída.

Como ambos os testes resultaram em amostras não normalmente distribuídas, o teste não paramétrico de Mann-Whitney-Wilcoxon foi conduzido para as amostras para indicar se há diferença significativa no número de consultas aos materiais.

Teste de Mann-Whitney-Wilcoxon

Os valores analisados por meio do teste não apresentaram uma diferença estatística. O valor de p , para ser comparado com um nível de significância de 95% ($\alpha = 0,05$), foi 0,6349.

Assim, $p = 0,6349 > \alpha = 0,05$, a hipótese nula ($H_{0_{cons}}$) não pôde ser refutada, fornecendo indícios de que não pôde se estabelecer uma relação de compreensibilidade com base no número de consultas aos materiais de cada abordagem.

Correlação entre Efetividade e Nível de Conhecimento dos Participantes

Para estabelecer se existe uma correlação entre a efetividade obtida por cada participante e o seu nível de conhecimento em UML foi utilizada a correlação de Spearman, uma vez que houve a conversão de escalas nominais ordinais para valores discretos.

Os seguintes valores foram calculados separadamente para cada diagrama aplicando a escala de Spearman (Figura 4.1):

Diagrama de Casos de Uso:

- Abordagem PLUS: $p = 0,32$ correlação positiva fraca; e
- Abordagem SMarty: $p = 0,16$ correlação positiva fraca.

Diagrama de Classes:

- Abordagem PLUS: $p = 0,24$ correlação positiva fraca; e
- Abordagem SMarty: $p = 0,41$ correlação positiva fraca.

Deseja-se que o valor da correlação seja o menor possível para que a facilidade de compreensibilidade da abordagem seja maior. Em relação ao diagrama de casos de uso, no caso do método PLUS, o nível de conhecimento do participante tem maior influência do que o nível de conhecimento de SMarty. Quanto ao diagrama de casos de classes, para a abordagem SMarty, o nível de conhecimento do participante tem maior influência do que o nível de conhecimento de PLUS.

Figura 4.1: Escala de Correlação de Spearman



Fonte: (Spearman, 1987)

Efetividade em Rastrear Elementos entre os Diagramas

Teste de Normalidade

Para o teste de normalidade das amostras dos acertos de elementos rastreados foi aplicado o teste de Shapiro-Wilk, tanto para as amostras rastreando elementos do diagrama de casos de uso para o de classes e o de classes para casos de uso. Em todos os casos as amostras obtiveram $p < (\alpha = 0,05)$, portanto não sendo normalmente distribuídas.

Assim, o teste não paramétrico de Mann-Whitney-Wilcoxon foi conduzido para as amostras para indicar qual abordagem obteve melhor efetividade na rastreabilidade de elementos entre os diagramas.

Teste de Mann-Whitney-Wilcoxon

Casos de Uso → *Classes*

O valor calculado para p foi 0,6442 e em comparação $p = 0,6442 > \alpha = 0,05$ confirmando que a hipótese nula ($H_{0_{efet}}$) não pode ser refutada. Assim, não há diferença estatística significativa entre a efetividade em rastrear elementos do diagrama de casos de uso para o de classes, usando a abordagem SMarty ou o método PLUS.

Classes → *Casos de uso*

O valor identificado no teste foi de $p = 0,176$ e em comparação $p = 0,176 > \alpha = 0,05$ confirmando que a hipótese nula ($H_{0_{efet}}$) não deve ser refutada. Isso significa que não há diferença estatística entre a abordagem SMarty o o método PLUS no rastreamento de elementos a partir do diagrama de classes para o diagrama de casos de uso.

4.3.2 Experimento #2:

As seguintes questões de pesquisa foram definidas para este estudo:

Q.P.1: Qual abordagem é mais efetiva para derivar configurações de produtos específicos a partir de diagramas de classes e componentes?

Q.P.2: Qual a influência do grau de conhecimento do participante na aplicação da respectiva abordagem para derivar produtos específicos a partir de diagramas de classes e componentes?

Q.P.3: Quanto a abordagem SMarty é efetiva na rastreabilidade de elementos entre os dois diagramas UML?

Efetividade das Abordagens

Teste de Normalidade

Para o teste de normalidade das amostras de efetividade foi aplicado o teste de Shapiro-Wilk, tanto para as amostras da efetividade com diagramas de casos de uso como de classes das duas abordagens. As amostras obtiveram $p < 0,05$, portanto não sendo normalmente distribuídas. Assim, o teste não paramétrico de Mann-Whitney-Wilcoxon foi conduzido para as amostras para indicar qual abordagem obteve melhor efetividade na configuração de produtos.

Teste de Mann-Whitney-Wilcoxon

Diagrama de Classes

O valor calculado para p foi 0,2539 e em comparação $p = 0,2539 > \alpha = 0,05$ confirmando que a hipótese nula ($H0_{efet}$) não pode ser refutada. Assim, não há diferença estatística significativa entre as amostras de efetividade na configuração de produtos a partir de diagramas de classes.

Diagrama de Componentes

O valor identificado no teste foi de $p = 0,3884$ e em comparação $p = 0,3884 > \alpha = 0,05$ confirmando que a hipótese nula ($H0_{efet}$) não pode ser refutada. Assim, não há diferença estatística significativa entre as amostras de efetividade na configuração de produtos a partir de diagramas de componentes.

Correlação entre Efetividade e Nível de Conhecimento dos Participantes

Para estabelecer se existe uma correlação entre a efetividade obtida por cada participante e o seu nível de conhecimento em UML foi utilizada a correlação de Spearman, uma vez que houve a conversão de escalas nominais ordinais para valores discretos.

Os seguintes valores foram calculados separadamente para cada diagrama aplicando a escala de Spearman (Figura 4.1):

Diagrama de Classes:

- Abordagem PLUS: $p = 0,32$ correlação positiva fraca; e

- Abordagem SMarty: $p = 0,41$ correlação positiva fraca;

Diagrama de Componentes:

- Abordagem SMarty: $p = 0,26$ correlação positiva fraca; e
- Abordagem Razavian e Khosravi: $p = 0,0$ sem correlação.

Deseja-se que o valor da correlação seja o menor possível para que a facilidade de compreensibilidade da abordagem seja maior. Em relação ao diagrama de classes, no caso da abordagem SMarty, o nível de conhecimento do participante tem maior influência do que o nível de conhecimento de PLUS. Quanto ao diagrama de componentes, para a abordagem SMarty, o nível de conhecimento do participante tem maior influência do que o nível de conhecimento de Razavian e Khosravi.

Efetividade em Rastrear Elementos entre os Diagramas

No começo do estudo, a intenção era analisar e comparar a capacidade e o apoio que cada abordagem estudada fornece ao rastrear elementos entre os diagramas UML. No entanto o método Ziadi et al não provê mecanismos para auxiliar no rastreamento de elementos entre os diagramas suportados pelo método. Por isso, nesta seção, apenas SMarty foi analisada nesse sentido.

Tabela 4.1: Respostas sobre Rastreabilidade de SMarty

Questão 1: Classes para Componentes		
Escala Likert	Qtde	(%)
Concorda totalmente	10	58,8
Concorda parcialmente	5	29,4
Discorda parcialmente	2	11,8
Discorda totalmente	0	0,0
Total	17	100
Questão 2: Componentes para Classes		
Escala Likert	Qtde	(%)
Concorda totalmente	13	76,5
Concorda parcialmente	4	23,5
Discorda parcialmente	0	0,0
Discorda totalmente	0	0,0
Total	17	100

Duas perguntas foram realizadas para os participantes que utilizaram a abordagem SMarty ambas com a escala Likert. As respostas são mostradas na Tabela 4.1.

- Questão 1: Supondo que em uma configuração de produto, a funcionalidade relacionada ao compartilhamento de mídia é excluída do diagrama de classes. As mudanças / impactos no diagrama de componentes podem ser identificados com o suporte da abordagem SMarty.
- Questão 2: Se o componente MediaMgr for excluído do diagrama de componentes em uma possível configuração do produto, é possível identificar os impactos que esta mudança teria no diagrama de classes com o auxílio da abordagem SMarty.

4.3.3 Experimento #3:

As seguintes questões de pesquisa foram definidas para este estudo:

Q.P.1: Qual abordagem é mais efetiva para derivar configurações de produtos específicos a partir de diagramas de classes e sequência?

Q.P.2: Qual a influência do grau de conhecimento dos participantes na aplicação da respectiva abordagem para derivar produtos específicos a partir de diagramas de classes e sequência?

Q.P.3: Quanto a abordagem SMarty é efetiva na rastreabilidade de elementos entre os dois diagramas UML?

Efetividade das Abordagens

Teste de Normalidade

Para o teste de normalidade das amostras de efetividade foi aplicado o teste de Shapiro-Wilk, tanto para as amostras da efetividade com diagramas de casos de uso como de classes das duas abordagens. As amostras obtiveram $p < 0,05$, portanto não sendo normalmente distribuídas.

Assim, o teste não paramétrico de Mann-Whitney-Wilcoxon foi conduzido para as amostras para indicar qual abordagem obteve melhor efetividade na configuração de produtos.

Teste de Mann-Whitney-Wilcoxon

Diagrama de Classes

O valor calculado para p foi 0,0299 e em comparação $p = 0,0299 < \alpha = 0,05$ confirmando que a hipótese nula ($H_{0_{efet}}$) pode ser refutada. Ou seja, há diferença estatística significativa entre as amostras de efetividade na configuração de produtos a partir de diagramas de classes, fornecendo indícios de que a abordagem SMarty fornece melhor apoio na configuração de produtos efetivos a partir de diagramas de classes.

Diagrama de Sequência

O valor identificado no teste foi de $p = 0,001846$ e em comparação $p = 0,001846 < \alpha = 0,05$ confirmando que não existem indícios de que a hipótese nula ($H0_{efet}$) deve ser aceita e assim, a abordagem SMarty fornece melhor apoio na configuração de produtos efetivos a partir de diagramas de sequência.

Correlação entre Efetividade e Nível de Conhecimento dos Participantes

Para estabelecer se existe uma correlação entre a efetividade obtida por cada participante e o seu nível de conhecimento em UML foi utilizada a correlação de Spearman, uma vez que houve a conversão de escalas nominais ordinais para valores discretos.

Os seguintes valores foram calculados separadamente para cada diagrama aplicando a escala de Spearman (Figura 7):

Diagrama de Classes:

- Método Ziadi et al: $p = 0,77$ correlação positiva forte; e
- Abordagem SMarty: $p = 0,27$ correlação positiva fraca.

Diagrama de Sequência:

- Método Ziadi et al: $p = 0,66$ correlação positiva forte; e
- Abordagem SMarty: $p = 0,05$ correlação positiva fraca.

Deseja-se que o valor da correlação seja o menor possível para que a facilidade de compreensibilidade da abordagem seja maior. Tanto para diagrama de classes quanto o de sequência, no caso do método Ziadi et al, o nível de conhecimento do participante tem maior influência do que o nível de conhecimento de SMarty.

Efetividade em Rastrear Elementos entre os Diagramas

No começo do estudo, a intenção era analisar e comparar a capacidade e o apoio que cada abordagem estudada fornece ao rastrear elementos entre os diagramas UML. No entanto o método Ziadi et al não provê mecanismos para auxiliar no rastreamento de elementos entre os diagramas suportados pelo método. Por isso, nesta seção, apenas SMarty foi analisada nesse sentido.

Duas perguntas foram realizadas para os participantes que utilizaram a abordagem SMarty ambas com a escala Likert. As respostas foram mostradas na Tabela 4.2.

- Questão 1: Supondo que em uma configuração de produto, as funcionalidades relacionadas ao Game Sprite fossem excluídos do diagrama de classes, as alterações/impactos no diagrama de sequencia são possíveis de serem identificados com apoio da abordagem X (Considerando que o digrama de classes é mais abstrato que o diagrama de sequência).
- Questão 2: Caso a funcionalidade Play Game ilustrado no diagrama de sequencia não existisse em uma possível configuração de produto, é possível identificar os impactos que essa mudança causaria no diagrama de classes (Considerando que o digrama de classes é mais abstrato que o diagrama de sequência).

Tabela 4.2: Respostas sobre Rastreabilidade de SMarty

Questão 1: Classes para Sequência		
Escala Likert	Qtde	(%)
Concorda totalmente	8	53,33
Concorda parcialmente	5	33,33
Discorda parcialmente	2	13,33
Discorda totalmente	0	0,0
Total	15	100
Questão 2: Sequência para Classes		
Escala Likert	Qtde	(%)
Concorda totalmente	6	40,0
Concorda parcialmente	5	33,3
Discorda parcialmente	3	20,0
Discorda totalmente	1	6,7
Total	15	100

4.4 Taxa de Mortalidade (D.18)

Experimento #1

No experimento 1 foi necessária a exclusão dos resultados de 3 participantes. Isso ocorreu por dois motivos: o participante entregou o formulário de respostas em branco ou o número obtido de efetividade foi muito abaixo da média de efetividades.

4.5 Ferramenta estatística para análise dos dados (D.19)

Experimento #1, Experimento #2 e Experimento #3:

A Correlação de Spearman foi calculada na ferramenta PSPP, enquanto todas as outras estatísticas foram calculadas na ferramenta R.

4.6 Tamanho do Efeito (D.20)

Experimento #1

Diagrama de Casos de Uso

O teste de Cohen foi aplicado nas amostras de efetividade do produtos configurados com o método PLUS e a abordagem SMarty a partir de diagrama de casos de uso, para saber a significância e diferença das amostras. O valor obtido de d foi -0,4844 e indica que a diferença dos valores é pequena.

Diagrama de Classes

Para o diagrama de classes, também foi aplicado o teste Cohen e o valor obtido de d foi 0,5168, indicando uma diferença média entre as amostras de efetividade na configuração de produtos.

Experimento #2

Diagrama de Classes

O teste de Cohen foi aplicado nas amostras de efetividade do produtos configurados com o método PLUS e a abordagem SMarty a partir de diagrama de classes, para saber a significância e diferença das amostras. O valor obtido de d foi -0,310 e indica que a diferença dos valores é pequena.

Diagrama de Componentes

Para o diagrama de componentes, também foi aplicado o teste Cohen e o valor obtido de d foi 0,109, indicando uma diferença insignificante entre as amostras de efetividade na configuração de produtos.

Experimento #3

Diagrama de Classes

O teste de Cohen foi aplicado nas amostras de efetividade do produtos configurados com o método Ziadi et al e a abordagem SMarty a partir de diagrama de classes, para saber a significância e diferença das amostras. O valor obtido de d foi -1,05 e indica que a diferença dos valores é grande.

Diagrama de Sequência

Para o diagrama de sequência, também foi aplicado o teste Cohen e o valor obtido de d foi -1,44, indicando uma diferença grande entre as amostras de efetividade na configuração de produtos.

Visto todos os valores, a maior diferença constatada pelo teste de Cohen foi de -1.05 no Experimento 1 com diagramas de Classes, que indicou uma diferença grande entre as amostras de efetividade comparadas. Enquanto que o valor 0.109 foi o menor valor, indicando uma diferença insignificante nos valores de efetividade no Experimento 2 com diagramas de componentes.

4.7 Resultados Obtidos do Ponto de Vista de Pesquisadores e de Profissionais (D.21)

Do ponto de vista de pesquisadores entende-se que a evolução de SMarty trará benefícios em médio prazo, pois as suas melhorias estão sendo implementadas na ferramenta SMartyModeling (Silva *et al.*, 2020) e poderão ser constatadas nos próximos 12 meses.

Já do ponto de vista dos profissionais, a implementação da ferramenta com base na melhoria de SMarty, por este estudo, traz uma *feature* imprescindível e que raras ferramentas CASE dos mais variados propósitos fornecem: rastreabilidade (Anquetil *et al.*, 2010; Cavalcanti *et al.*, 2012; Vale *et al.*, 2017) dos artefatos produzidos durante o processo de desenvolvimento de software. Essa *feature* possui um importante diferencial de impacto nos projetos de software já que permite, por exemplo, identificar inconsistências entre elementos de diagramas e requisitos, além de permitir a possibilidade de entender as decisões de projeto *round-trip*.

4.8 Implicações dos Tratamentos Desenvolvidos (D.22)

O tratamento avaliado e melhorado foi a abordagem SMarty no que diz respeito à configuração de produtos e rastreabilidade. Para tanto, os usuários de SMarty podem agora contar com essas duas novas *features* da abordagem, o que a torna particularmente mais atrativa do que as demais comparadas.

Ainda, tais *features* estão sendo implementadas na ferramenta SMartyModeling (Silva *et al.*, 2020), que permite a modelagem de LPS baseadas em UML.

Ademais, se observa a contínua melhoria da abordagem SMarty, o que contribui diretamente para os usuários do ponto de vista de gerência de variabilidades em LPS baseadas em UML.

4.9 Ameaças à Validade Identificadas no Experimento (D.23)

As ameaças à validade dos três experimentos são discutidas na Seção 5.4.

4.10 Fonte do Pacote Experimental (D.24)

Experimento #1

<https://doi.org/10.5281/zenodo.3569423>

Experimento #2

<https://doi.org/10.5281/zenodo.4308863>

Experimento #3

<https://doi.org/10.5281/zenodo.4304279>

4.11 Template Experimental Utilizado para Conduzir, Planejar ou Documentar o Experimento (D.25)

Experimento #1, Experimento #2 e Experimento #3:

Os estudos experimentais foram baseados no template proposto por Wohlin (Wohlin *et al.*, 2012)

4.12 Considerações Finais

Neste capítulo, obtemos importantes resultados de testes estatísticos.

Apesar de usar uma LPS mais complexa comparada a utilizada por Nepomuceno e Oliveira Jr (2018), resolver as variabilidades e configurar produtos específicos não é uma tarefa complexa, mesmo com o método PLUS, Rzavian e Khosravi ou Ziadi que não possuem diretrizes e um processo de apoio como o da abordagem SMarty.

Outra suposição que surge é a relativa simplicidade de configuração de produtos com PLUS, Ziadi e Razavian e Khosravi, já que seus diagramas são modelados com menos

estereótipos comparados à abordagem SMarty, fazendo com que os participantes tivessem bom desempenho.

Pudemos observar que para o experimento com diagramas de casos de uso e classes e com diagramas de classes e componentes, SMarty não se sobressai em relação ao Método PLUS. Cenário diferente ocorre na experimentação com diagramas de classes e sequência, em que SMarty se mostra melhor do que o método Ziadi et al.

Esses dados, servem como subsídios para análise mais detalhada mostrada no próximo capítulo.

Discussão dos Resultados, Meta Análise e Novas Diretrizes

5.1 Considerações Iniciais

Neste capítulo serão discutidos os resultados obtidos por meio da condução dos experimentos cujo protocolo foi detalhado no Capítulo 3 e análises realizadas no Capítulo 4.

Além da análise individual, com estatística descritiva, teste de normalização, teste de hipótese e tamanho da amostra de cada experimento realizado individualmente, a análise se estende a uma meta análise, ou seja, os dados obtidos por todos os experimentos foram analisados em conjunto para que outras conclusões fossem possíveis de serem obtidas.

Os pontos observados como possíveis ameaças aos experimentos executados foram relatados, como as Ameaças à Validade de Construção, à Validade de Conclusão, à Validade Interna e Externa.

Por fim, as novas diretrizes foram criadas para a abordagem SMarty, cinco para auxiliar na configuração de produtos (C.1, C.2, C.2.1 C.3 e C.4) e quatro para rastreabilidade (R.1, R.2, R.3 e R.4). Essas diretrizes são detalhadas e cada uma exemplificada com diagrama.

5.2 Discutindo os Resultados dos Experimentos

5.2.1 Estudo Experimental #1 (Diagramas de Casos de Uso e Diagramas de Classes)

Por meio da aplicação dos testes estatísticos e suas análises, verificamos que há indícios iniciais de que a abordagem SMarty, em sua versão atual, não se sobressai em relação à efetividade na configuração de produtos com base nos diagramas de casos de uso de uma LPS comparada ao método PLUS.

Quanto ao diagrama de classes, com as análises percebemos que os participantes que fizeram uso do método PLUS tiveram configurações de produtos mais efetivas do que aqueles que usaram a abordagem SMarty.

Uma suposição considerada é que a abordagem PLUS possui um número consideravelmente menor de estereótipos, o que torna o diagrama simplificado e conseqüentemente a efetividade foi maior dos produtos configurados pelos participantes que utilizaram o método PLUS.

Em relação ao número de consultas ao material instrucional de cada abordagem, este não influenciou no valor calculado de efetividade de cada abordagem.

Existem indícios de que o nível prévio de conhecimento do participante em LPS, variabilidade e UML pode estar relacionado à melhora da efetividade do uso nas abordagens na configuração de produtos, principalmente no que se refere à abordagem SMarty.

Assim, participantes menos experientes conseguiram configurar produtos com maior efetividade com SMarty. Este fato pode ter ocorrido devido o detalhamento da abordagem SMarty, refletindo também no diagrama modelado com esta abordagem, o que exigiu, conseqüentemente, menos consultas ao material de apoio pelos participantes.

Por fim, os participantes que utilizaram a abordagem SMarty obtiveram maior efetividade no rastreamento de elementos entre os diagramas. Por isso, entende-se que a abordagem SMarty fornece mais subsídios para o rastreamento de elementos entre os diagramas, no entanto esse aspecto ainda precisa ser melhorado.

5.2.2 Estudo Experimental #2 (Diagramas de Classes e Diagramas de Componentes)

Por meio da aplicação dos testes estatísticos e suas análises, verificamos que há indícios de que a abordagem SMarty, em sua versão atual, não se sobressai em relação à efetividade

na configuração de produtos com base nos diagramas de classes nem de componentes de uma LPS comparada ao método PLUS e Razavian e Khosravi.

Existem indícios de que o nível prévio de conhecimento do participante em LPS, variabilidade e UML pode estar relacionado à melhora da efetividade do uso nas abordagens na configuração de produtos, principalmente no que se refere à abordagem SMarty.

Assim, participantes menos experientes conseguiram configurar produtos com maior efetividade com PLUS a partir de diagramas de Classes e com a abordagem Razavian e Khosravi a partir de diagrama de Componentes. Esse cenário pode ter acontecido devido a simplicidade das abordagens PLUS e Razavian e Khosravi, já que os diagramas de classes e componentes da MM são extensos e um grande número de estereótipos pode causar confusão. Dessa maneira, resolver as variabilidades e configurar produtos específicos não foi uma tarefa complexa, mesmo com o método PLUS e Razavian e Khosravi que não possui diretrizes e um processo de apoio como o da abordagem SMarty.

5.2.3 Estudo Experimental #3 (Diagramas de Classes e Sequência)

Com relação à primeira questão de pesquisa, referente à efetividade das duas abordagens, os resultados evidenciaram vantagem da abordagem SMarty em relação à abordagem Ziadi et al tanto na configuração de produtos a partir do diagrama de classes como a partir de diagramas de sequência.

Apesar da abordagem Ziadi et al possuir poucos estereótipos para o diagrama de classes e sequência, eles não são dedutivos como os estereótipos da abordagem SMarty. Esse pode ser o motivo que levou os participantes que utilizaram a abordagem SMarty a configurar produtos mais efetivos do que aqueles que utilizaram Ziadi et al.

Existem indícios de que o nível prévio de conhecimento dos participantes em LPS, variabilidade e UML pode estar relacionado à melhora da efetividade do uso nas abordagens na configuração de produtos, principalmente no que se refere à abordagem Ziadi et al. Assim, participantes menos experientes conseguiram configurar produtos com maior efetividade com SMarty.

Isso também pode estar relacionado aos estereótipos claros e dedutivos de SMarty, além das diretrizes, que mesmo sem conhecimento prévio, os participantes que utilizam SMarty conseguem configurar produtos mais efetivos mesmo com pouco conhecimento.

Em relação à terceira questão de pesquisa, entende-se que a abordagem SMarty fornece subsídios para o rastreamento de elemento entre os diagramas. No entanto esse aspecto ainda precisa ser melhorado, já que a forma que a abordagem apoia da rastreabilidade

não é simples, necessitando de diretrizes claras e objetivas para que o participante consiga realizar tal tarefa.

Já Ziadi não pôde ser avaliada por não fornecer apoio para o rastreamento de elementos entre os diagramas.

Entendemos que os resultados deste experimento junto com os demais para diagramas de classes e de casos de uso e classes e componentes fornecem ainda mais subsídios para investir em uma abordagem que tenha como foco diagramas UML e, conseqüentemente, produza apoio ferramental para o ciclo de LPSs baseadas em UML. Tudo isso se mostra motivador para que a abordagem SMarty possa ser continuamente evoluída visando a reformulação e o teste de hipóteses em novos experimentos entre SMarty e outros métodos com os demais diagramas UML.

5.3 Meta Análise

Os estudos primários são processos metodológicos específicos, usados para avaliar as hipóteses formuladas pelo pesquisador sob condições estabelecidas na metodologia (de Almeida Biolchini *et al.*, 2007).

A análise secundária é a análise feita dos estudos primários, com o objetivo de estabelecer comparações, generalização, sumarizações entre os estudos primários. Segundo Glass (1976), a meta análise pode ser definida como uma análise estatística de uma coleção de análises resultantes de estudos primários com o propósito de integrar os resultados.

Nesta seção, os dados e análises estatísticas obtidos dos experimentos são organizados de forma gráfica e tabular a fim de gerar comparações e deduções que não são possíveis apenas com os estudos primários.

Uma meta análise tradicional começa com uma exaustiva busca na literatura para encontrar todos os artigos sobre o tema que será analisado (Miller, 2000). No caso deste trabalho, não fui necessária uma revisão de literatura, já que a meta análise é feita por meio de três estudos primários já relatados neste trabalho:

- Experimento #1 (Diagramas de Casos de Uso e Classes);
- Experimento #2 (Diagramas de Classes e Componentes);
- Experimento #3 (Diagramas de Classes e Sequência).

A partir destes estudos, segundo Miller (2000), é possível seguir uma das opções abaixo:

- estudos de comparação - testes difusos: usado para determinar se dois ou mais estudos produzem resultados significativamente diferentes, mas não revelam se a diferença possui causa sistêmica;
- estudos de comparação - testes focados: usando para determinar se dois ou mais estudos produzem resultados significativamente diferentes com base em uma teoria;
- estudos de combinação: usando quando se deseja determinar a potência de uma variável entre os estudos.

No caso deste trabalho foi feito um estudo de comparação. Todos os experimentos seguem uma mesma questão de pesquisa: Comparar a efetividade de SMarty em relação à outras abordagens na configuração de produtos específicos.

Muitos dos pontos avaliados individualmente agora são avaliados em conjunto, como:

- Média, mediana e desvio padrão das efetividades;
- conhecimento dos participantes em UML;
- conhecimento dos participantes em LPS;
- área de atuação dos participantes.

O número de consultas no material não foi considerado, já que foi contabilizada apenas no primeiro estudo. Nos Experimentos #2 e #3 não foi possível estabelecer tal número, isso ocorreu porque no Experimento #2 os participantes que utilizaram SMarty configuraram dois produtos, enquanto os participantes que utilizaram as outras abordagens (PLUS e Razavian e Khosravi) só configuraram um produto. No caso do Experimento #3, como parte aconteceu de forma online, não foi possível contabilizar o número de consultas ao material. Dessa maneira, como houve a contagem apenas no primeiro estudo não existia a possibilidade de uma análise em conjunto.

5.3.1 Estatística Descritiva

Nas Tabela 5.1, Tabela 5.2, Tabela 5.3 e Tabela 5.4, estão resumidos os valores de média, mediana e desvio padrão obtidos com a efetividade dos produtos configurados pelos mesmos, usando a abordagem SMarty e os métodos PLUS, Razavian e Khosravi e Ziadi et al respectivamente.

As maiores médias de efetividade foram registrados nos experimentos #2 e #3 pelos participantes que utilizaram a abordagem SMarty. No Experimento #2, com diagrama

de classes, a média de efetividade foi 0,92 enquanto no experimento 3, a média registrada foi ainda maior com o diagrama de classes, em 0,96 e com o diagrama de sequência 0,92.

Os menores valores de desvio padrão observados entre as efetividades dos produtos configurados ocorreram com os diagramas de classes e sequência dos participantes que utilizaram a abordagem SMarty, sendo respectivamente 0,06 e 0,07, esses baixos valores indicam uma homogeneidade da amostra.

Tabela 5.1: Valores de Média, Mediana e Desvio Padrão Observados com SMarty

	Experimento #1				Experimento #2				Experimento #3			
	Produto 1 C. de Uso		Produto 2 Classes		Produto 1 Classes		Produto 2 Compon.		Produto 1 Classes		Produto 2 Sequência	
	Cor.	Efet.	Cor.	Efet.	Cor.	Efet.	Cor.	Efet.	Cor.	Efet.	Cor.	Efet.
Média	8,90	0,89	7,35	0,73	9,29	0,92	8,82	0,88	9,60	0,96	9,20	0,92
Med.	10,00	1,00	9,00	0,90	10,00	1,00	10,00	1,00	10,00	1,00	9,00	0,90
D.P.	3,05	0,30	3,80	0,38	1,79	0,18	2,37	0,23	0,63	0,06	0,77	0,07

Tabela 5.2: Valores de Média, Mediana e Desvio Padrão Observados com o método PLUS

	Experimento #1				Experimento #2	
	Produto 1 C. de Uso		Produto 2 Classes		Produto 1 Classes	
	Cor.	Efet.	Corr.	Efet.	Cor.	Efet.
Média	9,47	0,94	9,39	0,93	8,70	0,87
Med.	10,00	1,00	10,00	1,00	10,00	1,00
D.P.	0,99	0,09	0,59	0,05	1,99	0,19

Tabela 5.3: Valores de Média, Mediana e Desvio Padrão Observados com o método Razavian e Khosravi

	Experimento #2	
	Produto 1 Componentes	
	Cor.	Efet.
Média	8,58	0,86
Med.	9,00	0,90
D.P.	1,90	0,19

Com todos esses dados agrupados e comparados, foi possível observar que considerando todos os experimentos, os participantes que utilizaram a abordagem SMarty se saíram melhor na maioria deles na configuração de produtos: No Experimentos #2 com diagrama de classes (SMarty x PLUS) e com diagramas de componentes (SMarty x Razavian

Tabela 5.4: Valores de Média, Mediana e Desvio Padrão Observados com o método Ziadi et al

	Experimento #3			
	Produto 1 Classes		Produto 2 Sequência	
	Cor.	Efet.	Cor.	Efet.
Média	8,22	0,82	7,66	0,76
Med.	8,00	0,80	7,00	0,70
D.P.	1,66	0,16	1,29	0,12

e Khosravi) e também no Experimento #3 (SMarty x Ziadi et al) na configuração de produtos.

5.3.2 Testes de Hipótese

O resultado do teste de normalidade Shapiro-Wilk aplicado na efetividade obtida com todas as abordagens de gerenciamento de variabilidades (SMarty, PLUS, Razavian e Khosravi e Ziadi et al) foi não-normal. Dessa maneira, o teste de hipóteses não-paramétrico Mann-Whitney-Wilcoxon foi escolhido para todas as amostras, os valores obtidos pode ser observados na Tabela 5.5.

Tabela 5.5: Valores Observados com o teste Mann-Whitney-Wilcoxon

Teste Mann - Whitney- Wilcoxon			
Experimento		p	Rejeitar H0?
#1	Casos de Uso	0,1898	Não
	Classes	0,02797	Sim
#2	Classes	0,2539	Não
	Componentes	0,3884	Não
#3	Classes	0,0299	Sim
	Sequência	0,001846	Sim

No Experimento #1, usando diagramas de casos de uso, p foi maior que 0,05 fazendo com que a hipótese nula não pudesse ser rejeitada. Já com diagrama de classes, p foi menor que 0,05 e, por isso, a hipótese nula foi rejeitada.

No segundo experimento, indiferente dos tipos de diagramas utilizados, classes ou componentes, podemos observar que p é maior que 0,05 e por isso a hipótese nula não pôde ser rejeitada.

Por último, no terceiro experimento, em ambos os casos, com diagramas de classes e sequência, o valor de p obtido foi menor do que 0,05 fazendo com que a hipótese nula fosse rejeitada.

Nas próximas seções serão discutidos os resultados causados por esses valores.

5.3.3 Formação dos Participantes

Os dados apresentados na Tabela 5.6, referem-se ao nível de formação de todos os participantes, dos experimentos #1, #2 e #3.

Os participantes melhor preparados são as que utilizam a abordagem Ziadi et al no experimento #3, com 3 participantes Mestrandos (representando 20% do total de participantes) e 12 graduados. Em seguida temos os participantes que utilizaram a abordagem SMarty, também do experimento #3, com 2 dos participantes mestrandos (representando 13,3 % do total) e o restante já graduados.

Por outro lado, o experimento que mais obteve participantes com menor nível de formação foi o #1, em que tanto os usuários da abordagem SMarty quanto do método PLUS eram ainda graduandos.

Tabela 5.6: Nível de formação dos participantes

Nível de Formação					
	Experimento	Graduando	Graduado	Mestrando	Total
#1	SMarty	20	-	-	20
	PLUS	23	-	-	23
#2	SMarty	14	-	3	17
	PLUS	14	-	3	17
	Razavian e Khosravi	14	-	3	17
#3	SMarty	-	13	2	15
	Ziadi et al	-	12	3	15

5.3.4 Nível de Conhecimento dos Participantes

O nível de conhecimento dos participantes em UML estão distribuídos entre “Nenhum”, “Básico”, “Moderado” e “Avançado” e está relacionado na Tabela 5.7.

Nota-se que mesmo possuindo os participantes com menor nível de formação, o experimento #1 conta com 3 participantes que utilizaram a abordagem SMarty e o método PLUS com conhecimento “Avançado” em UML.

Em relação ao conhecimento em LPS, os níveis estão distribuídos entre: “Nenhum”, “Leu”, “Básico” e “Moderado” e colocados na Tabela 5.8. O nível “Avançado” não aparece na tabela pois nenhum dos participantes assinalou esta alternativa.

No quesito conhecimento em LPS, os participantes que utilizaram o método Razavian e Khosravi no experimento #2 são os que tiveram a melhor média de conhecimento. Um dos participantes relatou ter conhecimento moderado e três conhecimento básico em LPS, o equivalente a 23.5% dos participantes.

Tabela 5.7: Conhecimento dos participantes em UML

Conhecimento UML					
	Experimento	Nenhum	Básico	Moderado	Avançado
#1	SMarty	-	10	10	3
	PLUS	-	12	8	3
#2	SMarty	3	1	4	1
	PLUS	-	12	4	-
	Razavian e Khosravi	-	6	10	1
#3	SMarty	-	9	6	-
	Ziadi et al	-	8	7	-

Tabela 5.8: Conhecimento dos participantes em LPS

Conhecimento LPS					
	Experimento	Nenhum	Leu	Básico	Moderado
#1	SMarty	16	7	-	-
	PLUS	15	7	1	-
#2	SMarty	5	9	2	1
	PLUS	5	11	1	-
	Razavian e Khosravi	6	7	3	1
#3	SMarty	7	7	1	-
	Ziadi et al	9	4	2	-

5.4 Ameaças à Validade

As ameaças discutidas na sequência foram identificadas e mitigadas nos experimentos. As ameaças fazem parte das diretrizes (D.23) definidas por Furtado (2018) para avaliação de qualidade de experimentos em LPS:

5.4.1 Ameaças à Validade de Conclusão

A principal ameaça de conclusão para este estudo está relacionada ao nível acadêmico dos participantes, visto que são alunos de graduação, sem experiência profissional. Outro ponto a ser considerado é que os participantes podem não ter reportado corretamente a quantidade de consultas realizadas no material. Outra ameaça relacionada à conclusão é a quantidade de participantes. Tal ameaça pode ser dividida em dois fatores:

- a adesão dos alunos da turma não foi total, pois muitos não se interessaram sobre o assunto e por não ser parte do conteúdo programático da graduação; e
- uma vez que os participantes foram divididos em dois blocos, o número para cada abordagem foi reduzido à metade. Logo, os dados obtidos a partir deste estudo só poderão ser considerados indicadores e não conclusivos.

5.4.2 Ameaças à Validade de *Constructo*

A viabilidade do estudo e a instrumentação foram testadas inicialmente com um projeto piloto, para analisar se são adequados para ser aplicadas no estudo real, e conseqüentemente não invalidar o experimento. Quanto ao nível de conhecimento, os participantes receberam um treinamento sobre os conceitos de LPS, variabilidade em LPS e as abordagens utilizadas em cada experimento, seja SMarty, PLUS, Razavian e Khosravi ou Ziadi et al, por isso entendemos que eles obtiveram a compreensão necessária para configurar os produtos.

5.4.3 Ameaças à Validade Interna

Algumas considerações podem ser feitas para o experimento quanto à validade interna:

- todos os participantes do estudo foram estudantes de graduação, não houve diferença significativa entre as habilidades do grupo;
- o treinamento quanto aos assuntos abordados no experimento nivelou o conhecimento quanto à LPS e variabilidade em LPS podendo, assim, ser considerado que as respostas fornecidas são válidas e significantes; e
- não houve influência entre os participantes durante o experimento, já que foi de forma presencial e sob supervisão de um ser humano para não haver interação entre participantes.

5.4.4 Ameaças à Validade Externa

Quanto à validade externa, podem ser detectadas ameaças quanto à instrumentação. Os diagramas da LPS MM não são comerciais, ou seja, não são de casos reais. Além disso, eles são bem simples de serem entendidos. Logo, em estudos adicionais, devem ser consideradas LPS reais e mais complexas para obter possivelmente resultados diferentes na indústria.

O nível de conhecimento dos participantes pode também ser uma ameaça, já que alguns têm mais conhecimento que outros sobre os temas relacionados (LPS e variabilidade em LPS), porém, autores como Falessi (Falessi *et al.*, 2017), discutem a importância do uso de estudantes em estudos experimentais e reforçam que nenhuma população deve ser considerada melhor que a outra.

5.5 Apresentação e Empacotamento dos Resultados

Toda a instrumentação e resultados deste estudo estão disponíveis como *Open Access*.

Seguimos os princípios de *Open Science* permitindo a disseminação e disponibilidade pública dos dados, bem como transparência, validade, auditabilidade e reprodutibilidade do estudo.

5.6 Novas Diretrizes para SMarty

Algumas diretrizes foram criadas para auxiliar o participante na configuração de um produto específico a partir de uma LPS e também para facilitar o rastreamento de elementos entre os diagramas suportados pela SMarty.

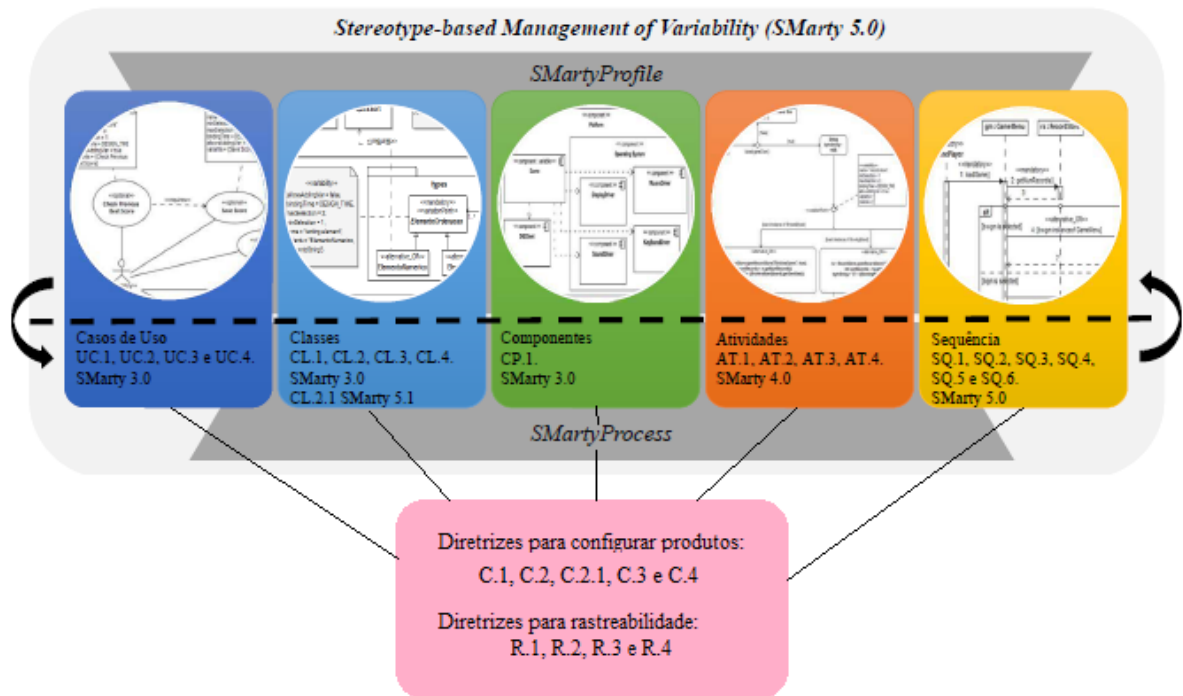
Na Figura 5.1 é ilustrada a visão Geral da abordagem SMarty com os estereótipos de cada diagrama. Nessa Figura, representado pelo quadro rosa, estão as novas diretrizes criadas. Este quadro está ligado à todos os diagramas suportados pela abordagem SMarty, já que as diretrizes servem para todos eles.

As novas diretrizes estão descritas a seguir.

5.6.1 Diretrizes para Configuração de Produtos

Com base nos dados coletados e *feedbacks* recebidos dos participantes a cada experimento, 5 diretrizes foram criadas para auxiliar na configuração de produtos, com o intuito de evoluir a abordagem SMarty.

Figura 5.1: Visão Geral da Abordagem SMarty com as novas diretrizes



Fonte: modificado de (Marcolino e OliveiraJr, 2015)

- C.1 Selecione para o produto a ser configurado todos os componentes, classes, atores, mensagens ou casos de uso que são obrigatórios, demarcados com o estereótipo «mandatory». Um exemplo está representado na Figura 5.2, em que o Caso de Uso *Check Out Customer* demarcado com o estereótipo «mandatory» está selecionado. Inclua também todas as interfaces obrigatórias, demarcadas com o estereótipo «mandatory». Na Figura 5.3 está representado parte de um diagrama de componentes com as interfaces demarcadas com o estereótipo «mandatory» selecionadas.

Figura 5.2: Exemplo da diretriz C.1

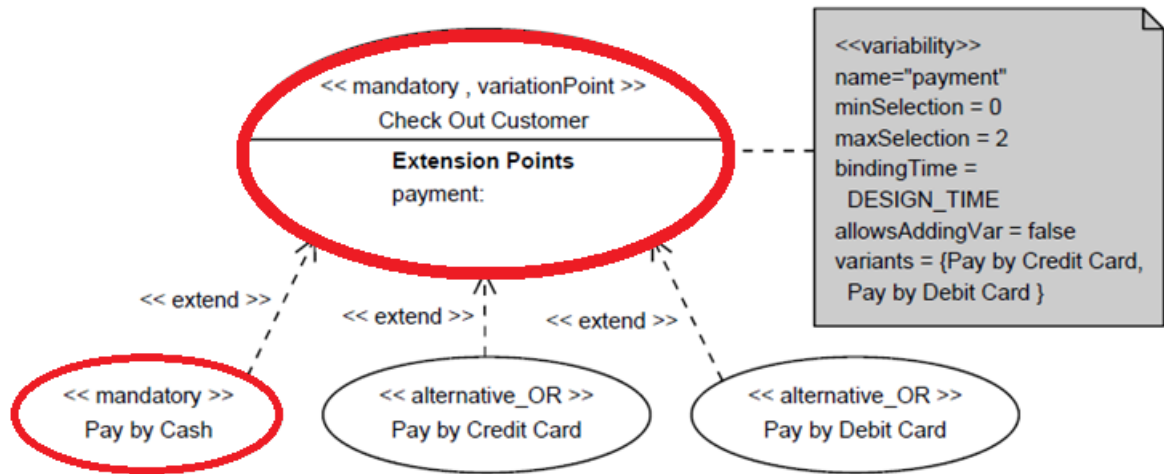
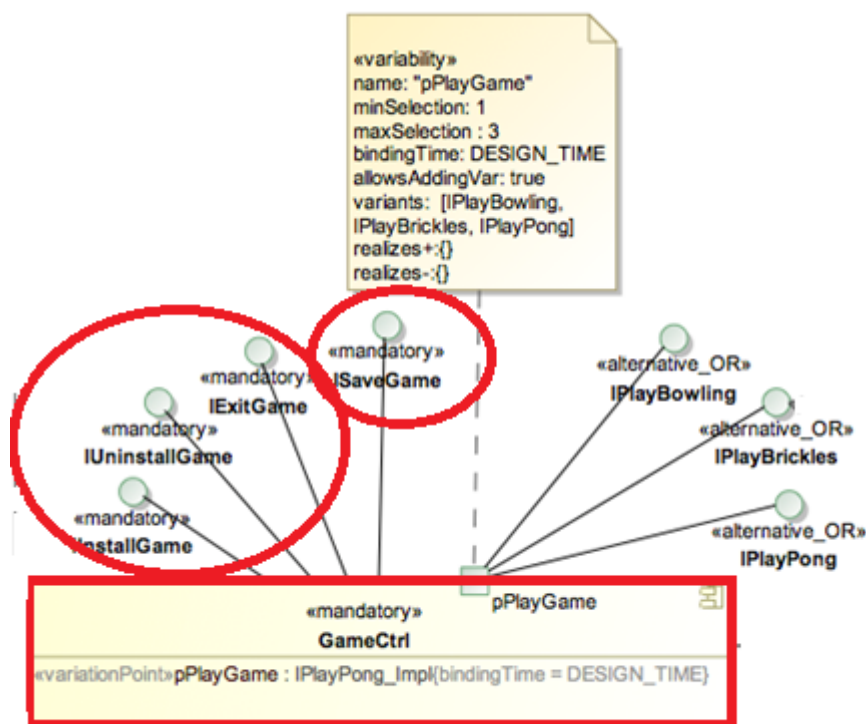


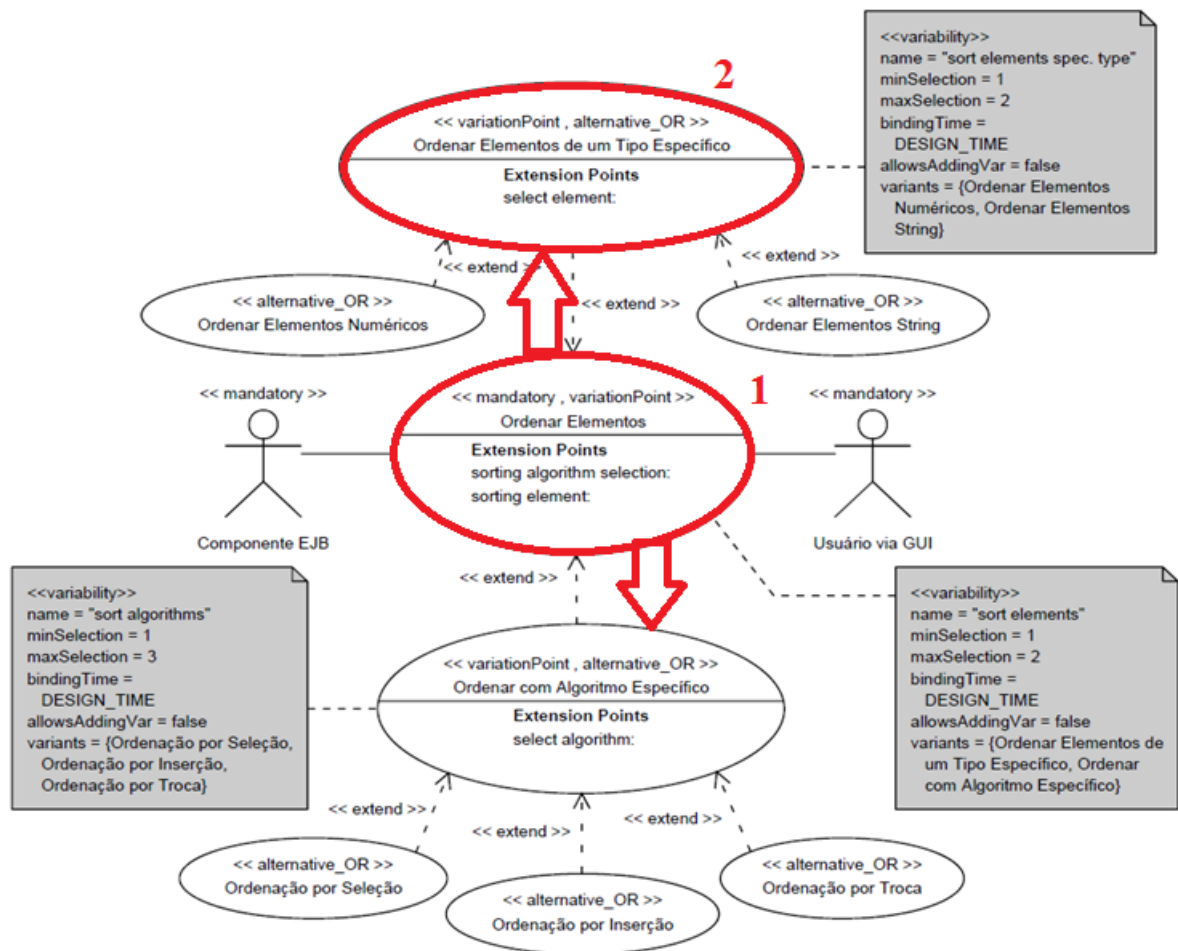
Figura 5.3: Exemplo da diretriz C.1



- C.2 Resolva os pontos de variação, com o estereótipo «variationPoint», de acordo com a notação de variabilidade, observando a quantidade mínima e máxima de variantes que devem ser escolhidas. Na Figura 5.4 um diagrama de Ordenação de

Elementos é usado para exemplificar esta diretriz. O Caso de uso "Ordenar Elementos" é um ponto de variação com duas possíveis variantes: "Ordenar Elementos de um Tipo Específico" e "Ordenar com Algoritmo Específico". Observando a notação de variabilidade, notamos que o minSelection é 1 e o maxSelection 2, ou seja, no mínimo 1 e no máximo 2 das variantes devem ser escolhidas. Caso a variante "Ordenar Elementos de um Tipo Específico" seja escolhida, devemos considerar que também se trata de um Ponto de Variação, e, por isso, o mesmo processo deve ser executado.

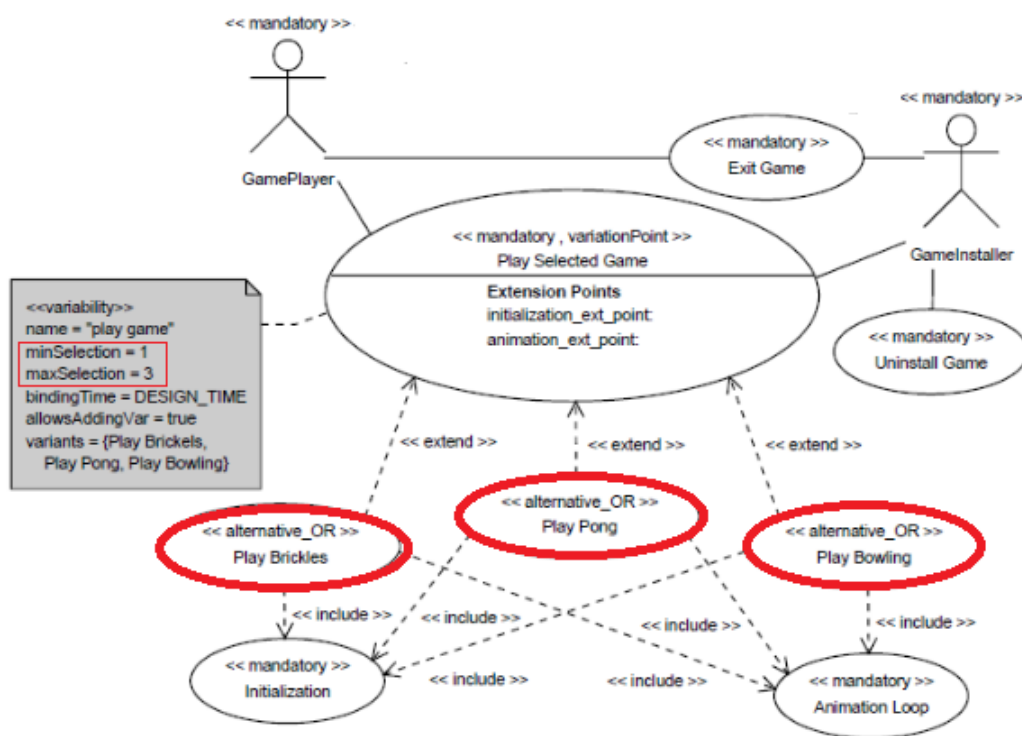
Figura 5.4: Exemplo da diretriz C.2



- C.2.1 Observe a relação das variantes por meio de seus estereótipos. Se forem «alternative_OR» pelo menos uma das variantes deverá ser escolhida para resolver o ponto de variação, se «alternative_XOR» somente uma das variantes

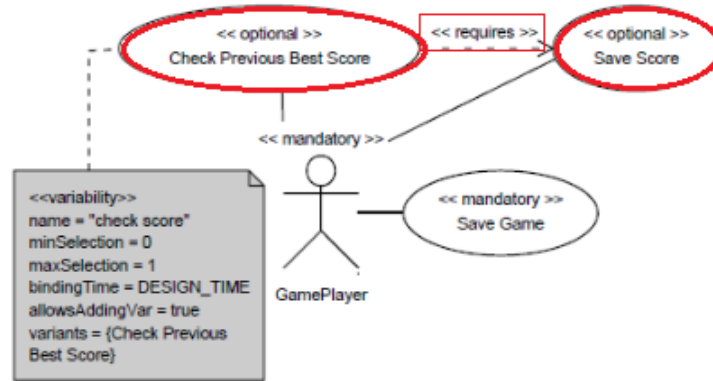
deverá ser escolhida para resolver o ponto de variação. No exemplo da Figura 5.5 o Caso de uso "Play Selected Game" é um Ponto de Variação com três variantes: "Play Brickles", "Play Pong" e "Play Bowling", a relação entre essas variantes deve ser observada, no caso, são todas demarcadas com o estereótipo «alternative_OR», ou seja, entre elas, pelo no mínimo uma deve ser escolhida. O mesmo pode ser observado na notação de variabilidade do Ponto de Variação pelo "minSelection" e "maxSelection".

Figura 5.5: Exemplo da diretriz C.2.1



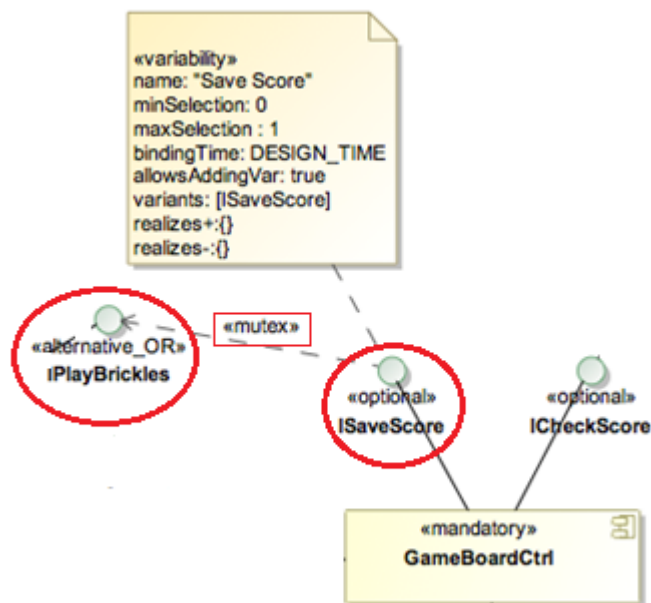
- C.3 Uma variante V1 que ao ser selecionada para fazer parte de um determinado produto, tiver um relacionamento com outra variante V2 com o estereótipo «requires», essa variante (V2) também deverá fazer parte do produto. Na Figura 5.6, existe uma relação de «requires», entre os Casos de Uso “Check Previous Best Score” e “Save Score”, isso quer dizer que se o primeiro Caso de uso for escolhido para uma configuração de produto, obrigatoriamente o segundo Caso de Uso também deve estar nessa configuração.

Figura 5.6: Exemplo da diretriz C.3



- C.4 Uma variante V1 que ao ser selecionada para fazer parte de um determinado produto, tiver um relacionamento com outra variante V2 com o estereótipo «mutex», essa variante (V2) não poderá fazer parte do produto. Uma Relação de «mutex» existe entre as Interfaces “ISaveScore” e “IPlayBrickles” Figura 5.7. Isso quer dizer que se a primeira Interface for escolhida para a Configuração de um Produto, a segunda não deve estar nessa Configuração.

Figura 5.7: Exemplo da diretriz C.4



5.6.2 Diretrizes para Rastreabilidade

Analizados os dados obtidos com a rastreabilidade de elementos entre os diagramas a cada experimento, foi percebida a necessidade de criar diretrizes para auxiliar esta tarefa. Por isso, 4 diretrizes para rastreabilidade foram criadas. Elas são apresentadas a seguir:

- R.1 Identifique a notação de comentário UML, com o estereótipo «variability» no elemento ao qual pretende-se rastrear. A Figura 5.8 representa uma notação de comentário usada pela abordagem SMarty.

Figura 5.8: Exemplo da diretriz R.1

```

    <<variability>>
    name: favourite Media
    minSelection: 0
    maxSelection: 1
    bindingTime: DESIGN_TIME
    allowsAddingVar: true
    variants: [Manage Favourite Media]
    realizes+: {}
    realizes-: {IManageFavouriteMedia}
  
```

- R.2 Para a identificação de rastreabilidade em níveis de maior abstração, procure o meta-atributo “realizes+”, que contém um conjunto de nomes da variabilidade. Este está identificado na Figura 5.9.

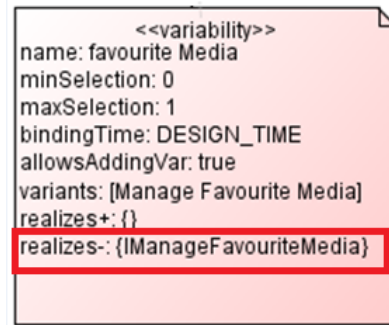
Figura 5.9: Exemplo da diretriz R.2

```

    <<variability>>
    name = "sending media"
    minSelection = 0
    maxSelection = 1
    bindingTime = DESIGN_TIME
    allowAddingVar = true
    Variants = {SenderMgr}
    realizes+: {Sending Media}
    realizes-: {}
  
```

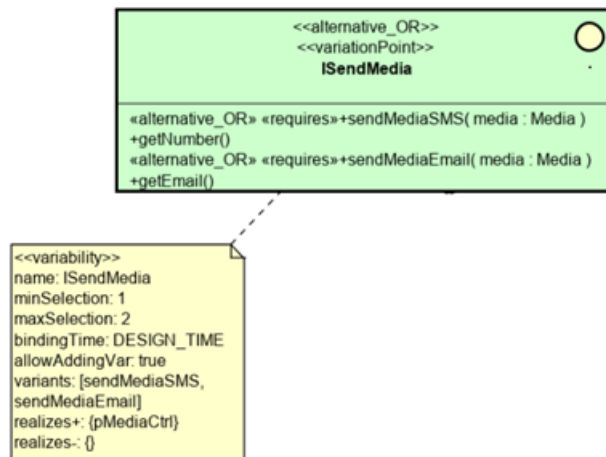
- R.3 Para rastrear elementos em diagramas de menor abstração, identifique o “realizes-” que contém um conjunto de nomes da variabilidade, como mostrado na Figura 5.10.

Figura 5.10: Exemplo da diretriz R.3



- R.4 Vá até o diagrama correspondente e procure pelo nome identificado no meta-atributo realizes (+ ou -).

Figura 5.11: Exemplo da diretriz R.4



Considerando a Figura 5.11, para determinar quais elementos essa variabilidade (*ISendMedia*) pode interferir, é necessário olhar o *realizes* na notação de variabilidade. Caso você queira saber onde essa variabilidade interfere em um nível de diagrama mais abstrato (por exemplo, no diagrama de classes), olhe para o *realizes +*, nesse caso está escrito “*pMediaCtrl*”, então é preciso procurar uma notação de variabilidade com este nome no diagrama de classes. A classe ou interface que essa notação estiver associada é o lugar que ocorre variação caso o componente “*ISendMedia*” seja alterado. O mesmo deve ser feito para diagramas menos abstratos, porém olhando para “*realizes -*”.

5.7 Considerações Finais

São diversas as razões que podem fazer com que um experimento seja inválido ou contenha limitações. Neste capítulo foram apontados os principais motivos que foram prejudiciais à execução perfeita dos experimentos.

Além disso, a análise e suposição dos motivos dos resultados foram levantados.

De maneira geral, pode-se dizer que há indícios que SMarty é mais efetiva que as demais abordagens avaliadas (PLUS, Razavian e Khosravi e Ziadi et al), no que se refere a efetividade em rastrear elementos entre os digramas, já que SMarty é a única que provê apoio nesta atividade.

Conclusão

A ideia de reusar partes de um sistema não é foco somente nas pesquisas acadêmicas, mas deve-se principalmente ao panorama competitivo da empresas, que buscam meios mais efetivos de reutilizar artefatos de software.

Nesse ponto, a abordagem de LPS mostra-se efetiva já que permite reutilização e customização em massa para uma produção de software muito mais econômica do que produzir produtos de software individualmente.

A variabilidade em uma LPS é o que permite a diversificação dos seus produtos e tem grande impacto em como um software é desenvolvido, estendido e mantido. A atividade de gerenciamento de LPS, assim, se torna umas das mais essenciais.

Dentre as abordagens baseadas em UML, uma das mais importantes é SMarty, composta de um perfil, SMartyProfile, e um processo, o SMartyProcess que, por meio das suas diretrizes, auxilia o usuário na identificação, representação, delimitação e rastreamento de variabilidades em modelos de LPS.

Este trabalho apresentou a condução de estudos experimentais buscando a caracterização da efetividade de SMarty com relação à efetividade de na configuração de produtos e rastreabilidade, frente a abordagens existentes na literatura.

No Experimento #1, os resultados evidenciaram vantagem do método PLUS em relação à abordagem SMarty na configuração de produtos a partir do diagrama de classes. Outra evidência foi o número de consultas ao material instrucional de cada abordagem não ter influenciado no valor calculado de efetividade de cada abordagem. Existem indícios para este experimento de que participantes menos experientes conseguiram configurar produtos com maior efetividade com SMarty. Ainda, entende-se que a abordagem SMarty

fornece mais subsídios para o rastreamento de elemento entre os diagramas, no entanto esse aspecto ainda precisa ser melhorado.

No Experimento #2 não houve indícios de que a abordagem SMarty fosse superior a PLUS na configuração de produtos com diagramas de classes e nem a Razavian e khosravi na configuração de produtos a partir de diagramas de componentes. Como no estudo anterior, neste o número de consultas ao material instrucional de cada abordagem não influenciou no valor calculado de efetividade de cada abordagem.

Já no Experimento #3 referente à efetividade das duas abordagens, os resultados evidenciaram vantagem da abordagem SMarty em relação à abordagem Ziadi et al tanto na configuração de produtos a partir do diagrama de classes como a partir de diagramas de sequência. Existem indícios de que o nível prévio de conhecimento em LPS, variabilidade e UML dos participantes pode estar relacionado à melhora da efetividade do uso nas abordagens na configuração de produtos, principalmente no que se refere à abordagem Ziadi et al. Assim, participantes menos experientes conseguiram configurar produtos com maior efetividade com SMarty.

Com os dados obtidos em todos os experimentos foi possível realizar uma meta análise, resumindo os resultados e valores calculados com o uso de testes estatísticos, para conseguir informações que não seriam possíveis com a análise dos estudos individuais.

De maneira geral, analisando o conjunto de dados da meta análise, houve indícios de que SMarty é superior às demais abordagens em todas as questões de pesquisa, já que considerando todos os experimentos, a abordagem SMarty foi melhor nas maiorias das questões respondidas.

Os participantes que configuraram produtos usando a abordagem SMarty, obtiveram melhores resultados de efetividade na maioria dos produtos.

A correlação entre o conhecimento dos participantes com a efetividade do produto configurado também levou a crer, que na maioria dos casos, os participantes que configuraram produtos com a abordagem SMarty mesmo com menos conhecimento configuraram produtos mais efetivos.

E por último: SMarty provê suporte para rastreabilidade de elementos entre os diagramas, o que faz com que os participantes tenham menos dificuldade nesta tarefa, apesar de ser necessária uma evolução de SMarty neste sentido.

6.1 Contribuições

Como resultado deste trabalho de mestrado e principal contribuição tem-se: a avaliação experimental de SMarty para diagramas de casos de uso, classes, componentes e sequência

da UML, além da criação de diretrizes para configuração de produtos e rastreamento de variabilidades.

Resultados e Contribuições quanto às Diretrizes

Apesar de todo suporte que a abordagem SMarty oferece, como uma vasta quantidade de estereótipos e apoio na rastreabilidade, configurar um produto a partir de uma LPS e rastrear elementos entre os diagramas UML nem sempre é uma tarefa trivial. Pensando nisso, Foram criadas seis diretrizes (C1, C2, C3, C3.1, C4 e C.5) para ajudar os participantes a configurar produtos de forma mais efetiva, e mais quatro diretrizes (R.1, R.2, R.3 e R.4) para auxiliar os participantes a rastrear elementos entre os diagramas UML suportados pela abordagem SMarty.

Resultados e Contribuições quanto à Avaliação Experimental

A execução dos experimentos contribuiu com obtenção de dados e estatísticas que contribuiu na criação de novas diretrizes para o SMartyProcess.

Essas informações foram obtidas gradualmente, ou seja, a cada experimento foi possível coletar importantes informações com o *feedback* dos participantes e os resultados analisados, com isso, diretrizes foram criadas para a configuração de produtos e rastreabilidade de elementos entre os diagramas UML.

A primeira contribuição neste tópico foi a avaliação experimental para modelos de casos de uso e classes, comparando a abordagem SMarty com o método PLUS em relação à configuração de produtos efetivos a partir da LPS AGM. Nesta avaliação PLUS foi mais efetiva que a abordagem SMarty em diagramas de classes.

Em seguida foi realizada uma avaliação experimental para modelos de casos de classes e componentes, também comparando a abordagem SMarty com o método PLUS e Razavian e Khosravi em relação a configuração de produtos efetivos a partir da LPS MM. Nesta avaliação a hipótese nula não pode ser rejeitada, nem para diagrama de classes, nem componentes, com evidência de que nenhuma abordagem (PLUS ou Razavian e Khosravi) é superior ou inferior a SMarty.

A última contribuição quanto à Avaliação experimental, foi com modelos de classes e sequência, em que a efetividade da Abordagem SMarty foi comparada a Ziadi et al quanto à configuração de produtos a partir da LPS AGM. Desta vez, SMarty se mostrou mais efetiva, tanto com configurações a partir do diagrama de classes, quanto com diagramas de componentes.

Resultados e Contribuições quanto ao Mapeamento Sistemático

A condução do Mapeamento Sistemático foi uma atualização do mapeamento realizado por Marcolino (Marcolino e OliveiraJr, 2015), com o objetivo de identificar avaliações experimentais com abordagens de gerenciamento de variabilidades.

Essa atualização foi necessária para saber se no período da atualização novos estudos foram realizados, com outras abordagens ainda não identificadas para realização dos estudos experimentais.

Publicações dos Resultados

- Thais S. Nepomuceno, Edson Oliveira Jr: **Configuring Software Product Line Specific Products with SMarty and PLUS: An Experimental Study on Use Case Diagrams**. *Simpósio Brasileiro de Qualidade de Software (SBQS)*, 2018, p. 81-90.
- Thais S. Nepomuceno, Edson Oliveira Jr, Raqueline Penteado, Marco A. Graciotto Silva, Avelino F. Zorzo: **Empirical Study on Product Configuration and Traceability in UML-based Product-Lines**. *Ibero-American Conference on Software Engineering (CibSE)*, 2020.
- Thais S. Nepomuceno, Edson Oliveira Jr, Ricardo Geraldi, Andreia Malucelli, Sheila Reinehr, Marco A. Graciotto Silva: **Software Product Line Configuration and Traceability: an Empirical Study on SMarty Class and Component Diagrams**. *IEEE Annual Computers, Software, and Applications Conference (COMPSAC)*, 2020.

6.2 Limitações

Este estudo apresentou algumas limitações como as discutidas na Seção 5.4 encontrando também dificuldade de obter participantes bem qualificados, bem como a escassez de LPSs industriais, dificultando a obtenção de uma análise mais próxima da realidade empresarial, e assim, que possa ser melhor aproveitada pela mesma.

A seguir as limitações são descritas de modo mais detalhado:

- falta de abordagens de gerenciamento de variabilidade que utilizam modelos de casos de uso, classes e componentes da UML no mapeamento sistemático, limitando a comparação entre as abordagens já conhecidas e experimentadas anteriormente;
- amostras de tamanhos restritos foram consequências da dificuldade em conseguir participantes com o necessário conhecimento para participar dos experimentos, além da restrição do horário em que podiam participar;

- dificuldade para encontrar LPSs advindas da indústria, ou de casos reais de estudo conduziu a utilização de LPS pedagógicas, limitando reduzindo a representatividade em relação a LPS reais;
- inexistência de validações experimentais similares para uma possível comparação de resultados e aprimoramento do pacote experimental.

6.3 Trabalhos Futuros

Como um primeiro ponto a ser destacado, os experimentos executados devem ser replicados a fim de mitigar as ameaças identificadas, como a quantidade de participantes, devem ser recrutadas mais turmas com pessoas que possuam conhecimento avançado em UML, a fim de de aumentar as amostras, gerando resultados mais generalizáveis. Além disso, convidar participantes com mais experiência na indústria, com desenvolvimento de software e não só alunos, para aumentar a gama de participantes com diferentes áreas de atuação.

As diretrizes desenvolvidas mostram-se eficiente, no entanto é necessário desenvolver mais diretrizes que auxiliem os participantes na configuração correta de produtos e na rastreabilidade de elementos. Tais diretrizes devem ser desenvolvidas especificamente para cada tipo de diagrama UML suportado pela abordagem SMarty, bem como devem ser avaliadas empiricamente.

Outro ponto a ser considerado é automatizar a forma de executar os experimentos. Muitas LPS são grandes e se tornam ainda maior dependendo do tipo de diagrama UML em que é configurada. Para reduzir o cansaço e fadiga dos participantes, automatizar o processo de experimentação tornaria a execução dos experimentos mais rápida e efetiva.

As LPS utilizadas neste trabalho não são casos reais, por isso, apesar da dificuldade de obter informações e regras de negócios provenientes da indústria, identificar e utilizar LPS originárias de empresas reais para condução dos estudos experimentais tornaria os resultados dos experimentos mais objetivos.

A atualização do Mapeamento Sistemático para identificar estudo de avaliação de Abordagens de Gerenciamento de Variabilidades é outro ponto a ser considerado, já que o resultado pode implicar em novos planejamentos de estudos, com abordagens ainda não avaliadas neste trabalho.

A última observação como trabalho futuro é planejar e conduzir um experimento com diagramas de casos de uso e atividades. Esse experimento não foi realizado neste trabalho

por causa da inexistência de LPS acadêmicas com tais diagramas e a falta de tempo hábil para desenvolver o diagrama e em seguida desenvolver o material e aplicar o experimento.

REFERÊNCIAS

- ALMEIDA, E. S. *Software reuse and product line engineering* Cham: Springer International Publishing, p. 321–348, 2019a.
- ALMEIDA, E. S. *Software reuse and product line engineering* Cham: Springer International Publishing, p. 321–348, 2019b.
- ANQUETIL, N.; KULESZA, U.; MITSCHKE, R.; MOREIRA, A.; ROYER, J.-C.; RUMMLER, A.; SOUSA, A. A model-driven traceability framework for software product lines. *Software and Systems Modeling*, v. 9, p. 427–451, 2010.
- BACHMANN, F.; CLEMENTS, P. C. *Variability in software product lines*. Relatório Técnico, CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 2005.
- BASIL, V. R. The role of experimentation in software engineering: past, current, and future. In: *Proceedings of IEEE 18th International Conference on Software Engineering*, 1996, p. 442–449.
- BONIFÁCIO, R.; BORBA, P. Modeling scenario variability as crosscutting mechanisms. In: *Proceedings of the 8th ACM International Conference on Aspect-oriented Software Development*, New York, NY, USA: ACM, 2009, p. 125–136.
- BONIFÁCIO, R.; TEIXEIRA, L.; BORBA, P. Hephaestus: A tool for managing spl variabilities. In: *Sessão de ferramentas, Terceiro Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software (SBCARS 2009)*, 2009, p. 1–9.
- BOSCH, J. Design and use of software architectures: Adopting and evolving a product-line approach. 2000.
- BOSCH, J.; FLORIJN, G.; GREEFHORST, D.; KUUSELA, J.; OBBINK, J. H.; POHL, K. Variability issues in software product lines. In: *4th International Workshop on Software Product-Family Engineering*, Berlin, Heidelberg: Springer-Verlag, 2002, p. 13–21.

- CAPILLA, R.; BOSCH, J.; KANG, K.-C. *Systems and software variability management: Concepts, tools and experiences*. Springer Publishing Company, Incorporated, 2013.
- CATAL, C. Barriers to the adoption of software product line engineering. v. 34, n. 6, p. 1–4, 2009.
- CAVALCANTI, Y.; MACHADO, I.; NETO, P.; LOBATO, L. *Handling variability and traceability over spl disciplines* 2012.
- CHEN, L.; ALI BABAR, M. A systematic review of evaluation of variability management approaches in software product lines. *Inf. Softw. Technol.*, v. 53, n. 4, p. 344–362, 2011.
- CHEN, L.; ALI BABAR, M.; ALI, N. Variability management in software product lines: A systematic review. 2009, p. 81–90.
- CLEMENTS, P.; NORTHROP, L. *Software product lines: Practices and patterns*. Addison-Wesley Professional, 2001.
- DE ALMEIDA BIOLCHINI, J. C.; MIAN, P. G.; NATALI, A. C. C.; CONTE, T. U.; TRAVASSOS, G. H. Scientific research ontology to support systematic review in software engineering. *Advanced Engineering Informatics*, v. 21, n. 2, p. 133 – 151, 2007.
- ECHEVERRÍA, J.; FONT, J.; CETINA, C.; PASTOR, O. Usability evaluation of variability modeling by means of common variability language. 2015, p. 1–8.
- FALESSI, D.; JURISTO, N.; WOHLIN, C.; TURHAN, B.; MÜNCH, J.; JEDLITSCHKA, A.; OIVO, M. Empirical software engineering experts on the use of students and professionals in experiments. *Empirical Software Engineering*, v. 1, n. 23, p. 452–489, 2017.
- FIORI, D.; GIMENES, I.; MALDONADO, J.; OLIVEIRAJR, E. Variability management in software product line activity diagrams. In: *International Conference on Distributed Multimedia Systems*, 2012, p. 89–94.
- FURTADO, V. R. *Diretrizes para avaliação de qualidade de quase-experimentos e experimentos controlados em linha de produto de software*. Dissertação de Mestrado, Programa de Pós-Graduação em Ciência da Computação - Universidade Estadual de Maringá, 2018.
- GALSTER, M.; WEYNS, D.; TOFAN, D.; MICHALIK, B.; AVGERIOU, P. Variability in software systems—a systematic literature review. *IEEE Transactions on Software Engineering*, v. 40, n. 3, p. 282–306, 2014.

- GLASS, G. V. Primary, secondary, and meta-analysis of research. *Educational Researcher*, v. 5, n. 10, p. 3–8, 1976.
- GOMAA, H. *Designing software product lines with UML: From use cases to pattern-based software architectures*. 2005.
- VAN GURP, J.; BOSCH, J.; SVAHNBERG, M. Managing variability in software product lines, p. 1–13. 2002.
- HALMANS, G.; POHL, K. Communicating the variability of a software-product family to customers. *Software and Systems Modeling*, v. 2, p. 15–36, 2003.
- JURISTO, N.; MORENO, A. M. *Basics of software engineering experimentation*. 1st ed. Springer Publishing Company, Incorporated, 2010.
- VAN DER LINDEN, F. J.; SCHMID, K.; ROMMES, E. *Software product lines in action: The best industrial practice in product line engineering*. 1st ed. Springer Publishing Company, Incorporated, 2007.
- LINDGREN, E.; MÜNCH, J. Raising the odds of success: the current state of experimentation in product development. *Information and Software Technology*, v. 77, p. 80 – 91, 2016.
- MARCOLINO, A.; OLIVEIRAJR, E. Avaliação experimental da abordagem smarty para gerenciamento de variabilidades em linhas de produto de software baseadas em UML. *SBQS: Proceedings of the 14th Brazilian Symposium on Software Quality*, v. 1, p. 339–353, 2015.
- MARCOLINO, A.; OLIVEIRAJR, E. Comparing smarty and plus for variability identification and representation at product-line UML class level: A controlled quasi-experiment. *Journal of Computer Science*, v. 13, p. 617–632, 2017.
- MARCOLINO, A.; OLIVEIRAJR, E.; GIMENES, I. Variability identification and representation in software product line UML sequence diagrams: Proposal and empirical study. In: *2014 Brazilian Symposium on Software Engineering*, 2014a, p. 141–150.
- MARCOLINO, A.; OLIVEIRAJR, E.; GIMENES, I.; BARBOSA, E. F. Empirically based evolution of a variability management approach at UML class level. In: *2014 IEEE 38th Annual Computer Software and Applications Conference*, 2014b, p. 354–363.

- MARCOLINO, A.; OLIVEIRAJR, E.; GIMENES, I.; MALDONADO, J. Towards the effectiveness of a variability management approach at use case level. 2013.
- MARCOLINO, A.; OLIVEIRAJR, E.; M.S. GIMENES, I.; BARBOSA, E. Variability resolution and product configuration with smarty: An experimental study on UML class diagrams. *Journal of Computer Science*, v. 13, p. 307–319, 2017.
- MARCOLINO, A. S. *Avaliação experimental da abordagem smarty para gerenciamento de variabilidade em linha de produto de software*. Dissertação de Mestrado, Programa de Pós-Graduação em Ciência da Computação - Universidade Estadual de Maringá, 2014.
- MCGREGOR, J. D.; NORTHROP, L. M.; JARRAD, S.; POHL, K. Initiating software product lines. *IEEE Software*, v. 19, n. 4, p. 24–27, 2002.
- MILLER, J. Applying meta-analytical procedures to software engineering experiments. *Journal of Systems and Software*, v. 54, n. 1, p. 29 – 39, 2000.
- MOON, M.; CHAE, H. S.; NAM, T.; YEOM, K. A metamodeling approach to tracing variability between requirements and architecture in software product lines. In: *7th IEEE International Conference on Computer and Information Technology (CIT 2007)*, 2007, p. 927–933.
- MUNEZERO, M.; YAMAN, S.; FAGERHOLM, F.; KETTUNEN, P.; MÄENPÄÄ, H.; MÄKINEN, S.; TIIHONEN, J.; RIUNGU-KALLIOSAARI, L.; TUOVINEN, A.; OIVO, M.; MÜNCH, J.; MÄNNISTÖ, T. *Continuous experimentation cookbook: An introduction to systematic experimentation for software-intensive businesses*. 2017.
- NEPOMUCENO, T.; OLIVEIRAJR, E. Configuring software product line specific products with smarty and plus: An experimental study on use case diagrams. 2018, p. 81–90.
- NGUYEN, Q. L. Non-functional requirements analysis modeling for software product lines. In: *2009 ICSE Workshop on Modeling in Software Engineering*, 2009, p. 56–61.
- OLIVEIRAJR, E.; GIMENES, I.; MALDONADO, J. Systematic management of variability in UML-based software product lines. *Journal of Universal Computer Science*, v. 16, p. 2374–2393, 2010.
- POHL, K.; BÖCKLE, G.; VAN DER LINDEN, F. J. *Software product line engineering: Foundations, principles, and techniques*. Berlin: Springer, 2005.

RAZAVIAN, M.; KHOSRAVI, R. Modeling variability in the component and connector view of architecture using UML. In: *2008 IEEE/ACS International Conference on Computer Systems and Applications*, 2008, p. 801–809.

RILLING, J.; CHARLAND, P.; WITTE, R. *Traceability in software engineering – past, present and future*. Relatório Técnico TR-74-211, IBM Technical Report, CASCON 2007 Workshop, 2007.

Disponível em http://www.ibm.com/ibm/cas/publications/cascon2007reports/cascon07_TR-74-211.pdf

RYU, D.; LEE, D.; BAIK, J. Designing an architecture of sns platform by applying a product line engineering approach. In: *2012 IEEE/ACIS 11th International Conference on Computer and Information Science*, 2012, p. 559–564.

SÁNCHEZ, P.; LOUGHRAN, N.; FUENTES, L.; GARCIA, A. Engineering languages for specifying product-derivation processes in software product lines. In: GAŠEVIĆ, D.; LÄMMEL, R.; VAN WYK, E., eds. *Software Language Engineering*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, p. 188–207.

SCHMID, K.; JOHN, I. A customizable approach to full lifecycle variability management. *Science of Computer Programming*, v. 53, n. 3, p. 259 – 284, 2004.

SILVA, L. F.; OLIVEIRAJR, E.; ZORZO, A. F. Feasibility analysis of smartymodeling for modeling uml-based software product lines. In: *Proceedings of the 22nd International Conference on Enterprise Information Systems*, INSTICC, SciTePress, 2020, p. 442–449.

SPEARMAN, C. The proof and measurement of association between two things. *The American Journal of Psychology*, v. 100, n. 3/4, p. 441–471, 1987.

Disponível em <http://www.jstor.org/stable/1422689>

TAWHID, R.; PETRIU, D. Towards automatic derivation of a product performance model from a UML software product line model. In: *WOSP'08: Proceedings of the 7th International Workshop on Software and Performance 2008*, 2008, p. 91–102.

TICHY, W. F. Should computer scientists experiment more? *Computer*, v. 31, n. 5, p. 32–40, 1998.

TRAVASSOS, G. *Introdução à engenharia de software experimental*. 2002.

Disponível em <https://books.google.com.br/books?id=4SnKZwEACAAJ>

VALE, T.; DE ALMEIDA, E. S.; ALVES, V.; KULESZA, U.; NIU, N.; DE LIMA, R. Software product lines traceability: A systematic mapping study. *Information and Software Technology*, v. 84, p. 1–18, 2017.

WEISS, D. M.; LAI, C. T. R. *Software product-line engineering: a family-based software development process*. 1999.

WOHLIN, C.; RUNESON, P.; HST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLN, A. *Experimentation in software engineering*. Springer Publishing Company, Incorporated, 2012.

YOUNG, T. J. *Using aspectj to build a software product line for mobile devices*. Tese de Doutorado, University of British Columbia, 2005.

Disponível em <https://open.library.ubc.ca/collections/ubctheses/831/items/1.0051632>

ZELKOWITZ, M. V.; WALLACE, D. Experimental validation in software engineering. *Information and Software Technology*, v. 39, n. 11, p. 735 – 743, 1997.

ZIADI, T.; HÉLOUËT, L.; JÉZÉQUEL, J.-M. Towards a UML profile for software product lines. 2003, p. 129–139.

ZIADI, T.; JEZEQUEL, J.-M. *Software product line engineering with the UML: Deriving products* Berlin, Heidelberg: Springer Berlin Heidelberg, p. 557–588, 2006.

Apêndice A: Avaliação Experimental das abordagens de Gerenciamento de Variabilidades: um Mapeamento Sistemático

Para a condução das avaliações experimentais de abordagens de gerenciamento de variabilidades baseadas em UML, um levantamento bibliográfico foi conduzido, inicialmente, para obter estudos primários que pudessem auxiliar no projeto e execução de tais conduções.

Dentre os estudos levantados, um mapeamento sistemático realizado por Marcolino (Marcolino e OliveiraJr, 2015), foi identificado com o mesmo assunto de interesse. Dessa maneira, um mapeamento foi conduzido em setembro de 2018 para recuperar estudos que corroborassem com o resultado anterior e possibilitassem o refinamento e ajustes dos estudos conduzidos atualizando o mapeamento sistemático já existente.

Os resultados do levantamento bibliográfico realizado apresentaram trabalhos que utilizam técnicas experimentais, em avaliações de abordagens de gerenciamento de variabilidades baseadas em UML em linhas de produtos de software.

Assim esse mapeamento serviu para colaborar nos novos estudos que avaliarão a efetividade dessas abordagens, no que tange a identificação e representação de variabilidades em modelos UML.

A.1 Introdução

Na Engenharia de LPS, o gerenciamento de variabilidades é uma das atividades mais importantes. Abordagens de gerenciamento de variabilidades são fundamentais para a configuração correta de novos produtos em uma LPS. Segundo Chen, Babar e Ali (2009) e Galster et al. (2013), a falta de avaliações experimentais da efetividade das abordagens de gerenciamento de variabilidades impossibilita a sua adoção por indústrias e na academia.

Esse mapeamento sistemático tem como objetivo identificar o estado da arte das avaliações experimentais conduzidas para abordagens de gerenciamento de variabilidades baseadas em UML, para que possam contribuir com os estudos empíricos para avaliar a efetividade de tais abordagens e também para servir de referencial de comparação entre os resultados. Por se tratar de uma atualização de um mapeamento sistemático já existente, os parâmetros de pesquisa como as questões de pesquisa, parâmetros de inclusão e exclusão, bases de busca e strings serão os mesmos que definidos por Marcolino (2014).

Esse documento está estruturado da seguinte forma: a Seção B.2 define a especificação do mapeamento sistemático, na Seção B.3 são definidos os processos de seleção dos estudos e suas etapas, na Seção B.4 são apresentadas a aplicação das definições para o mapeamento, aplicados para recuperação dos estudos, na Seção B.5 são apresentados os estudos recuperados e a meta-análise das informações pertinentes ao objetivo deste mapeamento sistemático e na Seção B.6 são apresentadas as conclusões.

A.2 Especificação do Mapeamento Sistemático

Este mapeamento sistemático tem como objetivo obter estudos empíricos que validem abordagens de gerenciamento de variabilidades baseadas em UML para contribuir com os estudos empíricos que serão conduzidos para avaliar a efetividade de tais abordagens.

A.2.1 Questões de Pesquisa

O propósito deste trabalho é identificar e caracterizar o que há de relevante na literatura que apresente avaliações empíricas para abordagens de gerenciamento de variabilidades baseadas em UML, assim, a questão de pesquisa para este mapeamento é: “Quais estudos apresentam avaliações empíricas para abordagens gerenciamento de variabilidades baseadas em UML?”.

Essa questão tem duas dimensões de pesquisa: A primeira está relacionada a identificação dos procedimentos experimentais planejados e executados, para refinamento e utilização em novas avaliações empíricas e a segunda dimensão refere-se a identificação dos resultados das avaliações conduzidas, para que possam ser comparadas com os resultados empíricos de outras avaliações.

A.2.2 Refinamento das Questões de Pesquisa

Com base na questão de pesquisa, três subquestões foram derivadas para guiar este estudo:

- Q1. Quais estudos apresentam validações empíricas para abordagens de gerenciamento de variabilidade baseadas em UML?
- Q2. Quais estudos apresentam o processo experimental de suas validações para abordagens de gerenciamento de variabilidade baseadas em UML?
- Q3. Quais estudos apresentam resultados semelhantes à identificação da efetividade das abordagens de gerenciamento de variabilidades baseadas em UML, no que tange a representação e identificação de variabilidades em tais modelos?

A.2.3 Critérios de Inclusão e Exclusão

Como em Marcolino (2014), foram adotados os seguintes critérios de inclusão e exclusão para seleção dos estudos neste mapeamento, com base nas questões de busca.

Critérios de Inclusão: Os critérios para a inclusão de estudos primários definidos para cada uma das questões de pesquisa do mapeamento foram:

- Critério de Inclusão 1: estudos que apresentam validações empíricas para abordagens de gerenciamento de variabilidade baseadas em UML;
- Critério de Inclusão 2: estudos que apresentam o processo experimental de suas validações para abordagens de gerenciamento de variabilidade baseadas em UML; e
- Critério de Inclusão 3: estudos que apresentam resultados semelhantes à identificação da efetividade das abordagens de gerenciamento de variabilidades baseadas em UML, no que tange a representação e identificação de variabilidades em tais modelos.

Critérios de Exclusão: foram adotados os seguintes critérios para exclusão:

- Critério de Exclusão 1: os estudos que não apresentam validações empíricas para abordagens de gerenciamento de variabilidade baseadas em UML;
- Critério de Exclusão 2: os estudos que não apresentam o processo experimental de suas validações para abordagens de gerenciamento de variabilidade baseadas em UML; e
- Critério de Exclusão 3: os estudos que não apresentam resultados semelhantes à identificação da efetividade das abordagens de gerenciamento de variabilidades baseadas em UML, no que tange a representação e identificação de variabilidades em tais modelos.

Além dos critérios acima, consideram-se também, como critérios de exclusão, independentemente das questões de pesquisa, os que seguem:

- Os estudos recuperados por meio eletrônico em formatos que não sejam PDF;
- (Portable Document Format), DOC/DOCX (Processador de Texto MicrosoftWord) ou ODT (Processador de Texto do Open Office);
- Estudos duplicados, encontrados previamente em outra(s) fonte(s);
- Estudos indisponíveis, que não puderam ser recuperados; e
- Estudos em outros idiomas, que não sejam o Inglês.

A.2.4 Esquema de Classificação

Visando a obtenção de resultados concisos, de acordo com questões de busca definidos neste mapeamento, definiu-se a estratégia para busca de estudos primários, baseadas nas fontes de busca, idiomas dos estudos e palavras-chave relacionadas.

As fontes de busca referem-se a locais e pessoas de onde se originam os estudos, escritos em idioma específico e localizados por meio de palavras-chave definidas de forma precisa.

- Fontes: base de dados eletrônicas ACM, Compendex, IEEE, ScienceDirect e Scopus, disponíveis nas URLs apresentadas na Tabela 1;
- Idioma dos Estudos: inglês, uma vez que é o idioma internacionalmente aceito em redações de estudos científicos;

- Tipos de Documentos: artigos de conferências, anais de congressos, artigos de periódicos, livros/capítulos de livros, capítulos de teses e dissertação e relatórios técnicos, em formato PDF, ODT ou DOC/DOCX;
- Ano de Publicação: estudos que foram publicados a partir do ano de 2014;
- Palavras-chave: “experimental validation”, “product line” e “UML”, com os seguintes termos e variações:
 - Experimental validation: “empirical validation”;
 - Product line: “product line”, “variability”, “variability management” e “variability approach”; e
 - UML: “unified modeling language”.

Tabela 1.1: Bases de busca

Bases de Busca	
Bases	Endereço
ACM	http://dl.acm.org
Compendex	http://ieeexplore.ieee.org
IEEE	http://www.engineeringvillage.org
ScienceDirect	http://www.sciencedirect.com
Scopus	http://www.scopus.com

A.3 Seleção dos Estudos

Nesta seção são apresentados os processos de seleção preliminar, visando refinar a quantidade de estudos recuperados, para assim ser realizado o processo de seleção final, que selecionará os mais importantes.

A.3.1 Processo de Seleção Preliminar

O processo de seleção preliminar se dará por meio da construção de strings de busca formatada com a combinação das palavras chave apresentadas na Seção A.2.4. Essas strings, apresentadas no formato exigido por cada uma das bases, na Tabela 2 serão aplicadas nas máquinas de busca e nas bases de dados indexadas, apresentadas na Tabela 1.

Após a recuperação dos mesmos, a leitura do resumo de tais estudos será feita, com a finalidade de identificar os relevantes para sua leitura na íntegra, considerando os critérios de inclusão e exclusão apresentados, para a filtragem dos mais relevantes.

Tabela 1.2: Strings de Busca

Bases	Strings de Busca
ACM	("experimental validation" OR "empirical validation") AND ("product line" OR "variabiity" OR "variability management" OR "variability approach") AND ("UML" OR "unified modeling language")
Compendex	((("experimental validation" OR "empirical validation" OR "evaluation") AND ("product line" OR "variabiity" OR "variability management" OR "variability approach") AND ("UML" OR "unified modeling language"))
IEEE	((("experimental validation" OR "empirical validation") AND ("product line" OR "variabiity" OR "variability management" OR "variability approach") AND ("UML" OR "unified modeling language"))
ScienceDirect	((("experimental validation" OR "empirical validation") AND ("product line" OR "variabiity" OR "variability management" OR "variability approach") AND ("UML" OR "unified modeling language"))
Scopus	(TITLE-ABS-KEY(("experimental" OR "empirical" OR "validation" OR "evaluation") AND TITLE-ABS-KEY(("product line" OR "variabiity" OR "variability management" OR "variability approach") AND TITLE-ABS-KEY(("UML" OR "unified modeling language")))) AND PUBYEAR>2014

A.3.2 Processo de Seleção Final

A seleção final avaliará a qualidade dos estudos e sua relevância, por meio de uma leitura completa dos estudos pré-selecionados, almejando considerar ou descartar resultados apresentados por cada um, garantindo assim, para a próxima etapa, a redação de uma análise crítica ressaltando a validação experimental de abordagens de gerenciamento de variabilidade baseadas em UML.

A.3.3 Extração dos Resultados

No processo de seleção final dos estudos, serão coletados alguns dados relevantes. Tais dados, a serem extraídos, são:

- Título: título do trabalho
- Autores: nome dos autores e co-autores do trabalho.
- Ano de Publicação: ano de publicação do trabalho;
- Publicação: origem da publicação;
- Fonte: onde o trabalho foi obtido;
- Quantidade de Páginas;
- Palavras-chave;
- Resumo; e
- Visão geral do trabalho: Análise sucinta do trabalho apresentado os pontos de destaque.

Após a leitura completa dos estudos selecionados, será redigido um relatório para sintetizar as análises críticas com relação ao tema da revisão, sendo este composto de sínteses dissertativas, bem como gráficos e tabelas ilustrativas.

A.4 Condução da Revisão

O mapeamento sistemático foi conduzido no mês 09/2018. Nesta seção são apresentadas as atividades realizadas na condução da mesma e especifica-se a estratégia utilizada na construção e aplicação das strings de busca, bem como os resultados das buscas efetuadas. Podemos dividir a condução do mapeamento sistemático, nas etapas que seguem:

- Definição das strings de busca;
- Aplicação das strings de busca e obtenção dos estudos nas fontes especificadas;
- Seleção preliminar dos estudos; e
- Seleção final dos estudos.

A.4.1 Definição das Strings de Busca

Na definição das strings de busca, combinou-se os sinônimos e as palavras-chave apresentadas na Seção A.2.4. Para tal combinação, foram utilizados os operadores lógicos “e”(AND) e “ou”(OR) como segue:

(“experimental validation”OR “empirical validation”) AND (“product line”OR “variability”OR “variability management”OR “variability approach”) AND (“UML”OR “unified modeling language”)

A variação destas para cada base de busca foi apresenta na Tabela 2.

A.4.2 Aplicação das Strings de Busca e Obtenção dos Estudos nas Fontes Especificadas

Com a aplicação das strings de busca nas suas respectivas bases foram recuperados um total de 80 estudos. Estes estão divididos em 4 bases de busca, já que na base ACM não foi obtido nenhum resultado e são apresentados na Tabela 3. A Figura 1 representa a distribuição dos estudos por base.

Tabela 1.3: Quantidade de Resultados por Base de Busca

Bases	Resultados	
ACM	0	0%
Compendex	18	22,5%
IEEE	2	2,5%
ScienceDirect	17	21,25%
Scopus	43	53,75%
Total	80	100%

A.4.3 Seleção Preliminar dos Estudos

Após a aplicação das strings de busca, os estudos recuperados, tiveram os seus títulos e resumos lidos e avaliados mediante os critérios de inclusão e seleção. Destes, 4 estudos foram selecionados para o processo de seleção final.

A.5 Estudos Selecionados

A Tabela 1.4 apresenta informações quanto aos estudos selecionados e lidos na íntegra.

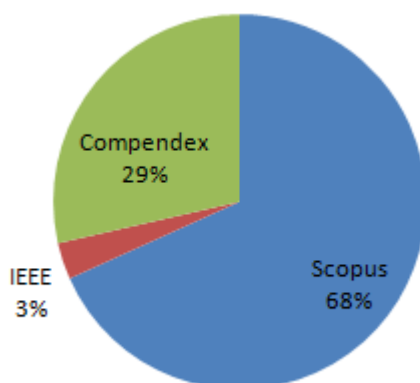


Figura 1.1: Distribuição de Estudos por Base

Tabela 1.4: Estudos Seleccionados

ID	Título	Autor(es)	Ano	Base	Publicado em
1	Variability Identification and Representation in Software Product Line UML Sequence Diagrams: Proposal and Empirical Study	Marcolino, OliveiraJr, Gimenes	2014	Compendex	Brazilian Symposium on Software Engineering
2	Towards the Effectiveness of the SMarty approach for variability management at Sequence diagram level	Marcolino, OliveiraJr, Gimenes	2014	IEEE	Internacional Conference on Enterprise Information Systems
3	Variability Resolution and Product Configuration with SMarty: An experimental study on UML diagrams	Marcolino et al.	2017	Scopus	Journal of Computer Science
4	Comparing SMarty and PLUS for variability identification and representation at product line UML class level: A Controlled Quase-Experiment	Marcolino, OliveiraJr	2017	Scopus	Journal of Computer Science

Além destes, outros dois estudos que não apareceram nas buscas foram considerados por se tratar do tema de interesse deste mapeamento sistemático e são apresentados na Tabela 1.5.

Tabela 1.5: Estudos Adicionados

ID	Título	Autor(es)	Ano	Publicado em
5	Towards the effectiveness of a variability management approach at use case level	Marcolino et al.	2013	Proceeding of the International Conferences on Software Engineering and Knowledge Engineering
6	Empiracally based evolution of a variability management approach at UML class level	Marcolino et al.	2014	Conference on Computers, Software and Applications

A.5.1 Síntese dos Estudos Selecionados

1 - (Marcolino, OliveiraJr, Gimenes, 2014): compara a abordagem SMarty com a Ziadi et al. em diagramas de sequência. As LPS utilizadas foram a Arcade Game Maker (AGM) e Banking. Neste estudo houve evidência de que a abordagem SMarty era mais efetiva para identificação e representação de variabilidades.

2 - (Marcolino, OliveiraJr, Gimenes, 2014): Compara as abordagens Ziadi et al. e SMarty, com o propósito de identificar a mais efetiva, em relação a capacidade de identificação e representação de variabilidades em Linha de Produto de Software em diagramas de sequência. As LPS utilizadas para tal avaliação foram Banking e AGM. A efetividade das duas abordagens foi próxima.

3 - (Marcolino et al., 2017): o estudo relata um experimento para comparar a efetividade da abordagem SMarty e do método PLUS no que se refere a resolução de variabilidades e configuração de produtos em diagramas de classes UML usando as LPS e-commerce e AGM. Neste experimento não houve diferença estatística entre as amostras de efetividade em relação à capacidade de interpretação e configuração de produtos corretos em diagramas de classes UML com o método PLUS e a abordagem SMarty.

4 - (Marcolino, OliveiraJr, 2017): compara o método PLUS e a abordagem SMarty para identificar qual é a mais efetiva em relação a capacidade de identificação e resolução de variabilidades em LPS usando diagramas de classes. As LPS usadas para o experimento foram a AGM e e-commerce. Como resultado, houve indícios de que o método PLUS é mais efetivo do que a abordagem SMarty.

5 - (Marcolino et al., 2013): no estudo uma avaliação foi conduzida para comparar o método PLUS e a abordagem SMarty em diagramas de casos de uso, utilizando a LPS e-commerce. Neste experimento, ficou evidenciado que SMarty era mais efetiva para identificação e resolução de variabilidades do que PLUS.

6 - (Marcolino et al., 2014): compara a abordagem SMarty e o método PLUS em dois experimentos com diagramas de classes utilizando as LPS AGM e e-commerce. A partir dos resultados houve indícios que o método PLUS era mais efetivo que a abordagem SMarty em diagramas de classes UML com as duas LPS observadas.

A.6 Conclusão

Com a condução deste mapeamento sistemático pode-se perceber que a avaliação empírica de abordagens de gerenciamento de variabilidades ainda não são comuns. Mesmo assim, os poucos resultados obtidos estão intimamente ligados com as questões de pesquisa deste mapeamento sistemático e ajudarão na condução de futuros experimentos para avaliar a efetividade das abordagens de gerenciamento de variabilidades baseadas em UML, bem como na comparação dos resultados.

Apêndice B: Instrumentação dos Experimentos

B.1 Materiais do Experimento #1

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)

Mestranda / Pesquisadora: Thais Santos Nepomuceno	Grupo de Pesquisa: GReater - Grupo de Pesquisa em Reuso Sistemático de Software e Experimentação Contínua.
Participante:	
Email para Contato:	Data:

Prezado(a) Senhor(a),

O grupo de pesquisa GReater eventualmente realiza estudos experimentais para caracterizar/avaliar uma determinada tecnologia de software. Estes estudos são conduzidos por alunos de Pós-graduação em Ciência da Computação (PCC) do Departamento de Informática (DIN) da Universidade Estadual de Maringá (UEM), sob a orientação do Prof. Edson Alves de Oliveira Júnior. Você foi previamente selecionado pelo seu perfil/conhecimento/experiência e está sendo convidado a participar desta pesquisa. Essa pesquisa será feita com base em dados coletados a partir de trabalhos práticos. Embora o trabalho prático faça parte da disciplina, você tem o direito de não permitir a utilização dos dados do seu trabalho na pesquisa.

1. Procedimentos

O estudo será realizado com data e hora marcada com os participantes pré-selecionados. Para participar do estudo normalmente será aplicado um formulário de caracterização de perfil, a fim de identificar seu nível de conhecimento/experiência. Em seguida, o estudo será executado de forma individual ou em grupos formados, seguindo sempre o planejamento do estudo feito pela pesquisadora responsável. Caso seja necessário, ao final do estudo será solicitado ao participante que responda um questionário de avaliação sobre a tecnologia de software que está sendo caracterizada/avaliada.

2. Tratamento de possíveis riscos e desconfortos

Serão tomadas todas as providências durante a coleta de dados de forma a garantir a sua privacidade e seu anonimato. Além disso, não existem riscos ou desconfortos que poderão afetar o participante durante a condução do estudo. Ex: fadiga, estresse, mal estar, dentre outros.

3. Benefícios e Custos

Espera-se que, como resultado deste estudo, você possa aumentar seus conhecimentos, de maneira a contribuir para o aumento da qualidade das atividades com as quais você trabalhe ou possa vir a trabalhar. Este estudo também contribuirá com resultados importantes para a pesquisa de um modo geral para o grupo de pesquisa GRSSE. Você não terá nenhum gasto ou ônus com a sua participação no estudo e também não receberá qualquer espécie de reembolso ou gratificação devido à autorização dos seus dados na pesquisa.

4. Confidencialidade da Pesquisa

Toda informação coletada neste estudo é confidencial e seu nome não será identificado de modo algum, a não ser em caso de autorização explícita para este fim. Quando os dados forem coletados, seu nome será removido dos mesmos e não será utilizado em nenhum momento durante a análise ou apresentação dos resultados.

5. Participação

Sua participação neste estudo é muito importante e voluntária, pois requer a sua aprovação para utilização dos dados coletados neste estudo. Segundo a Resolução 466/12 do Conselho Nacional de Saúde (CNS), o respeito devido à dignidade humana exige que toda pesquisa se processe após assentimento livre e esclarecido. Você tem o direito de não querer participar ou de sair deste estudo a qualquer momento, sem penalidades. Em caso de você decidir se retirar do estudo, favor notificar a pesquisadora responsável. Os pesquisadores responsáveis pelo estudo poderão fornecer qualquer esclarecimento sobre o mesmo, assim como tirar dúvidas.

Coordenador do Grupo de Pesquisa GReater: Prof. Edson Alves de Oliveira Júnior - edson@din.uem.br

Pesquisadora do Grupo de Pesquisa GReater: Mestranda Thais Santos Nepomuceno - thais.nepomuceno1@gmail.com

6. Declaração de Consentimento

Declaro que li e estou de acordo com as informações contidas neste documento e que toda linguagem técnica utilizada na descrição deste estudo de pesquisa foi explicada satisfatoriamente, recebendo respostas para todas as minhas dúvidas. Confirmando também que recebi uma cópia deste Termo (TCLE), compreendo que sou livre para não autorizar a utilização dos meus dados neste estudo em qualquer momento, sem qualquer penalidade. Declaro ter mais de 18 anos e concordo de espontânea vontade em participar deste estudo.

Obrigado pela sua colaboração!

Nome:

Questionário de Caracterização de Participante em Estudo Experimental

“Avaliação experimental de uma abordagem para gerenciamento de variabilidades em linhas de produto de software baseadas em UML”

Nas perguntas a seguir, quando duas ou mais alternativas forem válidas, marque a alternativa que mais se aplica ao seu caso.

1. Qual o seu nível de formação?

- | | |
|-------------------------------------|-----------------------------------|
| <input type="checkbox"/> Graduando | <input type="checkbox"/> Graduado |
| <input type="checkbox"/> Mestrando | <input type="checkbox"/> Mestre |
| <input type="checkbox"/> Doutorando | <input type="checkbox"/> Doutor |

2. Em qual setor atua?

- | | |
|---|---|
| <input type="checkbox"/> Acadêmico (ensino) | <input type="checkbox"/> Industrial (empresarial) |
|---|---|

3. Qual o nome da empresa/universidade que atua?

.....

4. Quanto tempo possui de experiência na área que atua?

.....

5. Qual a sua experiência com a notação UML com relação aos diagramas de classes e sequencia?

- Eu **nunca** modelei um software usando a UML.
- Minha experiência com a notação UML é básica.**
Eu modelo software somente no nível dos elementos mais comuns da UML como classes e herança; objetos;
- Minha experiência com a notação UML é moderada.**
Eu modelo software no nível dos elementos da opção anterior, além de: relacionamentos de dependência *include* e *extend*, e *extension points* em diagramas de casos de uso; polimorfismo, associação (uni e bi-direcionais), dependência, agregação e composição em classes; realizações e dependência em componentes, troca de mensagens em diagramas de sequência.

[] **Minha experiência com a notação UML é avançada.**

Eu modelo software que exige a utilização de todos os elementos de diagramas de casos de uso, classes, componentes e sequência, além de outros diagramas da UML como, por exemplo, diagramas de colaboração e atividades.

6. Qual a sua experiência com relação à abordagem de Linha de Produto de Software (LPS) e Gerenciamento de Variabilidade?

[] Eu **nunca** ouvi falar a respeito de LPS.

[] **Já lí**, de forma superficial, algo a respeito de LPS.

[] **Minha experiência com LPS é básica.**

Eu conheço os seguintes conceitos da abordagem: ciclo de desenvolvimento de LPS e suas atividades (engenharia de domínio e engenharia de aplicação). Porém, **não tenho experiência com gerenciamento de variabilidades.**

[] **Minha experiência com LPS é moderada.**

Eu conheço os conceitos da opção anterior, e com relação ao gerenciamento de variabilidades, eu sei o conceito de pontos de variação, variantes e os seus relacionamentos, além dos conceitos de resolução de variabilidades e tempos de resolução (*design time, link time, runtime*, entre outros).

[] **Minha experiência com LPS é avançada.**

Eu conheço os conceitos da opção anterior, além de alguns processos existentes de desenvolvimento de LPS (FODA, PLP, PLUS, PuLSE, entre outros). Com relação ao gerenciamento de variabilidades, eu sei os conceitos da opção anterior, além de: modelos de resolução; abordagens existentes para o gerenciamento de variabilidades, e representação de variabilidades (usando a UML, modelos de características, entre outras).

Assinatura do Participante	Local e Data
<hr/>	Cidade e Data

I. Linha de Produto de Software

Uma linha de produto de software (LPS) corresponde a um conjunto de sistemas de *software* que compartilham características (*features*) comuns e gerenciáveis, que satisfazem a necessidade de um segmento particular ou de uma missão. Este conjunto de sistemas é denominado também família de produtos. Os membros da família são produtos específicos desenvolvidos de maneira sistemática a partir da instanciação de uma infraestrutura comum de uma LPS, chamada núcleo de artefatos.

O núcleo de artefatos é formado por um conjunto de características comuns (similaridades) e características variáveis (variabilidades). As variabilidades podem estar associadas a diferentes níveis de abstração, como a descrição da arquitetura, o código fonte, etc., e auxiliam na geração de produtos específicos distintos em um mesmo domínio e, dessa forma, diminuem o custo e o tempo de desenvolvimento, reduzem riscos e perdas, além de reduzirem o *time to market* e justificarem o retorno de investimento (ROI).

O gerenciamento de variabilidades é uma das atividades mais importantes no gerenciamento de uma LPS, tendo sido foco de atenção por diversos pesquisadores, com pode ser percebido pelas diversas abordagens presentes na literatura.

As variabilidades podem estar associadas a diferentes níveis de abstração, dentre eles, a descrição da arquitetura, a documentação de projeto, o código fonte, o código compilado, o código ligado e o código executável.

Em síntese **variabilidade é a forma como os membros de uma família de produtos podem se diferenciar entre si, ou seja, é o que permite distinguir os diversos produtos de uma LPS.**

A variabilidade é descrita por pontos de variação e variantes:

- **Ponto de variação:** Um **local específico** de um artefato em que uma decisão de projeto ainda não foi tomada, ou seja, foi adiada;
- **Variante:** Corresponde a **uma alternativa** de projeto para resolver uma determinada variabilidade.
- **Restrições entre variantes:** define os relacionamentos entre duas ou mais variantes para que seja possível resolver um ponto de variação ou uma variabilidade.

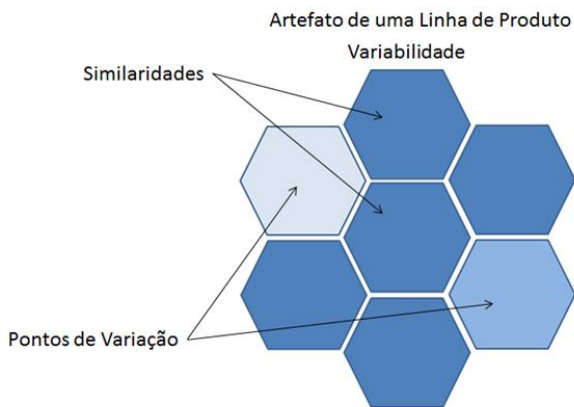
A aplicação destes conceitos é ilustrada na Figura 1.

A maioria das abordagens desenvolvidas para auxiliar no gerenciamento de variabilidades envolve diversos conceitos e modelos de representação. As abordagens “X” e “Y” apresentadas nos documentos 3.2 e 3.3, respectivamente, possuem como base a UML – *Unified Modeling Language*. Tais abordagens utilizam diversos conceitos da UML como o uso de estereótipos e diversos diagramas.

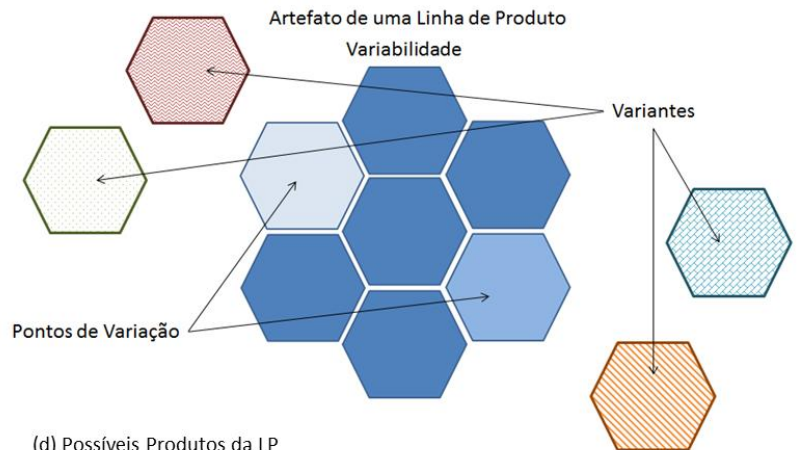
II. Abordagens para Linha de Produto de Software

Para a representação de linhas de produto de software e o gerenciamento de suas variabilidades, como mencionado anteriormente, existem diversas abordagens presentes na literatura. Entre os itens que estas abordagens contemplam, temos os que seguem na Tabela I, que serão posteriormente apresentados para cada abordagem.

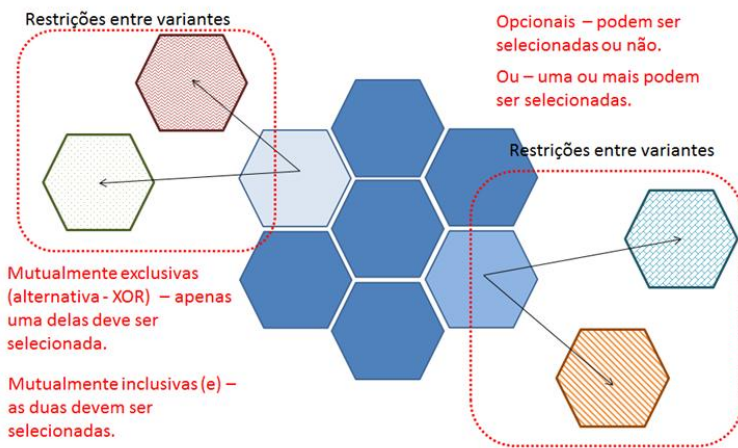
(a) Variabilidade



(b) Pontos de Variação e Variantes



(c) Variantes e suas Restrições



(d) Possíveis Produtos da LP

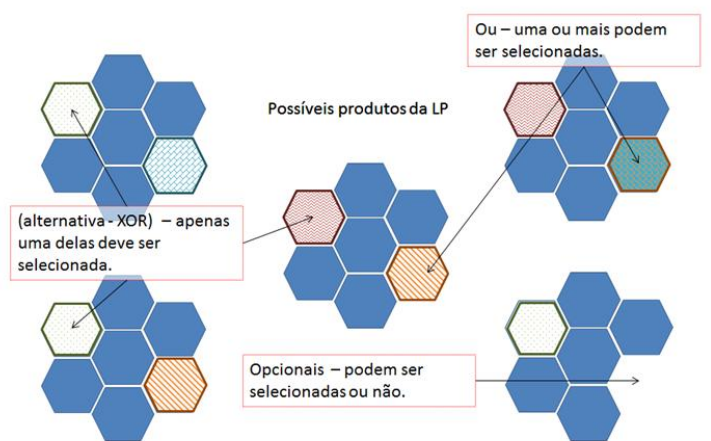


Figura 1 – Exemplo dos Conceitos de Variabilidade, Pontos de Variação, Variantes e Restrições entre Variantes.

Item	Identificação
Baseada em UML	Indica que a abordagem utiliza os modelos UML, meta-atributos, etc., como forma de representação da LPS e de suas variabilidades.
Perfil	Diversas abordagens apresentam um perfil específico que é formado por estereótipos e meta-atributos, geralmente derivados de uma linguagem de modelagem, como a UML.
Processo	O processo contempla a sistematização da utilização de um perfil para o gerenciamento de variabilidades, guiando o usuário no uso das definições do perfil utilizado.
Estereótipos	Estereótipos, como os da UML, são um padrão de mecanismo de extensão e são usados para distinguir diferentes tipos de elementos modelados. Em LPS são ferramentas úteis para identificar variabilidade, seus pontos de variação,

Conceitos Essenciais sobre Gerenciamento de Variabilidade em Linha de Produto de Software

	variantes e outros itens necessários ao seu gerenciamento.
Diretrizes	São os passos sistematizados definidos no processo, que permitem a aplicação facilitada do perfil da abordagem a que corresponde.

III. Rastreabilidade de Variabilidades

Uma LPS é projetada para atender todos os requisitos dos produtos de sua família de produtos. Esses requisitos e a própria LPS podem sofrer mudanças ao longo do tempo em decorrência de vários fatores, como por exemplo, evolução nos requisitos dos produtos, evolução no domínio de mercado para o qual os produtos foram projetados, evolução no processo de construção da LPS, evolução das tecnologias usadas para desenvolver os produtos. Para lidar com essas mudanças, a LPS deve ser modificada e evoluir, visando não ficar obsoleta, e se adequar aos novos requisitos que surgem.

A análise de impacto de mudanças consiste em uma atividade que visa compreender e identificar quais consequências essas mudanças causam ou causarão na LPS. A análise de impacto na LPS pode ser apoiada por meio de relações de rastreabilidade, as quais identificam relacionamentos entre artefatos criados durante todas as fases do desenvolvimento de software.

I. Visão Geral Abordagem X

Tabela I – Visão Geral Abordagem X.

Abordagem X			
Item	Sim	Não	Observação
Baseada em UML?	X		
Perfil?		X	
Processo?		X	
Estereótipos?	X		Estereótipos específicos, possuindo variações entre modelos.
Diretrizes?		X	Especificação de uso da abordagem através representação textual.

II. Estereótipos e Diretrizes

Nesta seção são apresentados os estereótipos para aplicação em casos de uso e classes por meio da Tabela II, em seguida exemplos do uso destes são apresentados, seguidos por especificações textuais do seu uso, identificando de forma conceitual as diretrizes.

Tabela II – Estereótipos da Abordagem X para Casos de Uso e Classes.

Estereótipo	Definição	Suporta modelos de	
		Casos de uso?	Classes?
<<kernel>>	Usado para representar elementos obrigatórios.	Sim	Sim
<<optional>>	Usado para representar elementos opcionais que podem ser selecionados ou não, para um produto específico.	Sim	Sim
<<alternative>>	Usado para representar elementos alternativos, mutuamente exclusivos ou inclusivos.	Sim	Não

II.1 Exemplos

Casos de Uso

Em linhas de produto é necessário descrever variações que ocorrem para diferentes membros. Uma forma de integrar variabilidade em casos de uso para LP é utilizando pontos de variação. Um **ponto de variação** é um local em um caso de uso, onde uma decisão pode ser tomada.

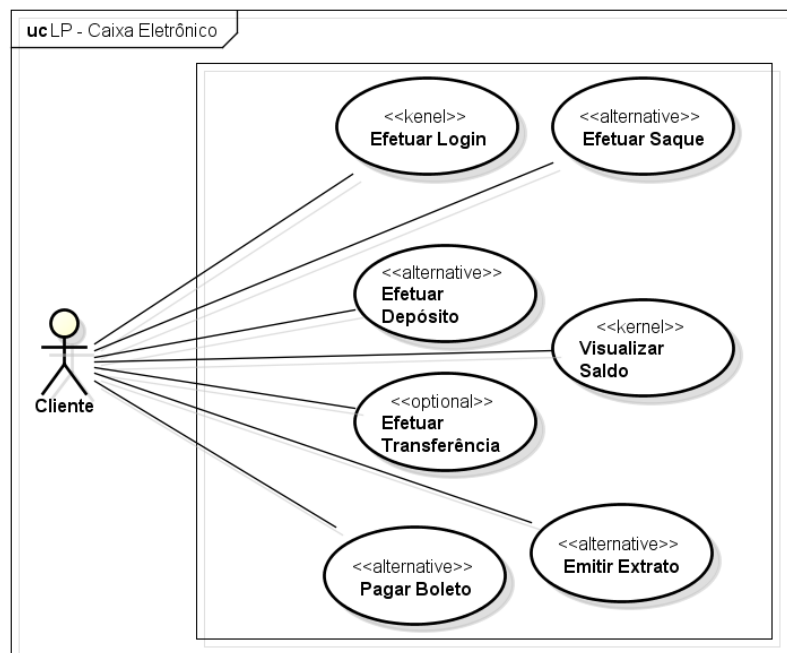


Figura 1 – Exemplo de Diagrama de Casos de Usos com a Abordagem X

Pontos de variação em casos de uso podem ser tratados de duas formas:

- Em casos mais simples, o ponto de variação é descrito no próprio caso de uso, identificando o local no caso de uso, onde a decisão ocorre, e podendo ser, inclusive a própria variante; e
- Em casos de uso mais complexos utilizam-se os conceitos de *extende include* para modelar pontos de variação.

Na Figura 1 podemos observar o uso dos estereótipos presentes na Tabela II. Para os casos de uso obrigatórios, **Efetuar Login** e **Visualizar Saldo** há a aplicação do estereótipo `<<kernel>>`, logo, esta variabilidade, e ponto de variação, é obrigatória e estará presente em todos os produtos dela derivadas. Já o caso de uso **Efetuar Pagamento** é opcional (`<<optional>>`), podendo ou não ser inseridos como funcionalidade do caixa. Os casos de uso **Efetuar Saque**, **Efetuar Depósito**, **Pagar Boleto** e **Emitir Extrato** são alternativos (`<<alternative>>`), ou seja, um caixa com a funcionalidade de efetuar saque, não pode, devido a uma regra da LP, estar presente em um mesmo produto que tenha a funcionalidade de efetuar depósito, sendo assim, são mutuamente exclusivos.

As regras aplicadas a esta LP são meramente demonstrativas, principalmente no que tange a opção alternativa para o exemplo acima.

Ainda quanto a opção `<<alternative>>` pode ser necessário formar um grupo de variantes assim estereotipadas, atendendo as restrições e regras de negócio aplicadas na LP. Desta forma, casos de uso podem ser inseridos para permitir tal agrupamento, como pode ser visto na Figura 2.

Na Figura 2, notamos que os casos de uso **Efetuar Saque**, **Efetuar Depósito**, **Pagar Boleto** e **Emitir Extrato**, anteriormente (Figura 1) representados como alternativos, passam a ser obrigatórios, e estão agrupados em um diagrama de caso de uso com o estereótipo `<<alternative>>`. Esta indicação mostra que, devido uma restrição de regras de negócio da LP, definidas anteriormente pelo gerente de LP, indica a necessidade de que produtos desta LP sejam produzidos com o grupo apresentado pelo caso de uso **Efetuar Transações Básicas** ou pelo grupo **Efetuar Transações Avançadas**.

Observe que, por ambos os grupos possuírem o estereótipo <<alternative>>, suas variantes passam a receber o estereótipo <<kernel>>. Assim, se selecionado ou um ou outro (relação mutuamente exclusiva – ou seja, se um for selecionado, o outro não será) as variantes estendidas ali farão parte obrigatória do produto.

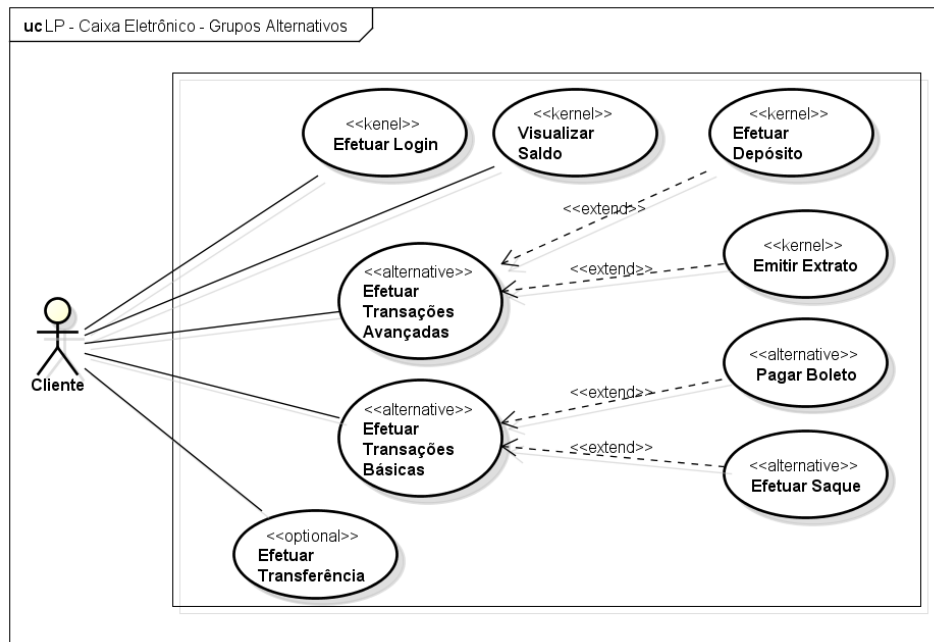


Figura 2 – Exemplo de Diagrama de Casos de Usos com a Abordagem X e Grupos de Variantes Alternativas.

Relações de Extensão (Extend)

A relação *extend*, em casos de uso para sistemas únicos são utilizados para estender um caso de uso base em diferentes caminhos, de acordo com a verificação de certas circunstâncias.

No exemplo apresentado na Figura 3, temos uma situação na qual o ponto de extensão **pagamento** é opcionalmente resolvido pelos pontos de extensão **Pagar com Cartão de Crédito** e/ou **Pagar com Cartão de Débito** e, obrigatoriamente, o **Pagar com Dinheiro**. Portanto, um produto desta linha deverá aceitar pagamento em dinheiro, de forma obrigatória (<<kernel>>) e poderá ter ambas as opções: pagamento com cartão de crédito e débito, ou somente uma delas (<<optional>>).

Desta forma, pontos de extensão podem ser usados em modelos de variabilidade de LP para apresentar variabilidades alternativas, variabilidades opcionais e para evolução futura (Ex.: Aceitar pagamento através de dinheiro eletrônico, no exemplo apresentado na Figura 6.).

Relações de Inclusão (Include)

Estas sequências comuns de interação refletem uma funcionalidade que é comum a mais de um caso de uso. Sequências de interação comuns pode ser extraídas de diversos casos de uso originais e transformadas em novos casos de uso, chamados casos de uso de inclusão

Em linhas de produto, as relações de *include* servem para representar casos de uso opcionais, ou obrigatórios. No primeiro caso, só será utilizado na LP, se a condição determinada para seu uso for verdadeira.

Na Figura 4, temos um exemplo de uso da relação *include*, onde dois casos de uso são opcionais,

Depositar Valor e **Visualizar Saldo** (*<<optional>>*), ou seja, podem ou não estarem presentes nos membros da LP; e ainda, três casos de uso obrigatórios (*<<kernel>>*).

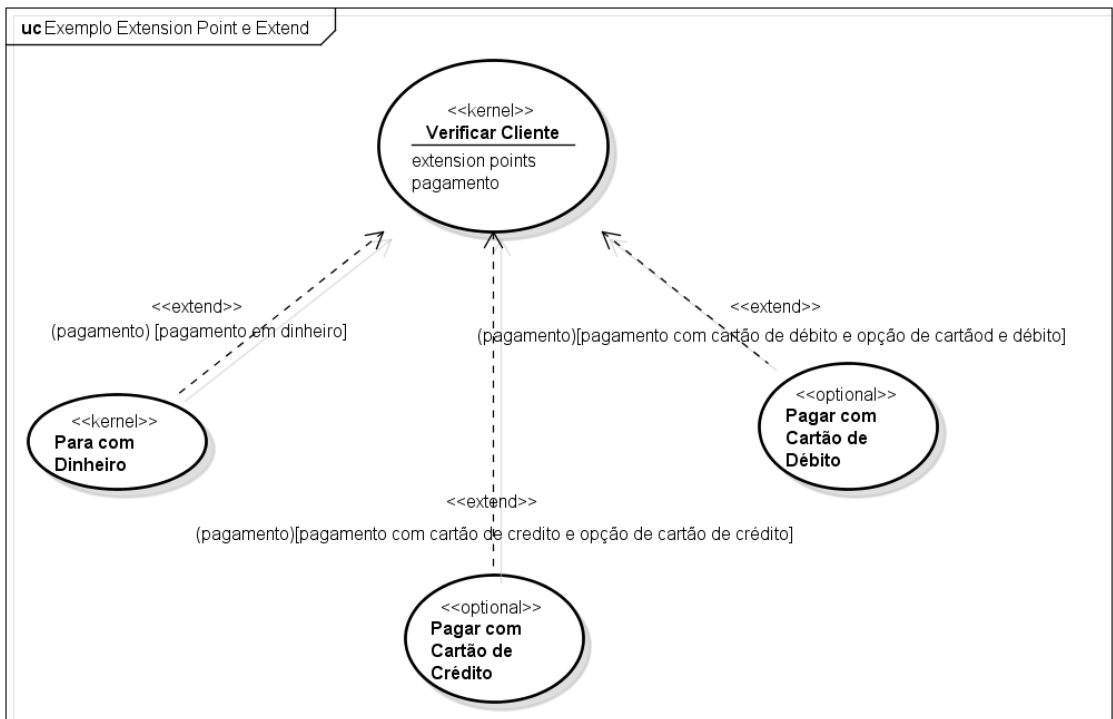


Figura 3 – Diagrama de Casos de Usos com Exemplo de Relação *Extend* a um *Extension Point*.

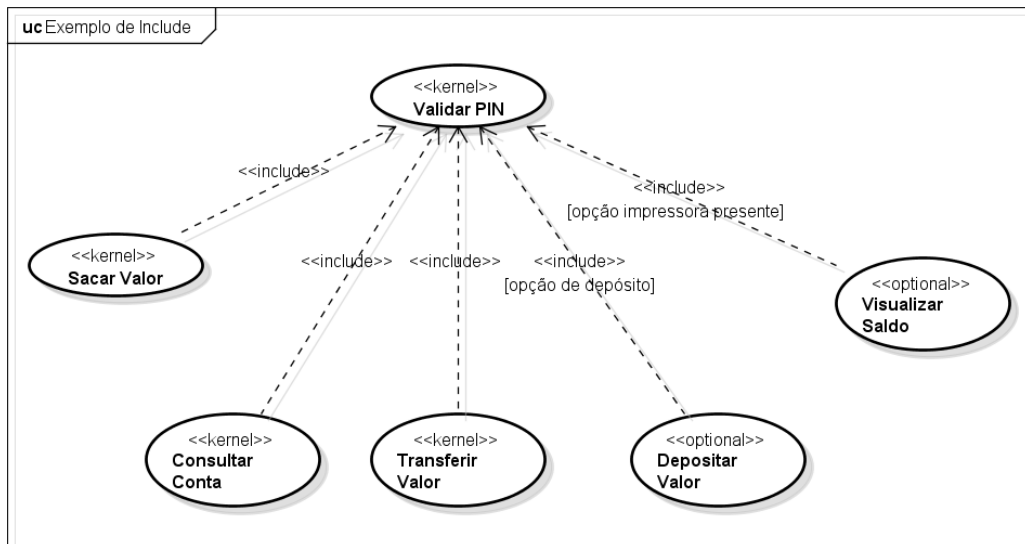


Figura 4 – Diagrama de Casos de Usos com Exemplo de Relação *Include*

Classes

Para a identificação da variabilidades e similaridades por meio da Abordagem X aplica-se um dos estereótipos apresentados na Tabela II, para cada classe existente no modelo e de acordo com a especificação da Linha de Produto a ser modelada.

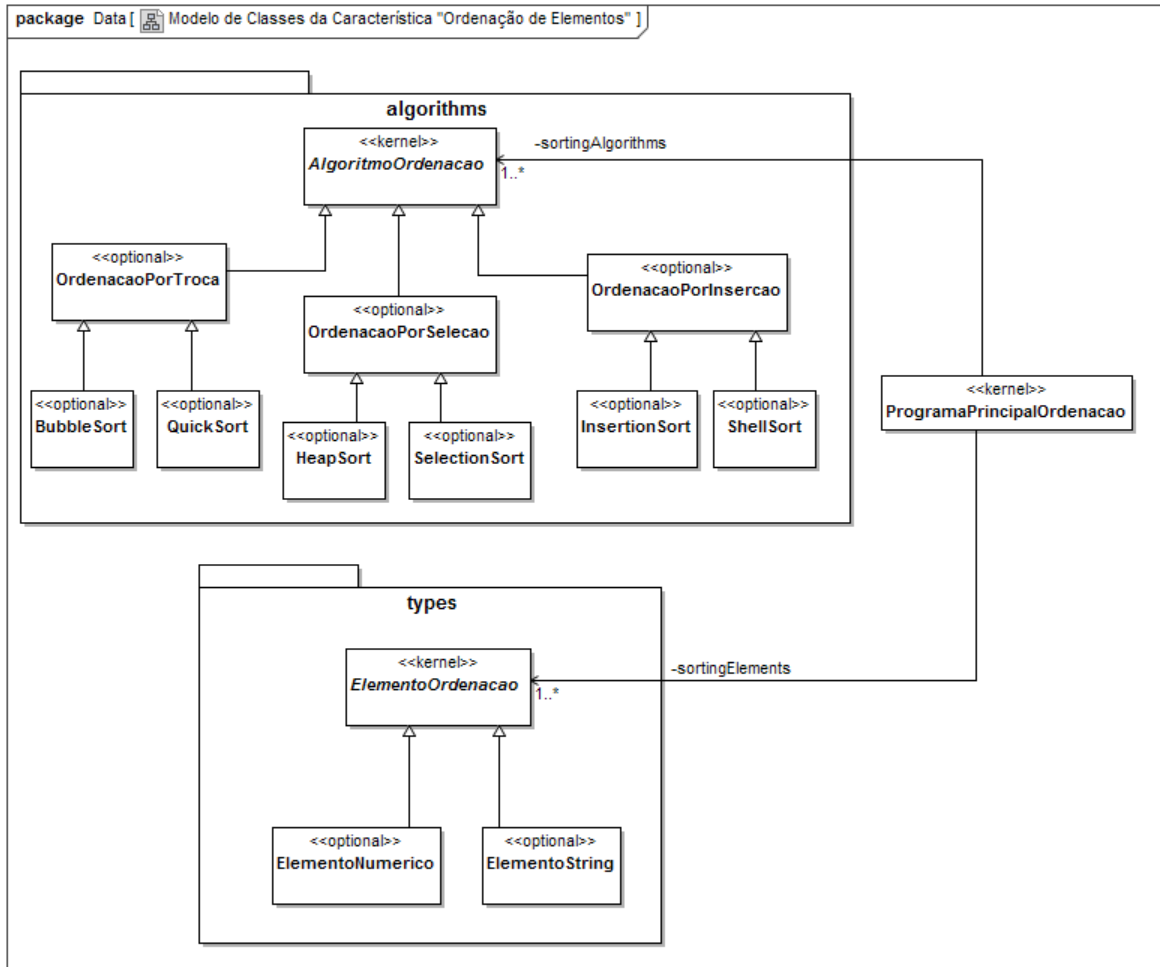


Figura 5 – Exemplo de Modelo de Variabilidade em Diagrama de Classes com a Abordagem X.

Para o exemplo de linha de produto de "Ordenação de Elementos" (Figura 5), as classes obrigatórias, ou seja, as que são similares para todos os produtos gerados a partir de tal LP são estereotipadas como <<kernel>>, como a classe **ProgramaPrincipalOrdenacao**. Para as classes que são alternativas, podendo ser selecionadas se mutuamente exclusivas, mutuamente inclusivas e também opcionais ocorre à aplicação do estereótipo <<optional>>, como para a classe **ElementoOrdenacao**.

Abordagem Y

I. Visão Geral Abordagem Y

Tabela I – Visão Geral Abordagem Y

Abordagem Y			
Item	Sim	Não	Observação
Baseada em UML?	X		
Perfil?	X		
Processo?	X		
Estereótipos?	X		Estereótipos específicos padrões para todos os modelos.
Diretrizes?	X		Diretrizes específicas para cada modelo.

II. Estereótipos e Diretrizes

Nesta seção são apresentados os estereótipos para aplicação em casos de uso, existentes no perfil da abordagem Y por meio da Tabela II.

Tabela II – Estereótipos da Abordagem Y para Casos de Uso.

Estereótipos Abordagem Y para casos de uso	
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 (em UML) entre variantes no qual a variante dependente (origem da dependência) só existirá em uma configuração se a variante relacionada (destino da dependência) existir.
<<mutex>>	Indica um relacionamento de dependência (em UML) entre variantes no qual a variante dependente (origem da dependência) só existirá em uma configuração se a variante relacionada (destino da dependência) obrigatoriamente não existir. São conhecidas como variantes mutuamente exclusivas.

II.1 Notação de Variabilidade

As variabilidades são identificadas através de notas UML, estereotipada com <<variability>>. Nestas

notas estão contidos os meta-atributos que seguem:

- **Name**: nome da variabilidade;
- **minSelection**: a quantidade mínima de variantes a serem selecionadas;
- **maxSelection**: a quantidade máxima de variantes a serem selecionadas;
- **bindingTime**: em qual momento será resolvida esta variabilidade ;
- **allowsAddingVar**: se permite incluir novas variantes para resolver o ponto de variação; e
- **variants**: quais as variantes para resolver o ponto de variação (casos de uso ligados ao ponto de variação).

• **realizes +**: representa a coleção de variabilidades de nível superior (menos abstrato) no qual é possível realizar a variabilidade;

• **realizes -**: representa a coleção de variabilidades de nível inferior (mais abstrato) no qual é possível realizar a variabilidade.

Estas notas são inseridas em todas as variabilidades em caso de uso e classes.

II.1 Exemplos

Casos de Uso

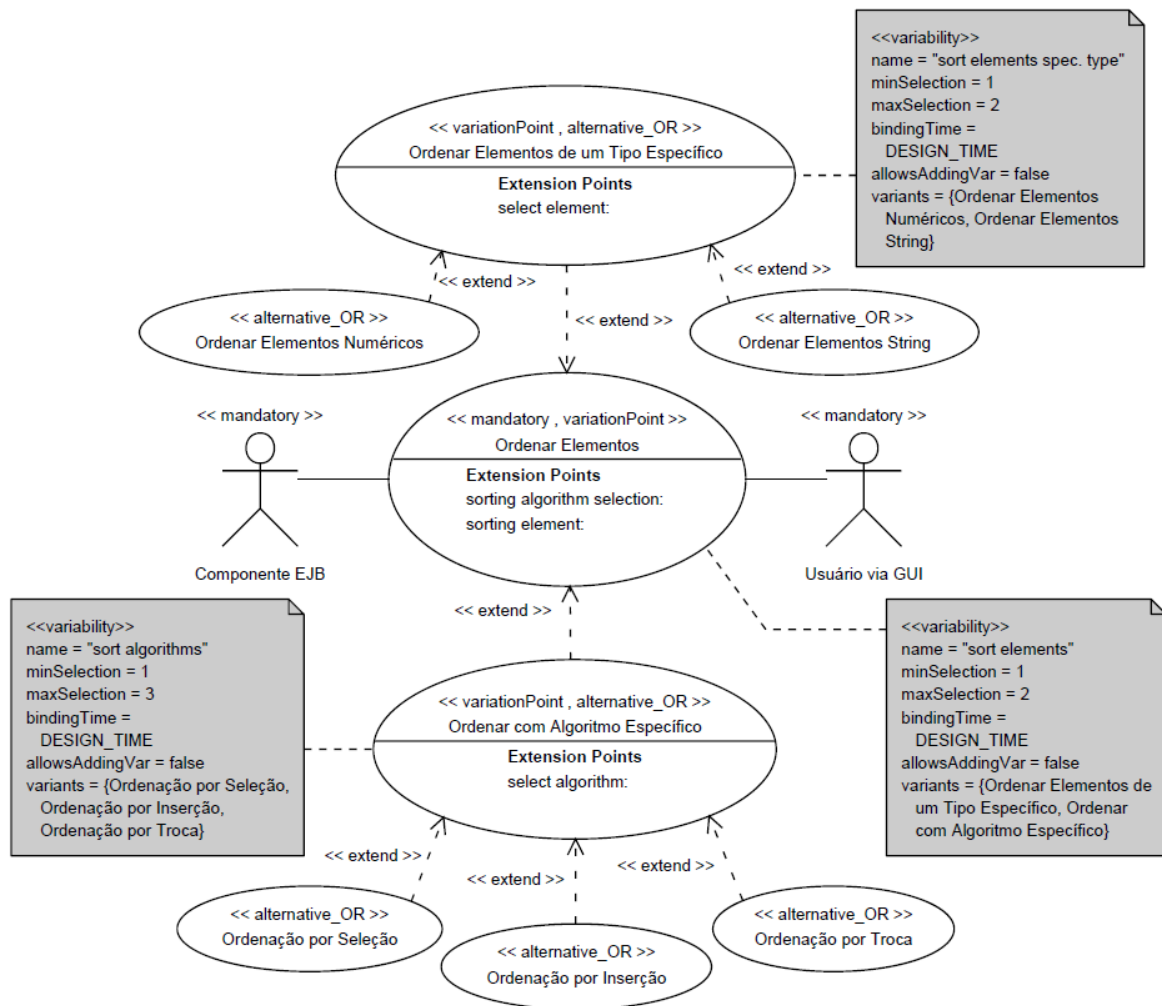


Figura 1 – Exemplo de Diagrama de Casos de Uso com a Abordagem Y.

Na Figura 1 observamos a aplicação da abordagem Y, e seus elementos. Passamos a analisar cada um deles, bem como as diretrizes presentes no processo da abordagem Y, que auxiliam sua utilização em outras LPs:

O caso de uso **Ordenar Elementos** é obrigatório `<<mandatory>>` e também é um ponto de variação `<<variationPoint>>`. Temos também dois casos de uso `<<alternative_OR>>` que são variantes do ponto de variação **Ordenar Elementos**. O que é facilmente notado por meio do item *variants* da nota da variabilidade de nome “*sort elements*”.

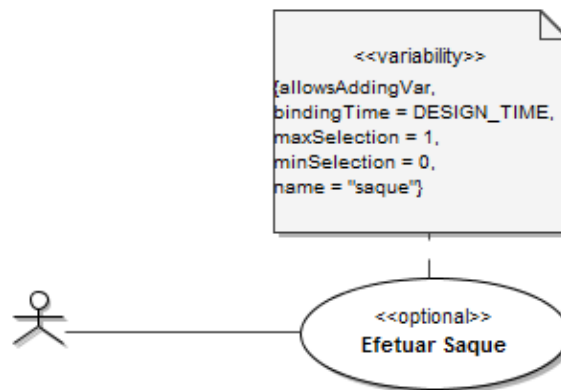


Figura 2 – Exemplo de Identificação de Variabilidade em Casos de Uso

Na Figura 2, que representa fragmento de um caso de uso, notamos que a variabilidade pode ser definida em um caso de uso único, como é o caso da variabilidade chamada “*saque*” identificada pelo comentário da UML estereotipado com `<<variability>>`.

Classes

Na Figura 3 observamos a aplicação da abordagem Y, e seus elementos. Passamos a analisar cada um deles, bem como as diretrizes presentes no processo da abordagem Y, que auxiliam sua utilização em outras LPs:

A classe **AlgoritmoOrdenacao** identifica uma classe obrigatória (`<<mandatory>>`) e representa também um ponto de variação (`<<variationPoint>>`), com três variantes. Estas variantes estão descritas no elemento comentário, relacionado a classe, por meio do *TaggedValue*(**variants**). As três variantes desta classe são **OrdenacaoPorTroca**, **OrdenacaoPorSelecao** e **OrdenacaoPorInsercao**. Todas estas são estereotipadas como `<<alternative_OR>>`, o que indica o tipo de restrição para tais variantes, neste caso, significa que ao menos uma ou todas elas podem solucionar o ponto de variação.

OrdenacaoPorTroca, **OrdenacaoPorSelecao** e **OrdenacaoPorInsercao**, além de variantes, são, por sua vez, pontos de variação (`<<variationPoint>>`), e assim, cada uma delas apresenta um comentário, que descreve as suas variantes (**variants**), bem como o nome da mesma (**name**). Neste caso, todas as variantes são marcadas como `<<alternative_OR>>` e, como anteriormente, uma delas, ao menos, deve ser selecionada ou todas.

A classe **ProgramaPrincipalOrdenacao**, representa uma classe obrigatória, portanto é marcada como `<<mandatory>>`, e estará presente em todos os produtos desta LP.

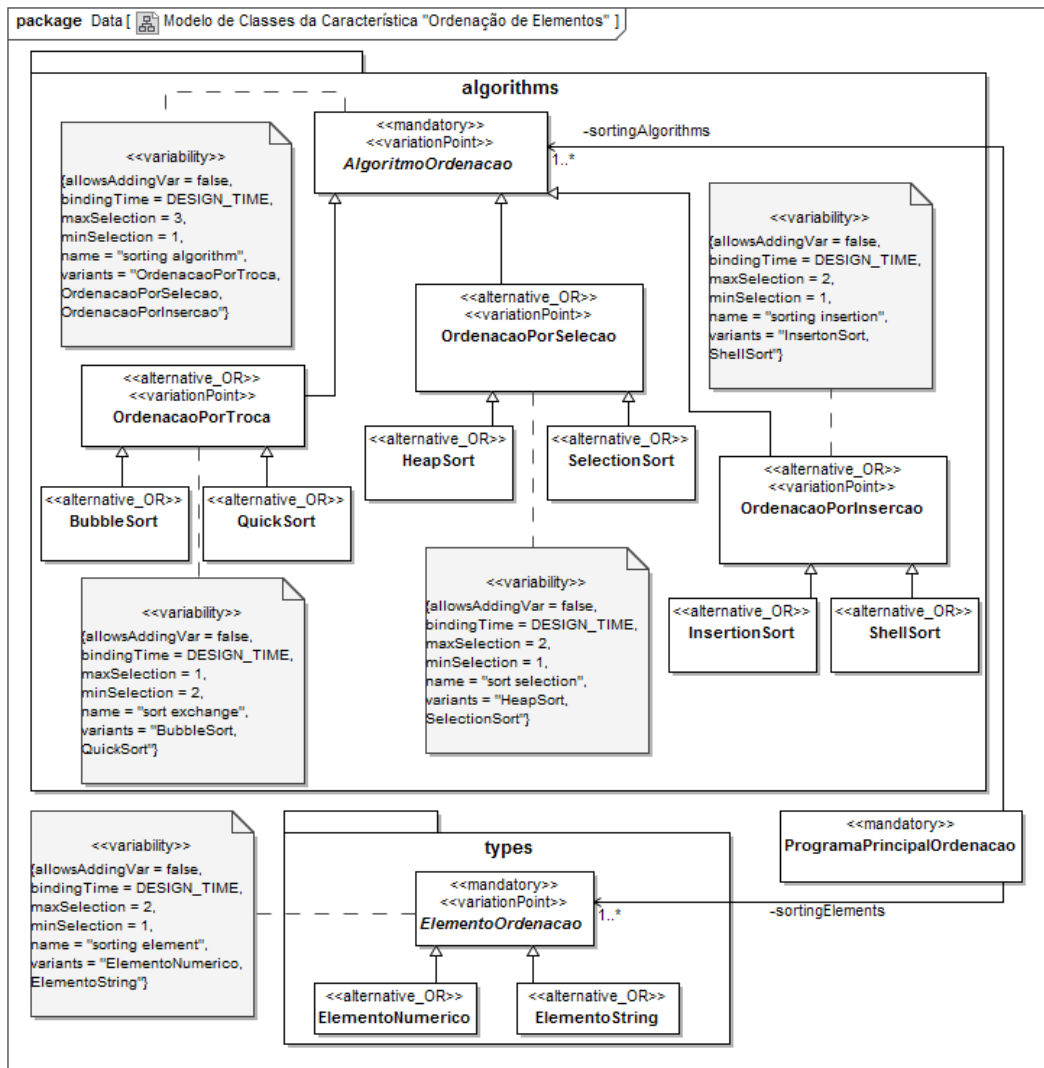


Figura 3 – Exemplo de Modelo de Variabilidade em Diagrama de Classes com a Abordagem Y.

A classe **ElementoOrdenacao**, também é obrigatória (*<<mandatory>>*) e representa um ponto de variação (*<<variationPoint>>*), logo possui o elemento comentário ligado a ela, com o estereótipo *<<variability>>*, que identifica os dados da variabilidade, que é nomeada, por exemplo, de “sortingelement” e possui duas classes variantes (**varinats**): **ElementoNumerico** e **ElementoString**, marcadas como variantes alternativas *<<alternative_OR>>*, onde, ambas podem ser selecionadas, ou ao menos uma.

Desta forma, as variabilidades são identificadas por meio do comentário UML, estereotipada com *<<variability>>*.

I. Identificação

Mobile Media (YOUNG, 2005) é uma LPS composta por aplicações (produtos) que manipulam músicas, vídeos e fotos para dispositivos móveis, como celulares e *palm tops*. Ela provê suporte para gerenciar (criar, excluir, visualizar, executar, enviar) diferentes tipos de mídia.

A Mobile Media surgiu da extensão de uma LPS já existente denominada Mobile Photo (YOUNG, 2005), por meio da inserção de novas propriedades multimídia, como manipulação de vídeos e músicas, que somente podem ser realizados em alguns tipos de aparelhos. De certa forma, pode-se dizer que a inserção das características opcionais e alternativas a determinados aparelhos caracterizou o surgimento da Mobile Media.

II. Similaridades e Variabilidades

Nesta seção são apresentadas as similaridades da LP, ou seja, os aspectos comuns a todos os produtos desta LP, bem como as variabilidades, que representam os aspectos que diferem de um produto em relação ao outro.

II.1 Similaridades

- Gerenciar Álbum: Permite ao usuário definir novos álbuns.
- Executar Mídia: Permite ao usuário assistir vídeo, ouvir música e/ou visualizar fotos.
- Adicionar Mídias no Álbum: Permite ao usuário adicionar novas fotos a álbuns definidos.
- Gerenciar Mídia: Permite ao usuário gerenciar vídeos, músicas e/ou fotos.
- Log in: o usuário deve fazer login antes de gerenciar mídia.

II.2 Variabilidades

- Enviar Foto via SMS: Permite a um usuário enviar uma foto para outro via *Short Messaging Service*.
- Enviar Foto via Email: Permite a um usuário enviar uma foto para outro via Email.
- Relacionar Foto com Registro na Agenda: Permite ao usuário associar um registro na sua lista de contatos com a foto do álbum.
- Mostrar Foto nas Chamadas Recebidas: Intercepta chamadas recebidas e mostra a foto associada ao contato.
- Tocar Melodia nas Chamadas Recebidas: Intercepta chamadas recebidas e toca uma melodia personalizada para aquele contato.

Os itens a seguir apresentam as características básicas e as características variáveis da Mobile Photo, agora denominada Mobile Media:

1. Criar Album de Fotos : Permite ao usuário definir novos álbuns de fotos para armazenar categorias de fotos no dispositivo. A persistência da informação do álbum é realizada utilizando RMS (J2ME Record Management System).

2. **Armazenar Foto:** Gerenciar a conversão e persistência dos arquivos de foto para o sistema de arquivos do dispositivo utilizando RMS.
3. **Adicionar/Deletar Foto:** Permite ao usuário excluir fotos permanentemente do dispositivo ou adicionar novas fotos a álbuns definidos.
4. **Rotular Foto:** Permite ao usuário determinar um texto para uma foto. Os rótulos aparecerão na lista de exibição e podem ser utilizados para uma futura funcionalidade relacionada à busca.
5. **Visualizar Foto:** Mostra uma foto selecionada na tela do dispositivo.
6. **Enviar Foto via SMS:** Permite a um usuário enviar uma foto para outro via Short Messaging Service.
7. **Enviar Foto via Email:** Permite a um usuário enviar uma foto para outro via Email.
8. **Relacionar Foto com Registro na Agenda:** Permite ao usuário associar um registro na sua lista de contatos com a foto do álbum.
9. **Mostrar Foto nas Chamadas Recebidas:** Intercepta chamadas recebidas e mostra a foto associada ao contato.
10. **Tocar Melodia nas Chamadas Recebidas:** Intercepta chamadas recebidas e toca uma melodia personalizada para aquele contato.

**Mobile Media:
Modelo de Identificação de Variabilidades (Abordagem X)**

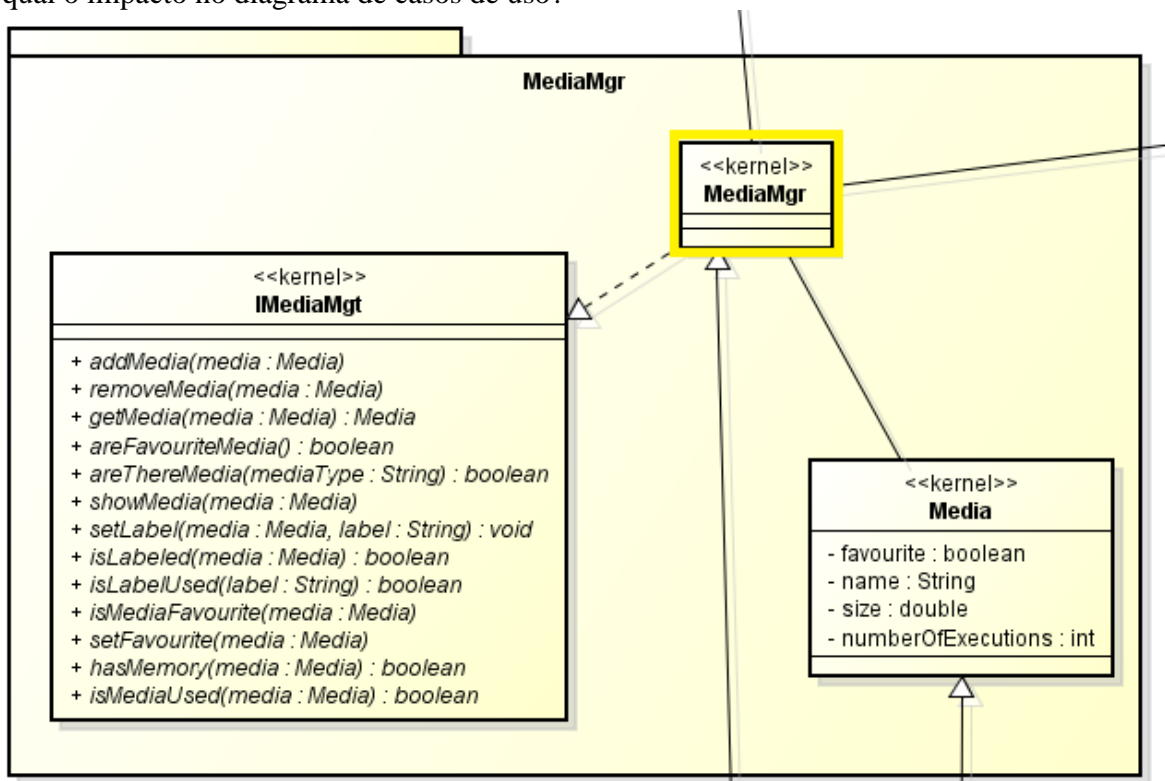
I. Configuração de Produtos

I.1 Com base nos Diagrama de Casos de Uso apresentado na figura 1 deste documento, da Descrição Geral da Linha de Produto Mobile Media do documento 4 e descrição das similaridades e variabilidades; modele um produtos específico possível de ser extraído do diagrama apresentado modelado conforme a **Abordagem X**.

I.2 Com base nos Diagrama de Classes apresentado na figura 2 deste documento, da Descrição Geral da Linha de Produto Mobile Media do documento 4 e descrição das similaridades e variabilidades; modele um produto específico possível de ser extraído do diagrama apresentado modelado conforme a **Abordagem X**.

I.3 Supondo que em uma configuração de produto, as funcionalidades relacionadas a música sejam excluídas no diagrama de caso de uso. Que alteração/impacto teria no diagrama de classes?

I.4 Existe um ponto de variação que é “Kernel” (MediaMgr) no diagrama de classes, caso ele seja excluído qual o impacto no diagrama de casos de uso?



Mobile Media:
Modelo de Identificação de Variabilidades (Abordagem Y)

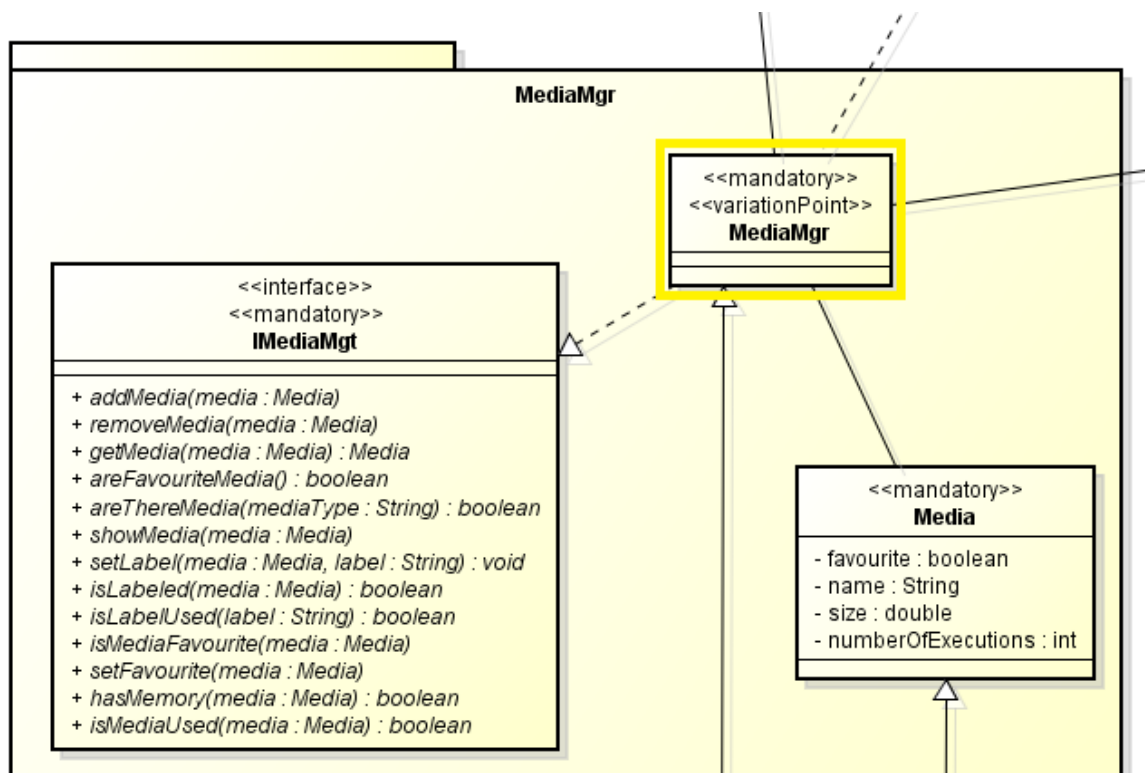
I. Configuração de Produtos

I.1 Com base no Diagrama de Casos de Uso apresentado na figura 1 deste documento, da Descrição Geral da Linha de Produto Mobile Media do documento 4 e descrição das similaridades e variabilidades; modele um produto específico possível de ser extraído do diagrama apresentado modelado conforme a **Abordagem Y**.

I.2 Com base no Diagrama de Classes apresentado na figura 2 deste documento, da Descrição Geral da Linha de Produto Mobile Media do documento 4 e descrição das similaridades e variabilidades; modele um produto específico possível de ser extraído do diagrama apresentado modelado conforme a **Abordagem Y**.

I.3 Supondo que em uma configuração de produto, as funcionalidades relacionadas a música sejam excluídas. Que alteração/impacto teria no diagrama de classes?

I.4 Existe um ponto de variação que é “mandatory” (MediaMgr) no diagrama de classes, caso ele seja excluído qual o impacto no diagrama de casos de uso?



B.2 Materiais do Experimento #2

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)

Mestranda / Pesquisadora: Thais Santos Nepomuceno	Grupo de Pesquisa: GReater - Grupo de Pesquisa em Reuso Sistemático de Software e Experimentação Contínua.
Participante:	
Email para Contato:	Data:

Prezado(a) Senhor(a),

O grupo de pesquisa GReater eventualmente realiza estudos experimentais para caracterizar/avaliar uma determinada tecnologia de software. Estes estudos são conduzidos por alunos de Pós-graduação em Ciência da Computação (PCC) do Departamento de Informática (DIN) da Universidade Estadual de Maringá (UEM), sob a orientação do Prof. Edson Alves de Oliveira Júnior. Você foi previamente selecionado pelo seu perfil/conhecimento/experiência e está sendo convidado a participar desta pesquisa. Essa pesquisa será feita com base em dados coletados a partir de trabalhos práticos. Embora o trabalho prático faça parte da disciplina, você tem o direito de não permitir a utilização dos dados do seu trabalho na pesquisa.

1. Procedimentos

O estudo será realizado com data e hora marcada com os participantes pré-selecionados. Para participar do estudo normalmente será aplicado um formulário de caracterização de perfil, a fim de identificar seu nível de conhecimento/experiência. Em seguida, o estudo será executado de forma individual ou em grupos formados, seguindo sempre o planejamento do estudo feito pela pesquisadora responsável. Caso seja necessário, ao final do estudo será solicitado ao participante que responda um questionário de avaliação sobre a tecnologia de software que está sendo caracterizada/avaliada.

2. Tratamento de possíveis riscos e desconfortos

Serão tomadas todas as providências durante a coleta de dados de forma a garantir a sua privacidade e seu anonimato. Além disso, não existem riscos ou desconfortos que poderão afetar o participante durante a condução do estudo. Ex: fadiga, estresse, mal estar, dentre outros.

3. Benefícios e Custos

Espera-se que, como resultado deste estudo, você possa aumentar seus conhecimentos, de maneira a contribuir para o aumento da qualidade das atividades com as quais você trabalhe ou possa vir a trabalhar. Este estudo também contribuirá com resultados importantes para a pesquisa de um modo geral para o grupo de pesquisa GRSSE. Você não terá nenhum gasto ou ônus com a sua participação no estudo e também não receberá qualquer espécie de reembolso ou gratificação devido à autorização dos seus dados na pesquisa.

4. Confidencialidade da Pesquisa

Toda informação coletada neste estudo é confidencial e seu nome não será identificado de modo algum, a não ser em caso de autorização explícita para este fim. Quando os dados forem coletados, seu nome será removido dos mesmos e não será utilizado em nenhum momento durante a análise ou apresentação dos resultados.

5. Participação

Sua participação neste estudo é muito importante e voluntária, pois requer a sua aprovação para utilização dos dados coletados neste estudo. Segundo a Resolução 466/12 do Conselho Nacional de Saúde (CNS), o respeito devido à dignidade humana exige que toda pesquisa se processe após assentimento livre e esclarecido. Você tem o direito de não querer participar ou de sair deste estudo a qualquer momento, sem penalidades. Em caso de você decidir se retirar do estudo, favor notificar a pesquisadora responsável. Os pesquisadores responsáveis pelo estudo poderão fornecer qualquer esclarecimento sobre o mesmo, assim como tirar dúvidas.

Coordenador do Grupo de Pesquisa GReater: Prof. Edson Alves de Oliveira Júnior - edson@din.uem.br

Pesquisadora do Grupo de Pesquisa GReater: Mestranda Thais Santos Nepomuceno - thais.nepomuceno1@gmail.com

6. Declaração de Consentimento

Declaro que li e estou de acordo com as informações contidas neste documento e que toda linguagem técnica utilizada na descrição deste estudo de pesquisa foi explicada satisfatoriamente, recebendo respostas para todas as minhas dúvidas. Confirmando também que recebi uma cópia deste Termo (TCLE), compreendo que sou livre para não autorizar a utilização dos meus dados neste estudo em qualquer momento, sem qualquer penalidade. Declaro ter mais de 18 anos e concordo de espontânea vontade em participar deste estudo.

Obrigado pela sua colaboração!

Nome:

Questionário de Caracterização de Participante em Estudo Experimental

“Avaliação experimental de uma abordagem para gerenciamento de variabilidades em linhas de produto de software baseadas em UML”

Nas perguntas a seguir, quando duas ou mais alternativas forem válidas, marque a alternativa que mais se aplica ao seu caso.

1. Qual o seu nível de formação?

- | | |
|-------------------------------------|-----------------------------------|
| <input type="checkbox"/> Graduando | <input type="checkbox"/> Graduado |
| <input type="checkbox"/> Mestrando | <input type="checkbox"/> Mestre |
| <input type="checkbox"/> Doutorando | <input type="checkbox"/> Doutor |

2. Em qual setor atua?

- | | |
|---|---|
| <input type="checkbox"/> Acadêmico (ensino) | <input type="checkbox"/> Industrial (empresarial) |
|---|---|

3. Qual o nome da empresa/universidade que atua?

.....

4. Quanto tempo possui de experiência na área que atua?

.....

5. Qual a sua experiência com a notação UML com relação aos diagramas de classes e sequencia?

- Eu **nunca** modelei um software usando a UML.
- Minha experiência com a notação UML é básica.**
Eu modelo software somente no nível dos elementos mais comuns da UML como classes e herança; objetos;
- Minha experiência com a notação UML é moderada.**
Eu modelo software no nível dos elementos da opção anterior, além de: relacionamentos de dependência *include* e *extend*, e *extension points* em diagramas de casos de uso; polimorfismo, associação (uni e bi-direcionais), dependência, agregação e composição em classes; realizações e dependência em componentes, troca de mensagens em diagramas de sequência.

[] **Minha experiência com a notação UML é avançada.**

Eu modelo software que exige a utilização de todos os elementos de diagramas de casos de uso, classes, componentes e sequência, além de outros diagramas da UML como, por exemplo, diagramas de colaboração e atividades.

6. Qual a sua experiência com relação à abordagem de Linha de Produto de Software (LPS) e Gerenciamento de Variabilidade?

[] Eu **nunca** ouvi falar a respeito de LPS.

[] **Já lí**, de forma superficial, algo a respeito de LPS.

[] **Minha experiência com LPS é básica.**

Eu conheço os seguintes conceitos da abordagem: ciclo de desenvolvimento de LPS e suas atividades (engenharia de domínio e engenharia de aplicação). Porém, **não tenho experiência com gerenciamento de variabilidades.**

[] **Minha experiência com LPS é moderada.**

Eu conheço os conceitos da opção anterior, e com relação ao gerenciamento de variabilidades, eu sei o conceito de pontos de variação, variantes e os seus relacionamentos, além dos conceitos de resolução de variabilidades e tempos de resolução (*design time, link time, runtime*, entre outros).

[] **Minha experiência com LPS é avançada.**

Eu conheço os conceitos da opção anterior, além de alguns processos existentes de desenvolvimento de LPS (FODA, PLP, PLUS, PuLSE, entre outros). Com relação ao gerenciamento de variabilidades, eu sei os conceitos da opção anterior, além de: modelos de resolução; abordagens existentes para o gerenciamento de variabilidades, e representação de variabilidades (usando a UML, modelos de características, entre outras).

Assinatura do Participante	Local e Data
<hr/>	Cidade e Data

I. Linha de Produto de Software

Uma linha de produto de software (LPS) corresponde a um conjunto de sistemas de *software* que compartilham características (*features*) comuns e gerenciáveis, que satisfazem a necessidade de um segmento particular ou de uma missão. Este conjunto de sistemas é denominado também família de produtos. Os membros da família são produtos específicos desenvolvidos de maneira sistemática a partir da instanciação de uma infraestrutura comum de uma LPS, chamada núcleo de artefatos.

O núcleo de artefatos é formado por um conjunto de características comuns (similaridades) e características variáveis (variabilidades). As variabilidades podem estar associadas a diferentes níveis de abstração, como a descrição da arquitetura, o código fonte, etc., e auxiliam na geração de produtos específicos distintos em um mesmo domínio e, dessa forma, diminuem o custo e o tempo de desenvolvimento, reduzem riscos e perdas, além de reduzirem o *time to market* e justificarem o retorno de investimento (ROI).

O gerenciamento de variabilidades é uma das atividades mais importantes no gerenciamento de uma LPS, tendo sido foco de atenção por diversos pesquisadores, com pode ser percebido pelas diversas abordagens presentes na literatura.

As variabilidades podem estar associadas a diferentes níveis de abstração, dentre eles, a descrição da arquitetura, a documentação de projeto, o código fonte, o código compilado, o código ligado e o código executável.

Em síntese **variabilidade é a forma como os membros de uma família de produtos podem se diferenciar entre si, ou seja, é o que permite distinguir os diversos produtos de uma LPS.**

A variabilidade é descrita por pontos de variação e variantes:

- **Ponto de variação:** Um **local específico** de um artefato em que uma decisão de projeto ainda não foi tomada, ou seja, foi adiada;
- **Variante:** Corresponde a **uma alternativa** de projeto para resolver uma determinada variabilidade.
- **Restrições entre variantes:** define os relacionamentos entre duas ou mais variantes para que seja possível resolver um ponto de variação ou uma variabilidade.

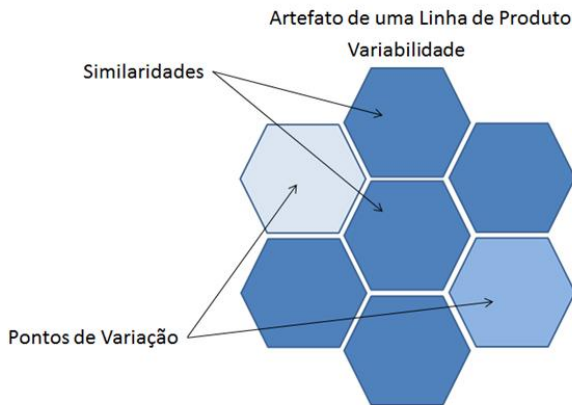
A aplicação destes conceitos é ilustrada na Figura 1.

A maioria das abordagens desenvolvidas para auxiliar no gerenciamento de variabilidades envolve diversos conceitos e modelos de representação. As abordagens “X” e “Y” apresentadas nos documentos 3.2 e 3.3, respectivamente, possuem como base a UML – *Unified Modeling Language*. Tais abordagens utilizam diversos conceitos da UML como o uso de estereótipos e diversos diagramas.

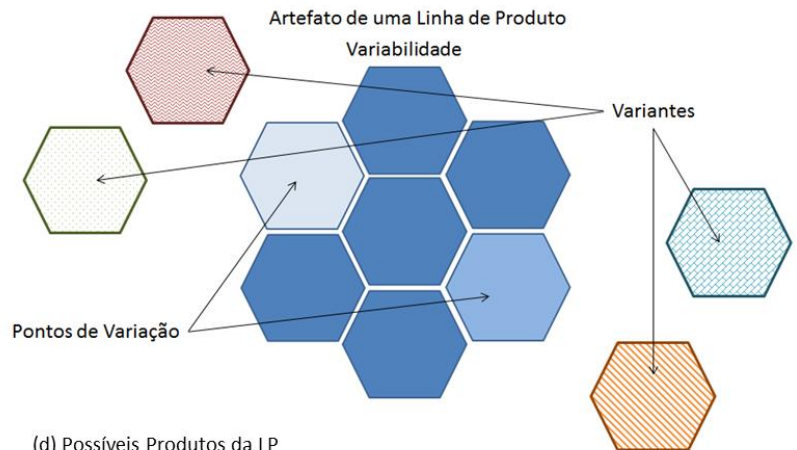
II. Abordagens para Linha de Produto de Software

Para a representação de linhas de produto de software e o gerenciamento de suas variabilidades, como mencionado anteriormente, existem diversas abordagens presentes na literatura. Entre os itens que estas abordagens contemplam, temos os que seguem na Tabela I, que serão posteriormente apresentados para cada abordagem.

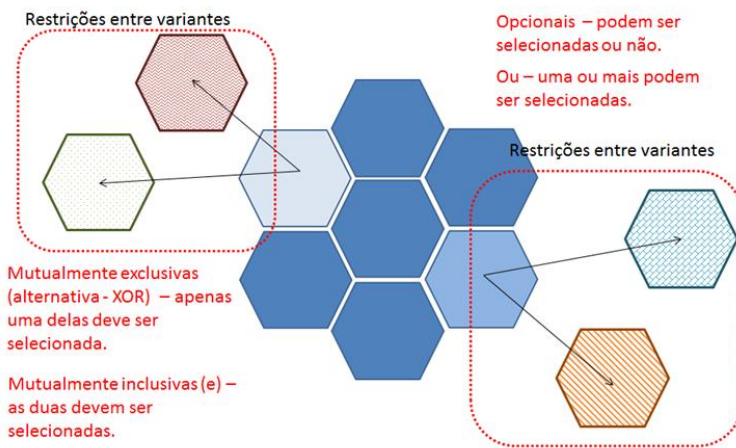
(a) Variabilidade



(b) Pontos de Variação e Variantes



(c) Variantes e suas Restrições



(d) Possíveis Produtos da LP

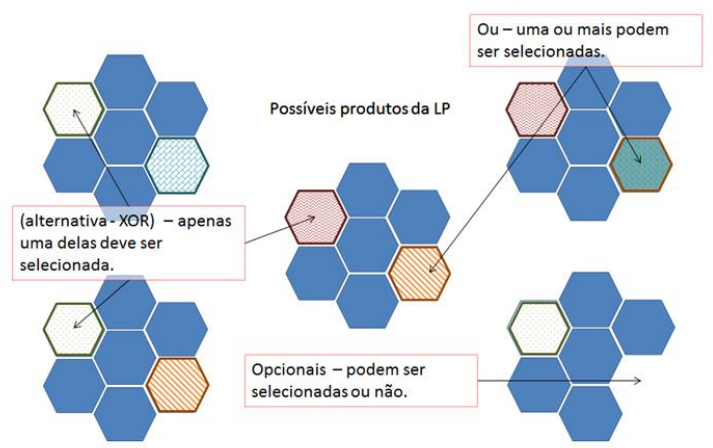


Figura 1 – Exemplo dos Conceitos de Variabilidade, Pontos de Variação, Variantes e Restrições entre Variantes.

Item	Identificação
Baseada em UML	Indica que a abordagem utiliza os modelos UML, meta-atributos, etc., como forma de representação da LPS e de suas variabilidades.
Perfil	Diversas abordagens apresentam um perfil específico que é formado por estereótipos e meta-atributos, geralmente derivados de uma linguagem de modelagem, como a UML.
Processo	O processo contempla a sistematização da utilização de um perfil para o gerenciamento de variabilidades, guiando o usuário no uso das definições do perfil utilizado.
Estereótipos	Estereótipos, como os da UML, são um padrão de mecanismo de extensão e são usados para distinguir diferentes tipos de elementos modelados. Em LPS são ferramentas úteis para identificar variabilidade, seus pontos de variação,

	variantes e outros itens necessários ao seu gerenciamento.
Diretrizes	São os passos sistematizados definidos no processo, que permitem a aplicação facilitada do perfil da abordagem a que corresponde.

III. Rastreabilidade de Variabilidades

Uma LPS é projetada para atender todos os requisitos dos produtos de sua família de produtos. Esses requisitos e a própria LPS podem sofrer mudanças ao longo do tempo em decorrência de vários fatores, como por exemplo, evolução nos requisitos dos produtos, evolução no domínio de mercado para o qual os produtos foram projetados, evolução no processo de construção da LPS, evolução das tecnologias usadas para desenvolver os produtos. Para lidar com essas mudanças, a LPS deve ser modificada e evoluir, visando não ficar obsoleta, e se adequar aos novos requisitos que surgem.

A análise de impacto de mudanças consiste em uma atividade que visa compreender e identificar quais consequências essas mudanças causam ou causarão na LPS. A análise de impacto na LPS pode ser apoiada por meio de relações de rastreabilidade, as quais identificam relacionamentos entre artefatos criados durante todas as fases do desenvolvimento de software.

I. Visão Geral Abordagem X

Tabela I – Visão Geral Abordagem X.

Abordagem X			
Item	Sim	Não	Observação
Baseada em UML?	X		
Perfil?		X	
Processo?		X	
Estereótipos?	X		Estereótipos específicos, possuindo variações entre modelos.
Diretrizes?		X	Especificação de uso da abordagem através representação textual.

II. Estereótipos e Diretrizes

Nesta seção são apresentados os estereótipos para aplicação em casos de uso e classes por meio da Tabela II, em seguida exemplos do uso destes são apresentados, seguidos por especificações textuais do seu uso, identificando de forma conceitual as diretrizes.

Tabela II – Estereótipos da Abordagem X para Casos de Uso e Classes.

Estereótipo	Definição	Suporta modelos de	
		Casos de uso?	Classes?
<<kernel>>	Usado para representar elementos obrigatórios.	Sim	Sim
<<optional>>	Usado para representar elementos opcionais que podem ser selecionados ou não, para um produto específico.	Sim	Sim
<<alternative>>	Usado para representar elementos alternativos, mutuamente exclusivos ou inclusivos.	Sim	Não

II.1 Exemplos

Classes

Para a identificação das variabilidades e similaridades por meio da Abordagem X aplica-se um dos estereótipos apresentados na Tabela II, para cada classe existente no modelo e de acordo com a especificação da Linha de Produto a ser modelada.

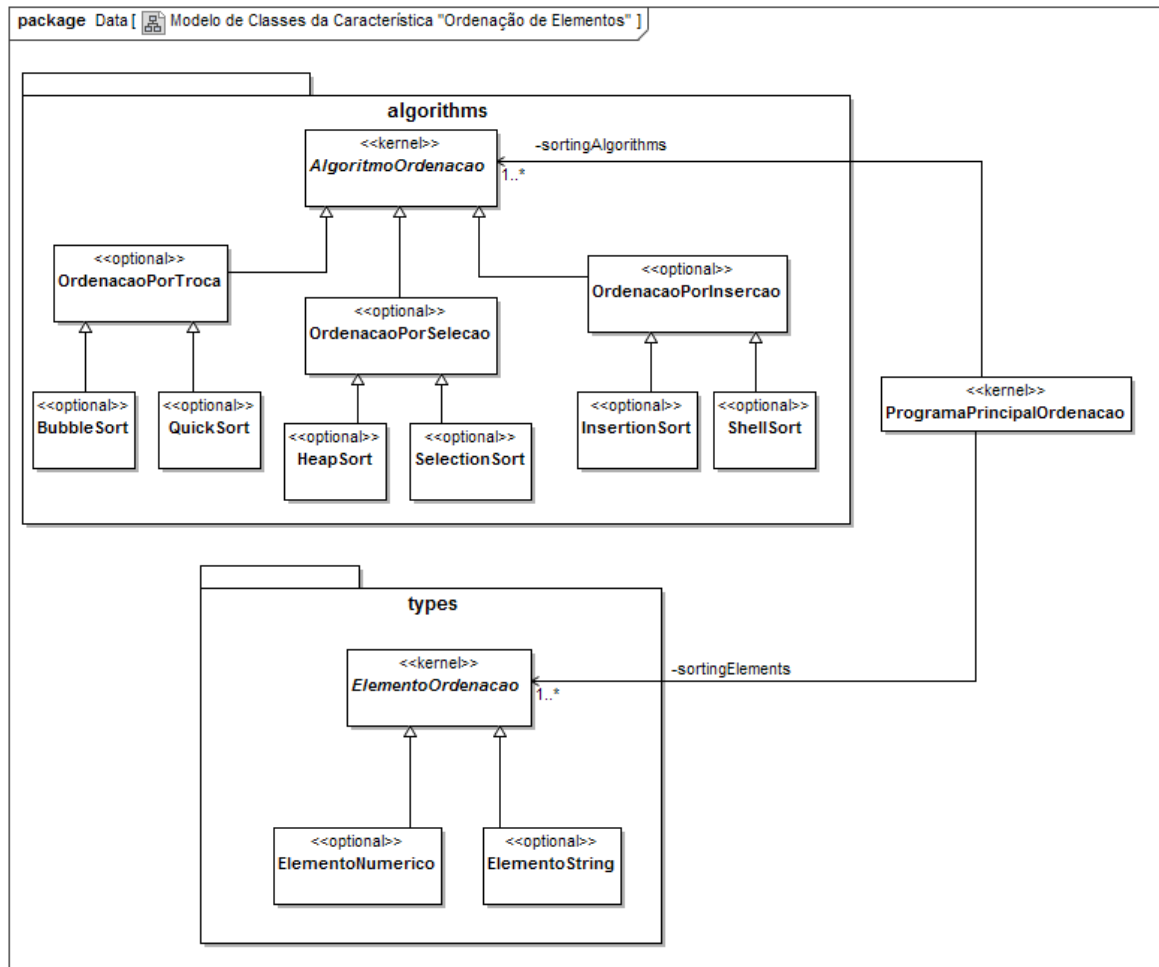


Figura 5 – Exemplo de Modelo de Variabilidade em Diagrama de Classes com a Abordagem X.

Para o exemplo de linha de produto de "Ordenação de Elementos" (Figura 5), as classes obrigatórias, ou seja, as que são similares para todos os produtos gerados a partir de tal LPS são estereotipadas como *<<kernel>>*, como a classe **ProgramaPrincipalOrdenacao**. Para as classes que são alternativas, podendo ser selecionadas se mutuamente exclusivas, mutuamente inclusivas e também opcionais ocorre à aplicação do estereótipo *<<optional>>*, como para a classe **ElementoOrdenacao**.

I. Visão Geral Abordagem Y

Tabela I – Visão Geral Abordagem Y

Abordagem Y			
Item	Sim	Não	Observação
Baseadaem UML?	X		
Perfil?	X		
Processo?	X		
Estereótipos?	X		Estereótipos específicos padrões para todos os modelos.
Diretrizes?	X		Diretrizes específicas para cada modelo.

II. Estereótipos e Diretrizes

Nesta seção são apresentados os estereótipos para aplicação em casos de uso, existentes no perfil da abordagem Y por meio da Tabela II.

Tabela II – Estereótipos da Abordagem Y para Casos de Uso.

Estereótipos Abordagem Y para casos de uso	
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 (em UML) entre variantes no qual a variante dependente (origem da dependência) só existirá em uma configuração se a variante relacionada (destino da dependência) existir.
<<mutex>>	Indica um relacionamento de dependência (em UML) entre variantes no qual a variante dependente (origem da dependência) só existirá em uma configuração se a variante relacionada (destino da dependência) obrigatoriamente não existir. São conhecidas como variantes mutuamente exclusivas.

II.1 Notação de Variabilidade

As variabilidades são identificadas através de notas UML, estereotipada com <<variability>>. Nestas

notas estão contidos os meta-atributos que seguem:

- **Name**: nome da variabilidade;
 - **minSelection**: a quantidade mínima de variantes a serem selecionadas;
 - **maxSelection**: a quantidade máxima de variantes a serem selecionadas;
 - **bindingTime**: em qual momento será resolvida esta variabilidade ;
 - **allowsAddingVar**: se permite incluir novas variantes para resolver o ponto de variação; e
 - **variants**: quais as variantes para resolver o ponto de variação (casos de uso ligados ao ponto de variação).
- **realizes+**: representa a coleção de variabilidades de nível superior (menos abstrato) no qual é possível realizar a variabilidade;
- **realizes-**: representa a coleção de variabilidades de nível inferior (mais abstrato) no qual é possível realizar a variabilidade.

II.1 Exemplos

Classes

Na Figura 1 observamos a aplicação da abordagem Y, e seus elementos. Passamos a analisar cada um deles, bem como as diretrizes presentes no processo da abordagem Y, que auxiliam sua utilização em outras LPSs:

A classe **AlgoritmoOrdenacao** identifica uma classe obrigatória (*<<mandatory>>*) e representa também um ponto de variação (*<<variationPoint>>*), com três variantes. Estas variantes estão descritas no elemento comentário, relacionado a classe, por meio do *TaggedValue(variants)*. As três variantes desta classe são **OrdenacaoPorTroca**, **OrdenacaoPorSelecao** e **OrdenacaoPorInsercacao**. Todas estas são estereotipadas como *<<alternative_OR>>*, o que indica o tipo de restrição para tais variantes, neste caso, significa que ao menos uma ou todas elas podem solucionar o ponto de variação.

OrdenacaoPorTroca, **OrdenacaoPorSelecao** e **OrdenacaoPorInsercacao**, além de variantes, são, por sua vez, pontos de variação (*<<variationPoint>>*), e assim, cada uma delas apresenta um comentário, que descreve as suas variantes (**variants**), bem como o nome da mesma (**name**). Neste caso, todas as variantes são marcadas como *<<alternative_OR>>* e, como anteriormente, uma delas, ao menos, deve ser selecionada ou todas.

A classe **ProgramaPrincipalOrdenacao**, representa uma classe obrigatória, portanto é marcada como *<<mandatory>>*, e estará presente em todos os produtos desta LPS.

A classe **ElementoOrdenacao**, também é obrigatória (*<<mandatory>>*) e representa um ponto de variação (*<<variationPoint>>*), logo possui o elemento comentário ligado a ela, com o estereótipo *<<variability>>*, que identifica os dados da variabilidade, que é nomeada, por exemplo, de “sortingelement” e possui duas classes variantes (**variants**): **ElementoNumerico** e **ElementoString**, marcadas como variantes alternativas *<<alternative_OR>>*, onde, ambas podem ser selecionadas, ou ao menos uma.

Desta forma, as variabilidades são identificadas por meio do comentário UML, estereotipada com *<<variability>>*.

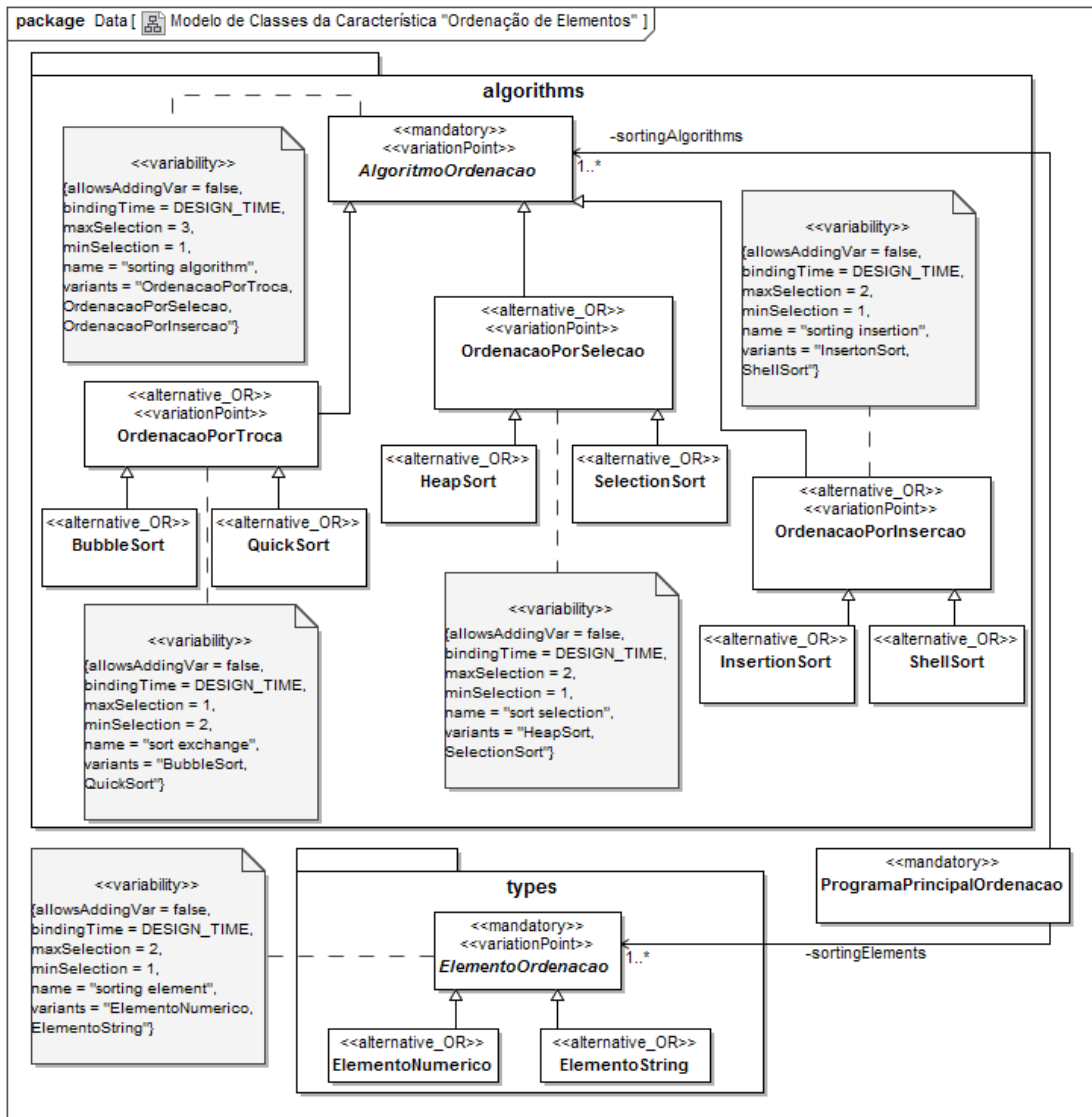


Figura 1 – Exemplo de Modelo de Variabilidade em Diagrama de Classes com a Abordagem Y.

Componentes, Portas e Interfaces

A Figura 2 apresenta um exemplo de representação de variabilidades na LPS “AGM”, em nível de componente. Todos os elementos identificados como obrigatórios foram anotados com o estereótipo <<mandatory>>. O componente **GameCtrl**, apresenta-se três interfaces referente aos jogos a serem selecionados, assim, uma ou mais interfaces devem ser selecionadas. O ponto de variação foi atribuído a uma porta, assim, as interfaces passam a ser suas variantes. A atribuição de ponto de variação para a porta *pPlayGame*, foi representada no compartimento do componente **GameCtrl**, adicionando o estereótipo <<variationPoint>> frente ao nome da porta. As interfaces *ISaveScore* e *ICheckScore*, foram identificadas como sendo opcionais, assim, foram anotadas com o estereótipo <<optional>>. Todos os pontos de variação e elementos opcionais possuem uma nota UML com o estereótipo <<variability>>, detalhando as propriedades de tal variabilidade de acordo com a abordagem Y. A representação da restrição

<<mutex>> foi forçada neste exemplo, com o intuito de mostrar um exemplo da aplicação desta restrição. Assim, a interface *ISaveScore* só poderá ser selecionada, caso a interface *IPlayBrickles* não seja selecionada.

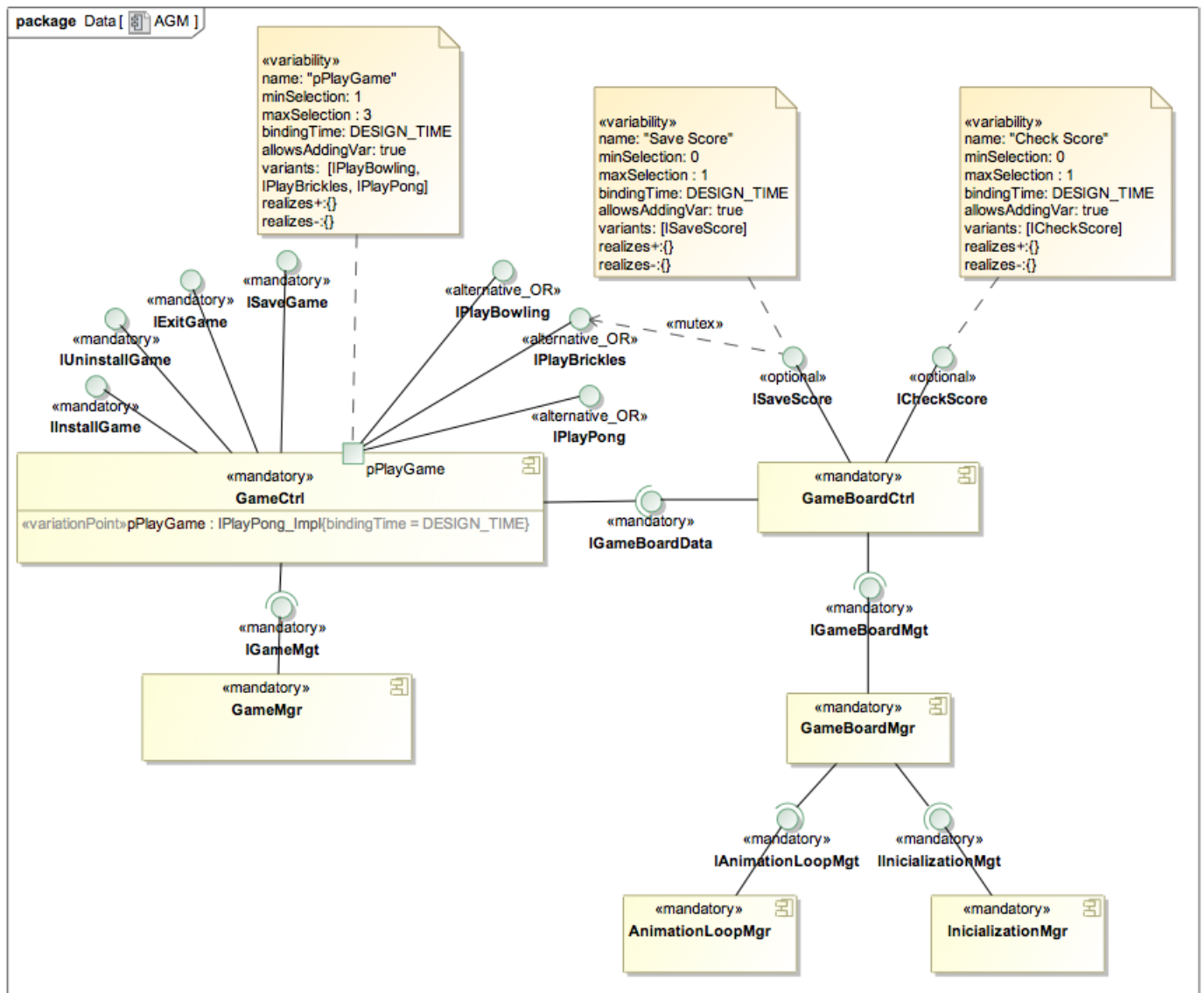
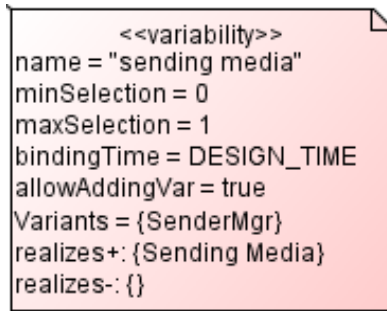


Figura 2 - LPS AGM

Rastreabilidade

A rastreabilidade para identificação dos níveis de variabilidade pode ser identificada por meio do meta-atributo *realizes* contido na notação de comentários UML, com o estereótipo *variability*.

Para a identificação de rastreabilidade em níveis de maior abstração, o meta-atributo *realizes* contém um sinal de adição (+). Para os níveis de menor abstração, o item *realizes* contém um sinal de subtração (-). Assim, a rastreabilidade das resoluções de variabilidade se torna bidirecional.



II.2 Diretrizes para configuração de produtos

D.1 Selecione para o produto a ser configurado todos os componentes que são obrigatórios, demarcados com o estereótipo <<mandatory>>;

D.2 Inclua também todas as interfaces obrigatórias, demarcadas com o estereótipo <<mandatory>>;

D.3 Resolva os pontos de variação, com o estereótipo <<variation point>>, de acordo com a notação de variabilidade, observando a quantidade mínima e máxima de variantes que devem ser escolhidas;

D.3.1 Observe a relação das variantes por meio de seus estereótipos. Se forem <<alternative_OR>> pelo menos uma das variantes deverá ser escolhida para resolver o ponto de variação, se <<alternative_XOR>> somente uma das variantes deverá ser escolhida para resolver o ponto de variação;

D.4 Uma variante V_1 que ao ser selecionada para fazer parte de um determinado produto, tiver um relacionamento com outra variante V_2 com o estereótipo <<requires>>, essa variante (V_2) também deverá fazer parte do produto;

D.5 Uma variante V_1 que ao ser selecionada para fazer parte de um determinado produto, tiver um relacionamento com outra variante V_2 com o estereótipo <<mutex>>, essa variante (V_2) não poderá fazer parte do produto.

II.3 Diretrizes para rastreabilidade

D.1 Identifique a notação de comentário UML, com o estereótipo <<variability>> no elemento ao qual pretende-se rastrear;

D.2 Para a identificação de rastreabilidade em níveis de maior abstração, procure o meta-atributo “realizes+”;

D.3 Para rastrear elementos em diagramas de menor abstração, identifique o “realizes-”;

D.4 Vá até o diagrama correspondente e procure pelo nome identificado no meta-atributo realizes. Você terá rastreado um elemento de um diagrama para o outro!

I. Visão Geral Abordagem Z

Tabela I – Visão Geral Abordagem Z.

Abordagem Z			
Item	Sim	Não	Observação
Baseada em UML?	X		Compatível com a versão 2.0
Possui um Perfil UML definido?		X	
Possui um Processo definido?		X	
Utiliza Estereótipos?	X		Estereótipos específicos, possuindo variações entre modelos.
Possui Diretrizes?		X	

II. Estereótipos

Nesta seção são apresentados os estereótipos utilizados para a identificação de variabilidades no diagrama de componentes, por meio da Tabela II, em seguida exemplos do uso destes são apresentados, seguidos por especificações textuais do seu uso, identificando de forma conceitual as diretrizes.

Tabela II – Estereótipos da Abordagem Z para diagrama de Componentes.

Estereótipos Abordagem Z para Componentes		
Estereótipo	Utilização	Exemplos
<<alt_vp>>	Usado para representar um ponto de variação alternativo, onde suas variantes são alternativas e são representadas pelo estereótipo <<variant>>.	Figura 1
<<opt_vp>>	Usado para representar um ponto de variação opcional, onde suas variantes são opcionais e são representadas pelo estereótipo <<variant>>.	--
<<variant>>	Usado para representar variantes associadas a um determinado ponto de variação.	Figura 1 e 2
<<optional>>	Usado para representar elementos que podem ou não ser selecionados.	Figura 1
<<altvvp>>	Utilizado para representar pontos de variação existentes em variantes, do tipo alternativo.	Figura 2
<<optvvp>>	Utilizado para representar pontos de variação existentes em variantes, do tipo opcional.	--

II.1 Exemplos

Componentes

Para a identificação da variabilidades e similaridades por meio da Abordagem Z aplica-se um dos estereótipos apresentados na Tabela II, para cada componente existente no modelo e de acordo com a especificação da Linha de Produto a ser modelada.

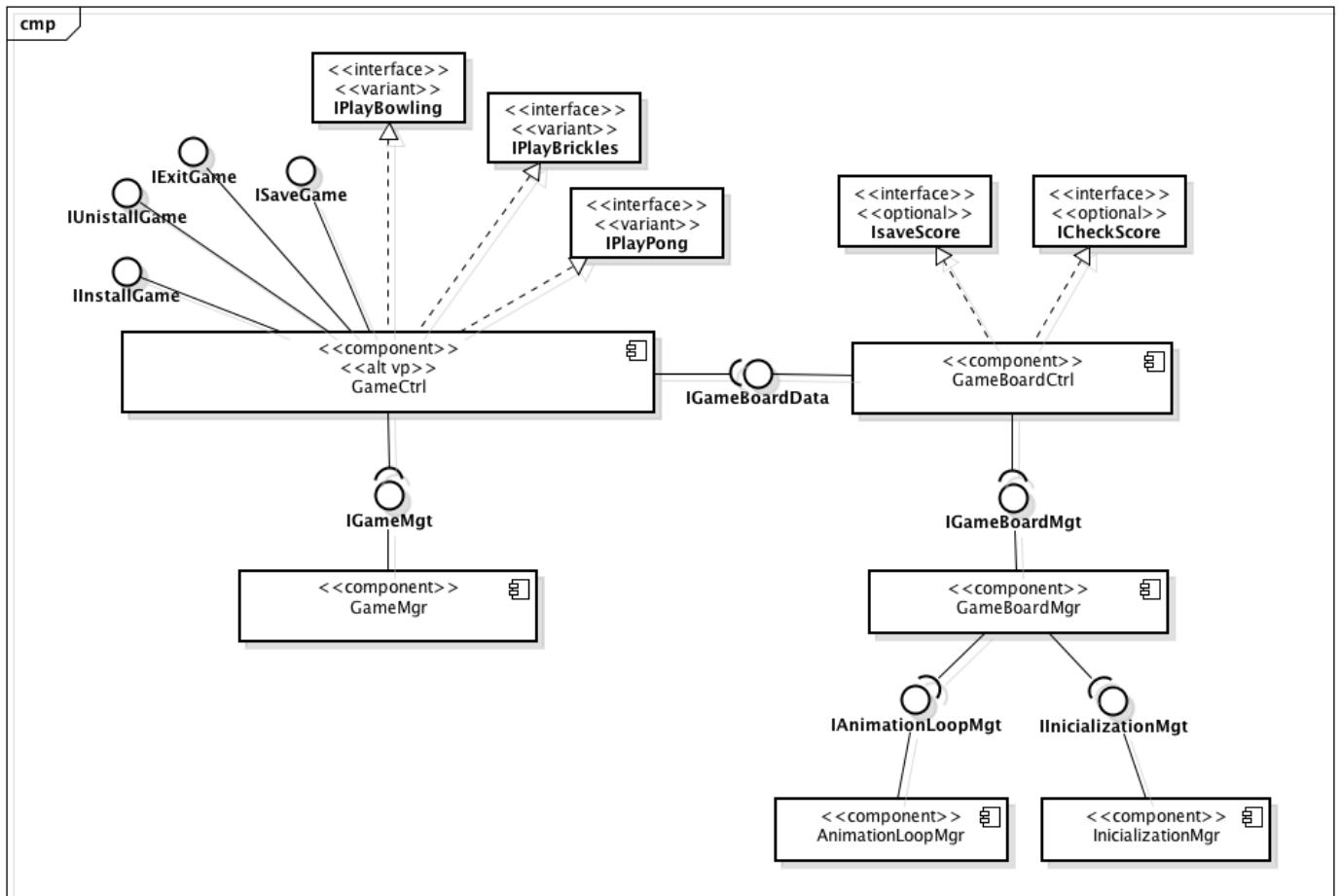


Figura 1 – Exemplo de Modelo de Variabilidade em Diagrama de Componentes com a Abordagem Z.

Para o exemplo de linha de produto "AGM", o componente identificado como ponto de variação, contendo variantes alternativas, está anotado com o estereótipo <<altvp>>, e suas variantes, com o estereótipo <<variant>>, indicando que ao menos uma variante deve ser selecionada. Os elementos considerados opcionais, neste caso as interfaces ISaveScore e ICheckScore, foram anotadas com o estereótipo <<optional>>, indicando que elas podem ou não existirem.

A Figura 2 mostra um exemplo, onde o componente **backgroundMgr** é um ponto de variação alternativo (<<altvp>>), que possui duas interfaces variantes, sendo elas **BackgroundColorMgr** e **BackgroundTypeMgr**, anotadas com o estereótipo <<variant>>.

A interface BackgroundColorMgt, além de ser uma variante, do componente **backgroundMgr**, também é um ponto de variação, assim, está anotada com o estereótipo <<altvvp>>. Tal ponto de variação possui 3 variantes: *red*, *White* e *black*, anotadas com o estereótipo <<variant>>.

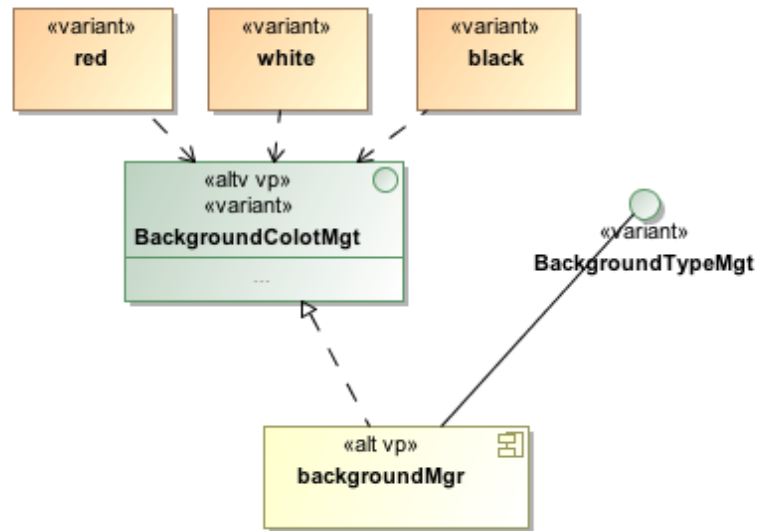


Figura 2 – Exemplo de Modelo de Variabilidade em Interfaces com a Abordagem Z.

I. Identificação

Mobile Media (YOUNG, 2005) é uma LPS composta por aplicações (produtos) que manipulam músicas, vídeos e fotos para dispositivos móveis, como celulares e *palm tops*. Ela provê suporte para gerenciar (criar, excluir, visualizar, executar, enviar) diferentes tipos de mídia.

A Mobile Media surgiu da extensão de uma LPS já existente denominada Mobile Photo (YOUNG, 2005), por meio da inserção de novas propriedades multimídia, como manipulação de vídeos e músicas, que somente podem ser realizados em alguns tipos de aparelhos. De certa forma, pode-se dizer que a inserção das características opcionais e alternativas a determinados aparelhos caracterizou o surgimento da Mobile Media.

II. Similaridades e Variabilidades

Nesta seção são apresentadas as similaridades da LP, ou seja, os aspectos comuns a todos os produtos desta LP, bem como as variabilidades, que representam os aspectos que diferem de um produto em relação ao outro.

II.1 Similaridades

- Gerenciar Álbum: Permite ao usuário definir novos álbuns.
- Executar Mídia: Permite ao usuário assistir vídeo, ouvir música e/ou visualizar fotos.
- Adicionar Mídias no Álbum: Permite ao usuário adicionar novas fotos a álbuns definidos.
- Gerenciar Mídia: Permite ao usuário gerenciar vídeos, músicas e/ou fotos.
- Log in: o usuário deve fazer login antes de gerenciar mídia.

II.2 Variabilidades

- Enviar Foto via SMS: Permite a um usuário enviar uma foto para outro via *Short Messaging Service*.
- Enviar Foto via Email: Permite a um usuário enviar uma foto para outro via Email.
- Relacionar Foto com Registro na Agenda: Permite ao usuário associar um registro na sua lista de contatos com a foto do álbum.
- Mostrar Foto nas Chamadas Recebidas: Intercepta chamadas recebidas e mostra a foto associada ao contato.
- Tocar Melodia nas Chamadas Recebidas: Intercepta chamadas recebidas e toca uma melodia personalizada para aquele contato.

Os itens a seguir apresentam as características básicas e as características variáveis da Mobile Photo, agora denominada Mobile Media:

1. Criar Album de Fotos : Permite ao usuário definir novos álbuns de fotos para armazenar categorias de fotos no dispositivo. A persistência da informação do álbum é realizada utilizando RMS (J2ME Record Management System).

2. **Armazenar Foto:** Gerenciar a conversão e persistência dos arquivos de foto para o sistema de arquivos do dispositivo utilizando RMS.
3. **Adicionar/Deletar Foto:** Permite ao usuário excluir fotos permanentemente do dispositivo ou adicionar novas fotos a álbuns definidos.
4. **Rotular Foto:** Permite ao usuário determinar um texto para uma foto. Os rótulos aparecerão na lista de exibição e podem ser utilizados para uma futura funcionalidade relacionada à busca.
5. **Visualizar Foto:** Mostra uma foto selecionada na tela do dispositivo.
6. **Enviar Foto via SMS:** Permite a um usuário enviar uma foto para outro via Short Messaging Service.
7. **Enviar Foto via Email:** Permite a um usuário enviar uma foto para outro via Email.
8. **Relacionar Foto com Registro na Agenda:** Permite ao usuário associar um registro na sua lista de contatos com a foto do álbum.
9. **Mostrar Foto nas Chamadas Recebidas:** Intercepta chamadas recebidas e mostra a foto associada ao contato.
10. **Tocar Melodia nas Chamadas Recebidas:** Intercepta chamadas recebidas e toca uma melodia personalizada para aquele contato.

Nome

Hora Inicial

__ : __

Hora Final

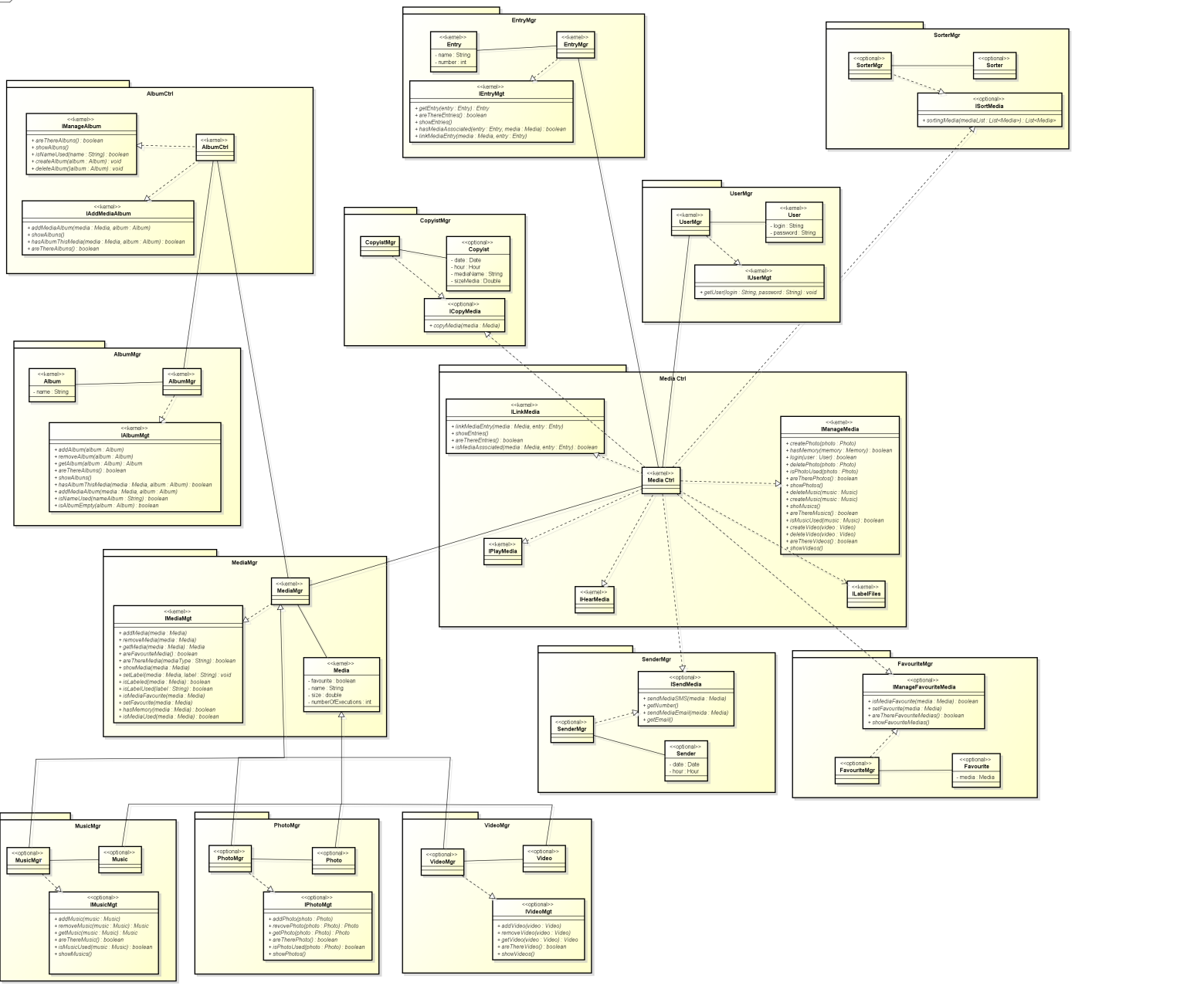
__ : __

Doc 5

Mobile Media:
Modelo de Identificação de Variabilidades (Abordagem X)

I. Configuração de Produto

I.1 Com base nos Diagrama de Classes fornecido, na Descrição Geral da Linha de Produto Mobile Media do documento 4; modele um produto específico possível de ser extraído do diagrama de Classes apresentado modelado conforme a **Abordagem X**.



Mobile Media:
Modelo de Identificação de Variabilidades (Abordagem Y)

I. Configuração de Produtos

I.1 Com base no Diagrama de Classes fornecido, na Descrição Geral da Linha de Produto Mobile Media do documento 4; modele um produto específico possível de ser extraído do diagrama apresentado modelado conforme a **Abordagem Y**.

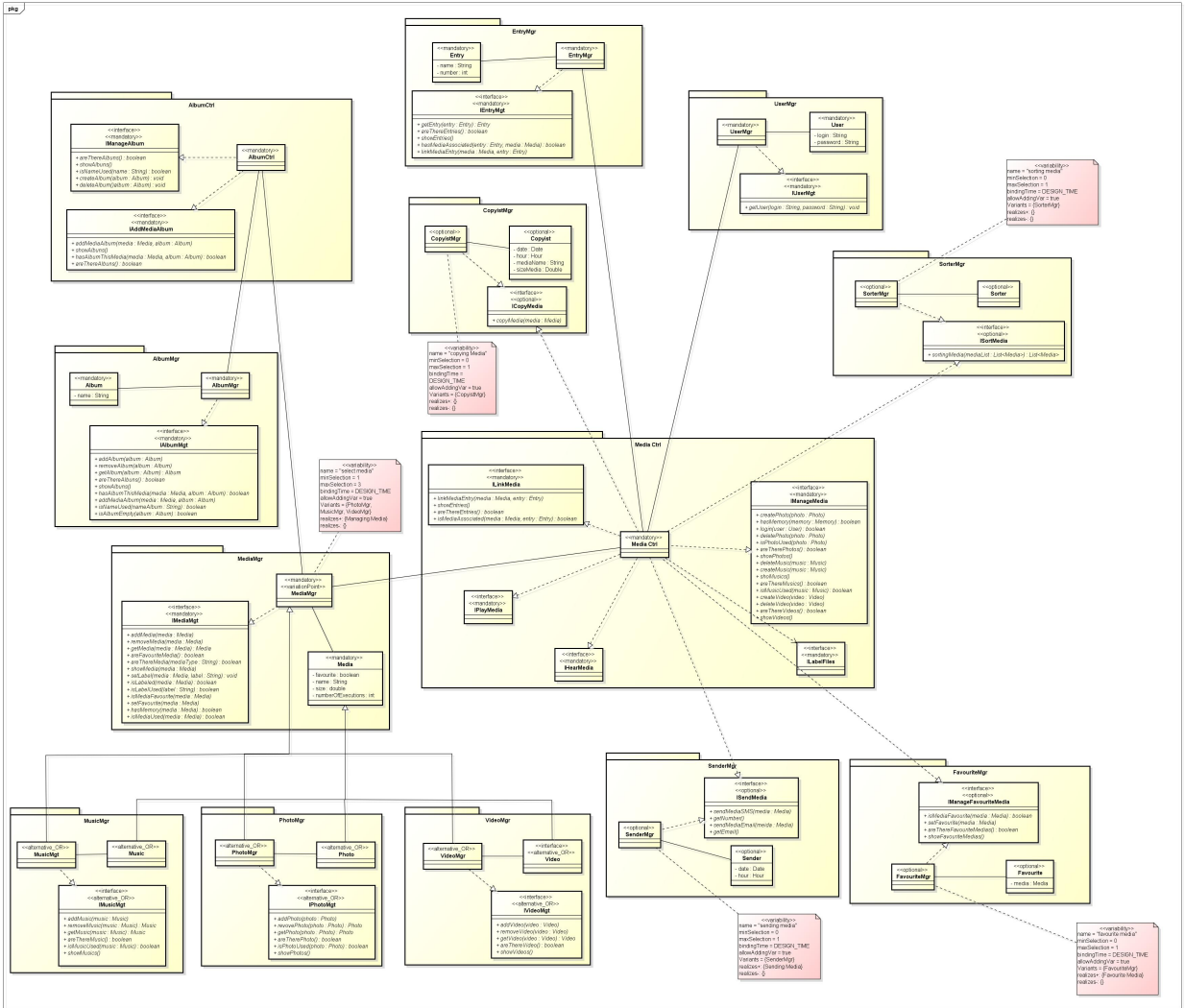
I.2 Com base no Diagrama de Componentes fornecido, na Descrição Geral da Linha de Produto Mobile Media do documento 4; modele um produto específico possível de ser extraído do diagrama apresentado modelado conforme a **Abordagem Y**.

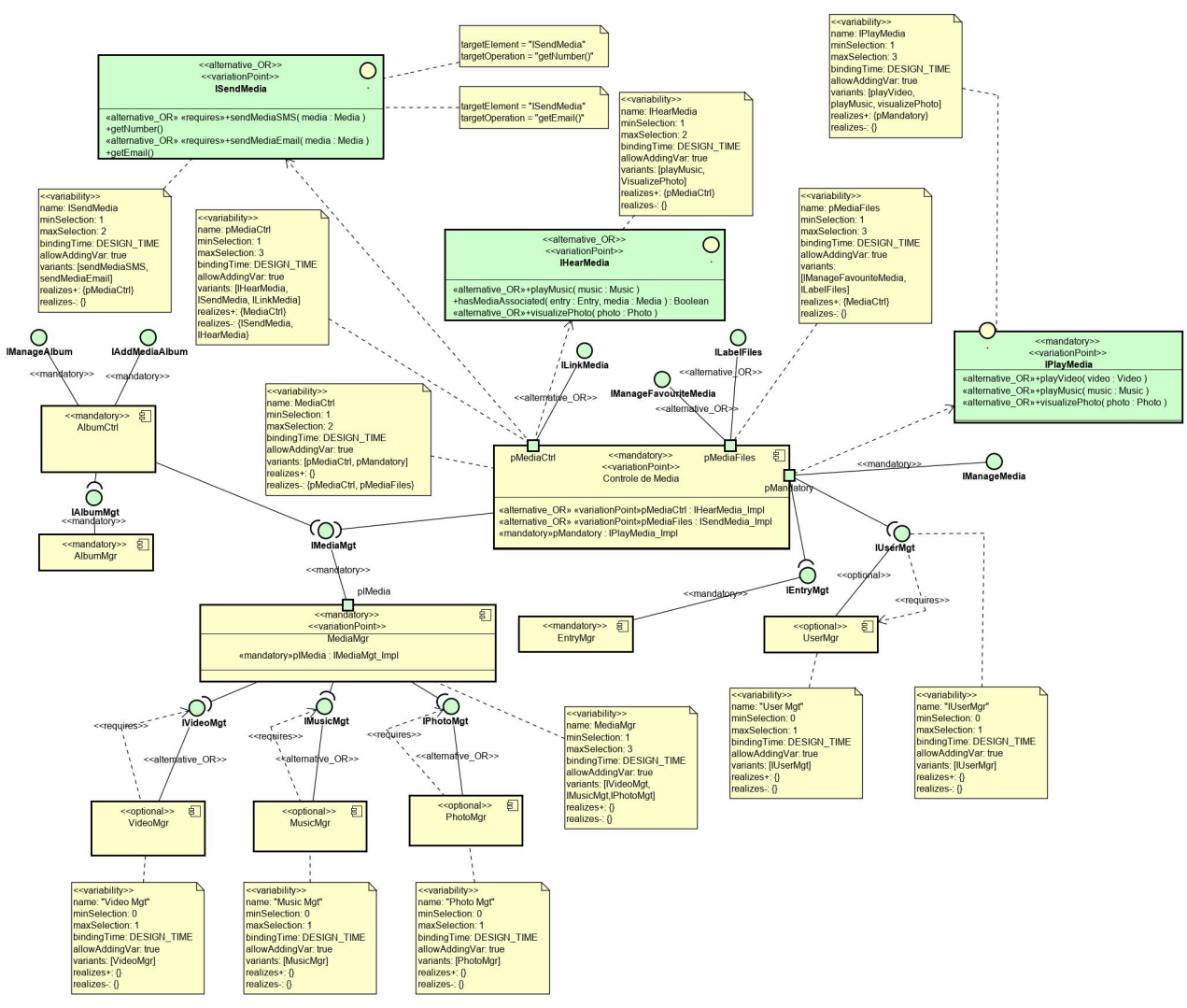
I.3 Supondo que em uma configuração de produto, as funcionalidades relacionadas ao compartilhamento de mídia fossem excluídos do diagrama de classes, as alterações/impactos no diagrama de componentes são possíveis de serem identificados com apoio da abordagem X.

- Concordo Totalmente
- Concordo Parcialmente
- Discordo Parcialmente
- Discordo Totalmente

I.4 Caso o componente **MediaMgr** seja excluído do diagrama de componentes em uma possível configuração de produto, mesmo que ele seja um ponto de variação obrigatório, é possível identificar os impactos que essa mudança causaria no diagrama de classes.

- Concordo Totalmente
- Concordo Parcialmente
- Discordo Parcialmente
- Discordo Totalmente





Nome

Hora Inicial

__ : __

Hora Final

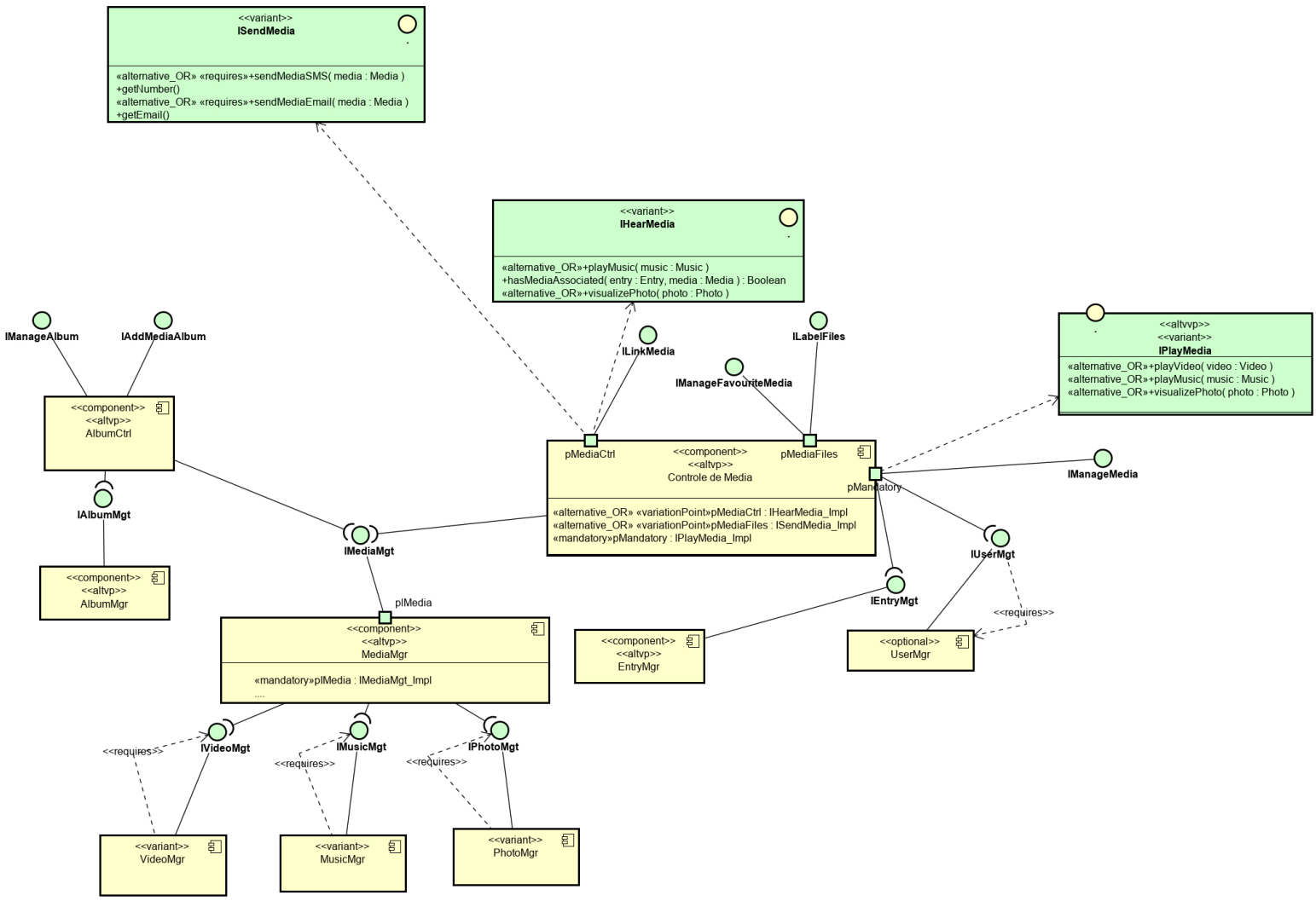
__ : __

Doc 5

Mobile Media:
Modelo de Identificação de Variabilidades (Abordagem Z)

I. Configuração de Produto

I.1 Com base nos Diagrama de Classes fornecido, na Descrição Geral da Linha de Produto Mobile Media do documento 4; modele um produto específico possível de ser extraído do diagrama apresentado modelado conforme a **Abordagem Z**.



B.3 Materiais do Experimento #3

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)

Mestranda / Pesquisadora: Thais Santos Nepomuceno	Grupo de Pesquisa: GReater - Grupo de Pesquisa em Reuso Sistemático de Software e Experimentação Contínua.
Participante:	
Email para Contato:	Data:

Prezado(a) Senhor(a),

O grupo de pesquisa GReater eventualmente realiza estudos experimentais para caracterizar/avaliar uma determinada tecnologia de software. Estes estudos são conduzidos por alunos de Pós-graduação em Ciência da Computação (PCC) do Departamento de Informática (DIN) da Universidade Estadual de Maringá (UEM), sob a orientação do Prof. Edson Alves de Oliveira Júnior. Você foi previamente selecionado pelo seu perfil/conhecimento/experiência e está sendo convidado a participar desta pesquisa. Essa pesquisa será feita com base em dados coletados a partir de trabalhos práticos. Embora o trabalho prático faça parte da disciplina, você tem o direito de não permitir a utilização dos dados do seu trabalho na pesquisa.

1. Procedimentos

O estudo será realizado com data e hora marcada com os participantes pré-selecionados. Para participar do estudo normalmente será aplicado um formulário de caracterização de perfil, a fim de identificar seu nível de conhecimento/experiência. Em seguida, o estudo será executado de forma individual ou em grupos formados, seguindo sempre o planejamento do estudo feito pela pesquisadora responsável. Caso seja necessário, ao final do estudo será solicitado ao participante que responda um questionário de avaliação sobre a tecnologia de software que está sendo caracterizada/avaliada.

2. Tratamento de possíveis riscos e desconfortos

Serão tomadas todas as providências durante a coleta de dados de forma a garantir a sua privacidade e seu anonimato. Além disso, não existem riscos ou desconfortos que poderão afetar o participante durante a condução do estudo. Ex: fadiga, estresse, mal estar, dentre outros.

3. Benefícios e Custos

Espera-se que, como resultado deste estudo, você possa aumentar seus conhecimentos, de maneira a contribuir para o aumento da qualidade das atividades com as quais você trabalhe ou possa vir a trabalhar. Este estudo também contribuirá com resultados importantes para a pesquisa de um modo geral para o grupo de pesquisa GRSSE. Você não terá nenhum gasto ou ônus com a sua participação no estudo e também não receberá qualquer espécie de reembolso ou gratificação devido à autorização dos seus dados na pesquisa.

4. Confidencialidade da Pesquisa

Toda informação coletada neste estudo é confidencial e seu nome não será identificado de modo algum, a não ser em caso de autorização explícita para este fim. Quando os dados forem coletados, seu nome será removido dos mesmos e não será utilizado em nenhum momento durante a análise ou apresentação dos resultados.

5. Participação

Sua participação neste estudo é muito importante e voluntária, pois requer a sua aprovação para utilização dos dados coletados neste estudo. Segundo a Resolução 466/12 do Conselho Nacional de Saúde (CNS), o respeito devido à dignidade humana exige que toda pesquisa se processe após assentimento livre e esclarecido. Você tem o direito de não querer participar ou de sair deste estudo a qualquer momento, sem penalidades. Em caso de você decidir se retirar do estudo, favor notificar a pesquisadora responsável. Os pesquisadores responsáveis pelo estudo poderão fornecer qualquer esclarecimento sobre o mesmo, assim como tirar dúvidas.

Coordenador do Grupo de Pesquisa GReater: Prof. Edson Alves de Oliveira Júnior - edson@din.uem.br

Pesquisadora do Grupo de Pesquisa GReater: Mestranda Thais Santos Nepomuceno - thais.nepomuceno1@gmail.com

6. Declaração de Consentimento

Declaro que li e estou de acordo com as informações contidas neste documento e que toda linguagem técnica utilizada na descrição deste estudo de pesquisa foi explicada satisfatoriamente, recebendo respostas para todas as minhas dúvidas. Confirmando também que recebi uma cópia deste Termo (TCLE), compreendo que sou livre para não autorizar a utilização dos meus dados neste estudo em qualquer momento, sem qualquer penalidade. Declaro ter mais de 18 anos e concordo de espontânea vontade em participar deste estudo.

Obrigado pela sua colaboração!

Nome:

Questionário de Caracterização de Participante em Estudo Experimental

“Avaliação experimental de uma abordagem para gerenciamento de variabilidades em linhas de produto de software baseadas em UML”

Nas perguntas a seguir, quando duas ou mais alternativas forem válidas, marque a alternativa que mais se aplica ao seu caso.

1. Qual o seu nível de formação?

- | | |
|-------------------------------------|-----------------------------------|
| <input type="checkbox"/> Graduando | <input type="checkbox"/> Graduado |
| <input type="checkbox"/> Mestrando | <input type="checkbox"/> Mestre |
| <input type="checkbox"/> Doutorando | <input type="checkbox"/> Doutor |

2. Em qual setor atua?

- | | |
|---|---|
| <input type="checkbox"/> Acadêmico (ensino) | <input type="checkbox"/> Industrial (empresarial) |
|---|---|

3. Qual o nome da empresa/universidade que atua?

.....

4. Quanto tempo possui de experiência na área que atua?

.....

5. Qual a sua experiência com a notação UML com relação aos diagramas de classes e sequencia?

- Eu **nunca** modelei um software usando a UML.
- Minha experiência com a notação UML é básica.**
Eu modelo software somente no nível dos elementos mais comuns da UML como classes e herança; objetos;
- Minha experiência com a notação UML é moderada.**
Eu modelo software no nível dos elementos da opção anterior, além de: relacionamentos de dependência *include* e *extend*, e *extension points* em diagramas de casos de uso; polimorfismo, associação (uni e bi-direcionais), dependência, agregação e composição em classes; realizações e dependência em componentes, troca de mensagens em diagramas de sequência.

[] **Minha experiência com a notação UML é avançada.**

Eu modelo software que exige a utilização de todos os elementos de diagramas de casos de uso, classes, componentes e sequência, além de outros diagramas da UML como, por exemplo, diagramas de colaboração e atividades.

6. Qual a sua experiência com relação à abordagem de Linha de Produto de Software (LPS) e Gerenciamento de Variabilidade?

[] Eu **nunca** ouvi falar a respeito de LPS.

[] **Já lí**, de forma superficial, algo a respeito de LPS.

[] **Minha experiência com LPS é básica.**

Eu conheço os seguintes conceitos da abordagem: ciclo de desenvolvimento de LPS e suas atividades (engenharia de domínio e engenharia de aplicação). Porém, **não tenho experiência com gerenciamento de variabilidades.**

[] **Minha experiência com LPS é moderada.**

Eu conheço os conceitos da opção anterior, e com relação ao gerenciamento de variabilidades, eu sei o conceito de pontos de variação, variantes e os seus relacionamentos, além dos conceitos de resolução de variabilidades e tempos de resolução (*design time, link time, runtime*, entre outros).

[] **Minha experiência com LPS é avançada.**

Eu conheço os conceitos da opção anterior, além de alguns processos existentes de desenvolvimento de LPS (FODA, PLP, PLUS, PuLSE, entre outros). Com relação ao gerenciamento de variabilidades, eu sei os conceitos da opção anterior, além de: modelos de resolução; abordagens existentes para o gerenciamento de variabilidades, e representação de variabilidades (usando a UML, modelos de características, entre outras).

Assinatura do Participante	Local e Data
<hr/>	Cidade e Data

I. Linha de Produto de Software

Uma linha de produto de software (LPS) corresponde a um conjunto de sistemas de *software* que compartilham características (*features*) comuns e gerenciáveis, que satisfazem a necessidade de um segmento particular ou de uma missão. Este conjunto de sistemas é denominado também família de produtos. Os membros da família são produtos específicos desenvolvidos de maneira sistemática a partir da instanciação de uma infraestrutura comum de uma LPS, chamada núcleo de artefatos.

O núcleo de artefatos é formado por um conjunto de características comuns (similaridades) e características variáveis (variabilidades). As variabilidades podem estar associadas a diferentes níveis de abstração, como a descrição da arquitetura, o código fonte, etc., e auxiliam na geração de produtos específicos distintos em um mesmo domínio e, dessa forma, diminuem o custo e o tempo de desenvolvimento, reduzem riscos e perdas, além de reduzirem o *time to market* e justificarem o retorno de investimento (ROI).

O gerenciamento de variabilidades é uma das atividades mais importantes no gerenciamento de uma LPS, tendo sido foco de atenção por diversos pesquisadores, com pode ser percebido pelas diversas abordagens presentes na literatura.

As variabilidades podem estar associadas a diferentes níveis de abstração, dentre eles, a descrição da arquitetura, a documentação de projeto, o código fonte, o código compilado, o código ligado e o código executável.

Em síntese **variabilidade é a forma como os membros de uma família de produtos podem se diferenciar entre si, ou seja, é o que permite distinguir os diversos produtos de uma LPS.**

A variabilidade é descrita por pontos de variação e variantes:

- **Ponto de variação:** Um **local específico** de um artefato em que uma decisão de projeto ainda não foi tomada, ou seja, foi adiada;
- **Variante:** Corresponde a **uma alternativa** de projeto para resolver uma determinada variabilidade.
- **Restrições entre variantes:** define os relacionamentos entre duas ou mais variantes para que seja possível resolver um ponto de variação ou uma variabilidade.

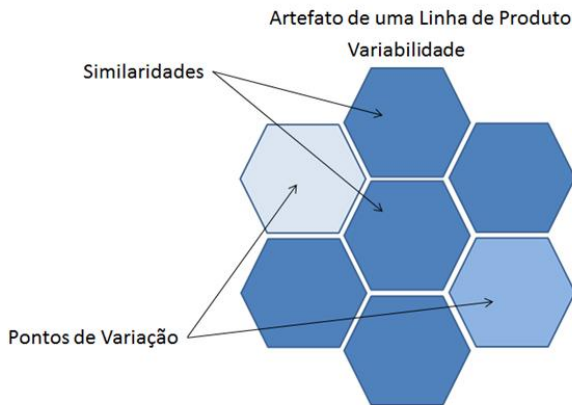
A aplicação destes conceitos é ilustrada na Figura 1.

A maioria das abordagens desenvolvidas para auxiliar no gerenciamento de variabilidades envolve diversos conceitos e modelos de representação. As abordagens “X” e “Y” apresentadas nos documentos 3.2 e 3.3, respectivamente, possuem como base a UML – *Unified Modeling Language*. Tais abordagens utilizam diversos conceitos da UML como o uso de estereótipos e diversos diagramas.

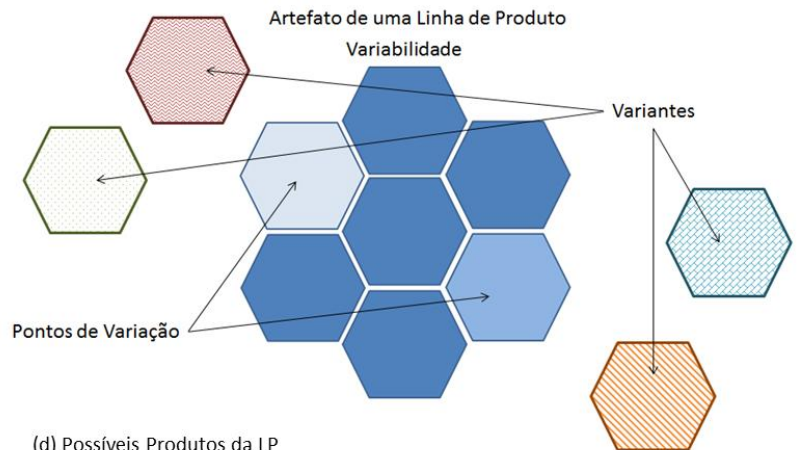
II. Abordagens para Linha de Produto de Software

Para a representação de linhas de produto de software e o gerenciamento de suas variabilidades, como mencionado anteriormente, existem diversas abordagens presentes na literatura. Entre os itens que estas abordagens contemplam, temos os que seguem na Tabela I, que serão posteriormente apresentados para cada abordagem.

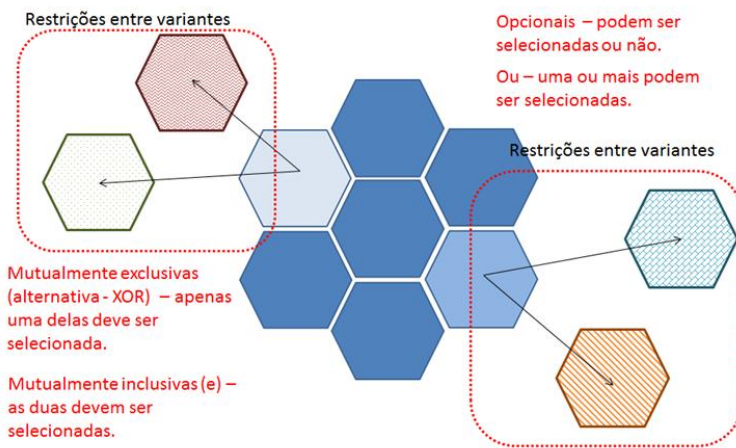
(a) Variabilidade



(b) Pontos de Variação e Variantes



(c) Variantes e suas Restrições



(d) Possíveis Produtos da LP

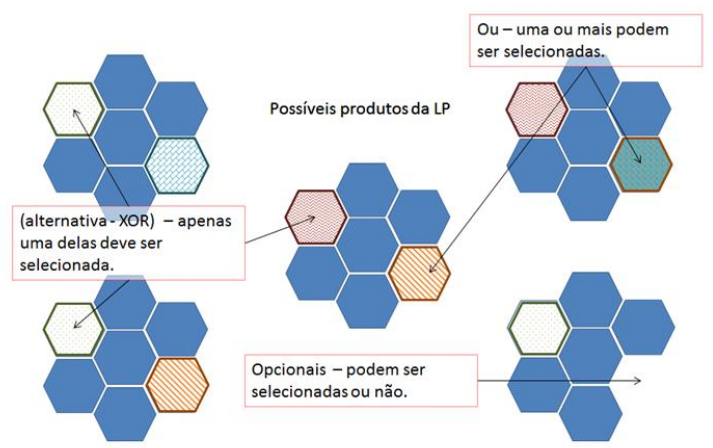


Figura 1 – Exemplo dos Conceitos de Variabilidade, Pontos de Variação, Variantes e Restrições entre Variantes.

Item	Identificação
Baseada em UML	Indica que a abordagem utiliza os modelos UML, meta-atributos, etc., como forma de representação da LPS e de suas variabilidades.
Perfil	Diversas abordagens apresentam um perfil específico que é formado por estereótipos e meta-atributos, geralmente derivados de uma linguagem de modelagem, como a UML.
Processo	O processo contempla a sistematização da utilização de um perfil para o gerenciamento de variabilidades, guiando o usuário no uso das definições do perfil utilizado.
Estereótipos	Estereótipos, como os da UML, são um padrão de mecanismo de extensão e são usados para distinguir diferentes tipos de elementos modelados. Em LPS são ferramentas úteis para identificar variabilidade, seus pontos de variação,

	variantes e outros itens necessários ao seu gerenciamento.
Diretrizes	São os passos sistematizados definidos no processo, que permitem a aplicação facilitada do perfil da abordagem a que corresponde.

III. Rastreabilidade de Variabilidades

Uma LPS é projetada para atender todos os requisitos dos produtos de sua família de produtos. Esses requisitos e a própria LPS podem sofrer mudanças ao longo do tempo em decorrência de vários fatores, como por exemplo, evolução nos requisitos dos produtos, evolução no domínio de mercado para o qual os produtos foram projetados, evolução no processo de construção da LPS, evolução das tecnologias usadas para desenvolver os produtos. Para lidar com essas mudanças, a LPS deve ser modificada e evoluir, visando não ficar obsoleta, e se adequar aos novos requisitos que surgem.

A análise de impacto de mudanças consiste em uma atividade que visa compreender e identificar quais consequências essas mudanças causam ou causarão na LPS. A análise de impacto na LPS pode ser apoiada por meio de relações de rastreabilidade, as quais identificam relacionamentos entre artefatos criados durante todas as fases do desenvolvimento de software.

Abordagem X

I. Visão Geral Abordagem X

Tabela I – Visão Geral Abordagem X.

Abordagem X			
Item	Sim	Não	Observação
Baseada em UML?	X		
Possui um Perfil UML definido?	X		
Possui um Processo definido?		X	
Utiliza Estereótipos?	X		Estereótipos específicos, possuindo variações entre os modelos UML.
Possui Diretrizes?		X	Especificação de uso da abordagem por meio representação textual.
Permite representação formal de variabilidade? Ex.: <i>Object Constraint Language (OCL)</i>	X		Algumas restrições são aplicadas utilizando OCL.

II. Estereótipos e Diretrizes

Nesta seção são apresentados os estereótipos utilizados para a identificação de variabilidades no diagrama de sequência, por meio da Tabela II, em seguida exemplos do uso destes são apresentados, seguidos por especificações textuais do seu uso, identificando de forma conceitual as diretrizes.

Tabela II – Estereótipos da Abordagem X para Classes e Sequência.

Estereótipos da Abordagem X Para Diagrama de Sequência				
Estereótipos	Utilização	Suporta diagramas Classes Sequencia? ?		Exemplos
<< <i>optionalLifeline</i> >>	Indica que o <i>lifeline</i> (linha de vida), nos diagramas de sequência, são opcionais, ou seja, podem ou não existir, de acordo com as variantes selecionados que usam ou não esta <i>lifeline</i> .	Não	Sim	Figura 2 e Figura 3.
<< <i>optionalInteraction</i> >>	Indica que o comportamento descrito na interação é opcional. Todo o fluxo existente em um elemento <i>interactionUse</i> “ref” com este estereótipo poderá ou não existir em um produto.	Não	Sim	Figura 3.
<< <i>optional</i> >>	Usado para representar elementos opcionais, como classes e pacotes, que podem ser incluídos ou não, em produtos específicos.	Sim	Não	
<< <i>variation</i> >>	Indica que a interação é um ponto de variação, que por sua vez estão relacionados a variantes inclusivas ou exclusivas.	Sim	Sim	Figura 2 e Figura 3.

<< <i>variant</i> >>	Indica que a interação é uma variante de comportamento no contexto de uma interação de variação. Apenas uma variante poderá ser selecionada para a resolução de um ponto de variação.	Sim	Sim	Figura 2 e Figura 3.
<< <i>virtual</i> >>	Indica que a interação é uma parte virtual, ou seja, pode ser redefinida por meio de outro diagrama de sequência, e este, por sua vez, pode representar variabilidades. É usado em casos atípicos, em que a LPS necessite modelar um comportamento que pode ser modificado.	Não	Sim	Figura 3.

II.1 Exemplos

Classes

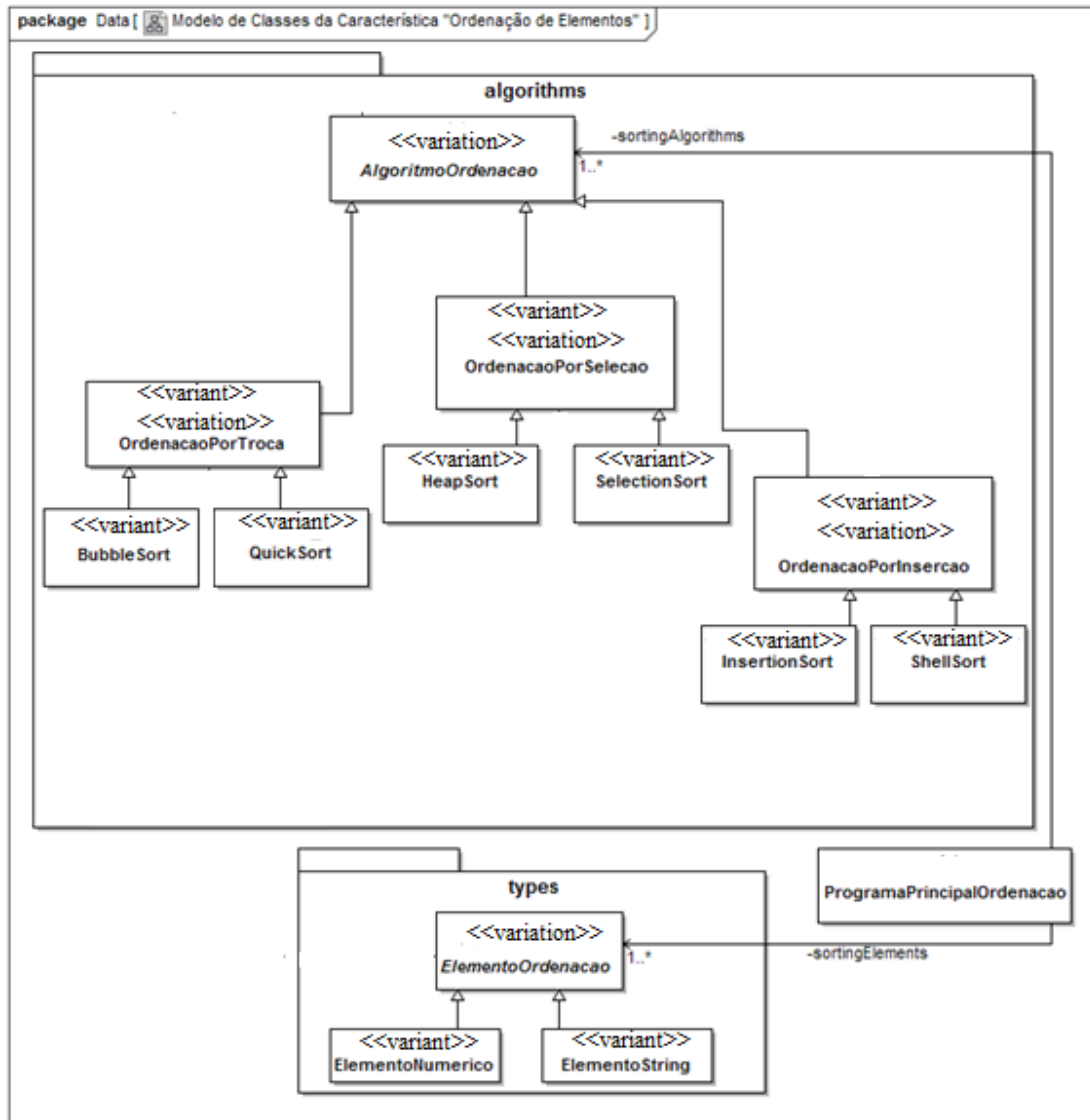
Na Figura 1 observamos a aplicação da abordagem X, e seus elementos. Passamos a analisar cada um deles, bem como as diretrizes presentes no processo da abordagem X, que auxiliam sua utilização em outras LPSs:

A classe **AlgoritmoOrdenacao** identifica uma classe obrigatória e representa também um ponto de variação (<<*variation*>>), com três variantes. Estas variantes estão descritas no elemento comentário, relacionado a classe, por meio do *TaggedValue* (**variants**). As três variantes desta classe são **OrdenacaoPorTroca**, **OrdenacaoPorSelecao** e **OrdenacaoPorInsercao**. Todas estas são estereotipadas como <<*variant*>>, o que indica o tipo de restrição para tais variantes, neste caso, significa que ao menos uma ou todas elas podem solucionar o ponto de variação.

OrdenacaoPorTroca, **OrdenacaoPorSelecao** e **OrdenacaoPorInsercao**, além de variantes (<<*variant*>>), são, por sua vez, pontos de variação (<<*variation*>>), e assim uma delas, ao menos, deve ser selecionada ou todas.

A classe **ProgramaPrincipalOrdenacao**, representa uma classe obrigatória, portanto estará presente em todos os produtos desta LPS.

A classe **ElementoOrdenacao**, também é obrigatória e representa um ponto de variação (<<*variation*>>) e possui duas classes variantes: **ElementoNumerico** e **ElementoString**, marcadas como variantes <<*variant*>>, onde, ambas podem ser selecionadas, ou ao menos uma.



Sequência

Para a identificação da variabilidades e similaridades por meio da Abordagem X aplica-se um dos estereótipos apresentados na Tabela II, para os elementos presentes nas interações necessárias entre os diversos objetos, no diagrama de sequência, de acordo com a especificação da Linha de Produto a ser modelada.

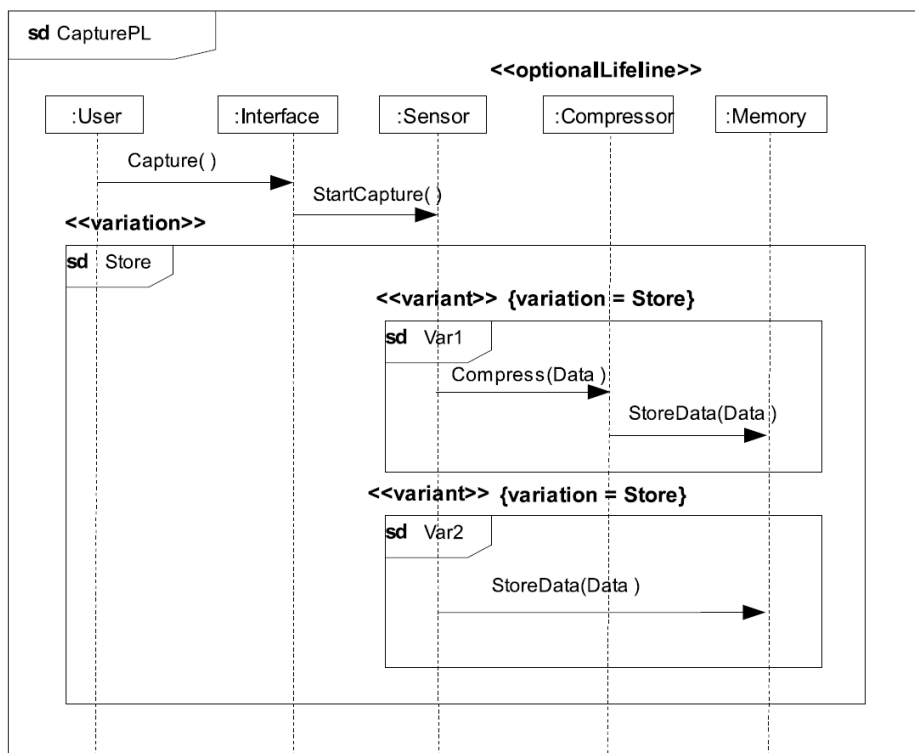


Figura 1 – Exemplo de Modelo de Variabilidade em Diagrama de Sequência com a Abordagem X.

A LPS apresentada corresponde a uma câmera fotográfica, que possui uma interface com usuário (*interface*), o sensor de captura (*sensor*), o compressor das imagens capturadas (*compressor*) e a memória (*memory*) onde tais imagens são armazenadas.

Para o exemplo de linha de produto de "Câmera Digital", os elementos representados pelas linhas de vida (*lifelines*) ou demais elementos sem a aplicação de estereótipos são considerados obrigatórios, ou seja, estarão presentes em todos os produtos.

Na Figura 1, o elemento **Compressor** recebe o estereótipo `<<optionalLifeline>>`, designando que o objeto apresentado por esta linha de vida é opcional, e será inserido ou não, segundo o ponto de variação e variantes a qual ele faz parte. Neste exemplo, somente se a variante de nome **Var1** for selecionada no produto, esta linha de vida deverá fazer parte do produto a ser criado.

A variabilidade **Store** é representada por meio do conjunto de *frame*(quadro), com os *interactionsOperators* (operadores de interação) de valores: **sd Store**, **sd Var1** e **sd Var2**. O *frame* com o *interactionOperator* **sd Store**, corresponde ao ponto de variação, recebendo, desta forma, o estereótipo `<<variation>>`, por sua vez, os *frames* inseridos no *frame* **sd Store**, com o *interactionOperator* **sd Var1** e **sd Var2**, correspondem as variantes, logo recebem o estereótipo `<<variant>>` com o meta-atributo `{variation = Store}`, onde, *Store* corresponde ao nome do ponto de variação a qual aquela variante pertence.

Para variabilidades opcionais, ou variabilidades com variantes que representam um fluxo de interações extenso, utilizam-se o elemento *interactionUse* "ref", conforme Figura 2, onde estes *interactionUses* serão substituídos por um diagrama de sequência especificado separadamente, permitindo o rastreamento dos presentes modelos, apresentados pelo item (a), (b) e (c) com outros diagramas, por meio dos meta-atributos *optionalPart*, *variationPart* e *virtualPart*, respectivamente, inseridos em um comentário da UML.

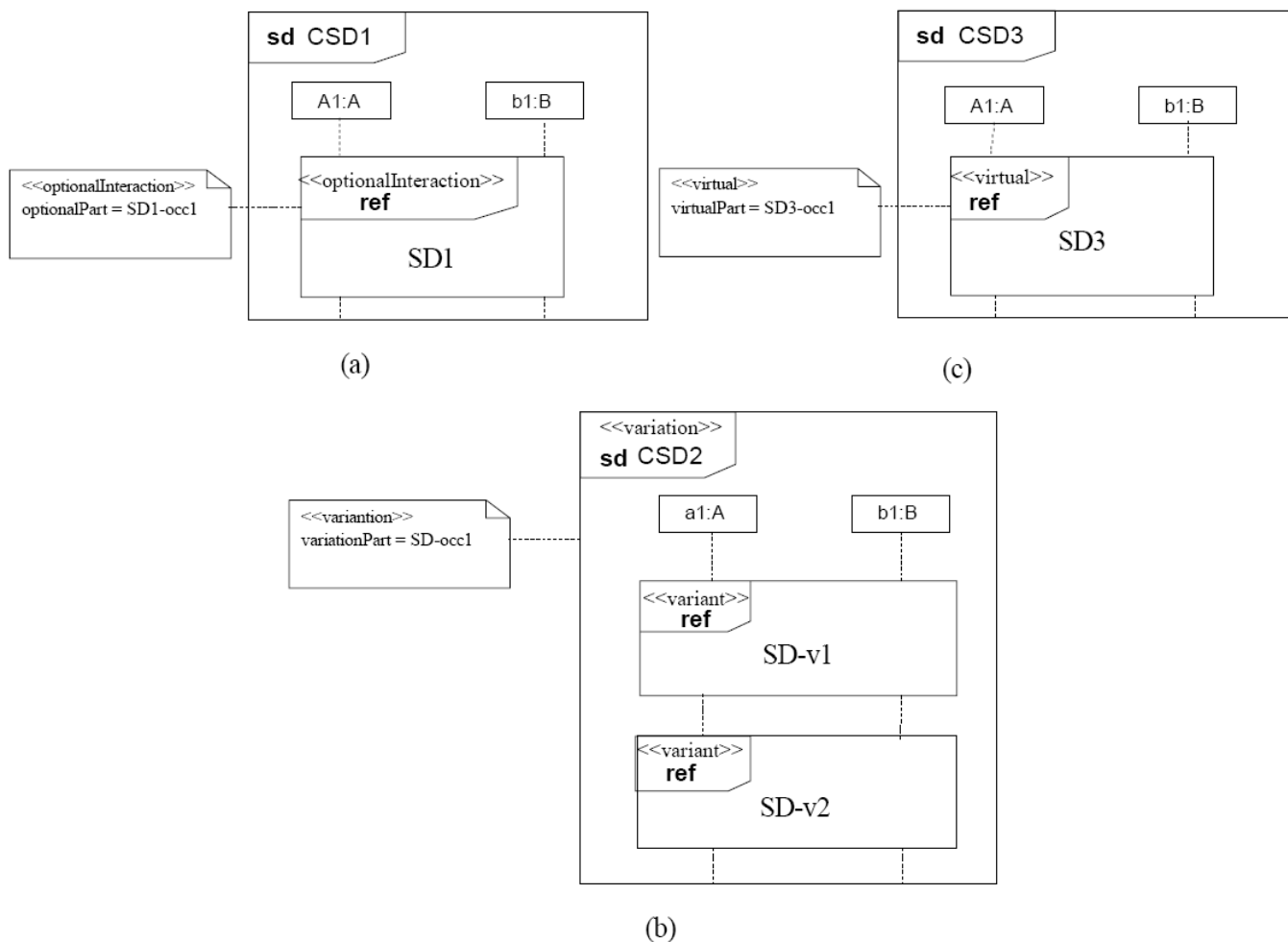


Figura 2 – Tipos de Representação de Variabilidades e seus componentes segundo a Abordagem X.

Na Figura 2, o item (a) corresponde a representação de uma interação opcional, ou seja, de uma variabilidade, que pode ser, inclusive a própria variante, podendo ou não ocorrer. Esta representação, bem como as demais representações (b e c) de variabilidade em interações, podem utilizar *lifelines* que podem ou não estar inseridas no fluxo normal da interação, logo, *lifelines* com o estereótipo `<<optionalLifetime>>` poderão ser utilizados se as variantes que os utilizam forem selecionadas.

Este tipo de variabilidade pode ser identificada mais facilmente quando inserida dentro de um *combinedFragment* do tipo opcional “**opt**”.

Os *lifelines* generalizados com os nomes A1:A e b1:B nos itens (a) e (c) e a1:A e b1:B, no item (b) representam os possíveis *lifelines* para a realização de uma interação que representa uma variabilidade. No exemplo, são demonstrados apenas dois, mas o número pode ser superior (mínimo 2).

O item (c) corresponde a uma interação que pode ser redefinida por outros diagramas de sequência, de acordo com a necessidade de criação de produtos específicos, ou seja, o *interactionUse* “**ref**” será substituído por outro diagrama de sequência que poderá possuir variabilidades ou não. Esta situação ocorre quando a LPS especifique que um comportamento será variável, e logo poderá ser substituído por outro, que por sua vez pode possuir variabilidades.

Ainda na Figura 2, no Item (b), temos a representação da mesma situação presente na Figura 1, onde uma variabilidade especificada pelo *frame* estereotipado como `<<variation>>` possui duas variantes

<<variant>>.

A Figura 3 apresenta a mesma variabilidade do diagrama da Figura 1, porém utilizando os conceitos do diagrama da Figura 2 – Item (b).

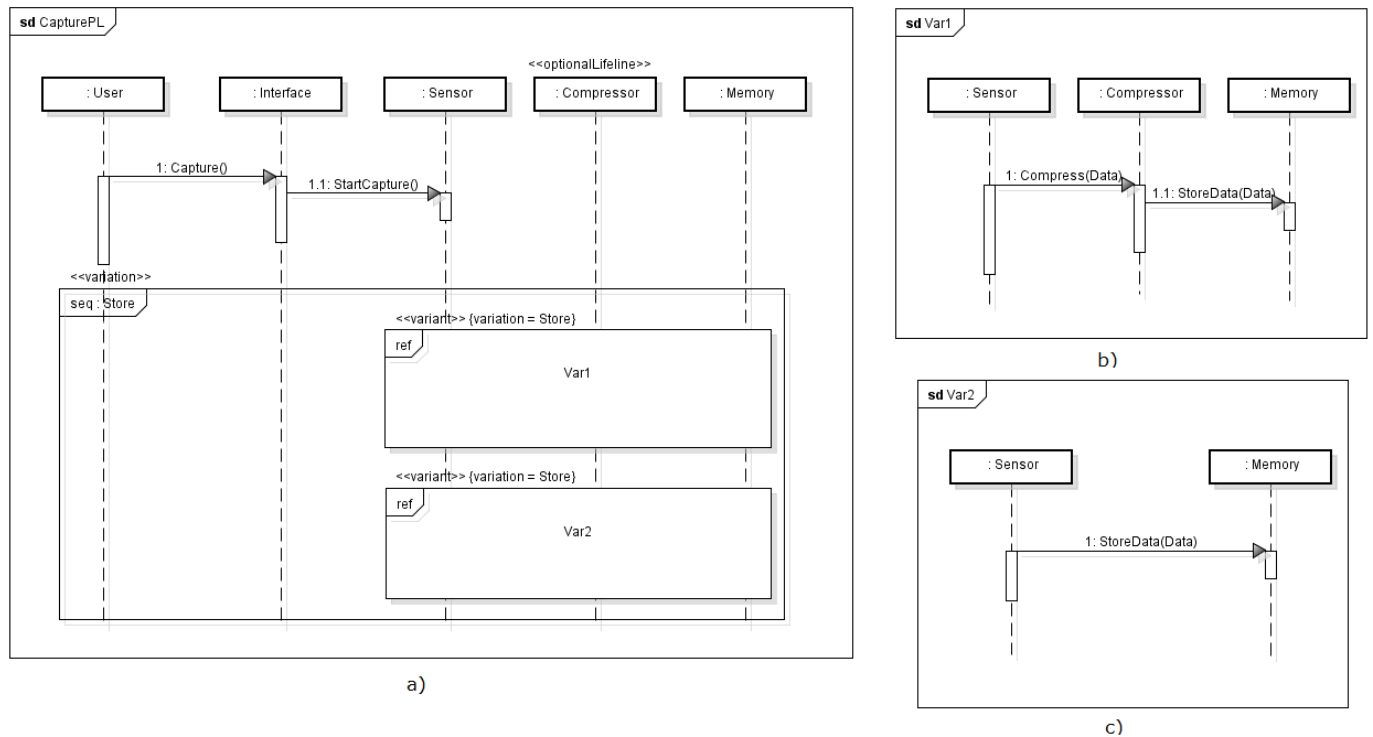


Figura 3 – Exemplo de Modelo de Variabilidade em Diagrama de Sequência com a Abordagem X com conceitos da Figura 2.

Como podemos observar, o diagrama de sequência representado no item a) apresenta uma variação (variabilidade) com duas variantes: Var1 e Var2, correspondentes ao item b) e c).

A variabilidade alternativa, especialmente definidos no recurso de abordagens orientadas a modelos de características é um caso particular do tipo de variabilidade <<variation>> onde cada produto deve escolher uma e apenas uma variante.

Em diagramas de sequência nos quais não aparecem os elementos utilizados por esta abordagem os mesmos devem ser acrescentados. Por exemplo: em um diagrama de sequência em que foi identificado um fluxo opcional, porém o mesmo não está inserido dentro de um *Frame* ou *InteractionUse* “*ref*”, estes elementos deverão ser desenhados para que a representação da variabilidade seja realizada como especificada na abordagem.

Pontos de variação podem ser variantes de outros pontos de variação.

I. Visão Geral Abordagem Y

Tabela I – Visão Geral Abordagem Y

Abordagem Y			
Item	Sim	Não	Observação
Baseadaem UML?	X		
Perfil?	X		
Processo?	X		
Estereótipos?	X		Estereótipos específicos padrões para todos os modelos.
Diretrizes?	X		Diretrizes específicas para cada modelo.

II. Estereótipos e Diretrizes

Nesta seção são apresentados os estereótipos para aplicação em casos de uso, existentes no perfil da abordagem Y por meio da Tabela II.

Tabela II – Estereótipos da Abordagem Y para Casos de Uso.

Estereótipos Abordagem Y para casos de uso	
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 (em UML) entre variantes no qual a variante dependente (origem da dependência) só existirá em uma configuração se a variante relacionada (destino da dependência) existir.
<<mutex>>	Indica um relacionamento de dependência (em UML) entre variantes no qual a variante dependente (origem da dependência) só existirá em uma configuração se a variante relacionada (destino da dependência) obrigatoriamente não existir. São conhecidas como variantes mutuamente exclusivas.

II.1 Notação de Variabilidade

As variabilidades são identificadas através de notas UML, estereotipada com <<variability>>. Nestas

notas estão contidos os meta-atributos que seguem:

- **Name**: nome da variabilidade;
 - **minSelection**: a quantidade mínima de variantes a serem selecionadas;
 - **maxSelection**: a quantidade máxima de variantes a serem selecionadas;
 - **bindingTime**: em qual momento será resolvida esta variabilidade ;
 - **allowsAddingVar**: se permite incluir novas variantes para resolver o ponto de variação; e
 - **variants**: quais as variantes para resolver o ponto de variação (casos de uso ligados ao ponto de variação).
- **realizes+**: representa a coleção de variabilidades de nível superior (menos abstrato) no qual é possível realizar a variabilidade;
- **realizes-**: representa a coleção de variabilidades de nível inferior (mais abstrato) no qual é possível realizar a variabilidade.

II.1 Exemplos

Classes

Na Figura 1 observamos a aplicação da abordagem Y, e seus elementos. Passamos a analisar cada um deles, bem como as diretrizes presentes no processo da abordagem Y, que auxiliam sua utilização em outras LPSs:

A classe **AlgoritmoOrdenacao** identifica uma classe obrigatória (*<<mandatory>>*) e representa também um ponto de variação (*<<variationPoint>>*), com três variantes. Estas variantes estão descritas no elemento comentário, relacionado a classe, por meio do *TaggedValue(variants)*. As três variantes desta classe são **OrdenacaoPorTroca**, **OrdenacaoPorSelecao** e **OrdenacaoPorInsercacao**. Todas estas são estereotipadas como *<<alternative_OR>>*, o que indica o tipo de restrição para tais variantes, neste caso, significa que ao menos uma ou todas elas podem solucionar o ponto de variação.

OrdenacaoPorTroca, **OrdenacaoPorSelecao** e **OrdenacaoPorInsercacao**, além de variantes, são, por sua vez, pontos de variação (*<<variationPoint>>*), e assim, cada uma delas apresenta um comentário, que descreve as suas variantes (*variants*), bem como o nome da mesma (*name*). Neste caso, todas as variantes são marcadas como *<<alternative_OR>>* e, como anteriormente, uma delas, ao menos, deve ser selecionada ou todas.

A classe **ProgramaPrincipalOrdenacao**, representa uma classe obrigatória, portanto é marcada como *<<mandatory>>*, e estará presente em todos os produtos desta LPS.

A classe **ElementoOrdenacao**, também é obrigatória (*<<mandatory>>*) e representa um ponto de variação (*<<variationPoint>>*), logo possui o elemento comentário ligado a ela, com o estereótipo *<<variability>>*, que identifica os dados da variabilidade, que é nomeada, por exemplo, de “sortingelement” e possui duas classes variantes (*variants*): **ElementoNumerico** e **ElementoString**, marcadas como variantes alternativas *<<alternative_OR>>*, onde, ambas podem ser selecionadas, ou ao menos uma.

Desta forma, as variabilidades são identificadas por meio do comentário UML, estereotipada com *<<variability>>*.

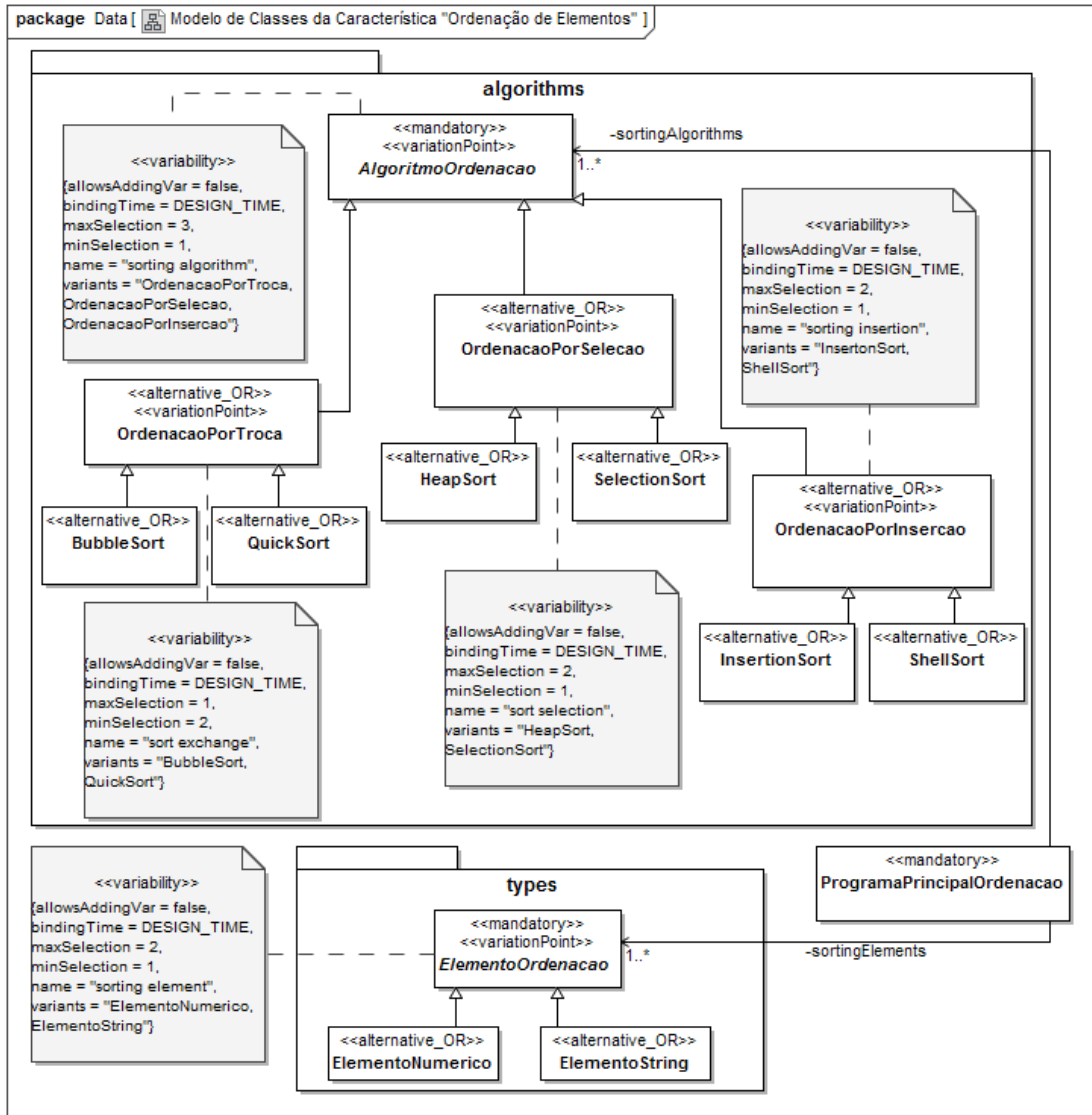


Figura 1 – Exemplo de Modelo de Variabilidade em Diagrama de Classes com a Abordagem Y.

Sequência

II. Estereótipos e Diretrizes

Nesta seção são apresentados os estereótipos para aplicação em diagrama de sequência, existentes no perfil da abordagem Y por meio da Tabela II, em seguida são apresentados exemplos do uso destes, seguidos pelas diretrizes para cada tipo de modelo.

Tabela II – Estereótipos da Abordagem Y para Sequência

Estereótipos Abordagem Y Para Sequência		
Estereótipo	Utilização	Exemplo
<<variationPoint>>	Representa o local em que ocorre uma variabilidade. Um ponto	Figura 1 e

	de variação está sempre associado a uma ou mais variantes.	3.
<<optional>>	A variante pode ou não estar presente na configuração de um produto da linha de produto. Variantes opcionais podem estar associadas a um ponto de variação, ou ser a própria variabilidade.	Figura 2.
<<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.	Figura 3.
<<alternative_XOR>>	Estão sempre associadas aos pontos de variação. Somente uma das variantes deverá ser escolhida para resolver o ponto de variação.	Figura 1.
<<variability>>	Indica uma variabilidade existente em um modelo UML.	Figura 1, 2 e 3.

II.1 Exemplos

Diagrama de Sequência

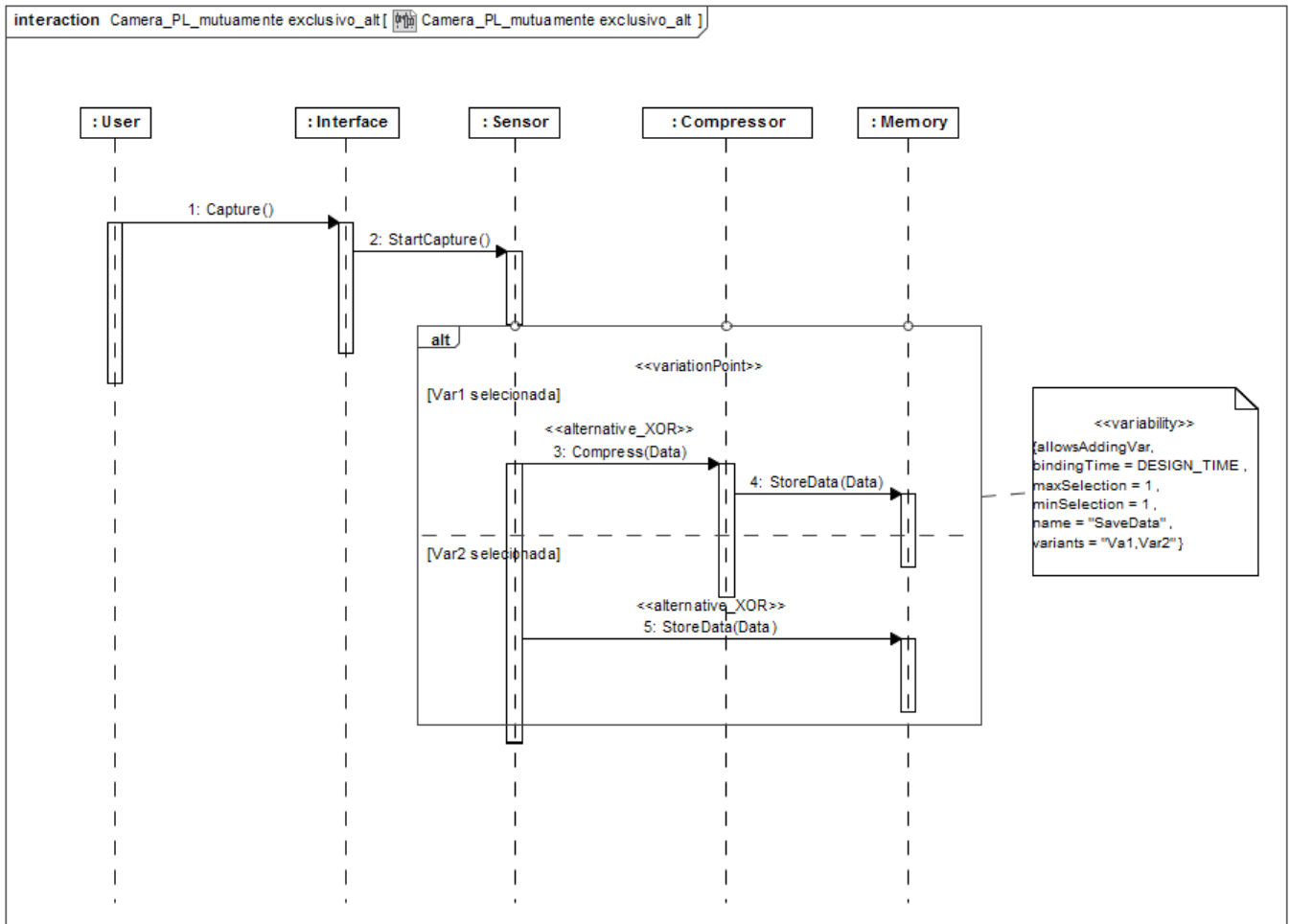


Figura 1 – Exemplo de Modelo de Variabilidade em Diagrama de Sequência com a Abordagem Y (variantemutuamente exclusiva).

Na Figura 1 observarmos a aplicação da abordagem Y, e seus elementos. Passamos a analisar cada um deles, bem como as diretrizes presentes no processo da abordagem Y, que auxiliam sua utilização em outras LPSs.

A LPS apresentada corresponde a uma câmera fotográfica, que possui uma interface com usuário (*interface*), o sensor de captura (*sensor*), o compressor das imagens capturadas (*compressor*) e a memória (*memory*) onde tais imagens são armazenadas.

As mensagens **Capture()** e **StartCapture()** são obrigatórias, e por convenção, elementos obrigatórios não deverão receber estereótipos.

O *CombinedFragment* com o *interactionOperator* "alt", indica que apenas um fluxo pode ser selecionado para a execução, logo, uma câmera que possuir ambas as variantes; representadas no conteúdo do *CombinedFragment* estereotipado por <<variationPoint>>, e também especificado como variabilidade, pelo comentário UML (<<variability>>), poderá ter ou não a opção de

compressão (mensagem **Compress(Data)**) de acordo com a solução para o ponto de variação.

Ainda no *CombinedFragment* primeira mensagem trocada internamente, em cada operando (Var1 e Var2) são estereotipados de acordo com o tipo de variação que sofrem. Nesta LPS, são estereotipados como **<<alternativa_XOR>>**, assim, no mínimo e no máximo haverá somente uma variante selecionada para o ponto de variação. No exemplo, são apresentadas duas opções, mas poderiam ser inseridos um número maior de variantes.

As variabilidades são identificadas por meio do comentário UML, estereotipada com **<<variability>>**. Estes comentários são inseridos em todas as variabilidades.

Diretrizes para Diagrama de Sequência -As diretrizes especificadas para auxiliar na identificação das variabilidades em diagramas de sequência são expressas abaixo:

SQ.1 Elementos de modelos de diagramas de sequência como *CombinedFragment* e possuidores do *interactionOperator* do tipo **"alt"** (*alternative*), ou seja, variantes exclusivas, sugerem pontos de variação marcados com **<<variationPoint>>** e serão relacionados a um comentário da UML especificando a variabilidade (**<<variability>>**). As variantes, correspondentes as mensagens serão estereotipadas como **<<alternative_XOR>>**;

SQ.2 Em diagramas de sequência, as duas possíveis ocorrências a seguir, sugerem pontos de variação opcionais:

- a) Elementos de modelos de diagramas de sequência como o *CombinedFragment* e possuidores do *interactionOperator* do tipo **"opt"** (*optional*), ou seja opcionais, sugerem variantes opcionais, sendo estereotipados como **<<optional>>**, e são relacionados a um comentário da UML especificando a variabilidade (**<<variability>>**). Os *lifelines* contidos neste *CombinedFragment* e que fazem parte da variabilidade deverão ser estereotipados também como **<<optional>>**;
- b) Troca de mensagens entre dois objetos não obrigatórios, ou entre um objeto obrigatório e outro não, sugerem uma variante opcional, estereotipada como **<<optional>>** e estarão relacionados a um comentário da UML especificando a variabilidade (**<<variability>>**). A(s) *lifeline(s)* correspondente(s) a esta variante será(ão) estereotipada(s) também como **<<optional>>**.

SQ.3 O elemento *interactionUse* "ref" sugere variantes alternativas inclusivas, onde um ou mais diagramas de sequência podem ser selecionados como variantes, para resolver um ponto de variação. Logo, o *interactionUse* "ref" será estereotipado como ponto de variação **<<variationPoint>>**, e também com o tipo de variabilidade (**<<alternative_OR>>**), estando relacionado ainda, a um comentário da UML, que identifica os elementos da variabilidade (**<<variability>>**);

SQ.4 as mensagens (*messages*) que são independentes dos fluxos contidos no *CombinedFragment* "alt", "opt", *interactionUse* "ref", ou não estejam relacionadas diretamente a uma variabilidade e seus elementos, são mantidas sem estereótipos e consideradas assim, obrigatórias;

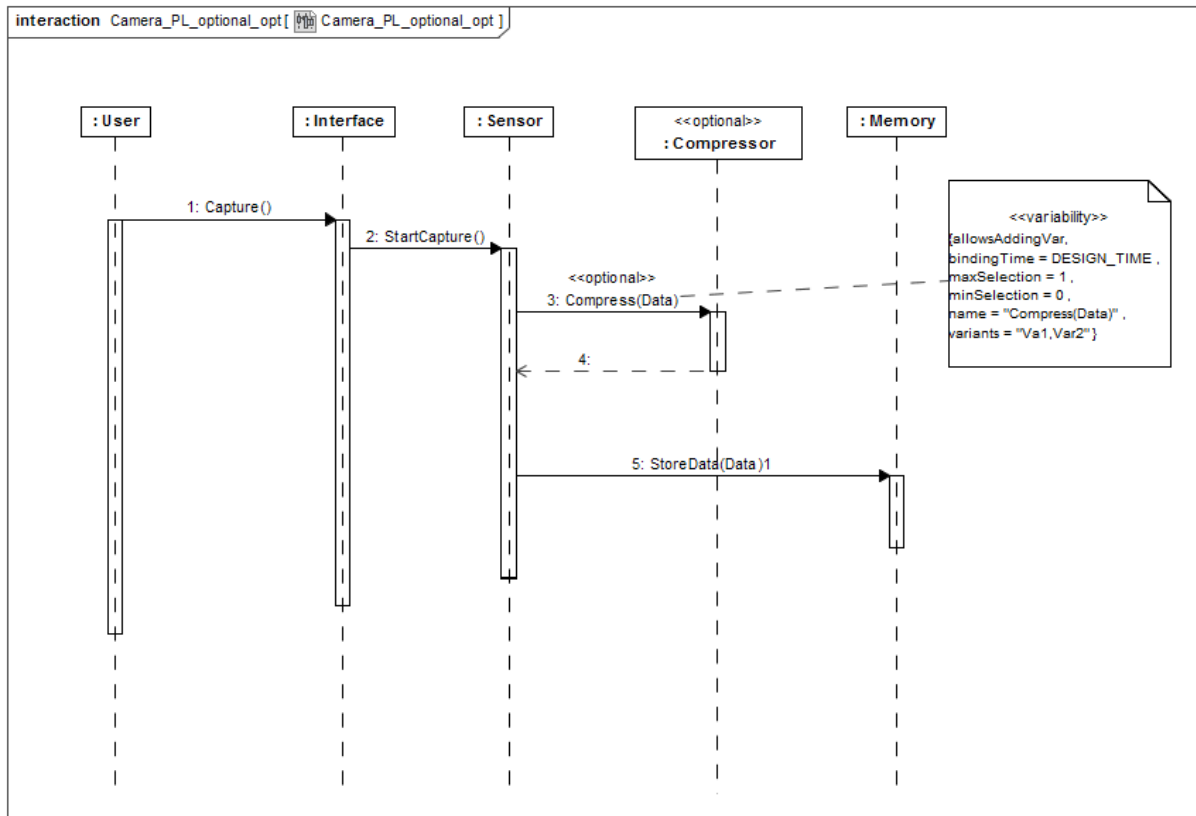
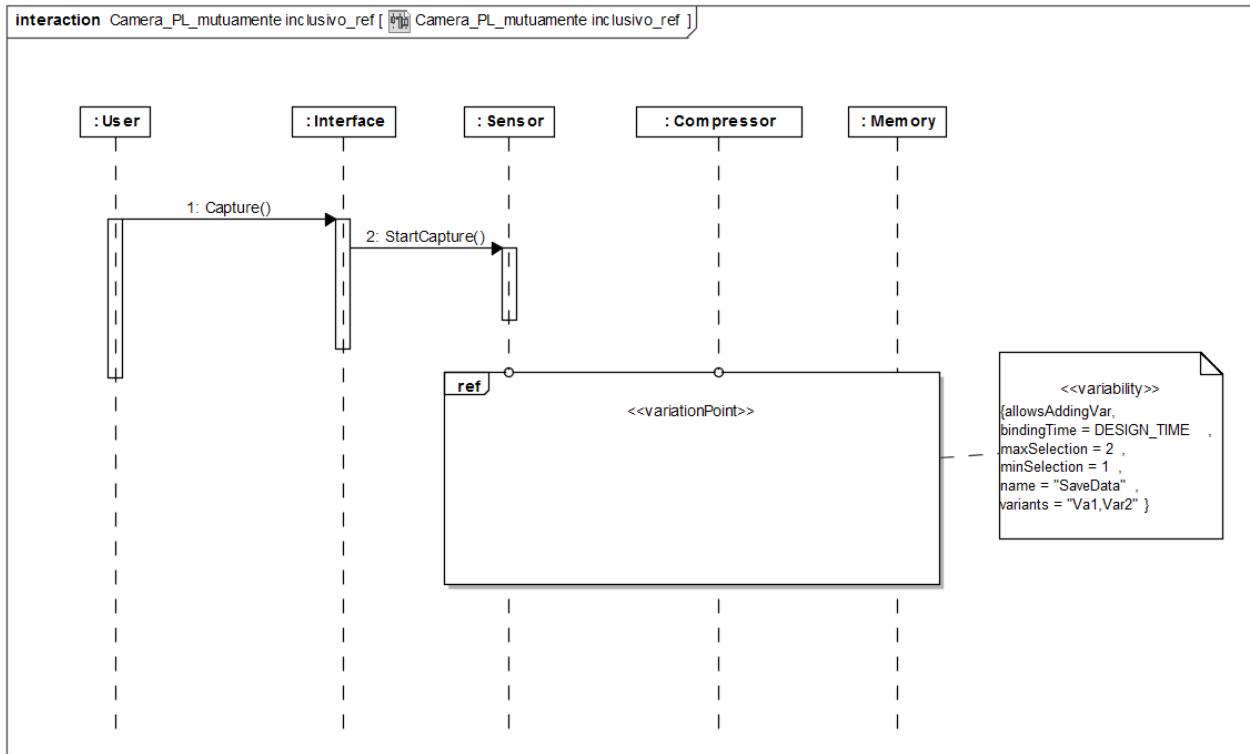


Figura 2 – Exemplo de Identificação de Variabilidade em Diagrama de Sequência (varianteopcional).

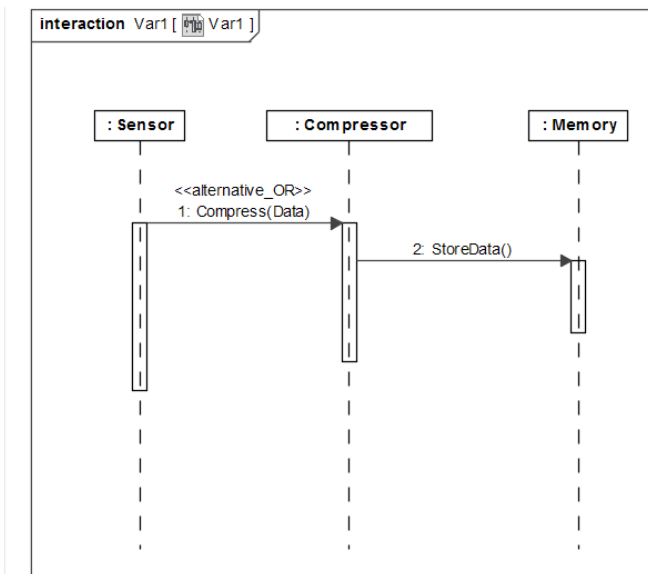
Na Figura 2 é apresentada a mesma funcionalidade da Figura 1, entretanto a compressão dos dados é modificada para atender uma nova LPS onde passa a ser considerado opcional. Assim, a mensagem **Compress(Data)** receberá o estereótipo <<optional>> e estará relacionado a um comentário da UML que define a variabilidade (<<variability>>).

A mensagem **Compress(Data)** é estereotipada como <<optional>>, pois a variante corresponde a apenas uma mensagem, caso haja mais de uma mensagem de fluxo opcional, apenas a primeira deverá ser estereotipada como <<optional>>. E como as variantes opcionais correspondem, na maioria dos casos, também ao ponto de variação, não necessitam ser estereotipadas como tal.

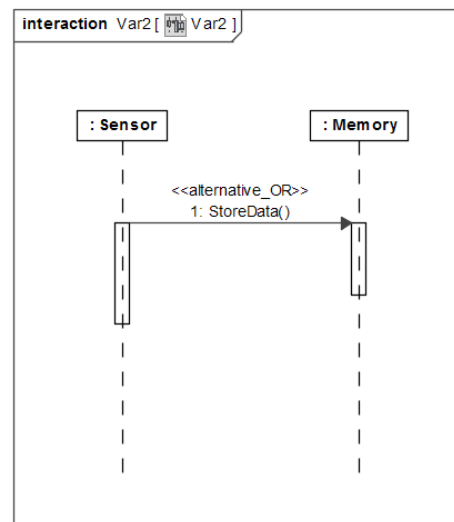
O elemento **Compressor**, que pertence a variabilidade opcional deve receber o estereótipo <<optional>>. Na existência de mais de um elemento representado por meio de uma *lifeline*, como o **Compressor**, estes fizerem parte de uma variabilidade opcional, deverão também, receber o estereótipo de opcional.



a)



b)



c)

Figura 3 – Exemplo de Identificação de Variabilidade em Diagrama de Sequência (variante alternativa inclusiva).

A Figura 3 apresenta um exemplo para a mesma variabilidade das Figuras 1 e Figuras 2, alterada para uma LPS que a considere alternativa. Vale lembrar que não há alterações na forma como uma variabilidade se apresenta em uma LPSs, ou seja, se identificada como alternativa não poderá se tornar opcional, a não ser que sejam modificadas para a criação de uma nova linha de produto.

Como podemos notar, o diagrama de sequência principal - item a) – corresponde ao diagrama da variabilidade, demonstrando o comentário UML estereotipado como **<<variability>>**, bem como

apresentado o *interactionUse* "ref" estereotipado de acordo com o tipo de variação que esta sofre, no caso, corresponde a uma variante inclusiva <<alternative_OR>>.

A Figura 4 apresenta o caso em que um diagrama de seqüência (*Frame* "sd") foi inserido dentro de outro diagrama, e em sua composição são apresentados dois *interactionUse* "ref".

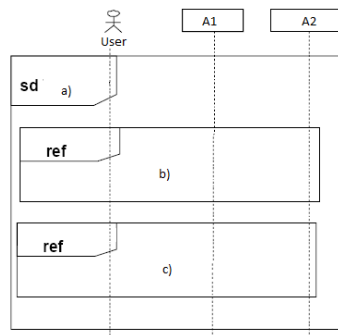


Figura 4 – Exemplo de Identificação de Variabilidade em Diagrama de Sequência (variante alternativa inclusiva) - *Frame*.

O item a) corresponde a um *Frame*, elemento que identifica um diagrama de seqüência. Inserido dentro deste encontram-se dois *interactionUse* "ref", identificados como item b) e c). Nesta representação, o *Frame* deverá ser estereotipado como ponto de variação <<variationPoint>> e estará relacionado ao comentário UML estereotipado como <<variability>>, contendo os dados da variabilidade, e os *interactionUse* b) e c), serão ambos estereotipados como alternativos exclusivos (<<alternative_XOR>>), pois condizem com as variantes. Logo, a quantidade de diagrama de seqüência que podem ser utilizados nesta variabilidade é definida pela quantidade de *interactionUse* "ref" presentes.

Nesta mesma representação, de acordo com a descrição da linha de produto, o *Frame* a) pode não representar uma variabilidade. Ou seja, somente os elementos internos deste (b) e c)) podem ser considerados variabilidades opcionais, logo os dois ou um deles receberá o estereótipo <<optional>> e estará relacionado a um comentário da UML que define a variabilidade (<<variability>>).

A Figura 5 apresenta exemplos de identificação de variabilidades nos elementos da Figura 4.

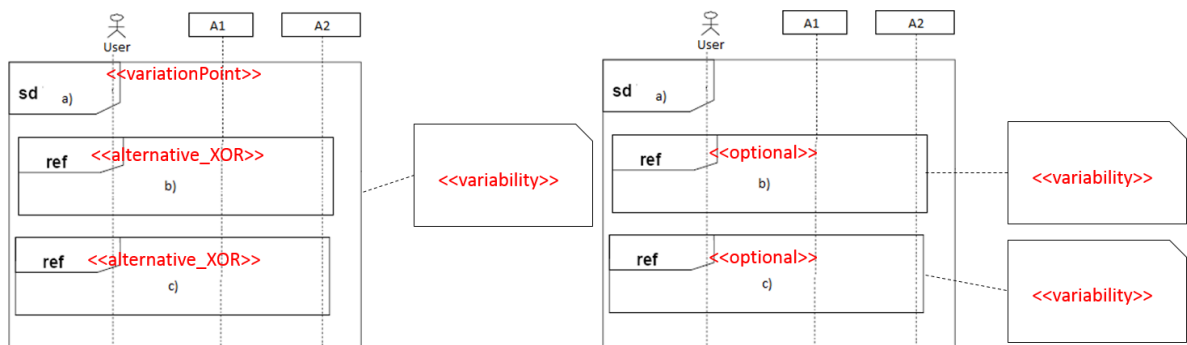
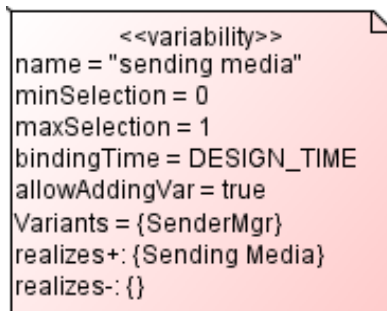


Figura 5 – Exemplo de Identificação de Variabilidade em Diagrama de Sequência com o Elemento *Frame*.

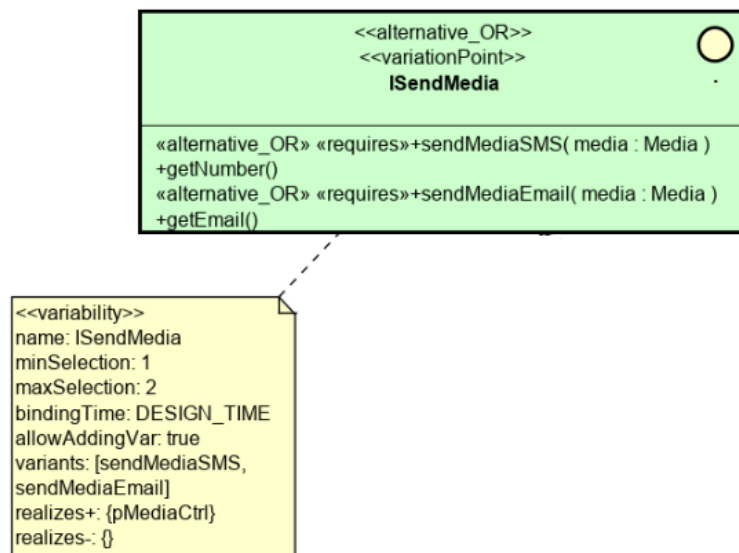
Rastreabilidade

A rastreabilidade para identificação dos níveis de variabilidade pode ser identificada por meio do meta-atributo realizaes contido na notação de comentários UML, com o estereótipo variability.

Para a identificação de rastreabilidade em níveis de maior abstração, o meta-atributo realizaes contém um sinal de adição (+). Para os níveis de menor abstração, o item realizaes contém um sinal de subtração (-). Assim, a rastreabilidade das resoluções de variabilidade se torna bidirecional.



Exemplo Rastreabilidade:



Para determinar quais elementos essa variabilidade (ISendMedia) pode interferir, é necessário olhar o realizes na notação de variabilidade. Caso você queira saber onde essa variabilidade interfere em um nível de diagrama mais abstrato (por exemplo, no diagrama de classes), olhe para o realizes +, nesse caso está escrito PMediaCtrl, então é preciso procurar uma notação de variabilidade com este nome no diagrama de classes. A classe ou interface que essa notação estiver associada é o lugar que ocorre variação caso o componente ISendMedia seja alterado. O mesmo deve ser feito para diagramas menos abstratos, porém olhando para realizes -.

II.2 Diretrizes para configuração de produtos com diagramas de classes

D.1 Selecione para o produto a ser configurado todas as classes que são obrigatórias, demarcadas com o estereótipo <<mandatory>>;

D.2 Resolva os pontos de variação, com o estereótipo <<variation point>>, de acordo com a notação de variabilidade, observando a quantidade mínima e máxima de variantes que devem ser escolhidas;

D.2.1 Observe a relação das variantes por meio de seus estereótipos. Se forem <<alternative_OR>> pelo menos uma das variantes deverá ser escolhida para resolver o ponto de variação, se <<alternative_XOR>> somente uma das variantes deverá ser escolhida para resolver o ponto de variação;

D.3 Uma variante V_1 que ao ser selecionada para fazer parte de um determinado produto, tiver um relacionamento com outra variante V_2 com o estereótipo <<requires>>, essa variante (V_2) também deverá fazer parte do produto;

D.4 Uma variante V_1 que ao ser selecionada para fazer parte de um determinado produto, tiver um relacionamento com outra variante V_2 com o estereótipo <<mutex>>, essa variante (V_2) não poderá fazer parte do produto.

II.3 Diretrizes para configuração de produtos com diagramas de sequencia

D.1 Selecione para o produto a ser configurado todas as Lifelines que são obrigatórias bem como as trocas de mensagens obrigatórias, demarcadas com o estereótipo <<mandatory>>;

D.2 Resolva os pontos de variação, com o estereótipo <<variation point>>, de acordo com a notação de variabilidade, observando a quantidade mínima e máxima de variantes que devem ser escolhidas;

D.2.1 Observe a relação das variantes por meio de seus estereótipos. Se forem <<alternative_OR>> pelo menos uma das variantes deverá ser escolhida para resolver o ponto de variação, se <<alternative_XOR>> somente uma das variantes deverá ser escolhida para resolver o ponto de variação;

D.3 Uma variante V_1 que ao ser selecionada para fazer parte de um determinado produto, tiver um relacionamento com outra variante V_2 com o estereótipo <<requires>>, essa variante (V_2) também deverá fazer parte do produto;

D.4 Uma variante V_1 que ao ser selecionada para fazer parte de um determinado produto, tiver um relacionamento com outra variante V_2 com o estereótipo <<mutex>>, essa variante (V_2) não poderá fazer parte do produto.

II.3 Diretrizes para rastreabilidade

D.1 Identifique a notação de comentário UML, com o estereótipo <<variability>> no elemento ao qual pretende-se rastrear;

D.2 Para a identificação de rastreabilidade em níveis de maior abstração, procure o meta-atributo “realizes+”;

D.3 Para rastrear elementos em diagramas de menor abstração, identifique o “realizes -”;

D.4 Vá até o diagrama correspondente e procure pelo nome identificado no meta-atributo realizes. Você terá rastreado um elemento de um diagrama para o outro!

Arcade Game Maker:

Descrição Geral da Linha de Produto

I. Identificação

A linha de produto de software (LPS) *Arcade Game Maker* (AGM) produz uma série de jogos arcade – ou seja, produtos com um ou mais jogos. Cada jogo é jogado por um único jogador que controla, parcialmente, os objetos que se movem. O objetivo é marcar pontos acertando obstáculos estáticos. Os jogos vão desde aqueles com obstáculos baixos até obstáculos altos e estão disponíveis para uma variedade de diferentes plataformas.

II. Similaridades e Variabilidades

Nesta seção são apresentadas as principais similaridades da LPS, ou seja, os aspectos comuns a todos os produtos desta LPS, bem como as variabilidades, que representam os aspectos que diferem de um produto em relação ao outro.

II.1 Similaridades

- Todos os jogos possuem elementos que recebem a interação do usuário (*Vector*);
- Todos os jogos possuem mesa (*Board*); e
- Os elementos não mencionados nas similaridades ou variabilidades da LPS são considerados obrigatórios para o correto funcionamento dos produtos.

II.2 Variabilidades

- Cada jogo, ao ser selecionado deverá ter sua mesa construída (*buildGameBoard*), logo a seleção de um ou mais jogos (mínimo 1 e máximo 3) necessita da construção de sua respectiva mesa. Assim, tal construção ocorrerá de acordo com o(s) jogo(s) que o produto suporta, devendo possuir no mínimo 1 mesa e máximo 3.
- A construção das mesas do jogo *Brickles* (*buildGameBoard()_Brickles*) e *Pong* (*buildGameBoard()_Pong*) poderá suportar ou não a existência de *pucks* de provisão (*PuckSupply*), que correspondente ao número de bolas no jogos.
- Todo o jogo deverá ser salvo (*saveGame*), para isso é necessário que a funcionalidade dos jogos presentes no produto estejam presentes, por exemplo, se o jogo *Brickles* estiver presente no produto, a opção para salvar este jogo e seu respectivo menu deverá estar acessível, e assim por diante. Desta forma, esta funcionalidade estará presente no produto, de acordo com o jogo que este possui, sendo no mínimo 1 e máximo 3; e
- Todo o jogo deverá ser carregado, para isso, cada método correspondente ao carregamento do jogo (*loadGame*) deverá estar presente nos produtos, de acordo com a seleção de seus respectivos jogos (mínimo 1 e máximo 3). A seleção desta funcionalidade é análoga a opção de salvar o jogo, a qual estará presente, de acordo com o jogo que compor o produto.

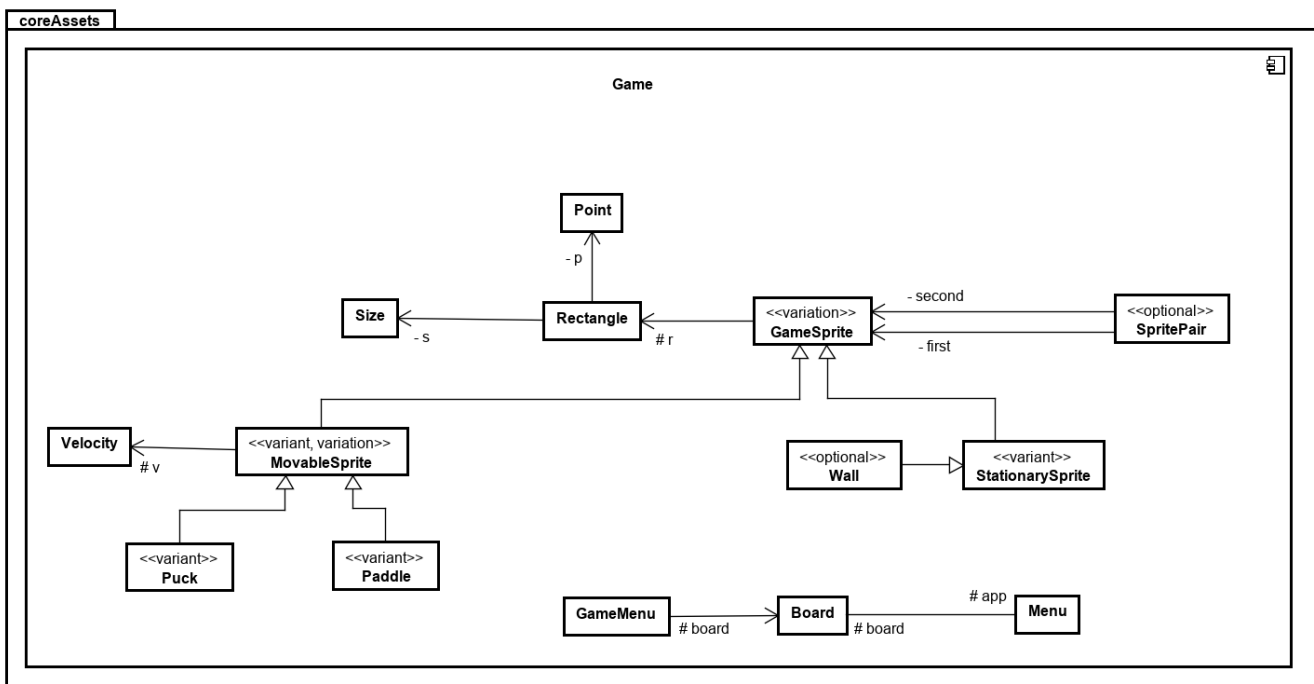
III. Descrição dos Elementos

Elemento	Descrição
<i>addMovablePiece()</i>	Adicionar peça móvel
<i>addStationaryPiece()</i>	Adicionar peça fixa
<i>Board</i>	Mesa/tabuleiro de jogo
<i>BottomPaddle()</i>	Elemento que movimenta a Puck do jogo, localizado na parte inferior da PongBoard.
<i>BowlingBoard</i>	Mesa/tabuleiro do jogo Bowling
<i>BowlingGameMenu</i>	Menu do Jogo de Bowling (Boliche)
<i>BricklesGameMenu</i>	Menu do Jogo Brickles
<i>buildGameBoard()</i>	Construir mesa/tabuleiro do jogo
<i>EndOfAlley()</i>	Fim da pista de boliche
<i>GamePlayer</i>	Jogador
<i>getGameOver()</i>	Obter fim de jogo
<i>getSaveData()</i>	Obter dados salvos
<i>Lane()</i>	Lona
<i>loadGame()</i>	Carregar jogo
<i>movableComponent</i>	Componente móvel
<i>New</i>	Novo
<i>playGame()</i>	Jogar jogo
<i>PongGameMenu</i>	Menu do Jogo Pong
<i>PuckSupply</i>	PuckRepresenta o principal elemento de um jogo como, por exemplo, a bola que derruba os BowlingPins no jogo Bowling, a bolinha que destrói os BrickPile no jogo Brickles, etc. PuckSupply é a quantidade de Pucks que o jogador tem direito em um jogo.
<i>rackPins()</i>	Empilhar pinos
<i>RecordStore</i>	Dados armazenados
<i>removeAllElements()</i>	Remover todos os elementos
<i>saveGame()</i>	Salvar jogo
<i>setMessage()</i>	Definir mensagem
<i>setSaveData()</i>	Definir dados salvos
<i>startMoving()</i>	Iniciar movimento
<i>stationaryComponents</i>	Componentes fixos
<i>TopPaddle()</i>	Elemento que movimenta a Puck do jogo, localizado na parte superior da PongBoard.
<i>Vector</i>	Vetor

Arcade Game Maker
Modelo de Identificação de Variabilidades (Abordagem X)

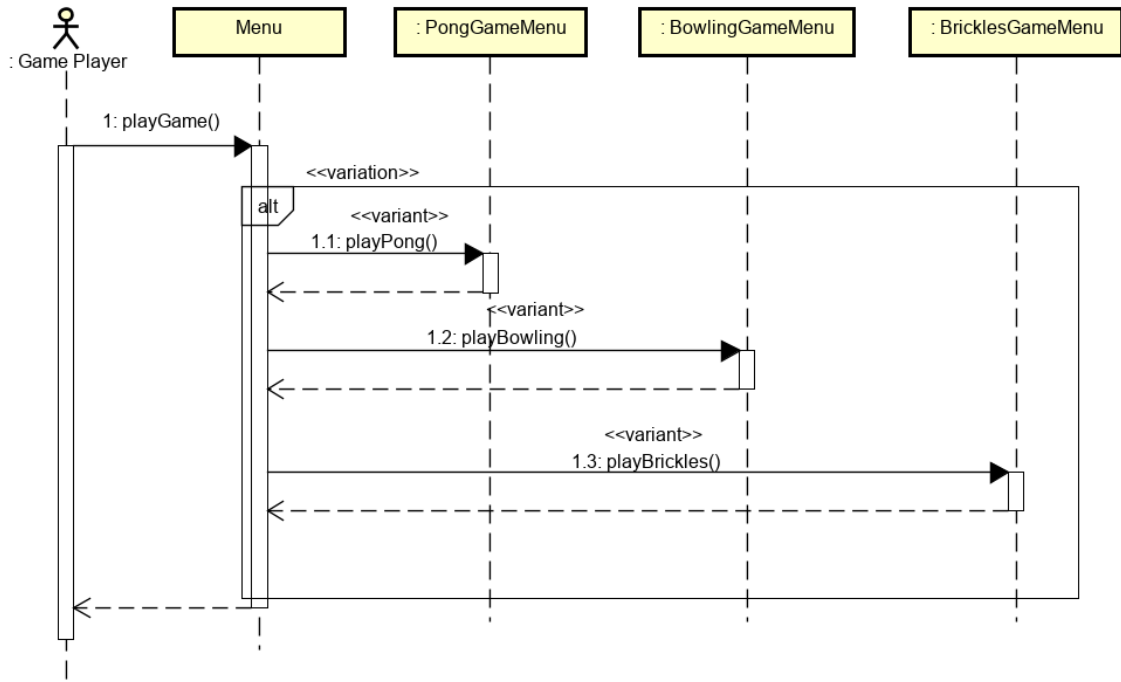
I. Configuração de Produto

I.1 Com base no Diagrama de Classes fornecido, na Descrição Geral da Linha de Produto Arcade Game Maker do documento 4; modele um produto específico possível de ser extraído do diagrama apresentado modelado conforme a **Abordagem X**.



I.2 Com base no Diagrama de Componentes fornecido, na Descrição Geral da Linha de Produto Arcade Game Maker do documento 4; modele um produto específico possível de ser extraído dos diagramas de sequencia apresentados modelado conforme a **Abordagem X**.

sd Sequence Diagram0

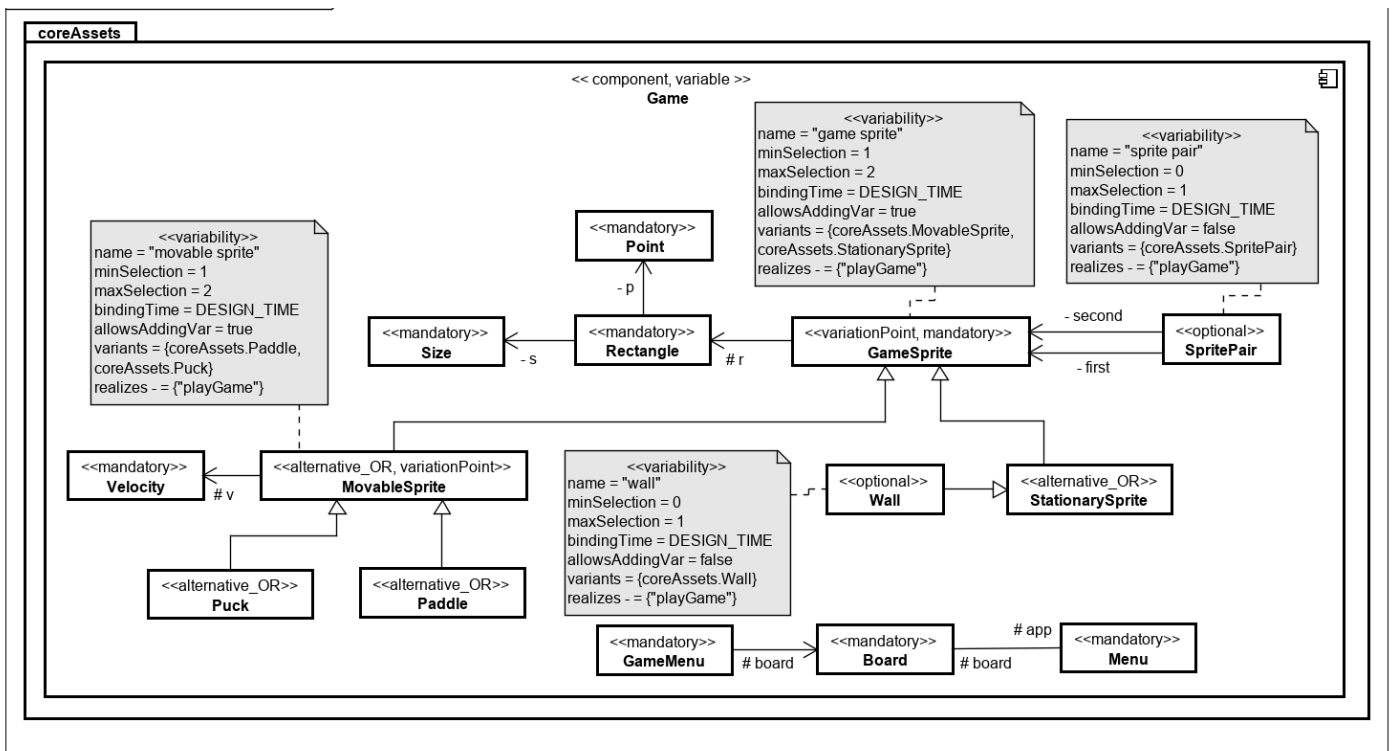


Doc 5

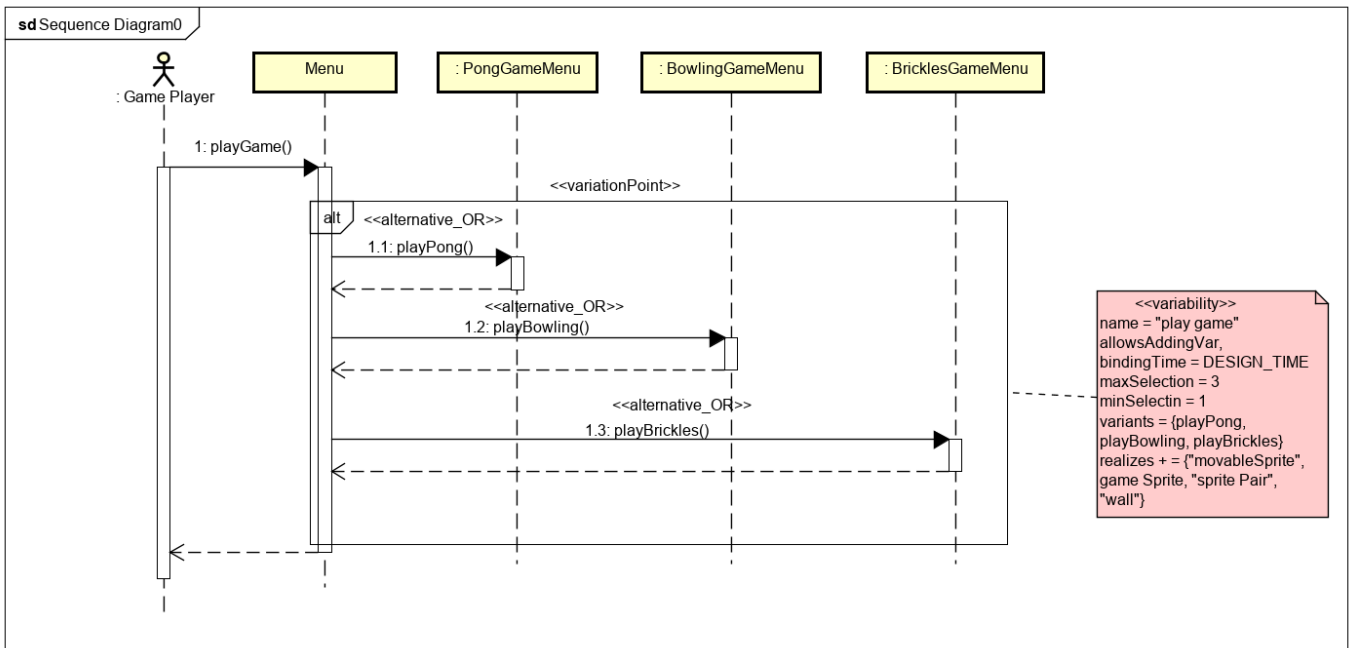
Mobile Media:
Modelo de Identificação de Variabilidades (Abordagem Y)

I. Configuração de Produtos

I.1 Com base no Diagrama de Classes fornecido, na Descrição Geral da Linha de Produto Arcade Game Maker do documento 4; modele um produto específico possível de ser extraído do diagrama apresentado modelado conforme a **Abordagem Y**.



I.2 Com base no Diagrama de Componentes fornecido, na Descrição Geral da Linha de Produto Arcade Game Maker do documento 4; modele um produto específico possível de ser extraído dos diagramas de sequencia apresentados modelado conforme a **Abordagem Y**.



I.3 Supondo que em uma configuração de produto, as funcionalidades relacionadas ao Game Sprite fossem excluídos do diagrama de classes, as alterações/impactos no diagrama de sequencia são possíveis de serem identificados com apoio da abordagem X (Considerando que o digrama de classes é mais abstrato que o diagrama de sequencia).

- Concordo Totalmente
- Concordo Parcialmente
- Discordo Parcialmente
- Discordo Totalmente

I.4 Caso a funcionalidade **Play Game** ilustrado no diagrama de sequencia não existisse em uma possível configuração de produto, é possível identificar os impactos que essa mudança causaria no diagrama de classes (Considerando que o digrama de classes é mais abstrato que o diagrama de sequencia)..

- Concordo Totalmente
- Concordo Parcialmente
- Discordo Parcialmente
- Discordo Totalmente