



## SYSTEM-ENV: PLANEJAMENTO AUTOMATIZADO DE AVALIAÇÃO DE LINHAS DE PRODUTO DE SOFTWARE

*André Felipe<sup>1</sup>; Anderson da Silva Marcolino<sup>2</sup>; Edson A. Oliveira Junior<sup>3</sup>*

**RESUMO:** Linha de Produto de Software (LPS) é uma abordagem de desenvolvimento que utiliza práticas de reuso de software de maneira sistemática e organizada. A efetividade de uma LPS pode ser constatada por meio de avaliações frequentes com relação às suas características e produtos possíveis de serem gerados. Tais avaliações devem ser realizadas de forma sistemática e mensurável, considerando métricas e atributos de qualidade. Este artigo apresenta a construção de um módulo inserido em um ambiente automatizado de avaliação de LPSs, permitindo apoiar engenheiros de LPS na organização de dados referentes às variabilidades de uma LPS baseada em UML por meio da organização e tratamento hierárquico de tais variabilidades. O módulo é capaz de processar arquivos XMI de modelos de LPS com variabilidades, organizar tais variabilidades de forma hierárquica e permitir a simulação da geração de produtos específicos de uma LPS de forma gráfica. Outros módulos como o de coleta de dados e análise estão em construção.

**PALAVRAS-CHAVE:** Linhas de Produto de Software, Avaliação de Linhas de Produto, Ambiente *System-Env*.

**ABSTRACT:** Software Product Line (SPL) is a development approach that uses software reuse practices in a systematic and organized way. The effectiveness of a SPL can be analyzed by means of frequent evaluations with respect to their features and possible products to be generated. Such evaluations should be conducted in a systematic and measurable basis by taking into account metrics and quality attributes. This paper presents the construction of a module of an automated environment for SPL evaluations, enabling SPL engineers to organize variability data from UML-based SPL models via a hierarchical organization and handling of such variabilities. Such a module is able to process XMI files with modeled SPL variability, organize variabilities in a hierarchical basis and simulate the generation of specific SPL products in a graphical way. In addition, several modules, such as data collection and analysis, are under construction.

**KEYWORDS:** Software Product Line, Product Line Evaluation, System-Env Environment.

### 1. INTRODUÇÃO

Linha de Produto de Software (LPS) (LINDEN et al., 2007) é uma prática de desenvolvimento considerando o reuso planejado e organizado de software, podendo levar em conta componentes e arcabouços (*frameworks*) (GOMAA, 2004; OLUMOFIN, 2007).

<sup>1</sup>Graduando em Informática na Universidade Estadual de Maringá (UEM) – Bolsista PIBIC Fundação Araucária cordeiroandrefelipe@gmail.com,

<sup>2</sup> Mestrando pelo Programa de Pós-Graduação em Ciência da Computação na Universidade Estadual de Maringá (PCC-UEM) andersonmarcolino@gmail.com,

<sup>3</sup> Docente do Departamento de Informática na Universidade Estadual de Maringá (DIN-UEM). edson@din.uem.br

Empresas de destaque como Nokia, Bosch e HP (Hewlett Packard) têm alcançado resultados significativos com a adoção de LPS em suas respectivas linhas de produção, de acordo com o *Hall of Fame*<sup>1</sup> do *Software Engineering Institute* (SEI, 2010). A adoção de LPS se justifica por permitir um desenvolvimento consistente, atendendo a customização de produtos por parte dos clientes (LINDEN et al., 2007).

Na fase inicial da adoção de uma LPS, realiza-se uma avaliação a respeito do domínio abordado pela linha. A avaliação é uma atividade essencial no ciclo de vida de uma LPS, pois visa estabelecer um entendimento baseado em medidas do comportamento geral de uma LPS (OLIVEIRA JUNIOR et al., 2010). Diante disso, é interessante a construção de um ambiente automatizado para avaliação de LPSs, permitindo a adoção de métricas e atributos de qualidade, a fim de coletar análises quantitativas e qualitativas.

Este artigo apresenta o Módulo de Planejamento (MP), integrante do ambiente experimental *Systematic Evaluation Method Environment for Software Product Lines* (*SystEM-Env*). Este módulo tem por finalidade permitir ao usuário organizar os dados referentes às variabilidades de uma LPS a partir de modelos UML de forma hierárquica e amigável para que se possa planejar e conduzir avaliações de LPS. O ambiente *SystEM-Env* tem como um dos seus pilares o método *SystEM-PLA* (OLIVEIRA JUNIOR et al., 2013) para avaliação de arquitetura de LPS. Tal método leva em consideração métricas para atributos de qualidade para arquiteturas de LPS, tais como, complexidade e extensibilidade.

Este artigo está organizado da seguinte forma: na Seção 2 é apresentada uma revisão bibliográfica a respeito de LPS e avaliação de LPS; na Seção 3 é apresentado o módulo de planejamento de avaliações de LPSs; na Seção 4 são apresentados exemplos de aplicação do módulo desenvolvido considerando a LPS *Arcade Game Maker* (AGM); na Seção 5 são mostrados trabalhos relacionados ao módulo e na Seção 6 é apresentada uma conclusão acerca deste artigo e direções para trabalhos futuros.

## 2. REVISÃO BIBLIOGRÁFICA

### 2.1. LINHA DE PRODUTO DE SOFTWARE E VARIABILIDADE

Uma Linha de Produto de Software (LPS) é definida como um conjunto de sistemas de software, ou família de produtos, que compartilham características comuns e gerenciáveis, que satisfazem as necessidades de um segmento particular do mercado ou de uma missão (LINDEN et al., 2007).

As características comuns, ou os artefatos de uma LPS, são classificados em duas categorias, os artefatos comuns e os variáveis existentes na LPS (LINDEN et al., 2007). Os artefatos comuns representam o núcleo do domínio, estando presentes em todos os produtos gerados. Já os artefatos variáveis referem-se às variabilidades, sendo justamente as características que podem diferenciar um produto dos demais dentro de uma mesma linha.

Para compreender o princípio de variabilidade, torna-se necessário entender o conceito de variante. Uma variante é definida como uma solução para a resolução de uma variabilidade. Uma variabilidade pode possuir muitas variantes, onde essas variantes podem ser combinadas, podendo até serem exclusivas em alguns casos, com a seleção

---

1 [http://www.sei.cmu.edu/productlines/plp\\_hof.html](http://www.sei.cmu.edu/productlines/plp_hof.html)

de uma variante impedindo a seleção de outra. Nesse contexto, aparecem também os pontos de variação, locais onde as variabilidades se manifestam (LINDEN et al., 2007).

Um dos pontos de maior esforço no desenvolvimento e manutenção de uma LPS está no gerenciamento das variabilidades, visto que considerando um conjunto extenso de variantes, onde muitas dessas variantes são exclusivas, a combinação dessas pode tornar-se uma atividade bastante complexa.

Diante dessa possível complexidade, tornam-se necessárias práticas de avaliação sobre o domínio e as variabilidades que serão abordadas pela respectiva LPS, além de definir as atividades que serão adotadas não somente no momento de criação da linha, mas também após a criação da mesma, em um processo constante de evolução (LINDEN et al., 2007).

A abordagem *Stereotype-based Management of Variability (SMarty)* (OLIVEIRA JUNIOR et al., 2010; FIORI et al., 2012; OLIVEIRA JUNIOR et al., 2013), que constitui uma abordagem para gerenciamento de variabilidades em LPS usando a *Unified Modeling Language (UML)*, foi a considerada para a atividade de gerenciamento de variabilidades a serem avaliadas. *SMarty* adota um perfil (*SMartyProfile*), composto por um conjunto de estereótipos que estendem o metamodelo da UML para modelar variabilidades em LPSs, e ainda, um processo sistemático, que define um conjunto de diretrizes para representar, delimitar e resolver variabilidades (*SMartyProcess*).

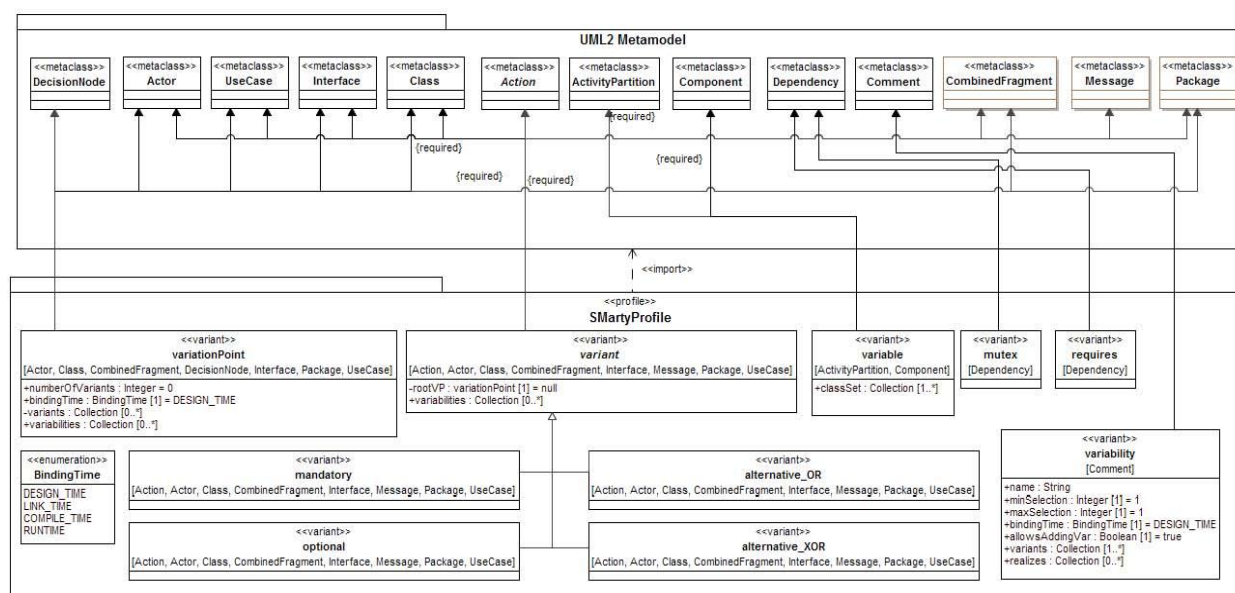


Figura 1: *SMartyProfile* e seus estereótipos e meta-atributos (OLIVEIRA JUNIOR et al., 2010).

## 2.2. O MÉTODO SYSTEM-PLA

O método *Systematic Evaluation Method for UML-based Software Product Line Architectures (SystEM-PLA)* é um método sistemático para avaliação de arquitetura de LPSs baseadas em UML (OLIVEIRA JUNIOR et al., 2013). O *SystEM-PLA* considera as variabilidades identificadas e representadas por meio da abordagem *SMarty* e conta com a possibilidade de aplicar métricas para que possam ser realizadas análises de trade-off com base nos atributos de qualidade da arquitetura de uma LPS.

O *SystEM-PLA* possui três fases distintas, Planejamento, Coleta de Dados e Análise de Dados e Documentação (Figura 2). A fase de Planejamento corresponde ao mapeamento dos possíveis produtos de uma LPS como, por exemplo, as variabilidades. A

fase de Coleta de Dados aplica métricas aos possíveis produtos gerados na fase de planejamento a fim de avaliá-los segundo critérios definidos no método, tais como extensibilidade e complexidade e análise dos atributos de qualidade. A fase de Análise dos Dados e Documentação realiza uma análise qualitativa e quantitativa dos dados coletados, tendo como resultado final, a documentação gerada sobre a avaliação da LPS.

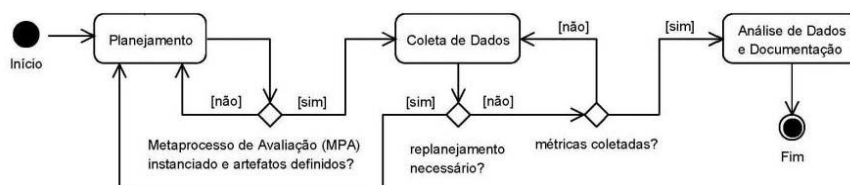


Figura 2: Fases do método SystEM-PLA (traduzido de OLIVEIRA JUNIOR et al., 2010).

### 3. MÓDULO DE PLANEJAMENTO DE AVALIAÇÃO DE LPS

O módulo desenvolvido corresponde à fase de Planejamento do *SystEM-PLA* (Figura 2), onde a avaliação de uma LPS inicia. O módulo tem por objetivo permitir que todo o ambiente de avaliação seja configurado para avaliar uma LPS e posteriormente realizar as demais fases do *SystEM-PLA*.

Para realizar a avaliação, informações a respeito das variabilidades da LPS devem ser exibidas. Assim, o módulo necessita de um *parser* que realize a leitura de arquivos XML, gerados a partir de ferramentas de modelagem UML. Os modelos de uma LPS salvos em ferramentas de modelagem precisam apresentar os estereótipos abordados pelo *SMartyProfile*. O processamento de arquivos XML com as variabilidades representadas permite que o avaliador de LPS possa visualizar graficamente as informações a respeito dos diversos modelos UML. Essas informações auxiliam na tomada de decisões no processo de avaliação.

O *SMartyParser* é a ferramenta que realiza o processamento de diagramas UML salvos em arquivos XML, considerando os estereótipos do *SMartyProfile*. Essa leitura apresenta informações quantitativas desse diagrama, tais como número de variabilidades, variantes, além de outras características (LANCELOTI et al., 2013). Por exemplo, é possível verificar com o *SMartyParser*, quantas variantes estão associadas à uma determinada variabilidade.

O módulo de planejamento utiliza o *SMartyParser* para coletar dados como variabilidades e variantes, exibindo-os em uma representação hierárquica e facilitando a análise dos possíveis produtos que podem ser gerados. O *SMartyParser* é essencial ao módulo pois a leitura do *parser* considera a UML estendida com o *SMartyProfile*, permitindo que o Módulo de Planejamento seja inserido no ambiente *SystEM-Env*, baseado no método *SystEM-PLA*.

O *SMartyParser* foi desenvolvido com base no *framework Open Core*<sup>2</sup>. O *parser* é constituído por outros dois arquivos, o *Metamodel Definition File* (MDF) e o *XMI Transformation File* (XTF) que especificam como será feito o processamento do arquivo XML contendo os diagramas UML.

A tecnologia utilizada para o desenvolvimento do módulo de planejamento é a Java SE<sup>3</sup> (*Standard Edition*), voltada principalmente para aplicações desktop e

2 <http://www.sdmetrics.com/OpenCore.html>

3 <http://www.oracle.com/technetwork/java/javase/overview/index.html>

4 [http://java.com/en/download/faq/java\\_webstart.xml](http://java.com/en/download/faq/java_webstart.xml)

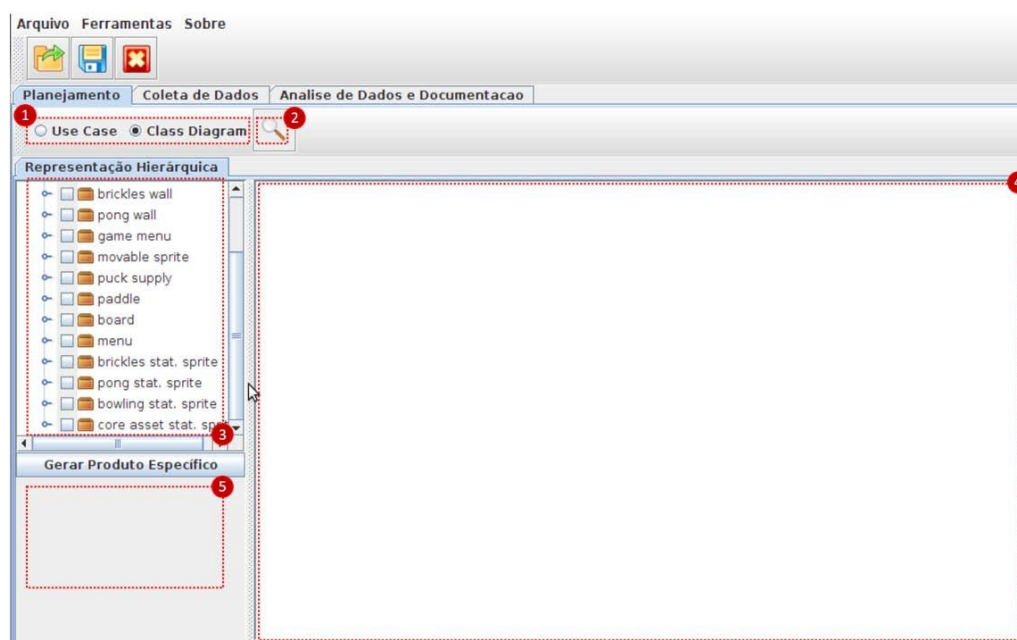
distribuídas. O módulo foi desenvolvido para ambiente desktop, pois Java possui o recurso *Java Web Start*<sup>4</sup> (JWS), que permite que aplicações desktop possam ser disponibilizadas em um container Web, como o Tomcat por exemplo, e clientes possam fazer o download de tais aplicações, executando-as em suas máquinas locais de forma transparente e automática, como se fossem aplicações locais. A manutenção de tais aplicações é facilitada, pois o JWS é responsável por verificar novas versões no container Web.

#### 4. EXEMPLO DE APLICAÇÃO DO MÓDULO DE PLANEJAMENTO À LPS AGM

Apresenta-se a seguir um exemplo de aplicação do módulo de planejamento para a LPS *Arcade Game Maker*

O módulo apresenta as seguintes funcionalidades:

- Leitura de Modelos UML salvos em XMI;
- Representação dos dados lidos do Modelo;
- Seleção de configurações de produtos;
- Suporte a abertura das imagens dos modelos; e
- Possibilidade de salvar e recuperar configurações realizadas.



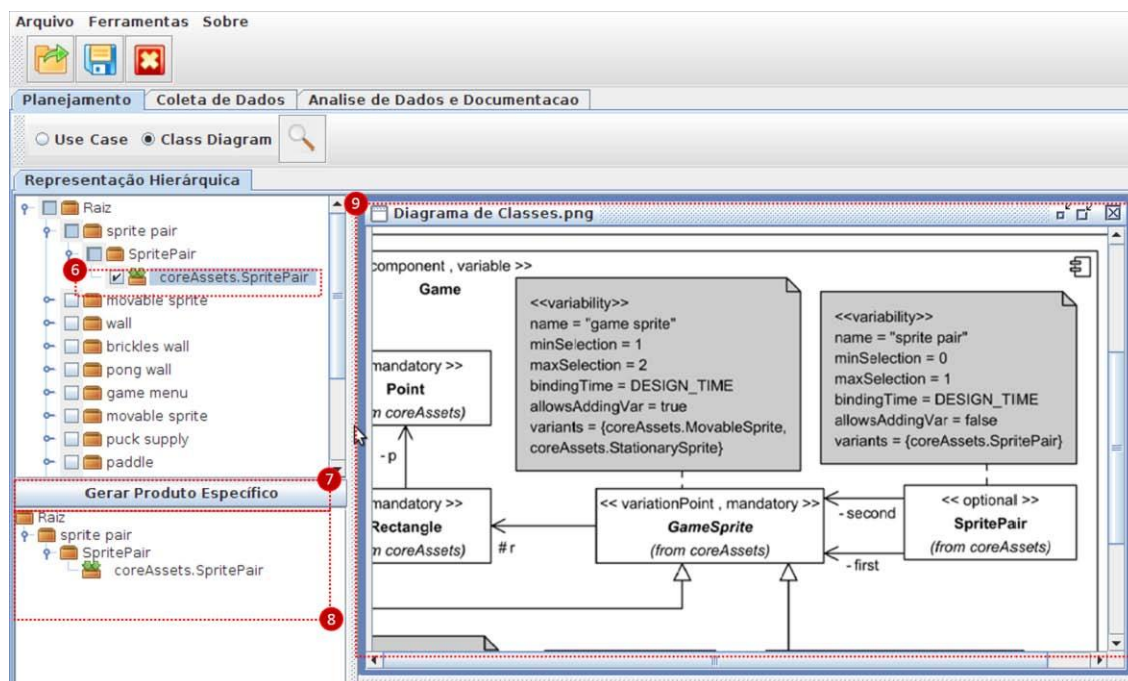
**Figura 3:** Interface após leitura inicial de um diagrama de classes UML

A Figura 3 apresenta a interface do módulo após a leitura de um modelo UML. Para realizar a leitura de um modelo, seleciona-se o modelo desejado (1) e clica no símbolo da lupa (2), na barra superior. O botão da lupa carregará informações na janela no lado esquerdo, contendo os dados lidos pelo *parser* de forma hierárquica (3). As janelas apresentadas em 4 e 5 referem-se às visualizações dos modelos salvos em imagens e representação das configurações selecionadas respectivamente. Detalhes dos passos 4 e 5 são mostrados nas figuras 4 e 5.

O arquivo XMI contendo os modelos UML é referenciado diretamente no código, pois no momento, estamos trabalhando apenas com a LP AGM. Entretanto, deseja-se estender essa funcionalidade para variados modelos UML de LPS, salvos em XMI e que

utilizem a abordagem SMarty. Isso permitiria a avaliação de LPS para qualquer modelo UML escolhido pelo usuário.

A Figura 4 apresenta o diagrama de classes utilizando a LPS AGM, após o processamento do arquivo XML correspondente. Na Figura 4, temos a variabilidade “SpritePair” Nessa situação apenas a variante “coreAssets.SpritePair” é responsável por resolver essa variabilidade (6).



**Figura 4:** Representação no Módulo de Planejamento do diagrama de classes da LP AGM considerando a variabilidade “sprite pair”

Na Figura 4, as ações 1, 2 e 3 da Figura 3 se repetem. Em seguida, ao selecionar uma ou mais variantes (6), os pontos de variação e as variabilidades associadas à variante também são selecionadas. Clicando em “Gerar Produto Específico” (7), uma nova representação hierárquica é apresentada no canto inferior esquerdo, contendo somente as configurações selecionadas (8). Para abrir um modelo UML associado a representação hierárquica, segue-se os seguintes passos: clique no menu “Ferramentas”, clique em “Abrir Diagrama” e selecione o arquivo e o diretório da imagem do modelo. Ao final, a imagem será inserida em (9).

A Figura 5 apresenta o diagrama de casos de uso da LPS AGM, juntamente com os dados lidos pelo *parser*. Considerando a variabilidade “Play Game”, são apresentadas possíveis configurações do produto (10). Selecionamos o jogo “Play Pong” como uma variante para resolver a variabilidade “play game” (11). Geramos a nova configuração clicando em “Gerar Produto Específico” (12). A imagem representando o modelo pode ser inserida em (13).

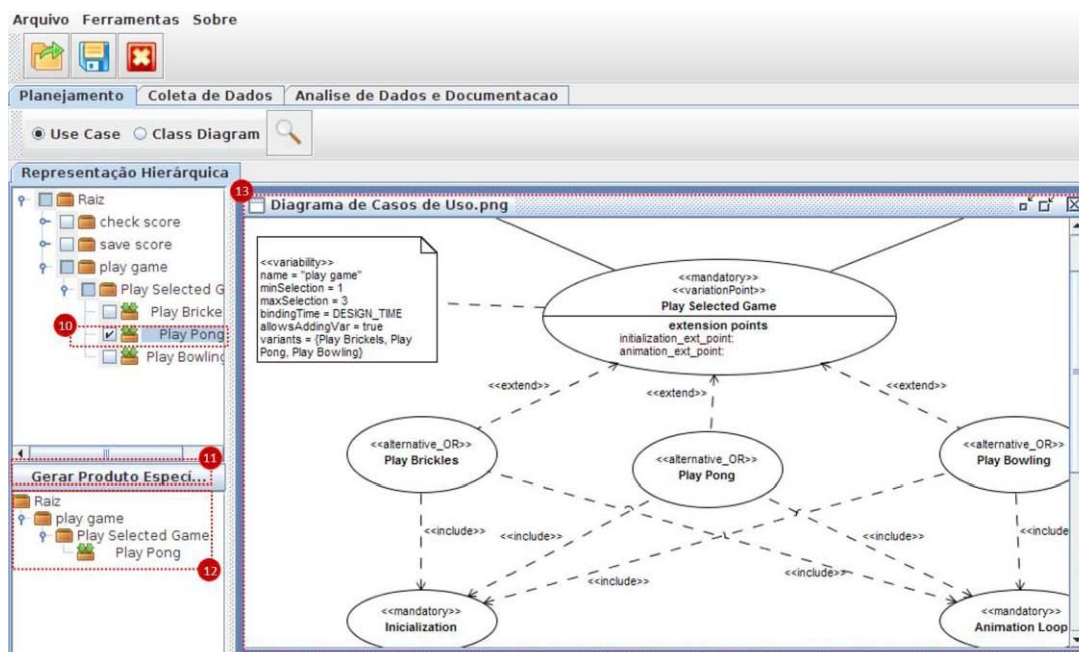


Figura 5: Representação no Módulo de Planejamento do diagrama de casos de uso.

As configurações selecionadas de um modelo UML podem ser salvas e recuperadas posteriormente. Para salvar uma configuração, aplicam-se os seguintes passos: selecione o menu “Arquivo”, clique em “Salvar configuração de produto”. Nessa etapa o usuário seleciona onde o arquivo será salvo e define um nome para o mesmo. Para abrir uma configuração de produto salva, clique novamente em “Arquivo” e clique em “Abrir configuração de produto”. Selecione a pasta onde o arquivo está salvo, selecione o arquivo desejado e clique em “Ok”. O arquivo será aberto em uma nova janela mostrando toda a configuração salva.

O módulo apresenta até esse momento, apenas funcionalidades relacionadas ao planejamento de uma LPS. O usuário pode repetir os passos mostrados na figuras 3, 4 e 5 indefinidamente, salvando as configurações quando achar necessário. Nas próximas etapas, o módulo permitirá uma coleta de dados com base em métricas estabelecidas e análise dos resultados coletados, ampliando com isso o escopo do System-Env.

## 5. TRABALHOS RELACIONADOS

As seguintes ferramentas foram estudadas e colaboraram para a construção do módulo:

**Pure::Variants**<sup>3</sup>: o Pure::Variants é um plugin suportado pelo Eclipse, criado para apoiar o desenvolvimento e implantação de LPSs, através do suporte nas atividades de análise, modelagem, implementação e implantação de uma LPS baseada em modelos de características.

**SPLIT**<sup>4</sup>: o Software Product Lines Online Tools é uma ferramenta web desenvolvida pela Universidade de Waterloo, Canadá que possibilita a modelagem de LPS com base nos modelos de características. Com o SPLIT, é possível realizar atividades como criação do modelo de características e configuração de possíveis produtos a partir de uma modelagem própria ou de modelagens existentes.

3 [http://www.pure-systems.com/pure\\_variants.49.0.html](http://www.pure-systems.com/pure_variants.49.0.html)

4 <http://www.splot-research.org>

Em comparação com as ferramentas Pure::Variants e SPLOT, o módulo de planejamento apresenta uma diferença quanto ao modelo considerado. Enquanto ambas as ferramentas consideram o modelo de características, o módulo apresentado neste artigo considera modelos UML, por meio de estereótipos estendidos. A escolha por modelos UML aconteceu devido a possibilidade de uma adoção facilitada pela indústria, dado ao amplo conhecimento da notação UML.

## 6. CONCLUSÕES E TRABALHOS FUTUROS

A abordagem de LPS utiliza o reuso de software de maneira sistemática e planejada, exigindo uma avaliação criteriosa da LPS, considerando o domínio e as variabilidades mapeadas na fase de avaliação.

O módulo apresentado neste artigo contribui para a avaliação de LPSs considerando a modelagem de variabilidades realizada em UML, exportadas em arquivos XML. O processamento de arquivos XML possibilita instanciar possíveis configurações de produtos pertencentes a essa linha. O módulo corresponde à fase de Planejamento do método *SystEM-PLA*, sendo integrante de um ambiente automatizado chamado *SystEM-Env*, ainda em desenvolvimento, que permitirá realizar todas as fases do *SystEM-PLA* em uma mesma aplicação.

Com a construção do módulo de planejamento, é possível realizar uma avaliação automatizada da arquitetura de uma LPS, nesse caso a LPS pedagógica do SEI *Arcade Game Maker* (AGM) bem como os produtos possíveis de serem gerados a partir dela. Como trabalhos futuros, será automatizado o processo de Coleta de Dados, a partir de métricas e atributos de qualidade já definidos, além de desenvolver o ambiente de Análise dos Dados e Documentação, englobando dessa forma as três fases do método *SystEM-PLA* e tornando o ambiente de avaliação *SystEM-Env* funcional e em produção.

## 7. AGRADECIMENTOS

Os autores agradecem à Fundação Araucária do Paraná e a CAPES pelo apoio financeiro.

## 8. REFERÊNCIAS

FIORI, D. R.; GIMENES, I. M. D. S.; MALDONADO, J. C. & OLIVEIRA JUNIOR, E. A. **Variability Management in Software Product Line Activity Diagrams**. In: THE INTERNATIONAL CONFERENCE ON DISTRIBUTED MULTIMEDIA SYSTEMS, Skokie, IL: Knowledge Systems Graduate School., 2012, v. 1, 89-94

GOMAA, H. **Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures**. Addison Wesley Longman Publishing Co., Inc., 2004.

LANCELOTI, L. A.; MALDONADO, J. C.; GIMENES, I. M. S.; OLIVEIRA JUNIOR, E. A.. **SMartyParser: an XMI Parser for UML-based Software Product Line Variability Models**. In: INTERNATIONAL WORKSHOP ON VARIABILITY MODELLING OF SOFTWARE-INTENSIVE SYSTEMS, 2013, Pisa. Proceedings of the 7th International Workshop on Variability Modelling of Software-intensive Systems. New York: ACM, 2013. v. 1. p. 48-52.



LINDEN, F. J. V. D.; SCHMID, K.; ROMMES, E. SOFTWARE PRODUCT LINES IN ACTION: **The Best Industrial Practice in Product Line Engineering**. Secaucus, NJ, USA: Springer-Verlag, 2007.

OLIVEIRA JUNIOR, E. A., GIMENES, I. M. S., MALDONADO, J. C. (2010). **Systematic Management of Variability in UML-based Software Product Lines**. Journal of Universal Computer Science (J.UCS), 16, 17, 2374–2393.

OLIVEIRA JUNIOR, E. A.; GIMENES, I. M. S.; MALDONADO, J. C.; MASIERO, P. C.; BARROCA, LEONOR . Systematic Evaluation of Software Product Line Architectures. **Journal of Universal Computer Science (JUCCS)**, v. 19, p. 25-52, 2013.

OLIVEIRA JUNIOR, E. A., GIMENES, I. M. S., MALDONADO, J. C. **Empirical Validation of Complexity and Extensibility Metrics for Software Product Line Architectures**. In: BRAZILIAN SYMPOSIUM ON SOFTWARE COMPONENTS, ARCHITECTURES, AND REUSE., Salvador, 2010. v. 1. p. 31-40.

OLUMOFIN, F. A **Holistic Method for Assessing Software Product Line Architectures**. Saarbrucken, Germany, Germany: VDM Verlag, 2007.