

System-Env: um Ambiente para Avaliação de Arquiteturas de Linhas de Produto de Software baseadas em UML

André F. R. Cordeiro¹, Edson A. Oliveira Junior¹

¹Departamento de Informática – Universidade Estadual de Maringá (UEM)
Av. Colombo, 5790 – 87.020-900 – Maringá – PR – Brasil

andrerc05@gmail.com, edson@din.uem.br

Abstract. *Evaluation of software product lines is an important activity in the context of product line engineering creation and evolution. Such an activity allows analyzing potential products that can be generated and possible risks related to the development of these products. This paper presents the development of SystEM-Env, an environment for UML-based software product line architecture evaluation. The environment was implemented, at first, based on the SystEM-PLA method for evaluating product line architectures. It allows variability management, generation of specific product configurations and application of metrics for performing trade-off analysis.*

Resumo. *A avaliação de linhas de produto de software é uma atividade importante no contexto de criação e evolução em engenharia de linha de produto. Tal atividade permite analisar potenciais produtos que podem ser gerados e os possíveis riscos relacionados ao desenvolvimento desses. Este artigo apresenta o desenvolvimento do SystEM-Env, um Ambiente para avaliação de arquiteturas de linhas de produto de software modeladas em UML. O ambiente foi implementado, a princípio, com base no método SystEM-PLA para avaliação de arquiteturas de linha de produto e permite o gerenciamento de variabilidades, a geração de configurações de produtos específicos e a aplicação de métricas para análises de trade-off.*

1. Introdução

Linha de Produto de Software (LPS) é uma área de pesquisa no contexto de Engenharia de Software que procura estudar e desenvolver mecanismos relacionados à customização em massa e ao desenvolvimento sistematizado de software em série [Linden *et al.* 2007]. Tais mecanismos procuram combinar diferentes características de

um domínio específico para a construção de vários produtos relacionados a este domínio. Essas características são classificadas em comuns, presentes em todos os produtos derivados da LPS e variáveis, que podem ou não estar presentes.

A criação de uma LPS vai além da combinação entre características comuns e variáveis. Questões técnicas, relacionadas à escolha da tecnologia utilizada, questões de domínio, como definição do escopo da LPS e questões organizacionais, como a avaliação do retorno de investimento (*Return On Investment - ROI*) podem ser levadas em consideração [Pohl *et al.* 2005].

Neste cenário de múltiplas questões, é necessário um método de avaliação sistemático que considere e equilibre tais questões. Com este objetivo, foi proposto o método *Systematic Evaluation Method for UML-based Software Product Line Architectures* (System-PLA) [Oliveira Junior *et al.* 2013] para avaliação de LPS, com ênfase na arquitetura. As LPSs consideradas pelo System-PLA são modeladas de acordo com a *Unified Modeling Language* (UML).

A avaliação arquitetural de LPS (ALPS) permite que questões relevantes sobre a estrutura da LPS possam ser verificadas nos estágios iniciais do processo de desenvolvimento, diminuindo os riscos e a complexidade de possíveis alterações que venham a ser realizadas.

Após a definição do System-PLA, verificou-se a necessidade do desenvolvimento de estudos para a automação deste método. Para realizar tal automatização, foi proposto o ambiente *Systematic Evaluation Method Environment for Software Product Lines* (System-Env), para avaliação arquitetural de LPS modeladas em UML, considerando o método System-PLA. Este artigo apresenta, assim, o desenvolvimento do System-Env para ALPSs.

Este artigo possui a seguinte estrutura: na Seção 2 é apresentada uma revisão da literatura com relação à LPS, além do método System-PLA; na Seção 3 é apresentado o ambiente System-Env para avaliação de ALPS; e na Seção 4 é apresentada a conclusão e direções para trabalhos futuros.

2. Revisão da Literatura

2.1 Linha de Produto de Software e Atividades Relacionadas

Linha de Produto de Software (LPS) pode ser definida como uma área de pesquisa que estuda um conjunto de sistemas de software que compartilham características comuns e variáveis, que satisfazem as necessidades de um segmento particular do mercado. [Linden *et al.* 2007; Pohl *et al.* 2005].

As atividades que envolvem a adoção e a manutenção de uma LPS podem ser divididas em dois tipos de atividades principais, Engenharia de Domínio e Engenharia

da Aplicação [Pohl *et al.* 2005]. Na Engenharia de Domínio são realizadas as atividades com o objetivo de definir o escopo da LPS e seu conjunto de artefatos. Esse conjunto de artefatos também é conhecido como núcleo de artefatos. Na Engenharia da Aplicação, são definidos os produtos da linha a partir da combinação dos artefatos presentes no núcleo de artefatos. A Figura 1 apresenta a interação entre essas atividades.



Figura 1. Atividades Essenciais presentes nas práticas de LPS. [Oliveira Junior 2010b].

A Figura 1 apresenta o conjunto das atividades essenciais envolvendo LPS. O Desenvolvimento do Núcleo de Artefatos apresentado corresponde à fase de Engenharia de Domínio. O Desenvolvimento do Produto corresponde ao conjunto das atividades da fase de Engenharia da Aplicação. Ambas as atividades estão relacionadas com a atividade de gerenciamento, realizada durante todo o ciclo de vida da LPS. É possível observar na Figura 1, a característica incremental e iterativa das atividades. Isto acontece porque os artefatos de ambas as fases são revisados constantemente, no processo de evolução e manutenção da LPS.

Empresas de grande porte como a HP (*Hewlett Packard*) reportaram resultados significativos sobre a adoção de LPS em seus processos de produção de software [SEI 2014a]. Entre os principais benefícios com a adoção de LPS, estão: redução dos custos com desenvolvimento, redução dos esforços com manutenção, melhoria nas estimativas do custo do software, menor tempo de envio do produto ao mercado [Linden *et al.* 2007, Pohl *et al.* 2005].

De forma sucinta, pode-se dizer que os produtos derivados de uma LPS são o resultado da combinação de características comuns e variáveis. As características comuns correspondem aos artefatos presentes em todos os produtos derivados da linha. As características variáveis correspondem aos artefatos que diferenciam os produtos de uma mesma LPS.

Além desta classificação quanto ao tipo de característica, outros conceitos são importantes no entendimento de LPS. Alguns destes conceitos são Variabilidades, Pontos de Variação e Variantes. As variabilidades representam as características variáveis, sendo os elementos que efetivamente diferenciam os produtos de uma LPS. O ponto de variação é o local onde ocorre uma variabilidade, a qual pode conter um ou mais pontos de variação associados. Já as variantes são as possíveis soluções que resolvem um ponto de variação e, conseqüentemente, uma variabilidade.

Atividades características de LPS como, por exemplo, o gerenciamento de variabilidades, evidenciam a necessidade de avaliação, principalmente no que se refere a produção de produtos específicos, a manutenção e a evolução da linha. Uma LPS com muitas variabilidades implica em muitos pontos de variação e muitas variantes. Gerenciar uma LPS nessas condições pode se tornar uma atividade extremamente complexa e suscetível a erros.

A avaliação de LPSs não é uma atividade trivial. Portanto, um dos principais elementos a serem avaliados em uma LPS é a sua arquitetura. A avaliação da Arquitetura de LPS (ALPS) pode ser feita de diversas formas [Oliveira Junior *et al.* 2013] como, por exemplo, realizando análises de *trade-off* baseadas em métricas [Barney *et al.* 2012].

O Método System-PLA [Oliveira Junior *et al.* 2013] realiza a avaliação de LPS no contexto da arquitetura a partir de análises de *trade-off*. Outros métodos presentes na literatura, tais como o *Architecture Tradeoff Analysis Method* (ATAM) [Kazman *et al.* 1998] e o *Holistic Product Line Architecture Assessment* (HOPLAA) [Olumofin e Mistic 2005] também realizam a avaliação de ALPSs. O ATAM considera arquiteturas de produtos únicos e o HOPLAA considera LPSs.

Neste trabalho, consideramos o método System-PLA como o método de avaliação de ALPSs base para a construção do System-Env. Mais detalhes do System-PLA e da construção do System-Env são apresentados adiante.

2.2 O Método System-PLA

O método System-PLA [Oliveira Junior *et al.* 2013] considera a avaliação da ALPSs nos estágios iniciais do processo de desenvolvimento. Os artefatos produzidos durante a engenharia de domínio de LPS são utilizados na avaliação e necessitam que os elementos de variabilidade sejam marcados com estereótipos UML. Para tanto, tais elementos são modelados de acordo com a abordagem *Stereotype-based Management of Variability* (SMarty) [Oliveira Junior *et al.* 2010a; Marcolino *et al.* 2014a; Marcolino *et al.* 2014b; Marcolino *et al.* 2013]. SMarty apresenta um conjunto de diretrizes e estereótipos. Esses estereótipos foram estendidos do metamodelo padrão da UML 2 e agrupados em um perfil UML, denominado SMartyProfile (Figura 2).

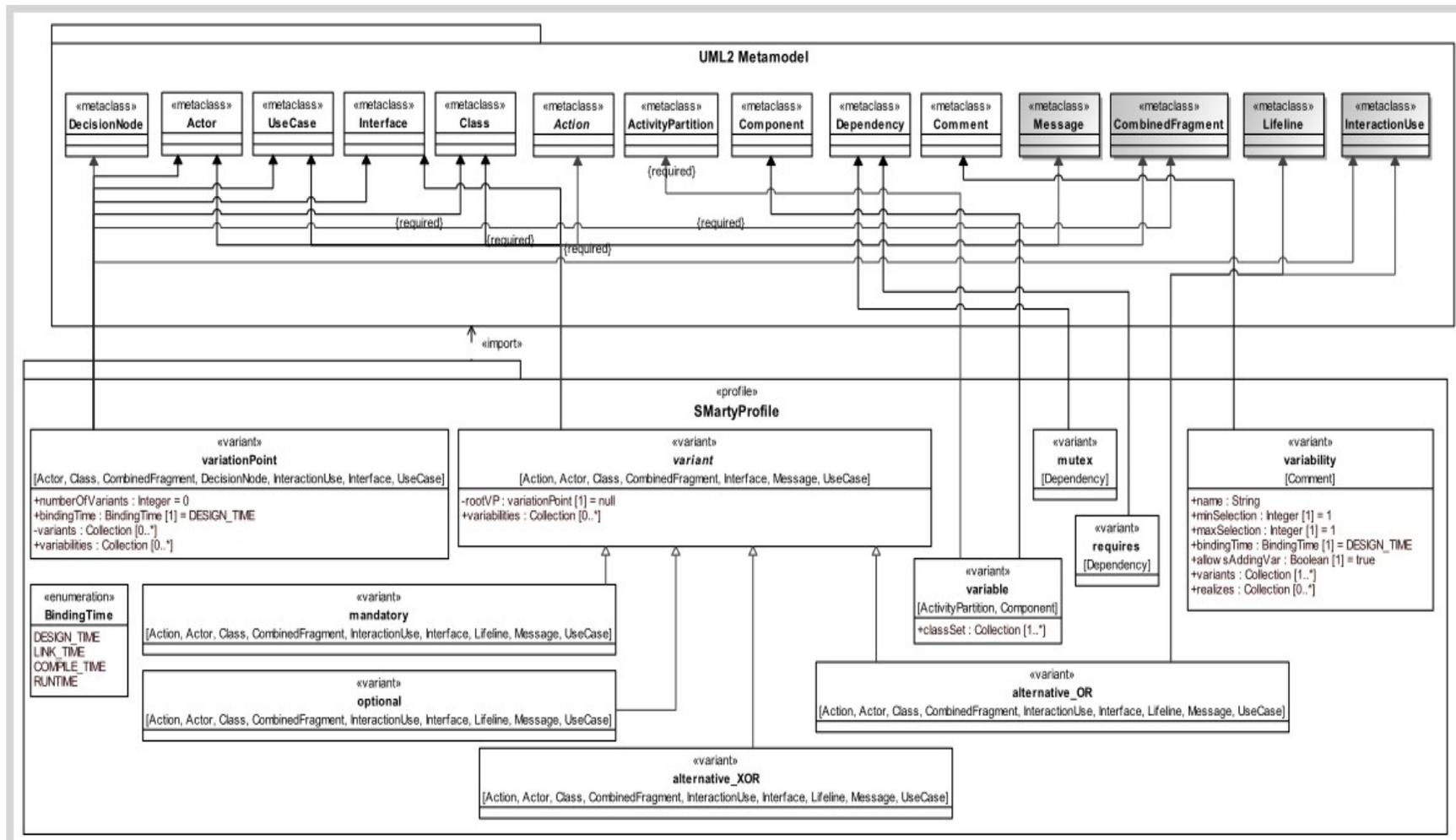


Figura 2. O Perfil SMartyProfile [Oliveira Junior *et al.* 2010a; Marcolino *et al.* 2014a; Marcolino *et al.* 2014b; Marcolino *et al.* 2013a]

A abordagem SMarty e o seu SMartyProfile são fundamentais para o SystemEM-PLA. Os estereótipos definidos no SMartyProfile refletem claramente quais elementos de LPS representam. Na Figura 2, os estereótipos *variability* (variabilidade), *variationPoint* (ponto de variação) e *variant* (variante) são exemplos de representação. Na região inferior é possível verificar o conjunto de estereótipos definidos. Esses estereótipos estão relacionados com os metamodelos da UML, que estão na região superior da Figura 2. A relação de um estereótipo com um metamodelo indica que uma instância deste metamodelo pode receber o respectivo estereótipo. Além disso, um estereótipo pode estar relacionado com mais de um metamodelo.

A avaliação de ALPSs considerando o SystemEM-PLA acontece em três fases (Figura 3): (i) Planejamento; (ii) Coleta de Dados; e (iii) Análise de Dados e Documentação. Cada fase do SystemEM-PLA considera diretrizes que devem ser seguidas para que a próxima fase seja iniciada.

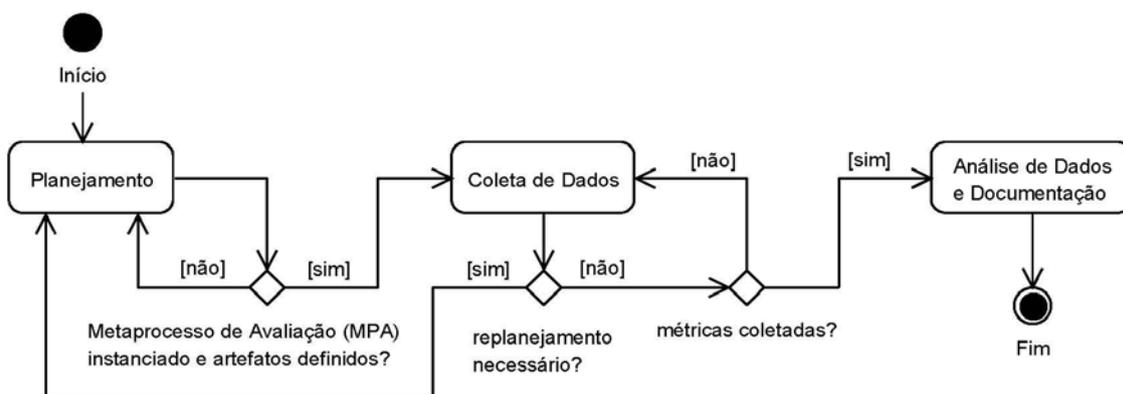


Figura 3. Fases do Método SystemEM-PLA. Traduzido de Oliveira Junior et al. (2013).

A Figura 3 apresenta as atividades realizadas ao longo das fases do SystemEM-PLA. Na fase de Planejamento os modelos UML de uma LPS são processados. Ao final desta fase, é verificado se o MetaProcesso de Avaliação (MPA), responsável por definir os artefatos que possibilitam a Avaliação da ALPS, está instanciado para os artefatos definidos. Tais artefatos incluem as metas de negócio, cenários, classificação de cenários, atributos de qualidade e métricas que avaliam se as metas de negócio estão satisfeitas a partir dos atributos de qualidade.

Caso os artefatos do MPA estejam instanciados, a fase de Coleta de Dados é iniciada. Nesta fase, métricas são aplicadas nas instâncias selecionadas para avaliar atributos de qualidade. A fase de Análise de Dados e Documentação recupera os

resultados coletados na fase anterior e procura selecionar as melhores instâncias considerando um conjunto de instâncias selecionadas a partir da análise de *trade-off*.

As atividades apresentadas na Figura 3 não determinam um prazo máximo de dias para a realização das atividades. Tal flexibilidade permite customizar o tempo de avaliação da ALPS de acordo o nível de exigência determinado pela equipe de desenvolvimento. Desde que os artefatos necessários a cada fase estejam instanciados, a sequência de fases do método pode acontecer satisfatoriamente.

Os Stakeholders principais envolvidos nas atividades de avaliação descritas na Figura 3 são o gerente, o arquiteto e o desenvolvedor da LPS. O gerente da LPS é o responsável direto por relacionar as metas de negócio da organização com a LPS, procurando assegurar que possíveis mudanças no domínio sejam refletidas diretamente na LPS. O arquiteto é responsável pela evolução/manutenção da LPS, procurando garantir que as especificações determinadas pelo gerente sejam aplicadas. O desenvolvedor é o profissional que desenvolve os artefatos da LPS, levando em consideração as instruções do arquiteto.

Apesar de apresentar os principais Stakeholders do processo de construção/evolução/manutenção da LPS, o SystEM-PLA não se limita a este número de papéis. A característica principal desejada e necessária para um possível Stakeholder é o seu entendimento avançado do domínio de atuação da futura LPS, visto a importância do entendimento do domínio para o sucesso da linha.

Como um dos trabalhos futuros de Oliveira Junior (2010b), foi proposto o desenvolvimento de um ambiente para automatizar as fases do SystEM-PLA. Este ambiente foi denominado SystEM-Env. Entre os objetivos para o SystEM-Env, estavam: (i) o gerenciamento de variabilidades; (ii) o gerenciamento de configurações de LPS; e (iii) a aplicação e coleta de métricas. Dessa forma, a próxima seção apresenta a construção do SystEM-Env.

3. O Ambiente SystEM-Env

O ambiente *Systematic Evaluation Method Environment for Software Product Lines* (SystEM-Env) tem como objetivo apoiar a avaliação de ALPSs baseadas em UML. A princípio, o ambiente foi desenvolvido com base no método SystEM-PLA. Portanto, o ambiente permite avaliações de LPSs por meio da análise de suas ALPSs e da geração de configurações de produtos específicos, além da aplicação de métricas e da realização de análises de *trade-off* multi-critérios.

O SystEM-Env foi desenvolvido utilizando a linguagem de programação Java, na plataforma *Standard Edition* (SE), com a utilização da biblioteca Swing, um *framework* para construção de interfaces gráficas em Java.

No SystemEM-Env, os resultados de uma fase são as entradas para a próxima fase, conforme a sequência de fases apresentada na Figura 3. A seguir, são apresentadas detalhadamente cada fase do método e, posteriormente, o seu desenvolvimento.

Cada fase do SystemEM-Env corresponde a um módulo. Para a avaliação dos módulos do SystemEM-Env, a LPS *Arcade Game Maker* (AGM) [SEI 2014b] foi utilizada. A AGM é uma LPS pedagógica disponibilizada para aprendizagem dos conceitos de LPS e experimentação. A avaliação dos módulos do SystemEM-Env aconteceu mediante processamento dos modelos UML da AGM e instânciação de possíveis configurações de produtos específicos. Cada módulo apresenta um conjunto de atividades apresentadas nas próximas subseções.

A Figura 4 apresenta o Modelo de Classes da AGM considerando a aplicação dos estereótipos definidos na abordagem SMarty. É possível perceber nesta figura, relações características na organização dos elementos da LPS, como a variante *Paddle* associada ao ponto de variação *MovableSprite*. A Figura 4 serve como ilustração para a apresentação dos módulos do SystemEM-Env como seguem.

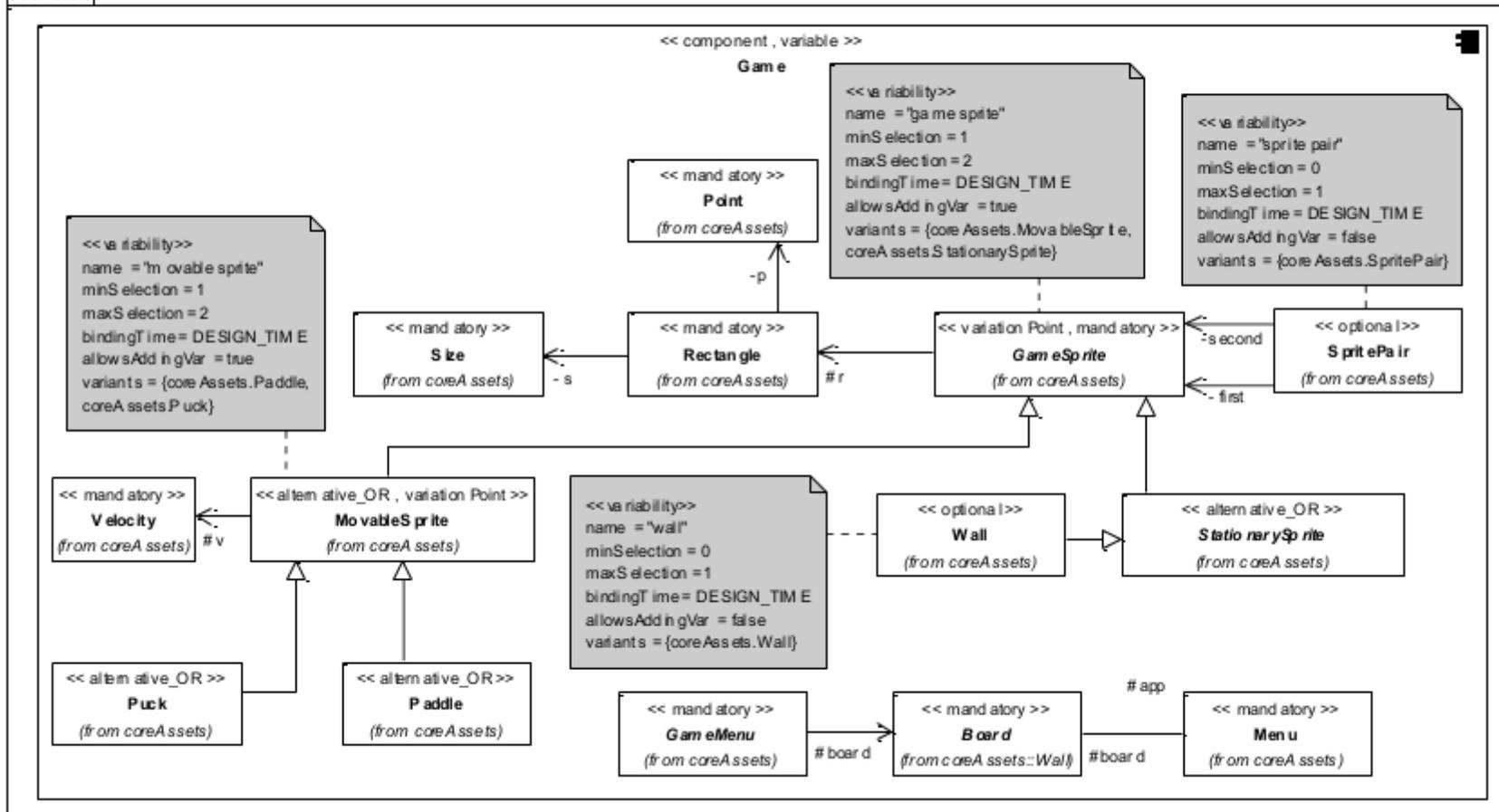


Figura 4. Modelo de Classes da AGM de acordo com a Abordagem SMarty [Oliveira Junior et al. 2010a].

3.1. Módulo de Planejamento

No módulo de Planejamento é realizado o processamento dos modelos UML de uma LPS, incluindo as suas variabilidades, pontos de variação e variantes. Para que a fase de Planejamento possa ser realizada, os modelos UML devem considerar a abordagem SMarty.

O objetivo principal deste módulo é permitir entender as variabilidades presentes nos modelos UML de forma hierárquica, separadas por tipos de modelos - casos de uso e classes atualmente – e criar configurações de produtos específicos [Cordeiro e Oliveira Junior 2013a; Cordeiro e Oliveira Junior 2013b]. A partir da hierarquia de variabilidades é possível selecioná-las e gerar tais configurações.

O módulo de Planejamento utiliza o SMartyParser [Lanceloti *et al.* 2013], um *parser* desenvolvido para a leitura dos modelos UML. Esse *parser* permite processar arquivos *XML Metadata Interchange* (XMI), provenientes de ferramentas de modelagem UML. O módulo extrai e envia as informações necessárias para o módulo de Planejamento, responsável por apresentá-las graficamente.

Quando o ambiente é executado, a tela da Figura 5 é apresentada.

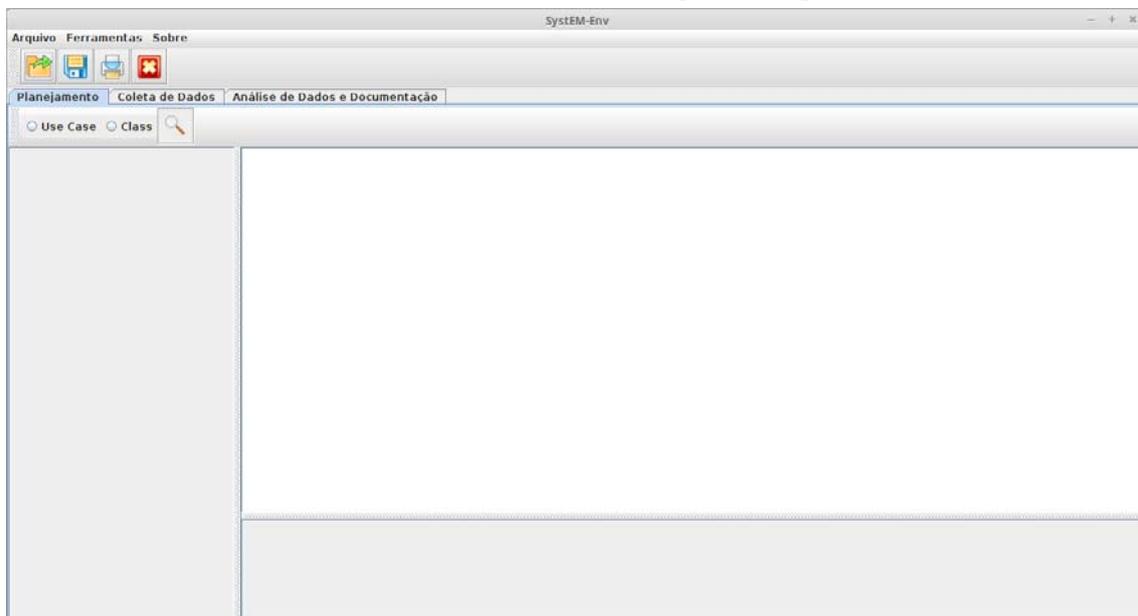


Figura 5: Tela inicial do Módulo de Planejamento System-Env.

Inicialmente, o usuário deve escolher o tipo de modelo UML que será analisado. Na região superior esquerda, são apresentados os modelos possíveis *Use Case* e *Class*. Após a escolha, a leitura do(s) modelo(s) é realizada. Tal atividade acontece mediante clique no botão com o ícone da **Lupa**. Após a leitura do modelo UML, todos os

elementos recuperados do arquivo XMI correspondente ao modelo são apresentados graficamente, em uma estrutura hierárquica na região superior esquerda. A Figura 6 apresenta o módulo de Planejamento do System-Env, correspondente a primeira fase do System-PLA.

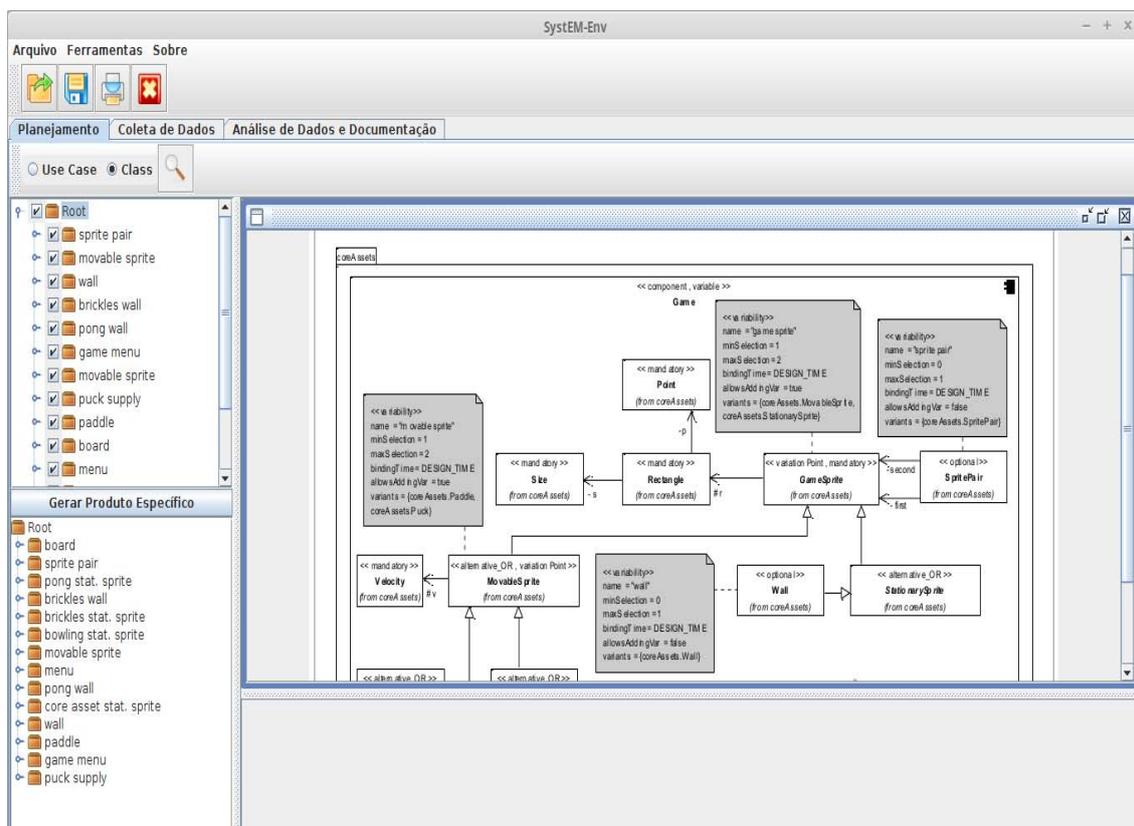


Figura 6. Módulo de Planejamento System-Env após o Processamento de Modelos UML de Classes da AGM.

Na Figura 6 é apresentado o estado da ferramenta após o processamento do arquivo XMI com a modelagem de classes da LPS. Do lado esquerdo superior estão as variabilidades em nível hierárquico, identificadas por seus respectivos nomes. Ao serem selecionadas tais variabilidades, essas passam a compor a configuração de um produto específico, na parte inferior esquerda. Ao clicar no botão “Gerar Produto Específico”, tal configuração é armazenada. No centro pode-se observar a imagem do diagrama de classes da AGM, destacado na Figura 4, que é carregado no módulo.

3.2. Módulo de Coleta de Dados

Após a criação de configurações de produtos específicos, tais configurações são enviadas ao módulo de Coleta de Dados. Neste módulo, métricas para avaliação da ALPS considerando as configurações são aplicadas e os resultados são coletados para análise.

Este módulo permite a definição e coleta de métricas para modelos UML de acordo com o *framework Open Core* SDMetrics [SDMETRICS 2014]. Para tanto, o módulo exige um arquivo XML com a descrição da métrica.

Métricas básicas [Oliveira Junior, Gimenes e Maldonado 2008] e para atributos de qualidade [Oliveira Junior e Gimenes 2014; Marcolino *et al.* 2013b] para ALPS foram definidas e estão disponíveis neste módulo. As métricas básicas avaliam características como a quantidade de variabilidades, pontos de variação e variantes existentes. As métricas para atributos de qualidade avaliam características qualitativas de uma configuração. Para tanto, este módulo dispõe de métricas de complexidade e de extensibilidade.

As métricas de complexidade avaliam uma configuração considerando os seguintes aspectos: Complexidade Ciclométrica (CC) e Métodos Ponderados por Classe (*Weighted Methods per Class – WMC*) [Marcolino *et al.* 2013b]. As métricas de extensibilidade avaliam o quanto a configuração escolhida é extensível com base em características de herança e em classes abstratas [Oliveira Junior e Gimenes 2014].

Uma observação importante com relação às métricas para atributos de qualidade está na sua dependência com relação às métricas básicas. As métricas para atributos de qualidade consideram as características retornadas pelas métricas básicas na avaliação da ALPS. As Tabelas 1 e 2 abaixo, apresentam as principais métricas básicas e para atributos de qualidade respectivamente, incorporadas pelo System-PLA.

Tabela 1. Subconjunto das Métricas Básicas System-PLA.

| Métrica | Descrição |
|---------------------|---|
| ClassVP | Classe é um ponto de variação |
| ClassAlternativeOR | Classe é uma variante inclusiva alternativa |
| ClassAlternativeXOR | Classe é uma variante exclusiva alternativa |
| ClassOptional | Classe é uma variante opcional |
| ClassMandatory | Classe é uma variante obrigatória |

| | |
|--------------------------|---|
| ClassTotalVP | Número de pontos de variação em um Diagrama de Classes |
| ClassTotalAlternativeOR | Número de variantes inclusivas alternativas em um Diagrama de Classes |
| ClassTotalAlternativeXOR | Número de variantes exclusivas alternativas em um Diagrama de Classes |
| ClassTotalOptional | Número de variantes opcionais em um Diagrama de Classes |
| ClassTotalMandatory | Número de variantes obrigatórias em um Diagrama de Classes |
| ClassTotalVariabilities | Número total de variabilidades em um Diagrama de Classes |
| PLTotalVariability | Número total de variabilidades em uma LPS |

Fonte: Oliveira Junior *et al.* 2008.

A Tabela 1 apresenta um subconjunto das métricas básicas consideradas pelo System-PLA na avaliação de ALPS. Foi apresentado um subconjunto das métricas por restrições quanto ao número de páginas, entretanto, as outras métricas suportadas pelo System-PLA podem ser encontradas em Oliveira Junior e Gimenes (2014); Marcolino *et al.* (2013) e Oliveira Junior *et al.* (2008).

As métricas apresentadas na Tabela 1 retornam principalmente características da LPS, como variabilidades, pontos de variação e variantes. A avaliação da ALPS é realizada mediante a avaliação dos produtos instanciados da LPS. Os produtos apresentam valores variados das características retornadas pelas métricas da Tabela 1.

As métricas apresentadas na Tabela 2 são dependentes das métricas básicas, para avaliar os atributos de qualidade da LPS.

Tabela 2. Subconjunto das Métricas para Atributos de Qualidade System-PLA.

| Métrica | Descrição |
|----------------------|--|
| CompClass | É o valor da métrica WMC para uma determinada Classe |
| CompVariabilityClass | É a soma da métrica CompVarPointClass de cada ponto de variação de uma determinada |

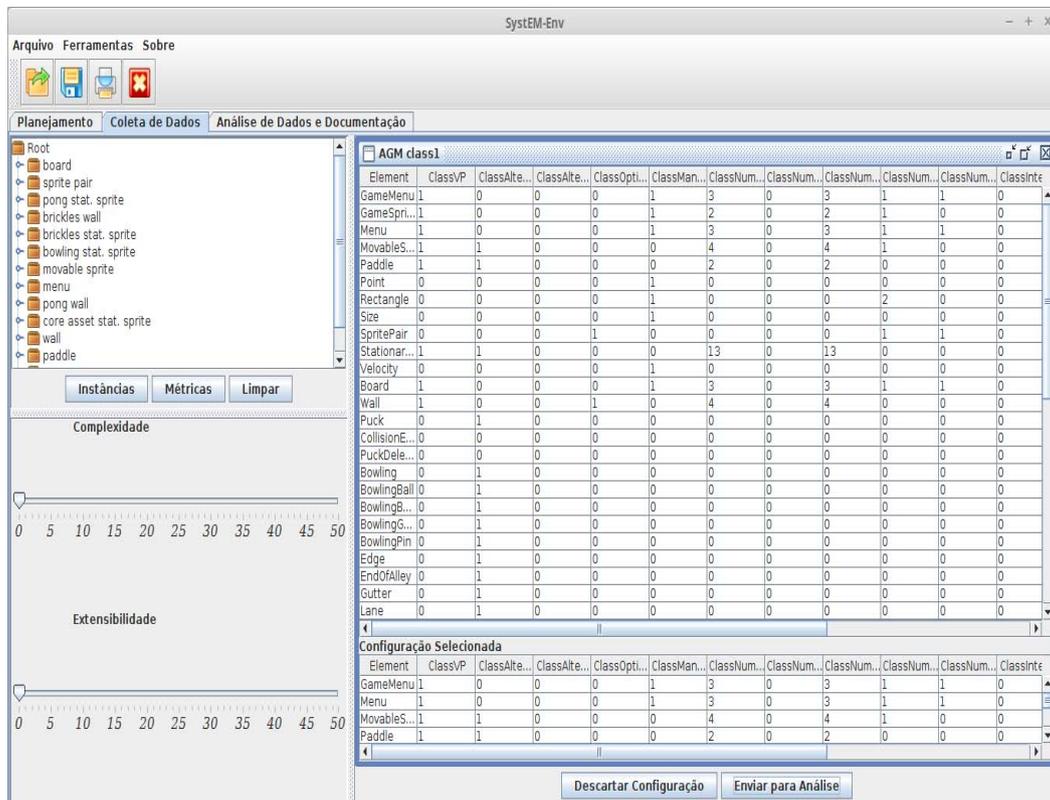
| | variabilidade |
|------------------------|---|
| CompPLA | É a soma da complexidade de cada componente de uma ALPS |
| ExtensClass | Mede o nível de extensibilidade de uma Classe |
| ExtensVariabilityClass | É a medida da métrica ExtensVarPointClass de cada ponto de variação associado a uma variabilidade |
| ExtensPLA | É a soma da extensibilidade de cada componente de uma ALPS |

Fonte: Oliveira Junior e Gimenes 2014; Marcolino *et al.* 2013; Oliveira Junior *et al.* 2008.

A Tabela 2 apresenta um subconjunto de métricas para os atributos de qualidade complexidade e extensibilidade suportados pelo System-PLA. Tais métricas permitem avaliar qualitativamente a ALPS e verificar se as metas de negócio estão sendo satisfeitas. Foi selecionado um subconjunto das métricas por limitações de páginas neste artigo, porém, as outras métricas suportadas pelo System-PLA podem ser encontradas em Oliveira Junior e Gimenes (2014); Marcolino *et al.* (2013) e Oliveira Junior *et al.* (2008).

O Módulo de Coleta de Dados aplica, então, todas as métricas definidas para o System-PLA, nas configurações de produtos específicos geradas e tabula os dados, apresentando os valores das métricas para cada configuração, além dos dados referentes ao modelo UML considerado na Avaliação. A Figura 7 apresenta o Módulo de Coleta de Dados com as métricas coletadas das configurações geradas. No centro da figura é possível perceber a presença de duas tabelas, uma tabela contendo os valores das métricas para todos os elementos do modelo UML avaliado, no caso o modelo de classes, e uma outra tabela (centro inferior) correspondente aos valores das métricas para uma determinada configuração selecionada. A apresentação das duas tabelas tem o objetivo de permitir a identificação de possíveis elementos relevantes que não foram selecionados.

Figura 7. Módulo de Coleta de Dados do SystEM-Env com Métricas Coletadas para uma



Configuração Selecionada.

O processo de aplicação e coleta das métricas considera a seguinte ordem de execução de atividades: ao clicar no botão **Instâncias**, são recuperadas as configurações geradas no módulo de Planejamento. Para que as métricas possam ser coletadas para tais configurações, clica-se no botão **Métricas**. Os resultados da aplicação e coleta são, então, apresentados nas respectivas tabelas. Ao final da aplicação das métricas e coleta de dados, é possível enviar tais configurações para o módulo de Análise de Dados e Documentação ou descartá-las. Para enviar tais configurações, clica-se no botão **Enviar para Análise** e para descartar, clica-se no botão **Descartar Configuração**.

3.3. Módulo de Análise de Dados e Documentação

Este módulo tem como objetivo analisar as configurações geradas e identificar quais as mais indicadas dentre um conjunto de configurações candidatas, de acordo com os

critérios selecionados para a avaliação da LPS. Neste trabalho, consideramos as métricas para atributos de qualidade como critérios de avaliação.

A escolha das configurações desejáveis com base nas metas de negócio de uma LPS baseia-se na aplicação da técnica *Multi-Criteria Decision Making* (MCDM) [Thurimella e Ramasway 2012]. Tal técnica considera como principais aspectos a definição de critérios base para que a avaliação aconteça e a definição de uma técnica de ranqueamento selecionada de acordo com as métricas para atributos de qualidade consideradas.

Após a definição dos critérios e seleção da técnica de ranqueamento, a avaliação é realizada e apresenta como resultado final, uma sequência decrescente das configurações selecionadas para avaliação. Neste trabalho, os critérios considerados para avaliação são as métricas para atributos de qualidade disponíveis no módulo de Coleta de Dados e a técnica para ranqueamento *Weighted Sum Model* (WSM) [Hshiong Tzeng e Jeng Huang 2011].

Neste módulo, as configurações selecionadas são analisadas com relação aos dados coletados a partir da aplicação das métricas. A análise acontece da seguinte maneira: inicialmente, são definidos os critérios para a análise dos dados. O System-Env considera, atualmente, a possibilidade de dois tipos de critérios, as métricas para atributos de qualidade e critérios quantificáveis relevantes, definidos pela equipe de avaliação. Para a definição dos critérios quantificáveis relevantes, o usuário clica no botão **+Critérios** e digita o nome do critério relevante e o seu respectivo valor. Para a escolha das métricas como critérios, o usuário seleciona a opção **Critérios** e escolhe a(s) métrica(s) que será(ão) utilizada(s).

Após a definição dos critérios, os resultados são apresentados em uma matriz conhecida como matriz de justificação [Thurimella e Ramaswamy 2012]. Nessa matriz, as configurações de produtos candidatas são os indicadores das linhas e os critérios são os indicadores das colunas. Por exemplo, um valor $A_{i,j}$ da matriz corresponde ao valor da configuração i considerando o critério j . Para a criação da matriz de justificação, seleciona-se a opção **Matriz**. A última atividade do módulo de análise corresponde à criação de um *ranking* das melhores opções considerando os critérios escolhidos para avaliação. Tal atividade acontece após a seleção da opção **Avaliação**. A Figura 8 apresenta o Módulo de Análise de Dados e Documentação.

Na Figura 8 três configurações derivadas do Modelo de Classes da Figura 4 foram avaliadas, considerando as métricas *CompClass* e *ExtensVariabilityClass*. Estes critérios avaliam a métrica WMC das classes envolvidas nas configurações de produto e o nível de extensibilidade dos elementos que resolvem as variabilidades presentes nas configurações de produto respectivamente.

SystemEM-Env

Arquivo Ferramentas Sobre

Planejamento Coleta de Dados **Análise de Dados e Documentação**

ExtensVarPointClass
 NumMethodsInterface
 CompClass
 CompVariationPointClassVariantAssociation
 ExtensVariabilityClass
 ExtensInterface
 CompVarPointClass
 CompVariationPointInter

Critérios
 + Critérios
 Matriz
 Avaliação

| Nome da Instância | Quantidade de Elementos da Instância |
|-------------------|--------------------------------------|
| Instância 0 | 19 |
| Instância 1 | 17 |
| Instância 2 | 18 |

| Instância | CompClass | ExtensVariabilityClass |
|-------------|-----------|------------------------|
| Instância 0 | 116.0 | 0.0 |
| Instância 1 | 104.0 | 0.0 |
| Instância 2 | 113.0 | 0.0 |

Adicionar Pesos

| Instância | Resultado |
|-------------|-----------|
| Instância 0 | 116.0 |
| Instância 2 | 113.0 |
| Instância 1 | 104.0 |

Figura 8. Módulo de Análise de Dados e Documentação do SystemEM-Env.

De acordo com a avaliação realizada, a Instância 0 apresentou o valor mais alto, para os critérios considerados. Na sequência, ficaram a Instância 2 e a Instância 1 respectivamente. Nesta análise específica, o critério **ExtensVariabilityClass** não influenciou na escolha da configuração, visto que apresentou o valor 0 para as três configurações.

4. Conclusão e Trabalhos Futuros

A implementação do System-Env para a avaliação de ALPSs baseadas em UML por meio da análise de produtos específicos exigiu o entendimento dos conceitos de LPS contextualizados no método System-PLA. A aprendizagem do funcionamento do SDMetrics e do *parser* também foram necessárias para que o processo de desenvolvimento dos módulos do System-Env.

A implementação do System-Env para automatizar as fases do método System-PLA apresenta evidências práticas sobre a possibilidade de automatização deste método de apoio à avaliação de LPS. Para ilustrar tais evidências a LPS AGM foi utilizada.

O fato da AGM ser uma LPS pedagógica com o objetivo de apresentar os conceitos iniciais de LPS, evidencia a necessidade de avaliação do System-Env com LPSs reais e com um maior número de características. Tais estudos permitiriam analisar efetivamente a possibilidade de incorporar a avaliação de ALPSs baseadas em UML em ambientes de desenvolvimento de software.

Como trabalhos futuros tem-se: (i) a avaliação do System-Env com LPSs reais; (ii) a total automatização do ambiente para avaliação dos possíveis produtos da LPS, com a geração automática de todas as configurações possíveis de produtos; e (iii) a avaliação automatizada de todas essas configurações. Para realizar tais trabalhos, considera-se a necessidade do planejamento e execução de estudos empíricos que possam fornecer dados que permitam avaliar a confiabilidade e capacidade de generalização dos resultados retornados pelo System-Env.

Agradecimentos

Os autores agradecem ao Programa Institucional de Bolsas de Iniciação Científica (PIBIC) do CNPq e da Fundação Araucária do Estado do Paraná pelo apoio financeiro despendido em Ago/2013-Jul/2014 e Ago/2012-Jul/2013, respectivamente.

Referências

Barney, S.; Petersen, K.; Svahnberg, M.; Aurum A.; Barney, H. Software Quality Tradeoffs: A Systematic Map. *Information and Software Technology*, 2012, Elsevier, n. 54, p. 651-662, 2012.

- Chen, L.; Babar, M. A. A Systematic Review of Evaluation of Variability Management Approaches in Software Product Lines. *Information and Software Technology*, 2011, v. 53, p. 344–362, 2011.
- Chen, L.; Babar, M. A.; Nour, A. Variability Management in Software Product Lines: A Systematic Review. In: *Proceedings of the 13th International Software Product Line Conference*, 2009, San Francisco. p. 81-90.
- Cordeiro, A. F. R.; Oliveira Junior, E. A. Um Módulo de Planejamento para um Ambiente Experimental de Avaliação de Linha de Produto de Software. In: *Anais do Congresso de Computação do Sul de Mato Grosso*, 2013, Rondonópolis, 2013a. v. 1. p. 21-32.
- Cordeiro, A. F. R.; Oliveira Junior, E. A. Módulo de Planejamento para o Ambiente Experimental SystEM-Env. In: *Anais do XXII Encontro Anual de Iniciação Científica*, 2013, Foz do Iguaçu-PR, 2013b. v. 1. p. 1-1.
- Hsiung Tzeng, G.; Jeng Huang, J. *Multiple Attribute Decision Making: Methods and Applications*. CRC Press, New York, 2011, 349p.
- Kazman, R.; Klein, M.; Barbacci, M.; Longstaff, T.; Lipson, H.; Carriere, J. The Architecture Tradeoff Analysis Method. In: *Proceedings International Conference in Engineering of Complex Computer Systems*. Monterrey CA, 1998; 68-78.
- 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: *Proceedings of the International Workshop on Variability Modelling of Software-intensive Systems*, 2013, Pisa. New York, 2013. v. 1. p. 48-52.
- Linden, F. J.; Schmid, K.; Rommes, E. *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.
- Marcolino, A. S.; Oliveira Junior, E. A.; Gimenes, I. M. S.; Barbosa, E. F. Empirically Based Evolution of a Variability Management Approach at UML Class Level. In: *Proceedings of the IEEE Annual Computer Software and Applications Conference*, 2014, Vasteras. 2014. v. 1. p. 354-363.(2014a).
- Marcolino, A. S.; Oliveira Junior, E. A.; Gimenes I. M. S. Variability Identification and Representation in Software Product Line UML Sequence Diagrams: Proposal and Empirical Study. In: *Proceedings of the Brazilian Symposium on Software Engineering*, 2014, Maceio. 2014 Brazilian Symposium on Software Engineering. v. 1. p. 141-150.(2014b).
- Marcolino, A. S.; Oliveira Junior, E. A.; Gimenes, I. M. S.; Maldonado, J. C. Towards the Effectiveness of a Variability Management Approach at Use Case Level. In: *Proceedings of the International Conference on Software Engineering and Knowledge*

- Engineering*, 2013, Boston, MA. Skokie, IL: Knowledge Systems Institute Graduate School, 2013. v. 1. p. 214-219. (2013a).
- Marcolino, A. S.; Oliveira Junior E. A.; Gimenes I. M. S.; Conte, T. U. Towards Validating Complexity-Based Metrics for Software Product Line Architectures. In: *Proceedings of the Brazilian Symposium on Software Components, Architectures and Reuse*, 2013, Brasília. 2013. v. 1. p. 69-94. (2013b).
- Oliveira Junior, E. A.; Gimenes, I. M. S. Empirical Validation of Product-line Architecture Extensibility Metrics. In: *Proceedings of the International Conference on Enterprise Information Systems*, 2014, Lisbon. v. 2. p. 111-118.
- Oliveira Junior, E. A.; Gimenes, I. M. S.; Maldonado, J. C.; Masiero, P. C.; Barroca, L. Systematic Evaluation of Software Product Line Architectures. *Journal of Universal Computer Science*, 2013, v. 19, p. 25-52, 2013.
- Oliveira Junior, E. A.; Gimenes, I. M. S.; Maldonado, J. C. Systematic Management of Variability in UML-based Software Product Lines. *Journal of Universal Computer Science*, 2010, v. 16, p. 2374-2393, 2010. (2010a)
- Oliveira Junior, E. A.. (2010). *SystEM-PLA: um Método Sistemático para Avaliação de Arquitetura de Linha de Produto de Software baseada em UML*". 2010. Tese - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2010. (2010b).
- Oliveira Junior, E. A.; Gimenes, I. M. S.; Maldonado, J. C. A Metric Suite to Support Software Product Line Architecture Evaluation. In: *Anais da Conferência Latinoamericana de Informática*, 2008, Santa Fe. 2008. p. 489-498.
- Olumofin, F.; Mistic, V. Extending the ATAM architecture evaluation to product line architectures. In: *Proceedings of the Working Conference on Software Architecture*. IEEE Computer Society Press: Los Alamitos CA, 2005; p. 45-56
- Pohl, K.; Bockle, G.; Linden, F. J. *Software Product Line Engineering: Foundations, Principles, and Techniques*. Secaucus, NJ, USA: Springer-Verlag, 2005.
- SDMETRICS. The Software Design Metrics tool for the UML. 2014. Disponível em: <http://www.sdmetrics.com/>. Acesso em 17/04/2014.
- SEI. Hall of Fame. Hewlett Packard: Owen Firmware Cooperative. Disponível em: <http://www.splc.net/fame/hp.html>. Acesso em 17/04/2014, (2014a).
- SEI. Arcade Game Maker Pedagogical Product Line. Disponível em: <http://www.sei.cmu.edu/productlines/ppl/>. Acesso em 17/04/2014, (2014b).
- Thurimella, A. K.; Ramaswamy, S. On adopting multi-criteria decision-making approaches for variability management in software product lines. In: *Proceedings International Software Product Line Conference*, v.2, New York USA, 2012; p. 32-35.