

Projeto e Implementação de um Módulo de Modelagem de *Features* para o Ambiente SMartyModeling

Henrique Fonseca Marques Luiz (PIBIC/CNPq/FA/UEM), Edson A. Oliveira Junior (Orientador), e-mail: ra105416@uem.br, edson@din.uem.br

Universidade Estadual de Maringá / Centro de Tecnologia / Maringá-PR

Ciência da Computação - Metodologia e Técnicas da Computação

Palavras-chave: linha de produto de software, *feature*, modelagem de *features*, SMarty.

Resumo:

Para se entender a importância de um Módulo de Modelagem de *Features* é importante saber em que meio ele se aplica. Uma *feature* é uma funcionalidade ou característica pertencente a um produto de uma Linha de Produto de Software. A modelagem de *features* é utilizada em um contexto de desenvolvimento de softwares em larga escala, voltados ao mesmo domínio, o que se chama de Linha de Produto de Software (LPS). O desenvolvimento de um software em uma LPS parte de um conjunto de características predefinidas, sendo essas classificadas como comuns ou variáveis. Pode-se expressar essas características por meio de um modelo de *features*, definindo como obrigatória aquelas características pertencentes a todos os produtos e como opcionais ou alternativas aquelas características particulares de cada produto instanciado. Toda essa modelagem fornece uma base para o desenvolvimento, portanto, se torna um passo fundamental para a organização das características de um produto em um contexto de LPS. A maioria das ferramentas de apoio à modelagem de *features* disponíveis no mercado não possui a funcionalidade de modelagem de *features* como suporte a LPS baseada em UML, sendo assim, esse projeto teve como proposta a implementação de um módulo complementar, de modelagem de *features*, à ferramenta SMartyModeling, voltada à modelagem de LPS baseadas em UML, com suporte às atividades de gerenciamento de variabilidades e rastreabilidade.

Introdução

Linha de Produto de Software (LPS) é uma abordagem de desenvolvimento que permite o reuso de software de forma sistemática para um domínio específico. Nos últimos anos, a abordagem de LPS vem definindo cada vez mais o Gerenciamento de Variabilidades (GV) como atividade fundamental para instanciação de produtos em larga escala (LINDEN et al.,2007).

A instanciação de um novo produto de uma LPS parte de um conjunto de características pré-definidas, podendo ser comuns ou variáveis. As características comuns estão presentes em todos os produtos derivados da LPS (similaridades) e as variáveis são responsáveis por diferenciar os produtos derivados de uma mesma LPS (variabilidades) (CAPILLA; BOSCH; KANG,2013).

A modelagem de *features* tem como objetivo organizar as características de um produto em uma LPS pertencentes a um mesmo domínio. Uma *feature* é um aspecto proeminente ou distinto que é visível pelo usuário, qualidade ou característica de um sistema ou sistema de software (KANG et al., 1990).

Na modelagem de *features*, os artefatos (*feature*) que são comuns a todos os produtos chamamos de *feature* obrigatória, já os artefatos variáveis são divididos em *feature* opcional e *feature* alternativa. *Feature* opcional é aquela que pode ou não estar presente em produto sem nenhuma relação de dependência, enquanto *features* alternativas são aquelas mutuamente exclusivas para um determinado produto. A representação dessas relações entre *features* é o grande fator variante das notações existentes (CZARNECKI et al.,2004).

SMartyModeling é um ambiente desenvolvido para a modelagem de LPS baseadas em UML, com suporte da abordagem SMarty (OLIVEIRAJR et al.,2013). A abordagem SMarty possibilita o Gerenciamento de Variabilidades em LPSs modeladas em UML, a partir de um conjunto de estereótipos e diretrizes, que aplicadas aos elementos dos modelos, permitem a representação das variabilidades.

Considerando o fato de que a modelagem de *features* é um passo indispensável para a engenharia de LPS e que SMartyModeling ainda não fornecia suporte a esse tipo de modelagem, o objetivo deste trabalho foi complementar a ferramenta, fazendo-a apoiar tal modelagem.

Materiais e métodos

Para o desenvolvimento do projeto foram feitos diversos estudos. Inicialmente foi realizada uma revisão de literatura para a familiarização com os conceitos relacionados à LPS, modelagem de *features*, gerenciamento de variabilidades e também uma revisão sobre a arquitetura de referência VMTTools-RA, tais estudos serviram principalmente para compreensão da estrutura das modelagens de *features*, e se possuem ferramentas de suporte.

Posteriormente foi estudado o ambiente (SMartyModeling) em que seria implementado o novo módulo, estudo realizado com o objetivo de definir uma maneira de representar, implementar e incluir o conceito de *feature* em SMartyModeling, tudo isso com base nas questões técnicas de desenvolvimentos da ferramenta. Para a contextualização de como é apresentada a modelagem de *features* foram analisadas ferramentas existentes que suportam tal recurso, mas que não apoiam a modelagem de LPSs baseadas em UML.

Resultados e Discussão

Na implementação do módulo foi utilizado a linguagem de programação Java, linguagem que já estava sendo utilizada no desenvolvimento do SMartyModeling, também foi utilizada uma biblioteca de diagramação chamada JGraphX, para ilustração dos diagramas, juntamente com o Swing, um framework que disponibiliza um conjunto de elementos gráficos. Diante da incorporação do módulo desenvolvido ao ambiente SMartyModeling, obteve-se um complemento de funcionalidade na ferramenta. Se tratando de uma ferramenta que tem como objetivo modelagem de LPSs, tal complemento foi de grande importância, já que, a modelagem de *features* é um passo fundamental para o contexto de LPS. Pode-se visualizar na Figura 1 um exemplo de diagrama de *features* modelado no ambiente SMartyModeling.

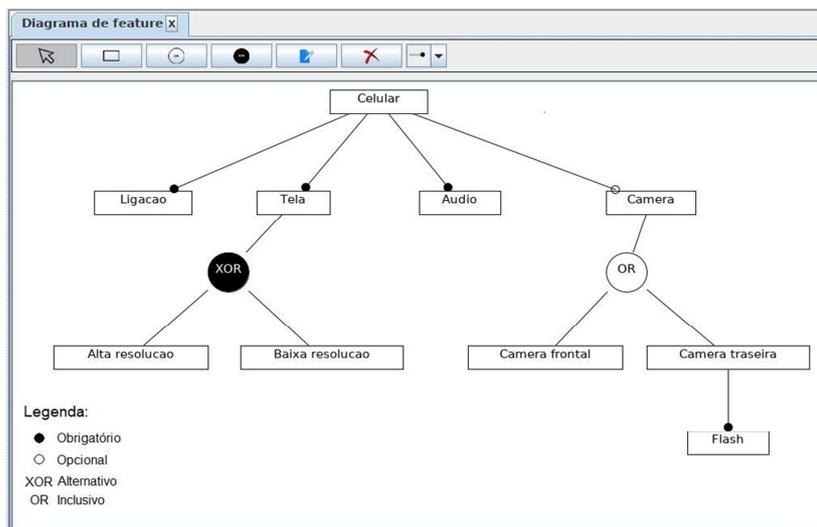


Figura 1: exemplo de modelagem

No exemplo da Figura 1, tem-se o conjunto das características do produto “Celular”. Cada característica é representada por uma *feature* que relaciona-se à outra *feature* por meio de um relacionamento. Observando os relacionamentos na Figura 1, tem-se que as *features* “Ligação”, “Tela” e “Áudio” são obrigatórias para qualquer produto derivado nesta LPS. A *feature* “Tela” possui duas *features* alternativas, “Alta resolução” e “Baixa resolução”, portanto na instanciação de um produto, somente uma delas poderá estar presente. Ainda no exemplo da Figura 1, existe um relacionamento opcional com a *feature* “Câmera”, que por sua vez tem um relacionamento inclusivo com “Câmera frontal” e “Câmera traseira”, o que significa que um Celular pode ter as duas *features* ao mesmo tempo. Caso a *feature* “Camera traseira” esteja presente, essa demanda um “Flash”.

O módulo foi avaliado de acordo com o *Technology Acceptance Model* (TAM) em relação à utilidade percebida, à facilidade de uso e à usabilidade desejável, por 13 pessoas com ampla experiência na área de LPS. Os resultados indicam que: 77% (38,5% concordam fortemente) concordam que o módulo é útil, 77% (30,8% concordam fortemente) concordam que ele é fácil de usar e 77% (30,8% concordam fortemente) entendem que o módulo possui usabilidade desejável. Várias melhorias foram identificadas e serão tratadas em projetos futuros como, por exemplo, as ações de *drag and drop*, menos caixas de diálogo e acentuação e pontuação no nome das *features*.

Conclusão

O Gerenciamento de Variabilidades está se tornando uma das principais atividades em uma abordagem de LPS, com isso, a modelagem de *features* está sendo cada vez mais utilizada. Em vista disso, a adição da funcionalidade de modelagem na ferramenta foi de extrema importância, portanto conclui-se que a complementação da ferramenta SMartyModeling foi realizada com êxito e os objetivos do projeto foram alcançados.

Agradecimentos

Os autores agradecem ao CNPq pelo apoio financeiro.

Referências

CAPILLA, R.; BOSCH, J.; KANG, K.-C. *Systems and Software Variability Management - Concepts, Tools and Experiences*. Springer, New York, NY, USA, 2013.

CZARNECKI, K. HELSEN S., EISENECKER, U. *Formalizing Cardinality-based Feature Models and their Specialization*. Universidade de Ciências Aplicadas de Kaiserslautern, Zweibrücken, Alemanha. Universidade de Waterloo, Canadá.

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. New York, Inc.

OLIVEIRAJR, E.; GIMENES, I.; MALDONADO, J.; MASIERO, P.; BARROCA, L. *Systematic Evaluation of Software Product Line Architectures*. *Journal of Universal Computer Science*, 2013, 19(1): 25-52.

KANG, K., COHEN, S., HESS, J., NOWAK, W. e PETERSON, S.. *Feature-oriented domain analysis (FODA) feasibility study*. Relatório Técnico CMU/SEI-90TR -21, Instituto de Engenharia de Software da Universidade Carnegie Mellon, de Pittsburgh, PA, 1990.