

Modelagem de Requisitos Não-Funcionais em Modelos de *Features* de Linha de Produto de Software Dinâmicas

Anderson G. Uchôa¹, Carla I. M. Bezerra^{1,2}

¹Campus Quixadá - Universidade Federal do Ceará (UFC)
Avenida José de Freitas Queiroz, 5.003 – Cedro - CEP 63.902-580 – Quixadá – CE – Brasil

²MDCC – Mestrado e Doutorado em Ciências da Computação
Universidade Federal do Ceará (UFC) Fortaleza – CE - Brasil

andersonuchoa@alu.ufc.br, carlailane@ufc.br

Abstract. *This paper proposes a representation of non-functional requirements and context requirements for the feature model of DSPLs. The representation extends the FODA notation of the feature model. The proposal will be implemented in DyMMer tool that already provides representation of features and context adaptations in a single feature model.*

Resumo. *Este trabalho propõe uma representação de requisitos não-funcionais e requisitos de contexto para o modelo de feature de LPSDs. A representação estende a notação FODA do modelo de feature. A proposta será implementada na ferramenta DyMMer que já prever a representação de features e adaptações de contexto em um modelo de feature único.*

1. Introdução

O modelo de *features* é um dos mais importante artefatos de uma Linha de Produtos de Software (LPS). Em pesquisas recentes, os modelos de *features* foram aplicados para a especificação de sistemas autônômicos e sistemas sensíveis ao contexto. Estes sistemas podem ser conceitualizados como uma Linha de Produtos de Software Dinâmicas (LPSDs) em que a ativação/desativação de funcionalidades e de configuração ocorrem em tempo de execução [Hallsteinsen et al. 2008].

O modelo de *features* é representado por diferentes níveis de abstrações: requisitos funcionais (RFs) e não-funcionais (RNFs), ambiente e restrições de contexto, componentes de tempo de execução, módulos de implementação, entre outros [Sanchez et al. 2015]. A representação de RFs no modelo de *features* tem sido extensivamente estudada. Recentemente, extensões para o modelo de *features* tem sido propostas *Feature-Softgoal Interdependency Graph* (F-SIG) [Jarzabek et al. 2006], *Extended Feature Model* (Ext-FM) [Benavides et al. 2005] e *Integrated Software Product Line Model* (ISPLM) [Siegmund et al. 2008] para a inclusão de RNFs em LPS, dentre estas identificamos *Feature Models and Quality Criteria* (FM-QC) [Sanchez et al. 2015] para LPSDs.

O modelo de *features* apresenta variações para representação de LPSs tradicionais e LPSs dinâmicas. O modelo de *features* estático, não pode modificar as características do sistemas, ou adaptar-se automaticamente, em tempo de execução, enquanto que no modelo de *features* dinâmico, as características do sistema podem ser adicionadas, removidas e modificadas em tempo de execução de maneira gerenciada [Bencomo et al. 2012].

Dessa forma, este trabalho propõe uma representação de RNFs e requisitos de contexto para o modelo de *features* de LPSDs. A representação estende a notação FODA [Kang et al. 1990] do modelo de *features*. A proposta será implementada na ferramenta DyMMer [Bezerra et al. 2016] que já prover a representação de *features* e adaptações de contexto em um modelo de *features* único. Para realizar a validação da representação proposta, será realizado uma extensão da DyMMer. Atualmente a DyMMer permite a edição dos modelos de *features* para a inserção de adaptações de contexto incluindo regras de adaptação de contexto para ativação e desativação de *features* [Bezerra et al. 2016].

2. Visão Geral da Proposta

O objetivo do presente estudo é prover uma extensão do modelo de *features* para LPSDs, baseado na representação de RNFs, restrições e adaptações de contexto. Conforme ilustrado na Figura 1, é apresentado a modelagem de *features* funcionais, *features* de qualidade, *features* abstratas, adaptações de contexto, restrições do modelo, e *binding time*. As definições para cada elemento da proposta do modelo de *features* em LPSDs, são:

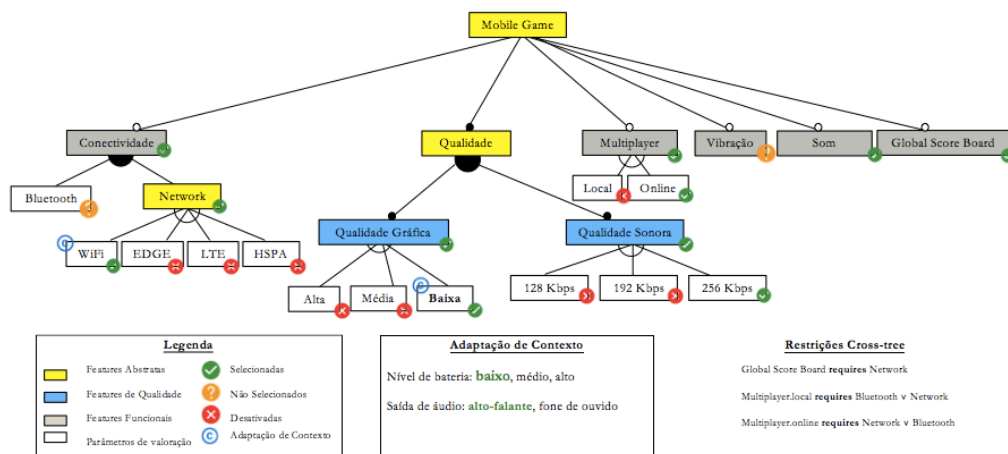


Figura 1. Modelo de *features* adaptado de [Pascual et al. 2015]

- **Features funcionais:** *features* podem ser classificadas em funcionais, qualidade e abstratas de acordo com os elementos de software que elas representam. Desse modo, *features* funcionais representam RFs da LPS podendo ser obrigatórias ou opcionais de acordo com as necessidades dos *stakeholders*. Exemplos de *features* funcionais na Figura 1 são *multiplayer*, *som*, *global score board* e *conectividade*;
- **Features abstratas:** correspondem as *features* usadas para organização do modelo de *features*, agrupamentos de *features* e variantes de parâmetros. No exemplo ilustrado na Figura 1, as *features* abstratas são representadas como *mobile game*, *qualidade* e *network*;
- **Features de qualidade:** correspondem aos RNFs ou atributos de qualidade de uma LPSD. Esses atributos de qualidade podem ser especificados em tempo de execução, assim como sofrerem adaptações de contexto. Esses RNFs podem ter conflitos em tempo de execução uma vez que um RNF pode impactar positivamente ou negativamente outro RNF, por exemplo, segurança e performance. As

features de qualidade podem ser quantitativas ou qualitativas, estão representadas como qualidade gráfica e qualidade sonora;

- **Adaptações de contexto:** são variações que ocorrem em tempo de execução, impactando a ativação e desativação de *features* de qualidade e de *features* funcionais, por exemplo, na Figura 1 temos duas adaptações de contexto: nível de bateria e saída de áudio. Quando o nível de bateria esta baixo, adaptações de contexto na qualidade gráfica e no tipo de conectividade com a internet são requeridas, da mesma forma que a *feature bluetooth* não é selecionada;
- **Parâmetros de valoração:** podem ser qualitativos ou quantitativos. Quantitativas representam parâmetros que pode ser mensurados, por exemplo, a *feature* de qualidade sonora é mensurada em níveis de Kbps, ao contrário da *feature* de qualidade gráfica, que seus parâmetros são qualitativos, ou seja, baixo, médio ou alto;
- **Restrições do modelo:** representam inclusão ou exclusão de *features* (e.g., se a *feature global score board* for selecionada *network* também é selecionada).

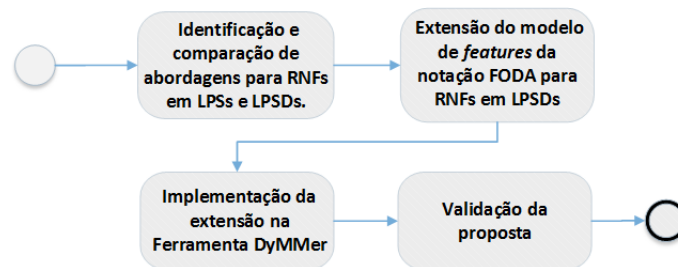


Figura 2. Passos de execução

A proposta será implementada na ferramenta DyMMer [Bezerra et al. 2016], que já prover a representação de *features* e adaptações de contexto em um modelo de *features* único. Atualmente a DyMMer permite a edição dos modelos de *features* para a inserção de adaptações de contexto, incluindo regras de adaptação de contexto para ativação e desativação de *features*. Desse modo, pretende-se adaptar a solução proposta para representação de RNFs para LPSDs na DyMMer. Para a construção dessa representação de RNFs em LPSDs serão realizados os passos de execução da Figura 2.

Tabela 1. Comparação de abordagens que representam RNFs

Crítérios	F-SIG	Ext-FM	ISPLM	FM-QC	Proposta
RNFs quantitativos e qualitativos	×	×	×	✓	✓
Restrições de contexto	×	×	✓	✓	✓
Configuração em <i>runtime</i>	×	×	Parcialmente	✓	Parcialmente
<i>Binding time</i>	×	×	×	✓	✓
Suporte a LPSDs	×	×	×	✓	✓
Suporte a ferramenta	-	✓	✓	✓	✓
Modelo de <i>features</i> único	✓	✓	×	×	✓
Avaliação estrutural do modelo de <i>features</i>	×	×	×	×	✓

A Tabela 1 apresenta o resultado da comparação entre as abordagens mapeadas que representam RNFs em LPSs e LPSDs. Os critérios estabelecidos para análise das abordagens encontram-se representadas na primeira coluna vertical contrapondo-se as abordagens avaliados, situados na primeira linha horizontal. O processo de análise foi

realizado por dois avaliadores, em que cada abordagem foi avaliada em conformidades aos critérios estabelecidos. Verificam-se deficiências entre as abordagens que não suportam LPSD na modelagem de RNFs quantitativos e qualitativos, representação de RNFs em um modelo de *features* único e avaliação estrutural do modelo de *features*. A abordagem FM-QC que apresenta suporte a LPSD, embora seja bem avaliada nos critérios estabelecidos, não realiza a avaliação estrutural do modelo de *features*, e divide o modelo de *features* em sub-módulos, aumentando a sua complexidade estrutural.

3. Conclusões e Próximos Passos

Neste artigo foi apresentada uma proposta para modelagem de RNFs em LPSDs. Através da representação proposta será possível representar RNFs qualitativos e quantitativos, restrições e adaptações de contexto, ativação e desativação de *features* em um único modelo de *features*. A solução será implementada a partir da ferramenta DyMMer. Até o momento foi realizado um mapeamento e comparação entre abordagens existentes para representação de RNFs em LPSs e LPSDs apresentado na Tabela 1. Além disso, uma visão geral da solução proposta foi estabelecida na Seção 2. Como próximo passo da pesquisa pretende-se realizar a implementação da extensão na DyMMer e a validação da mesma. Ao final, espera-se que a extensão da abordagem na DyMMer contribua para a evolução do estado da arte em LPSD, possibilitando a expansão da abordagem proposta.

Referências

- Benavides, D., Trinidad, P., and Ruiz-Cortés, A. (2005). Automated reasoning on feature models. In *Advanced Information Systems Engineering*, pages 491–503. Springer.
- Bencomo, N., Hallsteinsen, S., and Santana De Almeida, E. (2012). A view of the dynamic software product line landscape. *Computer*, 45(10):36–41.
- Bezerra, C. I., Monteiro, J. M., Andrade, R., and Rocha, L. S. (2016). Analyzing the feature models maintainability over their evolution process: An exploratory study. In *Proceedings of the Tenth International Workshop on Variability Modelling of Software-intensive Systems*, pages 17–24. ACM.
- Hallsteinsen, S., Hinchey, M., Park, S., and Schmid, K. (2008). Dynamic software product lines. *Computer*, 41(4):93–95.
- Jarzabek, S., Yang, B., and Yoeun, S. (2006). Addressing quality attributes in domain analysis for product lines. *IEEE Proceedings-Software*, 153(2):61–73.
- Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., and Peterson, A. S. (1990). Feature-oriented domain analysis (foda) feasibility study. Technical report, DTIC Document.
- Pascual, G. G., Lopez-Herrejon, R. E., Pinto, M., Fuentes, L., and Egyed, A. (2015). Applying multiobjective evolutionary algorithms to dynamic software product lines for reconfiguring mobile applications. *Journal of Systems and Software*, 103:392–411.
- Sanchez, L. E., Diaz-Pace, J. A., Zunino, A., Moisan, S., and Rigault, J.-P. (2015). An approach based on feature models and quality criteria for adapting component-based systems. *Journal of Software Engineering Research and Development*, 3(1):1–30.
- Siegmund, N., Kuhlemann, M., Rosenmüller, M., Kaestner, C., and Saake, G. (2008). Integrated product line model for semi-automated product derivation using non-functional properties. In *VaMoS*, pages 25–32.