

PROJETANDO NOVAS ARQUITETURAS DE LINHA DE PRODUTO DE SOFTWARE

Daniel Hiroshi Nouchi (PIC/ UEM), Thelma Elita Colanzi (Orientadora). E-mail: thelma.colanzi@gmail.com.br

Universidade Estadual de Maringá, CTC-DIN - Departamento de Informática,
Maringá, PR.

Ciências Exatas e da Terra / Ciência da Computação

Palavras-chave: Linha de produto de software, Projeto de arquitetura de linha de produto, Experimentos de engenharia de software.

Resumo

A reutilização de artefatos de software pela abordagem de Linha de Produto de Software (LPS) consiste no desenvolvimento de produtos de software de uma mesma família, diminuindo os custos de desenvolvimento de softwares e diminuindo o *time-to-market*. O principal artefato, do núcleo de artefatos, que irá desenvolver os produtos de uma LPS é a arquitetura da Linha de Produto de Software (PLA - *Product Line Architecture*). Com o objetivo de otimizar os projetos de PLA, foi proposta a abordagem da MOA4PLA (*Multi-Objective Approach for Product-Line Architecture Design*), que consiste na aplicação de algoritmos genéticos para otimização da modularização, da extensibilidade e da variabilidade dos projetos de PLA. Esses algoritmos são baseados na teoria de sobrevivência e adaptação do mais apto de Charles Darwin. Porém, esse tipo de otimização de projetos de PLA enfrenta o problema de escassez desses projetos para estudos experimentais. Por conta da baixa quantidade de projetos existentes serem acadêmicos e de pequeno porte, impedindo a generalização dos resultados obtidos, esse projeto visa modelar novas versões de projetos de PLA. Assim, contribuindo para aumentar o rol de projetos de PLA disponíveis para estudos experimentais, que poderão ser usados tanto no âmbito de estudos da UEM quanto por outros pesquisadores interessados no assunto.

Introdução

Uma LPS é um conjunto de sistemas que usam artefatos de software, compartilhando um conjunto de características comuns e gerenciadas, que satisfazem as necessidades de um segmento particular de mercado, e que são desenvolvidos a partir de um núcleo de artefatos comum. Essa otimização no desenvolvimento de software fez grandes empresas optarem por fazer a migração para o reuso de software de uma arquitetura de LPS em larga escala, abandonando assim o reuso de componentes individuais. Porém, apesar da arquitetura de LPS (do inglês *Product Line Architecture* - PLA) ser um dos principais artefatos de LPS para viabilizar o reuso, é difícil, para o arquiteto de software, achar uma PLA que atenda algumas propriedades arquiteturais como a reusabilidade, modularidade,

extensibilidade etc. (COLANZI, 2014). Dito isso, vê-se a necessidade de um método e de uma ferramenta que ajude o arquiteto a otimizar o projeto de uma PLA.

Por conta disso, um método que poderia ajudar nessa otimização poderia vir do campo de pesquisa conhecido como Engenharia de Software Baseada em Busca (SBSE - *Search Based Software Engineering*), pois esse método tem ajudado a resolver problemas similares na Engenharia de Software usando um algoritmo de busca, de forma eficiente. A SBSE é um campo de pesquisa que utiliza metaheurísticas para a otimização na Engenharia de Software. Para isso, a SBSE procura reformular problemas da Engenharia de Software em problemas de otimização para serem resolvidos por algoritmos de otimização, tais como os Algoritmos Genéticos (GOLDBERG, 1989), que são metaheurísticas inspiradas na teoria de Charles Darwin sobre a sobrevivência do mais apto, para maximizar ou minimizar uma função ou um grupo delas que afetam o problema da Engenharia de Software que será otimizado (COLANZI, 2014).

Nesse contexto, foi proposta a MOA4PLA (*Multi-Objective Approach for Product-Line Architecture Design*) - que de forma sistemática e automatizada visa otimizar uma PLA em termos de extensibilidade, modularização e etc. utilizando os algoritmos genéticos multiobjetivos (COLANZI, 2014). A MOA4PLA é um dos primeiros métodos a abordar essa ideia de pesquisa. A MOA4PLA possui operadores de busca (mutação e cruzamento) específicos para ajudar na otimização dos projetos de PLA em termos de modularização de características. Essa ferramenta possui um metamodelo que permite a manipulação desses projetos pelo algoritmo e um tratamento multiobjetivo para o problema de projetos de PLA. Esse tratamento é realizado utilizando um modelo de avaliação composto por funções de *fitness*, que utilizam métricas de software para medir algumas propriedades, como acoplamento, coesão, modularização e extensibilidade dos projetos de PLA. Ao final do processo, o engenheiro de software deverá escolher a melhor solução oferecida pela MOA4PLA, levando em consideração as suas necessidades, no meio das diversas possíveis soluções para o projeto.

A ferramenta OPLA-Tool (FREIRE et al., 2020) foi desenvolvida para instanciar a MOA4PLA, com sucesso no campo de SBSE, por meio de algoritmos evolutivos baseados em algoritmos genéticos. Essa ferramenta foi utilizada para estudos experimentais de alguns projetos acadêmicos de PLA de pequeno porte (COLANZI e VERGILIO, 2016, FREIRE et al. 2021).

Porém, por conta do uso de projetos acadêmicos de PLA de pequeno porte, acaba gerando resultados de menor abrangência e porte. Por isso a necessidade da utilização de mais projetos, de preferência de maior porte e reais, a fim de resolver a escassez de projetos de PLA para estudos experimentais e permitir a obtenção de resultados que permitam generalizar os achados sobre a eficácia dos métodos e técnicas da área da Engenharia de Software.

Materiais e Métodos

Neste projeto foram desenvolvidos projetos de PLA para ampliar a gama de projetos disponíveis para a realização de estudos experimentais de otimização de projeto de PLA por meio de algoritmos de busca.

Utilizando como base os projetos utilizados no trabalho de Colanzi (2014), foram modelados os projetos das LPS AGM3 e MM3, inspiradas nos projetos da AGM1 e da MM1, respectivamente. A AGM é uma LPS composta por 3 jogos, sendo eles: Brickles, Bowling and Pong. A arquitetura da MM é uma aplicação para dispositivos móveis que implementa funções para gerenciar mídias como fotos, músicas e vídeos.

A AGM3 foi inspirada na AGM1, pois ambas possuem 9 componentes, 14 interfaces, 20 classes, 4 variabilidades e 9 *features*.

A MM3 foi baseada na MM1, pois ambas possuem 11 componentes, 16 interfaces, 13 classes, 7 variabilidades.

Para realizar a implementação das PLAs foi utilizado a ferramenta CASE SMartyModeling, uma ferramenta usada para a modelagem de LPS utilizando diagramas UML.

Resultados e Discussão

As duas PLAs que foram desenhadas têm diferenças entre suas versões originais, as quais são explanadas a seguir, e as seguintes diferenças entre as AGM1 e AGM3, e as MM1 e MM3 foram anotadas.

As diferenças encontradas entre a AGM1 e AGM3 estão nos estereótipos das classes IAnimationLoopMgt, ISaveGame, IGameMgt e MovableSprites.

Nos projetos da MM1 e da MM3, a classe Album possui as mesmas funções, incluindo os métodos addAlbum(), removeAlbum(), getAlbum(), showAlbum(), hasAlbumThisMedia(), addMediaAlbum(), IsAlbumEmpty(), porém, na MM3, essas funções pertencem à interface AlbumManagement.

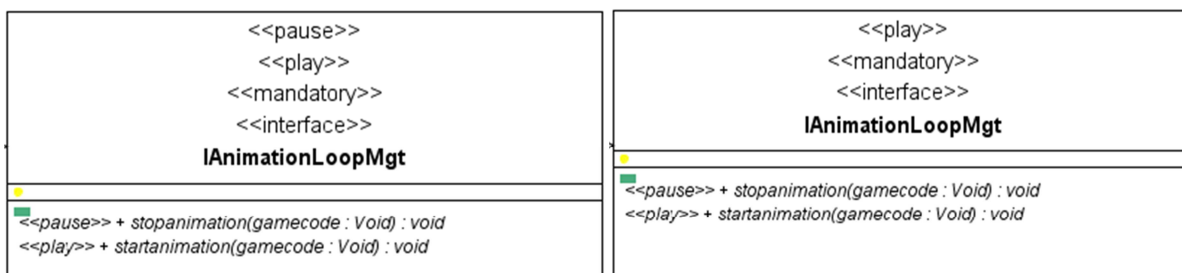


Figura 1 - Diferenças entre AGM1 (à esquerda) e AGM3 (à direita) na Classe IAnimationLoopMgt.

Os projetos desenhados estão disponíveis em:
<https://github.com/otimizes/OPLA-Tool>.

Conclusões

Podemos afirmar que o objetivo do estudo foi atingido, pois foi possível realizar o desenho de 2 novas PLAs e, assim, aumentar a gama de PLAs existentes para a realização de estudos experimentais de otimização de projeto de PLA por meio de algoritmos de busca.

Referências

Colanzi, T. E.. **Uma Abordagem de Otimização Multiobjetivo para Projeto Arquitetural de Linha de Produto de Software**. Tese de doutorado, Universidade Federal do Paraná, Curitiba, PR, 2014.

Colanzi, T. E.; Vergilio, S. R., **A feature-driven crossover operator for multi-objective and evolutionary optimization of product line architectures**. Journal of Systems and Software 121 : 126-143, 2016.

Freire, W. M. ; Massago, M. ; Zavadski, A. C. ; Amaral, A. M. M. M. ; Colanzi, T. E. . **OPLA-Tool v2.0: a Tool for Product Line Architecture Design Optimization**. In: Simpósio Brasileiro de Engenharia de Software (SBES 2020) - Trilha de Ferramentas, 2020, Natal. Simpósio Brasileiro de Engenharia de Software (SBES 2020) - Trilha de Ferramentas, 2020.

Freire, W. M. ; Tonhão, S. ; Bonetti, T. ; Shigenaga, M. ; Cadette, W. ; Felizardo, F. ; Amaral, A. ; Oliveira Jr, E. ; Colanzi, T. . **On the configuration of multi-objective evolutionary algorithms for PLA design optimization**. In: SBCARS'21: Brazilian Symposium on Software Components, Architectures, and Reuse, 2021, Joinville Brazil. 15th Brazilian Symposium on Software Components, Architectures, and Reuse. p. 11.

Goldberg, D. E.. **Genetic algorithms in search, optimization, and machine learning**. Addison-Wesley, New York, 1989. Harman, M., Mansouri, S.A., Zhang, Y.. **Search Based Software Engineering: A Comprehensive Analysis and Review of Trend Techniques and Applications**. Relatório Técnico TR-09-03, King's College London, Abril de 2009.