

Avaliação Quanti-Qualitativa de Arquiteturas de Linha de Produto de Software Otimizadas utilizando a OPLA-Tool

Luiz Fernando Okada (PIBIC/CNPq/FA/Uem), Thelma Elita Colanzi Lopes(Orientador), Aline M. M. M., Amaral (Coorientador),
e-mail: ra107247@uem.br
Universidade Estadual de Maringá / Centro de Tecnologia, PR.

Engenharia de Software

Palavras-chave: Engenharia de Software, OPLA-Tool, Avaliação qualitativa

Resumo:

A abordagem de LPS (Linha de Produto de Software) tem sido usada para promover o reuso de software e assim reduzir o tempo de desenvolvimento de um produto. Um dos principais artefatos de uma LPS é a Arquitetura de Linha de Produto (PLA - *Product Line Architecture*). O projeto de PLA é uma atividade difícil que pode ser otimizada utilizando algoritmos de busca. A abordagem MOA4PLA (*Multi-Objective Approach for Product-Line Architecture Design*) usa algoritmos de busca multiobjetivos para otimizar propriedades arquiteturais de um projeto inicial de uma PLA. Atualmente, a ferramenta OPLA-Tool utiliza essa abordagem e tem como entrada uma PLA e produz um conjunto de alternativas otimizadas como saída. Contudo as soluções podem possuir alguns problemas, tais como anomalias arquiteturais. Por esse motivo, foi criada a OPLA-Tool v2.0 que tem operadores de busca para evitar essas anomalias e corrigir outros problemas. Tendo isso em vista, é imprescindível fazer uma avaliação quanti-qualitativa das soluções geradas por esta nova versão da ferramenta. Sendo assim, neste trabalho foram feitas tais avaliações e como resultado foi verificado que: (i) a OPLA-Tool v2.0 evita o surgimento de algumas anomalias e ameniza a ocorrência de outras; e, (ii) os participantes da avaliação qualitativa percebem claramente as otimizações realizadas nas soluções geradas apesar de avaliarem as soluções de um modo diferente da avaliação realizada pela OPLA-Tool.

Introdução

O reuso de software, tem sido o foco de muitas pesquisas, pois promove o desenvolvimento de sistemas com menores custos de desenvolvimento e manutenção. A Linha de Produto de Software (LPS) está presente em muitas dessas pesquisas, visto que uma LPS é um conjunto de sistemas que compartilham características em comum e satisfazem um determinado segmento do mercado [1]. O principal artefato de uma LPS é sua arquitetura (*Product Line Architecture- PLA*), contudo obter uma PLA modular, extensível e reusável não é uma tarefa trivial, visto que o arquiteto tem que

seguir várias métricas e muitas vezes elas são conflitantes, sendo necessário que o arquiteto faça um *trade-off* entre tais métricas.

Tendo isso em vista, Colanzi [2] propôs a abordagem MOA4PLA (*Multi-Objective Approach for Product-Line Architecture Design*), a qual combina os princípios de LPS e Engenharia de Software Baseada em Busca. Essa abordagem permite um tratamento multiobjetivo para otimizar uma PLA. A ferramenta OPLA-Tool, que automatiza a MOA4PLA, utiliza como entrada uma PLA e produz como saída um conjunto de soluções que foram produzidas seguindo as métricas arquiteturais. Essas soluções são geradas usando o algoritmo de busca multiobjetivo NSGA-II (*Non-dominated Sorting Genetic Algorithm II*). Esse algoritmo é multiobjetivo, pois permite a otimização de diversas funções ao mesmo tempo. Além disso, esse algoritmo também é um algoritmo genético, ou seja, ele utiliza como operadores de busca: seleção, cruzamento e mutação.

Resultados experimentais demonstraram a efetividade da versão original da MOA4PLA e da OPLA-Tool para a otimização de projetos de PLA. No entanto, uma nova versão da ferramenta (OPLA-Tool v2.0) foi desenvolvida incluindo novos operadores de busca para melhorar a modularização de características em nível de classes, para prevenir anomalias arquiteturais e para melhorar o modo de combinação entre duas alternativas de projeto para uma mesma PLA. Sendo assim, é necessário avaliar quantitativamente e qualitativamente alternativas de PLAs geradas pela OPLA-Tool v2.0 utilizando os novos operadores de busca.

Materiais e métodos

Para a avaliação quantitativa realizada neste projeto foram executados dois experimentos: um utilizando a versão original da OPLA-Tool e outro a OPLA-Tool v2.0. Após obter as soluções de ambas versões, a avaliação consistiu em verificar quantas anomalias arquiteturais estavam presentes nas soluções não dominadas geradas nos dois experimentos. As anomalias arquiteturais consideradas nessa avaliação são: *Unused Interface*, *Unused Brick*, *Concern Overload*, *Large Class* e *Link Overload* [3]. Após a contagem das anomalias presentes nas soluções não dominadas de ambos os experimentos, foi feito um comparativo entre os resultados. Vale destacar, que ambos os experimentos utilizaram os mesmos parâmetros nas ferramentas, ou seja, a mesma taxa de mutação e cruzamento, o mesmo número de gerações, assim como o tamanho da população. As PLAs utilizadas como entrada nos experimentos foram: Arcade Game Maker (AGM), Mobile Media (MM) e Bilhetes Eletrônicos de Transporte (BET).

Na avaliação qualitativa, várias soluções produzidas pela OPLA-Tool v2.0 foram avaliadas por especialistas que responderam um questionário sobre as soluções apresentadas para eles. Após obter as respostas, foi feita a

análise delas utilizando um método chamado *coding* [4] e, por fim, foram formuladas teorias relacionadas às respostas obtidas.

Resultados e Discussão

A Tabela 1 mostra o resultado da análise quantitativa relacionando a média de anomalias arquiteturais encontradas nas soluções geradas pela versão original da OPLA-Tool e pela OPLA-Tool v2.0. Após analisar a tabela, conclui-se que a OPLA-Tool v2.0 preveniu que as anomalias *Unused Interface* e *Unused Brick* ocorressem nas soluções não dominadas produzidas, diferentemente do que ocorreu em algumas soluções geradas pela versão original da OPLA-Tool. Além disso, as anomalias *Concern Overload* e *Link Overload* continuaram ocorrendo em ambas versões da OPLA-Tool, contudo, essas anomalias, em geral, ocorrem com menos frequência nas soluções da OPLA-Tool v2.0.

Tabela 1 : Média de anomalias encontradas na avaliação quantitativa

Anomalia	OPLA-Tool original			OPLA-Tool v2.0		
	AGM	MM	BET	AGM	MM	BET
<i>Unused Interface</i>	6,31	0,2	0,1	0	0	0
<i>Unused Brick</i>	1,68	0	0,1	0	0	0
<i>Concern Overload</i>	2	1,1	5,42	2	1,83	5
<i>Large Class</i>	1,26	0,6	10	2	1,83	12
<i>Link Overload</i>	8,79	6,55	30,42	3,87	3,67	11,78

Durante a avaliação qualitativa, os comentários dos participantes evidenciam que a MOA4PLA, via OPLA-Tool v2.0, melhora os projetos de PLA fornecidos como entrada para o processo de otimização. No geral, os participantes perceberam as melhorias e ficaram satisfeitos com muitas das soluções. A partir do método *coding* formulou-se as 4 teorias sobre as soluções geradas pela OPLA-Tool v2.0, as quais são descritas abaixo.

A primeira teoria é que a OPLA-Tool v2.0 possui problemas relacionados à renderização das PLAs, visto que alguns dos participantes reclamaram da existência de pacotes com o nome estranho e da presença ou ausência de alguns relacionamentos entre classes.

A segunda teoria é que os participantes possuem viés e não avaliam as funções objetivo da mesma forma que a OPLA-Tool v2.0, por esse motivo, uma solução que possui o melhor *trade-off* entre os objetivos otimizados pode não ser escolhida como a melhor pelo avaliador, já que ele tende a

preferir as soluções mais otimizadas na propriedade arquitetural que ele considera mais importante.

A terceira teoria é que as otimizações feitas pela OPLA-Tool v2.0 são evidentes, pois os participantes conseguiram perceber as otimizações que a OPLA-Tool realizou sobre a PLA sem consultar os valores de *fitness* que elas tinham.

Por fim, a quarta teoria é que as interfaces das soluções da OPLA-Tool v2.0, em geral, são menores do que o esperado, visto que, os participantes julgaram que todas as soluções possuem interfaces pequenas.

Essas teorias mostram oportunidades de melhoria tanto da abordagem MOA4PLA como da ferramenta OPLA-Tool v2.0, gerando como resultado ideias de trabalhos futuros para o grupo de pesquisa no qual este projeto de iniciação científica está inserido.

Conclusões

Após avaliação quantitativa, conclui-se que a OPLA-Tool v2.0 de fato, evita anomalias arquiteturais citadas nesse projeto em suas soluções. Enquanto na avaliação qualitativa das soluções da OPLA-Tool v2.0, conclui-se que a ferramenta possui problemas que precisam ser resolvidos, as otimizações nas PLAs são evidentes e que a forma que a OPLA-Tool v2.0 avalia as PLAs é diferente dos participantes que avaliaram as soluções.

Agradecimentos

Agradecemos ao CNPq pelo apoio financeiro e ao Tiago T. Madrigar e João C. Neto pelo auxílio na realização do projeto.

Referências

- [1] LINDEN, F.; 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.
- [2] COLANZI, Thelma. **Uma abordagem de otimização Multiobjetivo para projeto arquitetural de linha de produto de software**. 2014. 215 Tese (Doutorado em Informática) - Universidade Federal do Paraná, UFPR, Curitiba, Paraná.
- [3] MADRIGAR, T; COLANZI, T. E; OIZUMI, W; GARCIA, A. Prevenção de Anomalias Arquiteturais na Otimização de Projeto de Linha de Produto. In: CIBSE, 23., 2020, Curitiba. **Proceedings [...]**. Assunção: Curran Associates, Inc, 2020. p.152-165.
- [4] CHARMAZ, K. **Constructing Grounded Theory A Practical Guide Through Qualitative Analysis**. Thousand Oaks, CA, USA: Sage, 2006.