



Universidade Federal de Pernambuco

Graduação em Ciência da Computação

Centro de Informática

Detectação de dependência de código entre funcionalidades através do JDT

Proposta de Trabalho de Graduação

**Aluno:** Vitor Maciel Fontes (vmf@cin.ufpe.br)

**Orientador:** Paulo Borba (phmb@cin.ufpe.br)

16 de Setembro de 2011

# 1. Introdução

*Software Product Lines* (SPL) se refere a técnicas de Engenharia de Software para a criação de produtos de software que possuem um conjunto de funcionalidades similares, e que satisfazem uma demanda de mercado [1].

Frequentemente, quando estas funcionalidades são implementadas, são utilizadas diretivas de pré-processamento para enclausurar o código relativo a cada uma delas [2]. Através destas diretivas, é permitido selecionar um conjunto de funcionalidades relativas a um determinado produto, sem precisar realizar alteração de código. [3] No entanto existe alguns problemas em relação a isto. Na maioria das vezes o código das funcionalidades possui dependência com outros trechos de código [6] e as diretivas sozinhas não oferecem nenhum suporte a detecção das dependências.

Tomando o trecho de código a seguir:

int x;

#ifdef A
m(x);
#endif

A funcionalidade “A” tem claramente uma dependência da variável “x”. Alterando a funcionalidade mandatória, podemos quebrar a funcionalidade “A”. O conceito de Virtual Separation of Concerns (VSoC) [4] propõe uma forma de diminuir estes problemas, separando o código relativo a funcionalidades diferentes, seja escondendo ou colorindo de formas diferentes. Mas somente isso não resolve o problema e foi proposto a utilização de Emergent Interfaces [5].

No entanto, estas EIs também apresentam problemas, elas não conseguem identificar vários casos de dependências. No exemplo acima, não conseguiria detectar dependência se a variável “x” acima fosse um atributo de classe. Estes casos não detectados diminuem a utilidade destas interfaces.

# 2. Objetivo

Este trabalho se propõe a detectar mais dependências de código, através da utilização de Árvores Sintáticas Abstratas (ASTs) geradas pela ferramenta JDT do Eclipse.

Posteriormente, o resultado deste trabalho será implantado na ferramenta CIDE EI [5], que faz uso de VSoC através de cores.

# 3. Estrutura do TG

1. Introdução
	1. Problema
	2. Objetivo
2. Conceitos
	1. Diretivas de pré-processamento
	2. Virtual Separation of Concern
	3. Emergent Interfaces
3. Implementação
	1. Árvores Sintáticas Abstratas
	2. JDT
	3. Visitor
4. Conclusão
	1. Principais Contribuições
	2. Trabalhos Futuros

# 4. Cronograma

|  |  |
| --- | --- |
|  | **Mês** |
| **Atividade** | *Setembro* | *Outubro* | *Novembro* | *Dezembro* |
| Estudo da Bibliografia e Estado da Arte. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementação |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Escrita do relatório |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Preparação da apresentação oral |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# Referências

 [1]  P. Clements and L. Northrop. *Software Product Lines: Prac- tices and Patterns.* Addison-Wesley, 2002.

[2]  V. Alves, P. M. Jr., L. Cole, P. Borba, and G. Ramalho. Extracting and Evolving Mobile Games Product Lines. In *Proceedings of the 9th International Software Product Line Conference (SPLC’05),* volume 3714 of *LNCS,* pages 70–81. Springer-Verlag, September 2005.

[3] K. Pohl, G. Bockle, and F. J. van der Linden. Software Product Line Engineering. Springer, 2005.

[4] Ka ̈stner, S. Apel, and M. Kuhlemann. Granularity in Soft- ware Product Lines. In *Proceedings of the 30th International Conference on Software Engineering (ICSE’08),* pages 311– 320, New York, NY, USA, 2008. ACM.

[5] M. Ribeiro, H. Pacheco, L. Teixeira, P. Borba. Emergent feature modularization. In Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications companion, SPLASH’10, pages 11–18, New York, NY, USA, 2010. ACM.

[6] M. Ribeiro, F. Queiroz, P. Borba, T. Toledo, C. Brabrand, S. Soares. On the Impact of Feature Dependencies when Maintaining Preprocessor-based Software. In 10th International Conference on Generative Programming and Component Engineering (GPCE 2011), Portland, Oregon, USA, October 2011.

Data e assinaturas

16 de Setembro de 2011

Paulo Henrique Monteiro Borba

(Orientador)

­­­­­­­­

Vitor Maciel Fontes

(Aluno)