Universidade Federal de Pernambuco

Graduação em Ciência da Computação

Centro de Informática

2011.2



Extensão de ferramenta de auxílio à manutenção de Linhas de Produto de Software

**Proposta de Trabalho de Graduação**

**Aluno:** Francisco Barreto Costa Pimentel Filho (fbcpf@cin.ufpe.br)

**Orientador:** Paulo Henrique Monteiro Borba (phbm@cin.ufpe.br)

Recife, 17 de setembro de 2011.

# Sumário

[Contexto 3](#_Toc303674274)

[Objetivos 6](#_Toc303674275)

[Cronograma 8](#_Toc303674276)

[Referências 9](#_Toc303674278)

[Assinaturas 10](#_Toc303674279)

# Contexto

Linhas de produto de software (LPS) consistem em um conjunto de sistemas que compartilham certa quantidade de features, as quais, geralmente, satisfazem as necessidades de um mercado particular [1]. LPS são comumente implementadas utilizando-se compilação condicional. Diretivas de compilação como #ifdef e #endif, contem o fragmento de código que poderá ser usado ou retirado, dependendo da feature sendo utilizada [2]. Essas diretivas de compilação podem deixar o código pouco legível e difícil de manter. Com o intuito reduzir esses efeitos pode-se utilizar técnicas de Virtual Separation of Concerns (VsoC), ou Separação Virtual de Interesses, que permite a desenvolvedores esconder linhas de código irrelevantes a tarefa atual sendo desenvolvida [2]. Dessa maneira, desenvolvedores podem focar em apenas uma feature, sem se distrair com as demais. Em contrapartida, desenvolvedores podem, inconscientemente, ao esconder uma determinada feature, introduzir erros nela, já que elas podem estar compartilhando elementos em comum, como variáveis e métodos. Logo, erros podem ser introduzidos no código da LPS, devido à falta de modularização de features, ou seja, a modificação de uma feature causa erros em outras [3].

Por exemplo, podemos imaginar uma LPS para aplicativos de mídia em dispositivos móveis. O sistema consiste, simplesmente, de um controlador para gerenciar operações com mídias (fotos e músicas) e outro módulo para a exibição dessas mídias, responsável por mostrar fotos em uma tela e tocar músicas [3]. Considerando-se esse caso, imaginemos que existem várias features para ordenar as diversas fotos presentes nesse sistema. Imaginemos agora, que em um dispositivo seja necessário implementar uma nova feature para ordenar fotos, como, por exemplo, ordenar segundo análises no histograma da imagem. Essa nova ordenação irá exigir que mudanças sejam realizadas no sistema. De acordo com o tipo do dispositivo móvel essa ordenação estará ou não presente no produto.

Manutenções nas operações de ordenação usuais podem acarretar em mudanças no comportamento da nova ordenação por histograma, considerando-se que o código dessa nova feature de ordenação esteja escondido por VsoC. Esse tipo de falha poderá ser percebida apenas no dispositivo mobile que apresenta esse tipo de feature.

Por exemplo, imagine o seguinte código que implementa uma seqüência simples para ordenação de fotos:

****

O desenvolvedor, ao usar VsoC, não irá ver o código contido entre os comentários //#if dispositovA e //#endif, podendo inserir código que venha a modificar o comportamento da feature de ordenação por histograma. Por exemplo, imagine que por uma questão de desempenho, o desenvolvedor adicione código para modificar a cor das fotos, não levando em consideração a feature de ordenação por histograma.



 As fotos retornadas, com as cores modificadas, não conseguem ser analisadas pelo método ordenacaoPorHistograma, de forma correta, retornando sempre um resultado diferente do esperado. Dessa forma o erro irá ser percebido apenas em dispositivos que implementas essa feature.

Emergent Interfaces (Interfaces Emergentes) visam reduzir esses problemas apontados, para que se possa, com maior segurança, se manter uma LPS. A abordagem consiste em realizar análises de fluxo de dados no código da LPS, para informar ao desenvolvedor em que outras features ou elementos de código uma determinada mudança irá influenciar. Isso irá possibilitar que falhas, como a apresentada, possam ser identificadas previamente [3].

# Objetivos

A ferramenta será um plug-in para o eclipse [6], devido ao grande suporte que essa plataforma oferece para a integração com a linguagem de programação Java. Além disso, ferramentas e frameworks que dão suporte ao desenvolvimento do plug-in estão disponíveis para essa linguagem de programação.

A versão do plug-in a ser desenvolvida já existe, entretanto, consiste de uma integração básica dos módulos: SOOT, CIDE e Emergent Interfaces. O SOOT é um framework que provê representações do código em Java, para que se possa efetuar uma análise de maneira simples e aplicar transformações no mesmo. Dessa forma, o SOOT é utilizado para efetuar a análise do fluxo de dados [4].O CIDE é um processador para implementar VsoC utilizando diferentes colorações no código para distinguir entre features diferentes[5]. Emergent Interfaces é usada para informar ao desenvolvedor as consequências que podem ser causadas na alteração de alguma feature.

Entretanto, o plug-in é simples e não possui uma boa integração com o desenvolvedor, pois não provê maneiras de esconder o código a partir da interfaces. Além disso, o plug-in não possui telas de personalização e configuração, para que o desenvolvedor possa adequar melhor a ferramenta ao desenvolvimento de uma LPS específica. A ferramenta também não possui diferentes visualizações para uma mesma interface (árvore, lista, tabelas), essas visualizações poderiam dar maior flexibilidade ao desenvolvedor, para que ele porra ordenar, de forma lógica, as diferentes features de uma LPS e suas dependências. O plug-in também não oferece visualizações para analisar features de uma LPS segundo complexidade e tamanho.

O plug-in existente realista análises simples, informando para o desenvolvedor de maneira não muito interativa. Melhorias nesse plug-in poderiam ser feitas com o intuito de aprimorar as análises do fluxo de dados efetuadas, além de melhorias na interação do desenvolvedor com a ferramenta, melhorando a GUI do mesmo.

Na fase inicial, uma pesquisa bibliográfica será realizada, com trabalhos anteriores publicados. Além de um estudo sobre as ferramentas utilizadas no plug-in, voltadas a melhorar a administração das features.

Posteriormente, uma extensão das funcionalidades do plug-in já existente irá ser implementada, para aprimorar o reconhecimento das dependências entre as features. Além disso, deverá ser feito um aprimoramento da interação entre o desenvolvedor e o plug-in, para que o mesmo possa, de maneira simples, manter uma LPS.

# Cronograma

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Agosto | Setembro | Outubro | Novembro |
| Estudo bibliográfico |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Estudo das Ferramentas |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Desenvolvimento de novas funcionalidades |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Teste com Usuário |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Escrita do Relatório Final |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Elaboração da Apresentação Oral |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

#

# Referências

[1] <http://www.sei.cmu.edu/productlines/> acessado em Setembro de 2011

[2] C. Kästner , S. Apel. Virtual Separation of Concerns - A Second Chance for Preprocessors. In *Journal of Object Technology.* Vol. 8, No. 6, Pa. 59-78, 2009.

[3] 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.

[4] Soot website:<http://www.sable.mcgill.ca/soot/#introduction> acessado em Setembro de 2011.

[5] Cide Website <http://wwwiti.cs.uni-magdeburg.de/iti_db/research/cide/> acessado em Setembro de 2011.

[6] Website do Eclipse <http://www.eclipse.org/> acessado em Setembro de 2011.

# Assinaturas

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Francisco Barreto Costa Pimentel Filho

**Orientando**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Paulo Henrique Monteiro Borba

**Orientador**