

Instituto de Matemática e Estatística da Universidade de São Paulo

Pointcuts abertos

Plano de estudos para a disciplina MAC5701 – Tópicos em Ciência da Computação

Cristiano Malanga Breuel

Orientador: Prof. Francisco Reverbel

São Paulo, abril de 2005.

Introdução

A programação orientada a aspectos (POA, ou AOP como é conhecida em inglês), é um paradigma de programação que permite modularizar interesses transversais de um programa ou sistema. Em outros paradigmas, como a orientação a objetos (OO), a implementação desses interesses ficaria espalhada através do código.

Um dos conceitos fundamentais para a programação orientada a aspectos são os *pointcuts*, ou especificações de conjuntos de junção. Sua finalidade é definir o conjunto de pontos do programa nos quais o código que efetuará o tratamento dos interesses transversais deve ser inserido. Todas as linguagens e *frameworks* (arcabouços) orientados a aspectos possuem uma sintaxe para definir *pointcuts*.

O objetivo do estudo cujo plano aqui se descreve é reunir e analisar os trabalhos que têm sido apresentados no sentido de permitir uma flexibilização da sintaxe de definição de especificações de conjuntos de junção. Com isso, estes trabalhos pretendem aumentar o poder expressivo da programação orientada a aspectos, além de permitir definições mais naturais e semanticamente ricas para conjuntos de junção. Damos a essas sintaxes flexíveis o nome de *pointcuts abertos*.

Justificativa

Na definição de *pointcuts*, algumas características são necessárias e/ou desejáveis:

- **Completeness:** um *pointcut* deve capturar todos os *joinpoints* aos quais o aspecto deve ser aplicado.
- **Resistência a mudanças:** mudanças no programa devem afetar o mínimo possível a definição dos *pointcuts*. Em particular, quando novos elementos são adicionados ao programa, o aspecto deve ser automaticamente aplicado onde for necessário.
- **Clareza de intenção:** a definição de um *pointcut* deve transmitir, tanto quanto possível, a intenção de quem o definiu. Ou seja, ao examinar tal definição, um programador sem conhecimento prévio deve ser capaz de entender os critérios que definem o *pointcut*.

As formas de definição de *pointcuts* disponíveis nas linguagens existentes nem sempre permitem alcançar esses objetivos. Muitas vezes é necessário recorrer a artifícios como valer-se de convenções de nomenclatura, ou listar *joinpoints* individuais na definição dos *pointcuts*.

Plano de Estudos

O objetivo do estudo proposto é reunir e analisar os trabalhos relacionados ao uso de formas abertas de especificação de *pointcuts*. O estudo incluirá a leitura dos artigos [3], [4], [7], [8] e [9]. Pretende-se também buscar exemplos em que os *pointcuts* abertos proporcionam soluções melhores no uso de programação orientada a aspectos. Em particular, tenciona-se investigar a aplicabilidade do conceito de *pointcut* aberto à área de

programação orientada a aspectos dinâmica e a arcabouços como o JBoss AOP [5] e AspectWerkz [2].

Além disso, pretende-se identificar áreas em que seja possível contribuir para o avanço da tecnologia de aspectos, no que diz respeito à definição de *pointcuts*.

Referências

- [1] Aspectj project. <http://eclipse.org/aspectj/>.
- [2] AspectWerkz. <http://aspectwerkz.codehaus.org>.
- [3] CHIBA, Shigeru; NAKAGAWA, Kiyoshi. **Josh: An Open AspectJ-like Language**. In AOSD'04, mar., 2004, Lancaster, Inglaterra.
- [4] EICHBERG, Michael; MEZINI, Mira; OSTERMAN, Klaus. **Pointcuts as functional queries**. In The Second ASIAN Symposium on Programming Languages and Systems (APLAS 2004), Springer, LNCS.
- [5] JBoss AOP. <http://www.jboss.org/products/aop>.
- [6] KICZALES, Gregor; LAMPING, John; MENDHEKAR, Anurag; MAEDA Chris; Lopes, CRISTINA V.; LOINGTIER, Jean-Marc; IRWIN, John. **Aspect-Oriented Programming**. In Proceedings of the European Conference on Object-Oriented Programming (ECOOP), 1997.
- [7] KLOSE, Karl; OSTERMANN, Klaus. **Back to the Future: Pointcuts as Predicates over Traces**. Workshop on Foundations of Aspect-Oriented Languages (FOAL'05), Chicago, USA, 2005
- [8] KOPPEN, Christian; STÖRZER, Maximilian. **PCDiff: Attacking the fragile pointcut problem**. European Interactive Workshop on Aspects in Software (EIWAS), Setembro/2004.
- [9] LIEBERHERR, Karl J.; PALM, Jeffrey; SUNDARAM, Ravi. **Expressiveness and Complexity of Crosscut Languages**. Workshop on Foundations of Aspect-Oriented Languages (FOAL'05), Chicago, USA, 2005