

Análise de Ferramentas de Mineração de Aspectos

Leandro M. Almeida, Thereza P.P. Padilha
Laboratório de Inteligência Computacional – Centro Universitário Luterano de Palmas,
Palmas TO, Brazil 50732-970, + 55 63 2232067
e-mail: [leandro, thereza]@ulbra-to.br

Abstract

Currently, there are many tools developed to incorporate aspects in legacy systems. This work presents an analysis of four aspect mining tools (FEAT, JQuery, Aspect Browser and AMT/AMTEX) using JHotDraw framework

Keywords: *aspect mining, crosscutting concerns, aspect-oriented programming.*

1. Introdução

Após o surgimento da programação orientada a aspectos (POA), cada vez mais se busca pela incorporação dessa nova tecnologia em sistemas legados. Isso se deve principalmente pelo fato da utilização do paradigma orientado a objetos possuir algumas limitações, tais como o entrelaçamento e o espalhamento de código com diferentes objetivos [6]. A presença dessas limitações acarreta em sistemas de difícil compreensão e, conseqüentemente, com um planejamento de evolução complexo e ineficaz. Para a incorporação da POA em sistemas legados, é necessário identificar no conjunto de classes do sistema, trechos de códigos que poderiam ser reformulados em aspectos [3]. Devido à complexidade de determinados sistemas e, conseqüentemente, o seu tamanho em termos de classes, métodos e linhas de código, a análise manual torna-se uma tarefa bastante árdua e tendenciosa a erros.

Para automatizar o processo de identificação de códigos candidatos a aspectos, existem diversas ferramentas de mineração de aspectos. Dentre essas ferramentas, destacam-se: *Aspect Browser* [1], *AMT/AMTEX* [4], *JQuery* [2] e *FEAT* [5]. As ferramentas *AMT/AMTEX* e *Aspect Browser* utilizam métodos léxicos que, basicamente, pesquisam a ocorrência de determinados tipos de textos ou dados. Uma limitação desse tipo de método é a necessidade de uma convenção de nomes nas classes do sistema. As ferramentas *FEAT* e *JQuery*, por sua vez, utilizam métodos exploratórios e explanatórios que provêem um conjunto de informações a respeito de detalhes de implementação que efetivamente auxiliam o usuário na identificação de aspectos. Para isso ambas as ferramentas possuem particularidades na formulação de consultas. Diante desse contexto, este trabalho apresentará uma análise de alguns experimentos realizados com as ferramentas citadas.

2. Experimentos Realizados

Foram utilizadas 195 classes pertencentes ao *framework* de criação de desenhos *JHotDraw*. Buscou-se por classes que realizam armazenamento e alteração do atributo figura, que contém informações a respeito de um desenho criado pelo usuário. Essa busca consiste em identificar os pontos no sistema que serão afetados com a inclusão de um aspecto de sincronização para o atributo figura, para assim, facilitar a implantação do aspecto.

Na ferramenta FEAT, a mineração foi bastante facilitada pela quantidade de informações a respeito dos detalhes de implementação. Essas informações são obtidas por meio de consultas intuitivas com base nos componentes de uma determinada classe ou do conjunto de classes como um todo. A quantidade de possibilidades de consultas é relativamente grande e simples, não necessitando expressar consultas em formato literal. A *JQuery* possibilita várias formas de consultas e retorna detalhes de implementação. Porém, para a realização de consultas é necessário que o usuário tenha um conhecimento da sintaxe de uma linguagem particular criada pelos seus idealizadores.

AMT/AMTEX é a ferramenta precursora na mineração de aspecto e, basicamente, fornece uma visão gráfica do espalhamento e entrelaçamento, em um nível simplificado, de código. *Aspect Browser* representa de forma visual o espalhamento e o entrelaçamento de código a partir de algumas consultas realizadas pelo usuário, mas não fornece múltiplos tipos de consultas e nem tão pouco disponibiliza uma forma clara de realização de consultas.

Na Tabela 1, são resumidas algumas características relevantes para a escolha de uma ferramenta a ser usada no processo de detecção de códigos candidatos a aspectos.

Tabela 1 – Algumas Características Analisadas nas Ferramentas de Mineração de Aspectos

Ferramenta	AMT/AMTEX	Aspect Browser	FEAT	JQuery
Característica				
Funcionamento com IDEs	Não	Não	Sim, Eclipse	Sim, Eclipse
Métodos de mineração	Léxico	Léxico	Exploratório e Explanatório	Exploratório e Explanatório
Formato das consultas, baseado em	Expressões regulares	Expressões regulares	Métodos, atributos, etc	Prolog
Complexidade de consultas	Alta	Alta	Baixa	Média
Manipulação e análise de possíveis aspectos encontrados	Não	Não	Sim	Não

Referências Bibliográficas

1. Griswold, W., Kato, Y., Yuan, J. Aspect Browser: Tool Support for Managing Dispersed Aspects. Workshop on Multi-Dimensional Separation of Concerns (ICSE), Ireland, 2000.
2. Janzen, D., Volder, K. Navigating and Querying Code Without Getting Lost. Proceedings of AOSD, ACM, UK, p.78-187, 2003.
3. Irwin, J., Kickzales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C. V., Loingtier, J. Aspect-Oriented Programming. Proceedings of ECOOP, IEEE, Finland, p.220-242, 1997.
4. Hannemann, J., Kiczales, G. Overcoming the Prevalent Decomposition in Legacy Code. Workshop on Advanced Separation of Concerns in Software Engineering at (ICSE), Toronto, 2001.
5. Robillard, M., Murphy, G. Concern graphs: Finding and describing concerns using structural program dependencies. *Proceedings of ICSE*, ACM, Florida, p. 406-416, 2002.
6. Tarr, P., Ossher, H., Harrison, W., Sutton, S. N Degrees of Separations: Multi-Dimensional Separation of Concerns. *Proceedings of ICSE*, ACM, California, p. 107-119, 1999.