

MAC5701

Tópicos em Ciência da Computação

Plano de Estudos: Reutilização de Software

Claudia de Oliveira Melo
email: claudia@ime.usp.br
Orientadora: Ana Cristina Vieira de Melo

09 de abril de 2004

1 Introdução

Este documento descreve o plano de estudos que será realizado como parte da disciplina MAC5701 - Tópicos em Ciência da Computação, na área de Engenharia de Software e Reutilização. A pesquisa será utilizada na elaboração da minha proposta de dissertação de mestrado, que abordará Classificação, Armazenamento e Seleção de Componentes Java para a construção de um ambiente de reuso efetivo.

1.1 Motivação

A produtividade de software aumentou nas últimas décadas, mas não o suficiente para acompanhar o que a demanda da indústria exige. Esse fato, comumente chamado de crise de software, continua a existir, mesmo após muitos estudos na área de Engenharia de Software e Inteligência Artificial.

A reutilização apareceu como uma das poucas abordagens reais para tratar esse problema, aumentando a produtividade e a qualidade que a indústria necessitava. Vários avanços foram conseguidos em bibliotecas, técnicas de classificação, criação e distribuição de componentes reutilizáveis, ambientes de apoio à reutilização, entre outros. Mesmo assim o grau de reutilização manteve-se muito aquém das expectativas.

Novos paradigmas e técnicas têm surgido para viabilizar o aumento do grau de reutilização, aprofundando-se em um dos quatro tipos de artefatos de software:

1. Reuso de Dados;

2. Reuso de Arquitetura;
3. Reuso de Projeto;
4. Reuso de Código.

A minha pesquisa está direcionada em paradigmas e técnicas para a Reutilização de Código.

2 Classificação, Armazenamento e Seleção de Componentes

No processo de reuso existe uma primeira fase de geração do componente, seguida da definição da interface, classificação e armazenamento do componente, para posterior busca. Cada uma destas operações necessita atenção no que diz respeito à abstração, configuração, parametrização, entre outros, de forma a obter uma maior capacidade de reutilização.

Quer os componentes tenham sido pensados para serem freqüentemente reutilizados, quer tenham sido desenhados para situações concretas, eles são passíveis de reutilização em determinado caso concreto. No processo de produção pretende-se identificar as características relevantes de cada componente para reutilização. Estas características são depois atribuídas em uma representação que permita a sua classificação, que traduz o resultado do processo de produção. Este é um dos mais complexos passos de todo o processo de reutilização. Se a classificação for incorreta, o componente nunca mais será encontrado, ou só será selecionado para situações impróprias. Por outro lado, se a classificação for muito geral, o componente será selecionado quase sempre, anulando as vantagens da classificação em si.

Além disso, as estruturas de armazenamento devem ser dimensionadas para uma biblioteca de componentes crescente e complexa. De nada adiantará algoritmos inteligentes se a informação for mal estruturada, pois os resultados das buscas serão ruins. Um bom repositório deve facilitar o trabalho dos algoritmos de busca de componentes, aumentando a performance das buscas e aumentando a satisfação do usuário.

3 Plano de Estudos

O plano de estudos consiste em pesquisar as seguintes áreas relacionadas à Classificação, Armazenamento e Seleção de Componentes Orientados a Objetos:

- *Métodos de Classificação e Seleção de Componentes OO*: estudar os métodos clássicos (ex: palavra-chave, facetas, casamento de interfaces e comportamento), descobrir novos métodos (ex: semântica) e definir quais deles serão utilizados como ponto de partida.
- *Métodos de Armazenamento de Componentes*: estudar técnicas de armazenamento de componentes para aumentar a precisão e a flexibilidade das buscas.
- *Reuso e o Ciclo de Vida Orientado a Objetos*: Os princípios de orientação a objetos e seus principais elementos como: encapsulamento, herança e polimorfismo, resultam de uma preocupação direta com a reutilização e manutenção de software. É necessário descobrir como tirar proveito desses elementos para a construção de um ambiente de reuso efetivo.
- *Métodos formais*: estudar o papel dos métodos formais na melhoria do processo de reutilização e como aplicá-los.
- *Linguagem Java*: estudar o conjunto de elementos OO que a linguagem oferece, pois a implementação que testará as novas técnicas de reutilização será escrita para reuso de código Java.

Referências

- [1] Ted J. Biggerstaff and Alan J. Perlis. *Software Reusability - Vol. I. Concepts and Models*. Addison-Wesley, 1989.
- [2] M. R. Girardi and B. Ibrahim. A similarity measure for retrieving software artifacts.
- [3] Scott Henninger. An evolutionary approach to constructing effective software reuse repositories. *ACM Transactions on Software Engineering and Methodology*, Vol. 6, No. 2, Pages 111-140, 2003.
- [4] Sametinger J. *Software Engineering with Reusable Components*. Springer, 1997.
- [5] Hafedh Mili, Fatma Mili, and Ali Mili. Reusing software: Issues and research directions. *IEEE Transactions on Software Engineering*, 21(6):528-561, 1995.
- [6] R. T. Mittermeir, H. Pozewaunig, A. Mili, and R. Mili. Uncertainty aspects in component retrieval. *Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems – IPMU98*, 1998.
- [7] Young Park. Organizing and retrieving class components based on types of reuse. *ACM Transactions on Software Engineering*, 1999.

- [8] Vijayan Sugumaran and Veda C. Storey. A semantic-based approach to component retrieval. *The DATA BASE for Advances in Information System - Summer 2003*. (Vol 34, No. 3), 2003.