UNIVERSIDADE FEDERAL FLUMINENSE
INSTITUTO DE COMPUTAÇÃO
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Luiz Laerte Nunes da Silva Junior
Thiago Nazareth de Oliveira

Vertical Code Completion

Niterói
2010

Luiz Laerte Nunes da Silva Junior
Thiago Nazareth de Oliveira

Vertical Code Completion

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal Fluminense como parte dos requisitos para obtenção do Grau de Bacharel em Ciência da Computação.

Orientadores: Leonardo Murta
Co-orientador: Alexandre Plastino

Niterói
2010

Luiz Laerte Nunes da Silva Junior
Thiago Nazareth de Oliveira

Vertical Code Completion

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal Fluminense como parte dos requisitos para obtenção do Grau de Bacharel em Ciência da Computação.

Aprovado em Junho de 2010

BANCA EXAMINADORA

Prof. Leonardo Murta, D.Sc.
Orientador
UFF

Prof. Alexandre Plastino, D.Sc.
Co-Orientador
UFF

Prof. - , M.Sc.
UFF

Prof. - , D.Sc.
UFF

Niterói
2010

RESUMO

VCC VERTICAL CODE COMPLETION AQUI ENTRA O RESUMO

Palavras Chave:
Engenharia de Software, Code Completion, Mineração de Dados, Mineração de Padrões Sequenciais.

ABSTRACT

VCC VERTICAL CODE COMPLETION AQUI ENTRA O ABSTRACT

Keywords:
Software Engineering, Code Completion, Data Mining, Sequence Mining.

LISTA DE ACRÔNIMOS

VCC: *Vertical Code Completion*

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

CAPÍTULO 1 - INTRODUÇÃO

Capítulo 1

CAPÍTULO 2 - REVISÃO DA LITERATURA

2.1 INTRODUÇÃO

Neste capítulo são abordados alguns conceitos e técnicas de Mineração de Dados, além da aplicação dessas técnicas em problemas de Engenharia de Software, citando trabalhos desenvolvidos nessa área.

2.2 MINERAÇÃO DE DADOS

Desde 1960, bases de dados evoluíram de simples processamento de arquivos a sofisticados sistemas de banco de dados. Isso permitiu que empresas e centros de pesquisas acumulassem grandes quantidades de dados históricos nos anos 70 e 80 [3].

Porém, essa enorme quantidade de dados não refletia uma grande riqueza de conhecimento, já que a análise desses dados sem ferramentas adequadas ultrapassava a habilidade humana para compreensão de um volume tão extenso de informações armazenadas. Consequentemente, importantes decisões eram frequentemente tomadas somente por intuição, simplesmente pela falta dessas ferramentas que poderiam extrair conhecimentos valiosos dos repositórios de dados [3].

Isso motivou diversos estudos a partir do início dos anos 90, que resultaram na criação do campo de pesquisa de Mineração de Dados, área que se refere ao processo de descoberta de novas informações e conhecimento, no formato de regras e padrões, a partir de grandes bases de dados [5]. A partir daí, foram desenvolvidas ferramentas para analisar e descobrir importantes padrões de dados, contribuindo para diversas áreas, tais como [3]: pesquisas médicas, negócios estratégicos, biologia molecular, entre outras.

Existem quatro principais tarefas em Mineração de Dados [3]: classificação, regras de associação, clusterização e padrões seqüências. Em geral, essas tarefas podem ser classificadas em duas categorias: mineração preditiva e mineração descritiva [3].

Na mineração preditiva, deseja-se prever o valor desconhecido de um determinado atributo, a partir da análise histórica dos dados armazenados na base. Na mineração descritiva, padrões

e regras descrevem características importantes dos dados dos quais se está trabalhando.

2.2.1 TAREFAS DE MINERAÇÃO DE DADOS

2.2.1.1 CLASSIFICAÇÃO

A tarefa de classificação tem por objetivo identificar, entre um conjunto pré-definido de classes, aquela a qual pertence um elemento a partir de seus atributos. Para inferir a qual classe esse elemento pertence, é necessária uma base de treinamento.

Um sistema de um banco, que tem por objetivo inferir se um cliente será ou não um bom pagador, com base nos dados de clientes antigos e nas características do elemento que está sendo classificado, é um exemplo de utilização da tarefa de classificação.

2.2.1.2 REGRAS DE ASSOCIAÇÃO

Uma regra de associação representa um padrão de relacionamento entre itens de dados do domínio da aplicação que ocorre com uma determinada frequência na base. Ela é extraída a partir de uma base de dados que contêm transações, que são formadas por conjunto de itens do domínio da aplicação.

A rede de lojas Wal-Mart, após aplicar esta técnica em sua base de dados, descobriu que parte significativa das compras de homens às sextas-feiras à noite, que incluí fraldas, incluí também cerveja, ou seja, a regra de associação minerada diz que a compra de fraldas está associada a compra de cervejas nas condições descritas acima. Após uma análise mais detalhada sobre esse padrão extraído, soube-se que os pais ao comprarem fraldas para seus bebês, aproveitavam também para comprar cerveja para o final de semana. Esse é um exemplo clássico de extração de regras de associação. Outro exemplo, que foi extraído de uma base de dados médica após a aplicação dessa técnica, descreve que pacientes aidéticos que contraem a doença candíase também têm pneumonia.

2.2.1.3 CLUSTERIZAÇÃO

A tarefa de clusterização é usada para agrupar (clusterizar) elementos de uma base de dados através de seus atributos ou características, de forma que elementos mais similares fiquem no mesmo cluster e elementos menos similares fiquem em clusters distintos.

Esta técnica é muito utilizada em sistemas de grandes operadoras de cartão de crédito, separando os clientes em grupos de forma que aqueles que apresentam o mesmo comportamento

de consumo fiquem no mesmo grupo. A separação desses clientes por grupo pode ser usada para fazer algum tipo de marketing apropriado ao grupo ou na detecção de fraudes no caso de um cliente apresentar um comportamento diferente do esperado para o seu perfil.

2.2.1.4 PADRÕES SEQUENCIAIS

Nesta seção detalharemos com maior profundidade a técnica de extração de padrões sequenciais, visto que essa será aplicada em nosso trabalho.

Existem muitas aplicações envolvendo dados sequenciais, e a ordem com que esses dados aparecem é muito importante para análise e entendimento de alguns padrões. Exemplos típicos incluem sequências de compras de um cliente, sequências biológicas e sequências de eventos na ciência e na engenharia. Padrões sequenciais representam sequências de eventos ordenados, que aparecem com significativa frequência em uma base de dados. Um exemplo de padrão seqüencial é: "clientes que compram uma câmera digital Canon comumente compram uma impressora HP colorida dentro de um mês".

Uma sequência é uma lista ordenada de eventos e seu tamanho é determinado pelo seu número de itens. Uma sequência **s** é representada por $< e_1 e_2 e_3 ... e_n >$, onde e_k é dito um evento ou elemento da sequência **s** e e_1 ocorre antes de e_2, que ocorre antes de e_3 e assim sucessivamente. Por sua vez, um evento ou elemento da sequência é representado por **e** = $(i_1 i_2 i_3 ... i_m)$, onde i_k é um item do domínio da aplicação.

Podemos dizer que um evento em uma base de dados de uma loja de vendas é uma compra feita por um cliente, e o item do domínio da aplicação são os produtos que pertencem à compra do cliente.

Além disso, outras definições são importantes. Uma sequência pode ser parte de outra sequência maior. Nesse caso, a sequência $\alpha = < a_1 a_2 ... a_n >$ é chamada de subsequência de outra sequência $\beta = < b_1 b_2 ... b_m >$, e β é uma supersequência de α, denotado como $\alpha \subseteq \beta$, se existirem inteiros $1 \leq j_1 < j_2 < ... < j_n \leq$ m tais que $a_1 \subseteq b_{j1}$, $a_2 \subseteq b_{j2}, ..., a_1 \subseteq b_{jn}$. Por exemplo, se $\alpha = < (ab), d >$ e $\beta = < (abc), (de) >$, onde a, b, c, d e e são itens, então α é uma subsequência de β e β é uma supersequência de α.

Entretanto, padrões sequenciais se tornam inúteis se não pudermos classificá-los corretamente. Nesse contexto o suporte se apresenta como métrica de avaliação. Podemos definir o suporte de uma sequência como a quantidade de vezes que a mesma ocorre em determinada base de dados. Este suporte pode ser absoluto, representado apenas por um número inteiro, ou relativo, informando o percentual de vezes que a mesma sequência ocorreu. Dessa forma, uma

sequência de eventos é dita frequente se a quantidade de vezes que essa sequência ocorrer for superior ao suporte mínimo, formando assim um padrão sequencial.

2.3 MINERAÇÃO DE DADOS NA ENGENHARIA DE SOFTWARE

Tarefas de mineração de dados têm sido muito utilizadas na engenharia de software, principalmente aplicando suas técnicas em repositórios de dados para obter informações sobre a evolução de software ao longo do tempo, com o objetivo de aumentar a qualidade do processo de desenvolvimento de software [4, 6, 2, 7, 1].

[4] utilizou um algoritmo classificador, a partir do aprendizado em uma base de treinamento, para classificar a relação de relevância de manutenção entre dois arquivos. Esta relação é utilizada no contexto de manutenção de software.

Em seu trabalho, [6] aplica mineração de dados para encontrar relações de dependências no código fonte para ajudar os desenvolvedores em tarefas de modificação. Em especial, é utilizada a técnica de extração de regras de associação para determinar padrões de mudanças no código fonte.

[2] utiliza técnicas de mineração de dados num repositório UML versionado para extrair regras de associação que possam identificar elementos do modelo UML que foram modificados juntos no passado e que provavelmente precisarão ser modificados juntos no futuro.

Em [1], um algoritmo de clusterização foi utilizado num sistema de controle de versões para identificar classes semanticamente relacionadas. A partir do gráfico gerado pelo algoritmo, pôde-se analisar que mudanças em classes de certo cluster eram frequentemente envolvidas com mudanças em classes de outro cluster.

CAPÍTULO 3 - VERTICAL CODE COMPLETION

Cap 3.

CAPÍTULO 4 - IMPLEMENTAÇÃO

Capítulo 4

CAPÍTULO 5 - RESULTADOS EXPERIMENTAIS

Capítulo 5

CAPÍTULO 6 - CONCLUSÕES

Capítulo 6

REFERÊNCIAS BIBLIOGRÁFICAS

[1] T. Ball, J. Kim, A.A. Porter e et al. If your version control system could talk. *Workshop on Process Modelling and Empirical Studies of Software Engineering*, 1, 1997.

[2] C. Dantas, L. Murta e C. Werner. Mining change traces from versioned uml repositories. *XXI Simpósio Brasileiro de Engenharia de Software (SBES 2007)*, 1, October de 2007.

[3] J. Han e M. Kamber. *Data Mining: Concepts and Techniques (2nd edition)*. Morgan Kaufmann, 2006.

[4] J.S. Shirabad, T. Lethbridge e S. Matwin. Supporting software maintenance by mining software update records. *International Conference on Software Maintenance (ICSM)*, 1:22–31, November de 2001.

[5] R. Srikant e R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. Em *Proceedings of the 5th International Conference on Extending Database Technology - EDBT*, 1996.

[6] A.T.T. Ying, G.C. Murphy, R. Ng e et al. Predicting source code changes by mining change history. *IEEE Transactions on Software Engineering (TSE)*, 30(9):574–586, 2004.

[7] T Zimmermann, P. Weisgerber, S. Diehl e et al. Mining version histories to guide software changes. *International Conference on Software Engineering (ICSE)*, 1:563–572, May de 2004.