

MARACATU

A component search tool



**Especificação, Projeto e Implementação de uma
Arquitetura para um Engenho de Busca de Componentes**

Vinicius Cardoso Garcia



Agenda

- Introdução
- Especificação da Arquitetura
 - Características e Requisitos Técnicos
- Projeto da Arquitetura
 - Eclipse *Plug-in* e MARACATU *Server*
- Implementação da Arquitetura
- Trabalhos Relacionados
- Conclusões e trabalhos futuros

Introdução



- Reuso (McIlroy, 1968)
 - utilização de produtos de software em uma situação **diferente** daquela para qual estes produtos foram **originalmente construídos**
- Repositórios de componentes
 - gerenciamento, planejamento, processos de reuso, entre outros (Morisio, 2002; Ravichandran, 2003)
- Atual -> reuso **black-box**
 - Adoção de forma intrusiva e como soluções isoladas
- Ideal -> **white-box**

Especificação da Arquitetura



- Requisitos para um **eficiente** engenho de busca e recuperação de componentes de software (Lucredio, 2003)
 - Elevada precisão e recuperação
 - Segurança
 - Formulação de consultas
 - Descrição do componente
 - Familiaridade no repositório
 - Interoperabilidade
 - Performance
- Entretanto, esses requisitos são referentes a um mercado de componentes utilizando componentes **black-box**

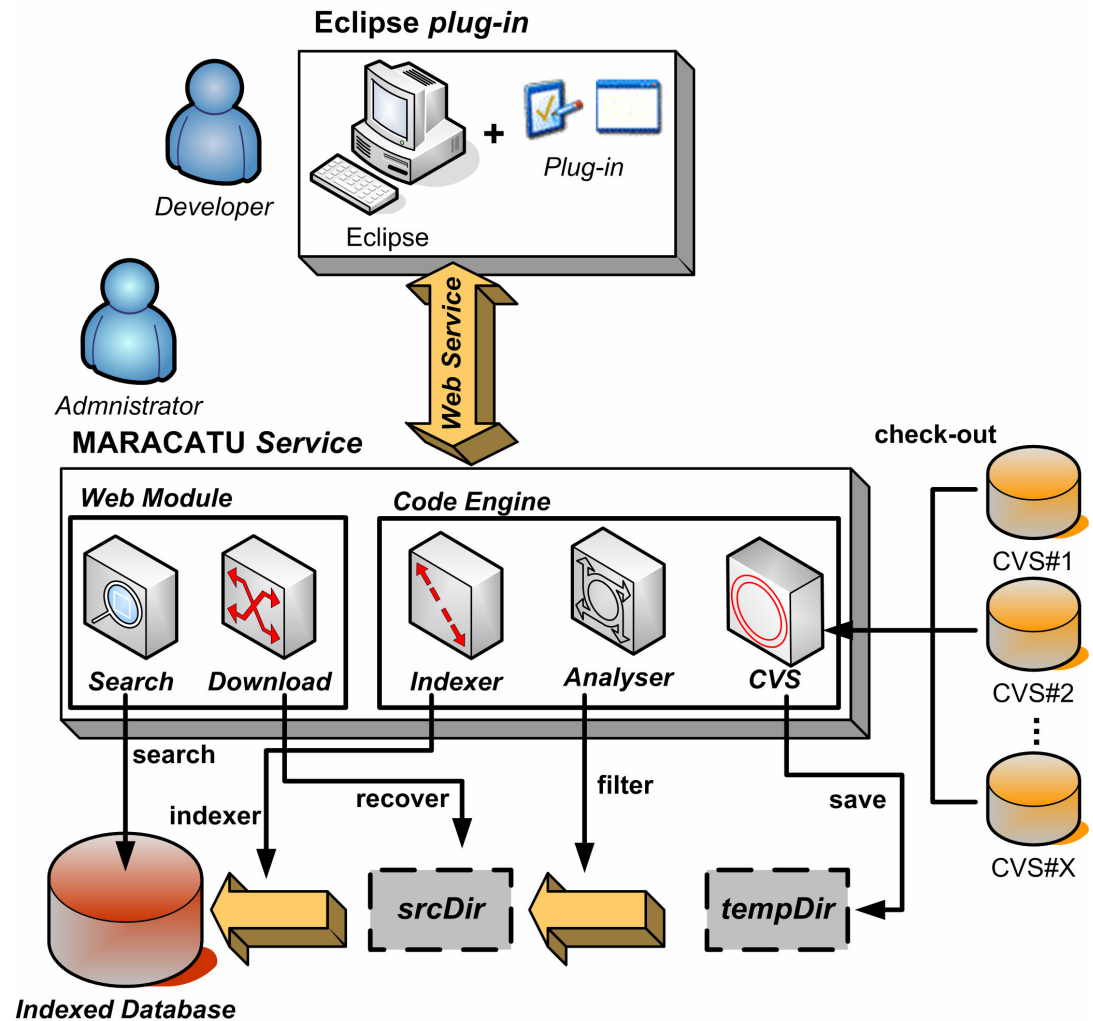
Características e Requisitos Técnicos



- Busca em repositórios de código fonte de projetos
 - localização, *check-out* e indexação dos componentes; e busca e recuperação de componentes
- Requisitos do MARACATU
 - Repositórios CVS na *Web* ou na *Intranet*
 - Selecionar Repositórios
 - Armazenar Componentes Localmente
 - Atualizar Repositórios
 - Indexar os Componentes
 - Buscar por Palavra-Chave
 - Exibir resultado da busca

Projeto da Arquitetura

- A arquitetura do MARACATU foi projetada para ser extensível a diferentes tecnologias de componentes, provendo a capacidade de adição de novas características na indexação, ranqueamento, busca e recuperação dos componentes.



Projeto da Arquitetura



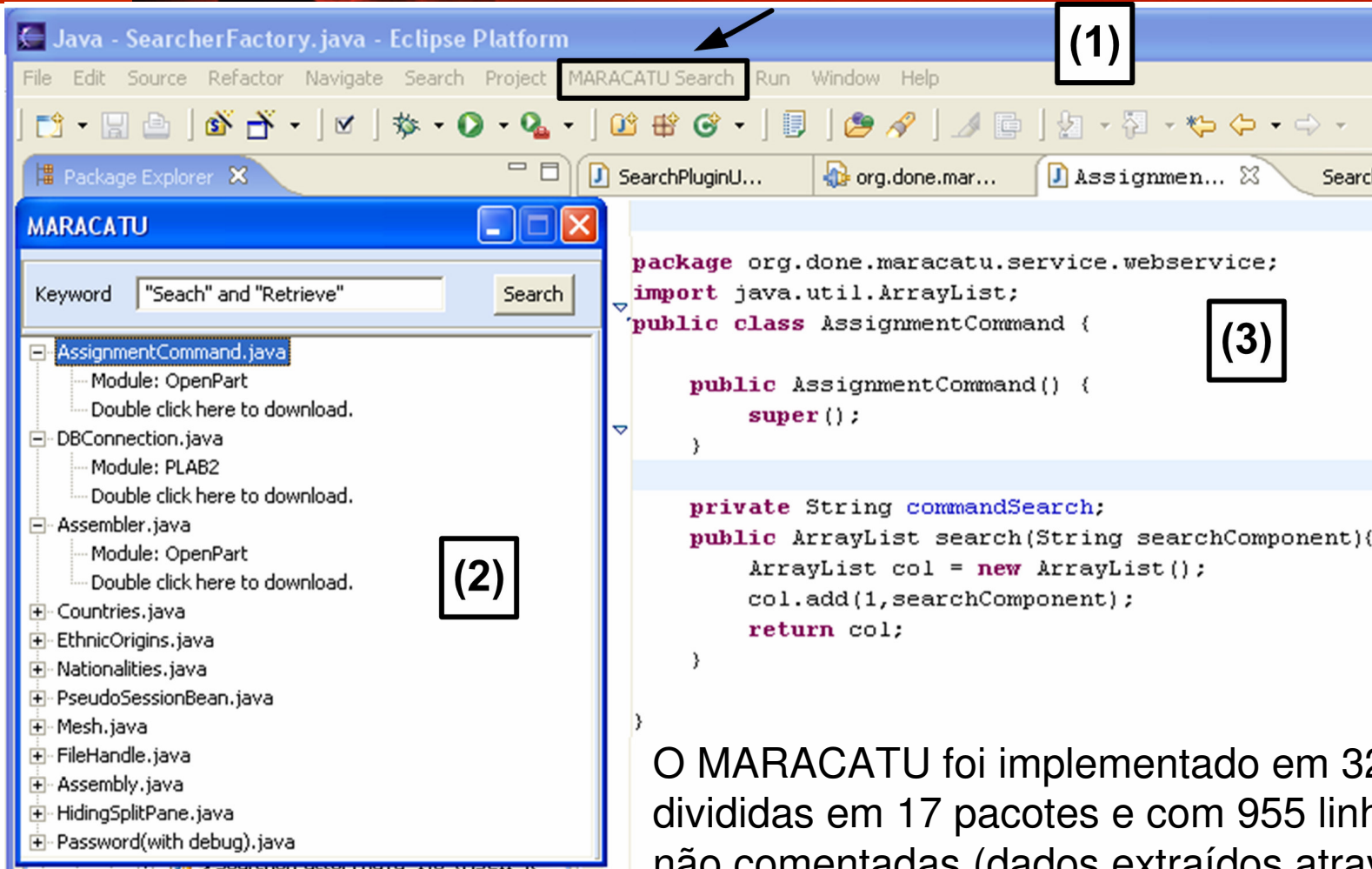
- A arquitetura do MARACATU contém dois sub-sistemas principais:
- Eclipse *Plug-in*
 - interface visual com o desenvolvedor
 - responsável por repassar as consultas para o sub-sistema *MARACATU Service*
 - se comporta como um cliente *Web Service* que consome os serviços disponíveis pelo *MARACATU Service*
- *MARACATU Service*
 - responsável pela execução dos serviços requisitados
 - dividido em sub-módulos: *CVS*, *Analyser*, *Indexer*, *Download* e *Search*

MARACATU *Service*



- **CVS**
 - é responsável por recuperar o código fonte dos repositórios CVS cadastrados
 - monitora os repositórios com o objetivo de recuperar as atualizações.
- **Analyser**
 - responsável por realizar uma análise no código e avaliar se o mesmo é adequado para ser reutilizado.
- **Indexer**
 - responsável por indexar os arquivos Java que passaram pela análise do módulo *Analyser*
- **Download**
 - auxilia o processo de download (check-out) do código fonte para a máquina do desenvolvedor
- **Search**
 - recebe os parâmetros da consulta solicitada pelo desenvolvedor, interpreta, busca no índice, e retorna um conjunto de entradas do índice.

Implementação da Arquitetura



The screenshot shows the Eclipse IDE interface. At the top, the title bar reads "Java - SearcherFactory.java - Eclipse Platform". The menu bar includes "File", "Edit", "Source", "Refactor", "Navigate", "Search", "Project", "MARACATU Search", "Run", "Window", and "Help". An arrow points to the "MARACATU Search" menu item, which is labeled with a circled (1). Below the menu bar is a toolbar with various icons. The Package Explorer on the left shows a tree structure for "MARACATU" with several Java files listed, including "AssignmentCommand.java", "DBConnection.java", "Assembler.java", "Countries.java", "EthnicOrigins.java", "Nationalities.java", "PseudoSessionBean.java", "Mesh.java", "FileHandle.java", "Assembly.java", "HidingSplitPane.java", and "Password(with debug).java". This list is labeled with a circled (2). The main editor window displays the source code for "AssignmentCommand.java", which is labeled with a circled (3). The code includes package declarations, imports, and class definitions.

```
package org.done.maracatu.service.webservice;
import java.util.ArrayList;
public class AssignmentCommand {

    public AssignmentCommand() {
        super();
    }

    private String commandSearch;
    public ArrayList search(String searchComponent){
        ArrayList col = new ArrayList();
        col.add(1,searchComponent);
        return col;
    }
}
```

O MARACATU foi implementado em 32 classes, divididas em 17 pacotes e com 955 linhas de código não comentadas (dados extraídos através da ferramenta JavaNCSS).

Trabalhos Relacionados (1)



- Agora (Seacord, 1998)
 - protótipo desenvolvido pelo SEI/CMU
 - objetivo é criar um repositório, indexado e disponível na internet, de produtos de software classificados por tipos de componentes (por exemplo, Javabeans ou ActiveX)
 - combina técnicas de introspecção com mecanismos de busca da Web

- Koders (<http://www.koders.com>)
 - primeiro motor de busca de projetos *Open Source*
 - o objetivo é apoiar os desenvolvedores de software através da comunidade de 15,5 milhão de projetos de software livre
 - se conecta diretamente nos sistemas de controle de versão (como CVS e Subversion)
 - reconhece cerca de 30 linguagens de programação e 20 licenças de software
 - projeto mantido pela Microsoft Architecture Resource Center

Trabalhos Relacionados (2)



- Strathcona (Holmes, 2005)
 - localiza exemplos de códigos que auxiliam desenvolvedores no processo de codificação
 - exemplos extraídos de repositórios através de pesquisa baseada em seis diferentes heurísticas
 - A escolha das classes é baseada na similaridade da estrutura do código que o desenvolvedor está escrevendo
- (Ye, 2002)
 - localiza e apresenta para o desenvolvedor, de forma autônoma, informações sobre componentes que sejam relevantes para as atividades que estão sendo realizadas no momento
 - personalizados de acordo com o conhecimento e ambiente do desenvolvedor
 - Avaliações empíricas mostraram que este tipo de estratégia é realmente efetiva em promover o reuso
 - exploração de um novo estilo de colaboração entre homem-máquina

Conclusões e trabalhos futuros



- De McIlroy até hoje (Lucrecio, 2004) a área de busca e recuperação de componentes vem evoluindo
- O MARACATU permite, não apenas a recuperação de componentes **black-box**, como também, componentes **white-box** e **trechos de código** existentes
- Módulo extrator
 - componentes existentes possam ser **recuperados** e **classificados** em facetas pré-definidas
 - Com a utilização de **facetras**, as buscas poderão ser otimizadas
- Outras funcionalidades
 - incorporação de conteúdo **semântico**
 - melhorias nos algoritmos de **ranqueamento** do engenho de busca
 - definição de **métricas**
- Projeto piloto: analisar a viabilidade de utilização da ferramenta no **ambiente industrial**.

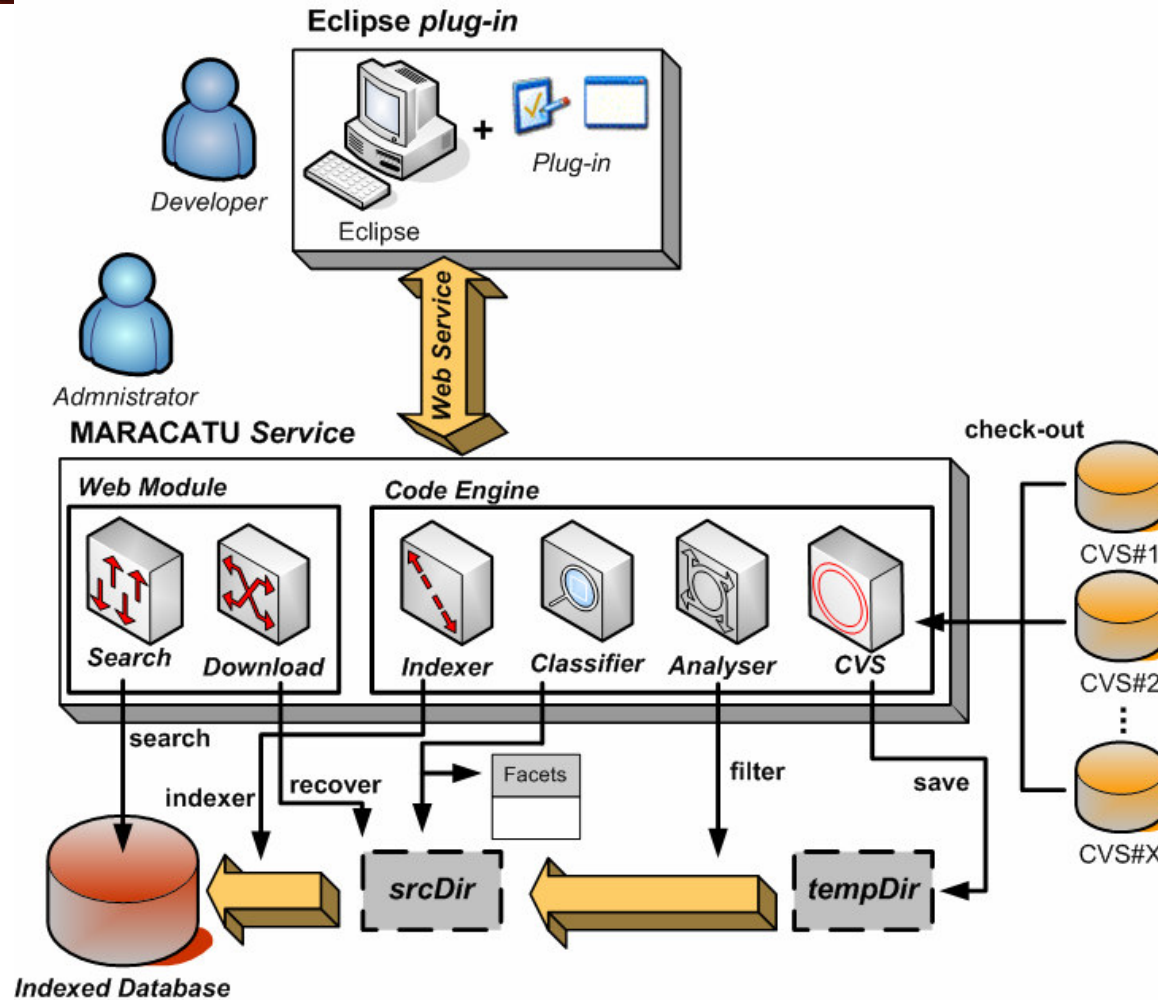
MARACA2

R:SE



R:SE

REUSE IN SOFTWARE ENGINEERING GROUP



References



- R. Holmes and G. C. Murphy. Using structural context to recommend source code examples. In Proceedings of the 27th ICSE, pages 117–125, New York, NY, USA, 2005. ACM Press.
- D. Lucrédio, E. S. d. Almeida, and A. F. d. Prado. A Survey on Software Components Search and Retrieval. In R. Steinmetz and A. Mauthe, editors, 30th IEEE EUROMICRO Conference, Component-Based Software Engineering Track, pages 152–159, Rennes - France, 2004. IEEE/CS Press.
- M. D. McIlroy. Software Engineering: Report on a conference sponsored by the NATO Science Committee. In NATO Software Engineering Conference, pages 138–155. NATO Scientific Affairs Division, 1968.
- M. Morisio, M. Ezran, and C. Tully. Success and Failure Factors in Software Reuse. IEEE Transactions on Software Engineering, 28(04):340–357, 2002.
- T. Ravichandran and M. A. Rothenberger. Software Reuse Strategies and Component Markets. Communications of the ACM, 46(8):109–114, 2003.
- R. C. Seacord, S. A. Hissam, and K. C. Wallnau. Agora: A Search Engine for Software Components. Technical report, CMU/SEI - Carnegie Mellon University/Software Engineering Institute, 1998.
- Y. Ye and G. Fischer. Supporting Reuse By Delivering Task-Relevant and Personalized Information. In Proceedings of the 24th ICSE, pages 513–523, Orlando, Florida, USA, 2002.