Infraestrutura para automatização do Analizo e ferramenta para visualização de software com DSM

Joenio Costa
DCC UFBA
joenio@colivre.coop.br

Daniela Feitosa
DCC UFBA
daniela@colivre.coop.br

***Resumo*—Relatório técnico sobre a implementação de uma infraestrutura para automatização do *Analizo* e uma ferramenta para visualização de software com *Design Structure Matrices*.**

I. Introdução

A solução implementada é composta do *Analizo*[1] e duas novas ferramentas chamadas *CodeJuicer* e *Visualizo*, juntas elas fornecem um ambiente Web que possibilitam através do fornecimento de uma URL o download, análise e extração de informações contidas no código fonte hospedado na URL, armazenamento das informações e visualização através de *Design Structure Matrix (DSM)*[2] exibindo o inter-relacionamento entre as entidades encontradas no código fonte.

A figura 1 apresenta a arquitetura da solução e as seções seguintes trazem detalhes sobre cada componente.

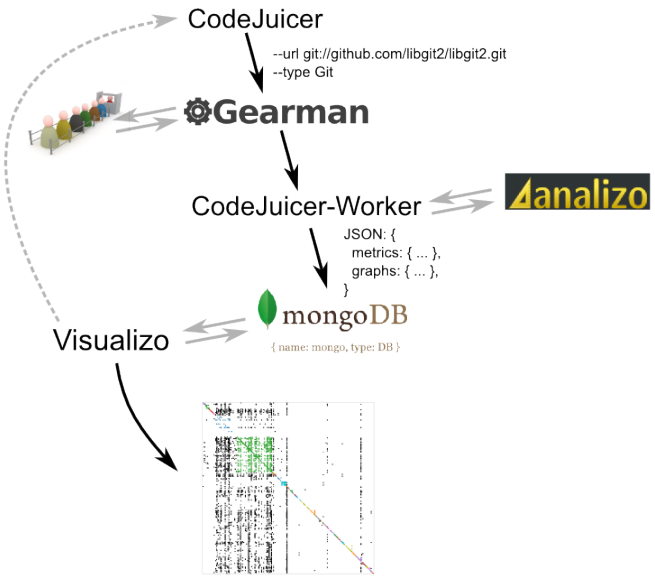

Figura 1. Arquitetura da solução

II. Analizo

O Analizo[1] é um conjunto de feramentas livres para análise, extração e visualização de software extensível e independente de linguagem. Ele é capaz de analisar o código fonte de um software, extrair métricas e informações dos relacionamentos entre as suas entidades, prover visualizações, bem como fornecer dados para ferramentas externas.

III. CodeJuicer

Ferramenta desenvolvida para automatizar o uso do *Analizo*, responsável por gerenciar o download e análise de código fonte, além de armazenar e disponibilizar tais informações. A figura 2 possui uma visão geral da arquitetura interna do *CodeJuicer*.

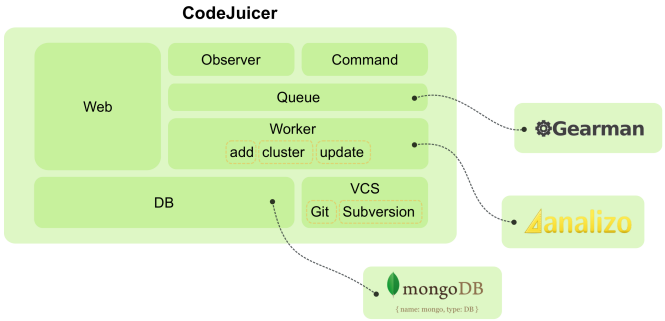

Figura 2. Arquitetura interna do CodeJuicer

Todas as tarefas realizadas pelo *CodeJuicer* são executadas em segundo plano e passam por uma fila gerenciada pelo *Gearman*[2], um motor de gestão de tarefas escrito em e C independente de linguagem. Os dados gerados são armazenados num banco de dados *MongoDB*[3], um banco de dados de documentos *NoSQL* que utiliza o formato *JSON* nativamente.

Segue abaixo detalhes de cada componente de sua arquitetura interna:

CodeJuicer::Observer *daemon* que verifica periodicamente por atualizações nas URLs, obtém atualizações, executa análise e atualiza informações no banco de dados.

CodeJuicer::DB simples interface de acesso ao banco de dados *MongoDB*.

CodeJuicer::VCS camada de abstração de acesso a repositório remoto de código fonte, atualmente suporta *Git*[4] e *Subversion*[5].

CodeJuicer::Worker é o ponto central de execução de todas as tarefas, conecta cada componente do *CodeJuicer*.

[1] http://analizo.org

[2] http://www.gearman.org
[3] http://www.mongodb.org
[4] http://git-scm.com
[5] http://subversion.apache.org

CodeJuicer::Cmd::Command interface de linha de comando para as funções do *CodeJuicer*, permite adicionar tarefas na fila do *Gearman*.

Codejucier::Web interface Web para consulta aos dados em *JSON* armazenados no banco de dados *MongoDB*.

A seguir um exemplo de uso da sua inteface de linha de comando para adicionar repositórios de código fonte:

```
codejuicer add --url <URL> --type <TYPE>
```

Isto irá adicionar uma tarefa na fila do *Gearman* para que no momento certo este execute o *CodeJuicer::Worker* que irá então concluir as seguintes atividades: a) baixar o código fonte, b) extrair informações através do *Analizo* e c) guardar as informações extraídos no banco de dados *MongoDB*.

O banco de dados utilizado pelo *CodeJuicer* é nomeado *codejuicer* e possui as seguintes *collections* (o termo collection no MongoDB é o equivalente a tabela num bancos de dados relacional):

repositories: URLs cadastradas no *CodeJuicer* são armazenados nessa collection, informações de processamento e progresso são atualizadas constantemente.

- **_id** Chave primária composta pelo hash sha1 da URL.
- **type** Tipo do repositório: Git ou Subversion.
- **url** URL do repositório.
- **status** O CodeJuicer atualiza este campo a medida que executa as etapas do processamento: adding, added, updating ou updated.
- **progress** Campo numérico variando de 0 a 100 indicando o progresso atual.
- **added_at** Data de quando o repositório foi adicionado ao CodeJuicer.
- **updated_at** Data da última atualização do repositório pelo CodeJuicer.
- **error** Em caso de erro este campo armazena a mensagem de erro gerada.

metrics: contém as métricas extraídas a partir da análise do codigo fonte.

_id Hash sha1 da URL.
global_metrics Métricas globais extraídas pelo Analizo.
modules_metrics Métricas por módulo extraídas pelo Analizo.

graphs: armazena o grafo de dependência entre módulos do sistema e o grafo representando uma *Design Structure Matrix*.

_id Hash sha1 da URL.
dsm Grafo representando a DSM.
callgraph Grafo de chamada/dependencia entre módulos (*não implementado ainda*).

Mais detalhes sobre a implementação do *CodeJuicer* e seus detalhes internos podem ser obtidos em http://github.com/joenio/codejuicer.

IV. Visualizo

Ferramenta que implementa uma visualização de software utilizando *Design Structure Matrices (DSM)* a partir de informações disponibilizadas pelo *CodeJuicer*. Foi desenvolvido com tecnologias Web e baseia-se fortemente na biblioteca *D3.js*[6], uma biblioteca JavaScript para manipular documentos de dados provendo componentes poderosos de visualização, combinando HTML, SVG e CSS.

O *Visualizo* é composto por uma interface para cadastro de URL (Figura 3) e outra para visualização da *DSM* (Figura 4).

Figura 3. Interface para cadastro de URL do Visualizo

Após inserir a URL de um repositóro no *Visualizo*, este irá executar o *CodeJuicer* para efetuar o download do código fonte e execução do *Analizo* para extração de métricas e outras informações necessárias para criar a visualização (figura 4).

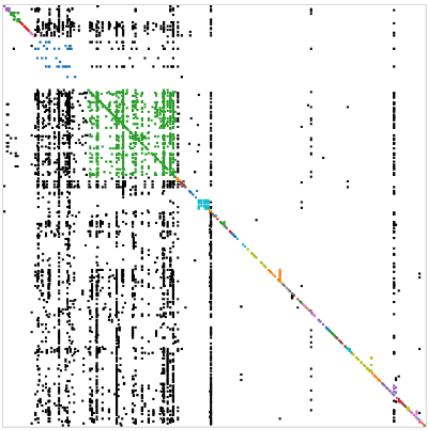

Figura 4. Design Structure Matrix gerada pelo Visualizo

As cores das células (figura 4) indicam que os arquivos dependentes estão num mesmo diretório. As células na cor preta indicam que os arquivos dependentes estão em diretórios diferentes.

Além da visualização da *DSM* é possível consultar nesta mesma interface algumas informações sobre o software analisado, opções de reordenação e uma upção de zoom da matrix *DSM*.

O *Visualizo* está disponível como software livre e pode ser obtida em http://github.com/joenio/visualizo.

Referências

[1] A. T. J. C. J. M. P. M. L. R. R. L. A. C. C. F. Kon, "Analizo: an extensible multi-language source code analysis and visualization toolkit," p. 6, 2010.
[2] A. M. J. R. C. Baldwin, "Exploring the structure of complex software designs an empirical study of open source and proprietary code," p. 40, 2004.

[6] http://d3js.org