

Geração Automática de Documentação a Partir de Código Fonte

Christiano de Oliveira Braga¹
Marcelo Moreira
Arndt von Staa²

Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro

{cbraga, moreira, arndt}@inf.puc-rio.br

Abstract

Program documentation is fundamental for the production and evolution of quality software. After investigating several documentation tool proposals, we developed our own tool which addresses the main attributes of an ideal solution and generates an HTML hypertext. We implemented **Documentu** integrating the transformation system Draco-PUC and the software engineering environment Talisman. This latter provides a Text database with an ample and evolvable set of relations, and provides an internal data manipulation language which can be used to write validators and linearizers used to generate the HTML text.

Keywords

Documentation, transformation systems, CASE tools, software engineering, tool integration

Resumo

A produção de documentação de boa qualidade é uma tarefa crítica no processo de desenvolvimento de software. Após analisarmos algumas propostas existentes, implementamos uma ferramenta, que suporta os principais conceitos encontrados nesta análise, gerando a documentação sob a forma de um hipertexto HTML. A ferramenta **Documentu** foi arquitetada e construída integrando o sistema transformacional Draco-PUC e o ambiente Talisman que provê uma base de fragmentos de textos e relações, além de uma linguagem de manipulação de dados que permite criar validadores e linearizadores utilizados para gerar o texto HTML.

Palavras-chave

Documentação, aplicação de sistemas transformacionais, ferramentas CASE, integração de ferramentas, engenharia de software

¹ Apoiado por CAPES e CENPES

² Apoiado por: CNPq, Bolsa de Pesquisador 300029/92-6 e CENPES

1 Introdução

Há muito tempo o problema associado à documentação de sistemas de software tem sido motivo de estudos [Horowitz85] [Garg90]. A produção e manutenção de documentação de boa qualidade deve ser encarada como uma tarefa importante e deve ser realizada ao longo de todo o processo de desenvolvimento de aplicações. Neste trabalho apresentamos uma ferramenta que produz documentos em HTML a partir de diversos módulos de código fonte em C++.

A documentação gerada por **Documentu** é extraída do código fonte e das informações existentes nos comentários. Os comentários devem conter marcadores em conformidade com o proposto em [Staa95], que descreve regras para a inserção de especificação de *design* em código C ou C++. **Documentu** utiliza a sintaxe do programa para gerar o esqueleto do hipertexto, e utiliza os comentários para detalhar e complementar este esqueleto. Por esta razão, é possível produzir documentação baseada exclusivamente no código fonte sem comentários. Além disso a estrutura da documentação é um reflexo fiel do que efetivamente se encontra implementado, permitindo, assim, também a inspeção do sistema quanto à correção de sua organização.

A extração de informação do código e dos fragmentos de comentários é realizada através de análise sintática do código fonte realizada pelo sistema transformacional Draco-PUC [Leite94]. Esta informação é transformada para satisfazer as regras de arquivos de importação aceitos por Talisman. Este integra toda a documentação, estabelecendo relações entre os diversos elementos independentemente do módulos em que se encontrem. Além disso, Talisman [Staa93] é utilizado para realizar validações estruturais e para gerar a documentação em formato aceitável por um *browser* WWW. Finalmente, o *browser* é utilizado para explorar a documentação.

2 A arquitetura da ferramenta Documentu

A figura a seguir ilustra a arquitetura geral de **Documentu**.

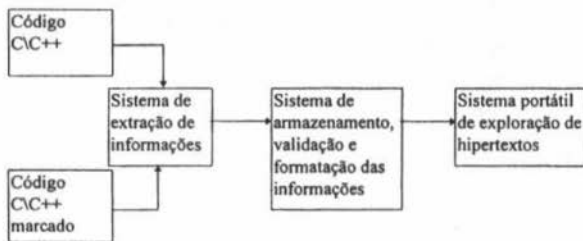


Figura 1 - Visão global da ferramenta

2.1 Draco-PUC

Draco-PUC [Leite94] é um sistema transformacional que propõe o desenvolvimento de software através da criação e da reutilização de domínios, num alto nível de abstração. Um domínio em Draco-PUC é definido por uma gramática escrita numa linguagem no estilo de BNF. A partir desta gramática é produzido o *parser* do domínio. Um *unparser* ou *pretty-*

printer é construído pelo processador a partir de informações de formatação de *lay-out* contidas no arquivo de definição da linguagem.

Draco-PUC cria uma árvore de sintaxe abstrata especial (DAST), armazenando a informação recuperada pelo *parser*. Uma vez criada esta estrutura, pode-se aplicar transformações que mapeiam informações da DAST para uma nova estrutura no mesmo domínio (transformações horizontais), ou para um domínio diferente (transformações verticais). Uma transformação em Draco-PUC tem um *left-hand-side* (LHS), que representa o padrão de busca a ser substituído pelo *right-hand-side* (RHS) da transformação. O RHS deve satisfazer o domínio destino da transformação.

Regras de transformação são agrupadas em Conjuntos de Transformações, que tem pontos de controle. O conjunto de conjuntos de transformações constitui um Transformador. A figura a seguir mostra todos os pontos de controle possíveis no sistema de transformação.

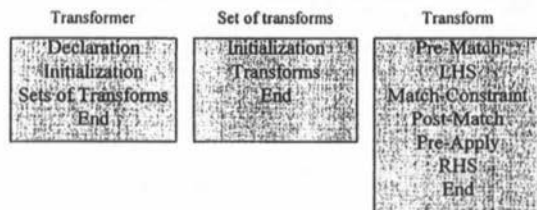


Figura 2 - Os pontos de controle das transformações de Draco-PUC

2.2 Talisman

Talisman [Staa93] é uma ferramenta CASE que utiliza repositórios para armazenar os fatos relativos a sistemas ou componentes. Talisman utiliza duas bases de dados interdependentes: a base de definição e a base de software. A base de definição configura a instância de ambiente em uso. A base de software é um banco de dados que armazena todas as informações a respeito de um dado sistema ou conjunto de sistemas. Este banco de dados é composto de fragmentos de texto e referências a fragmentos de texto que estão armazenados nesta mesma base de dados. Uma linguagem de representação é caracterizada por relações entre seus elementos (módulos, classes e funções, por exemplo) e também por formulários (validadores e linearizadores). Esses formulários podem ser incorporados ao ambiente em tempo de execução, personalizando o ambiente.

Os validadores de Talisman verificam a consistência entre os dados existentes dentro da base de software e os linearizadores são responsáveis pela geração de relatórios com as informações contidas dentro da base.

Para desenvolver um sistema utilizando Talisman, são utilizados formulários de edição. Dependendo da linguagem de definição que está sendo utilizada, fragmentos de texto podem ter significados diferentes. Por exemplo, um repositório de dados num DFD pode ser mapeado para uma tabela num modelo de entidades e relacionamentos. Uma vez que a base de software tenha sido povoada, devem ser aplicados validadores para realizar as verificações de consistência das informações dentro da base. Com a informação validada, são aplicados linearizadores para produzir a documentação ou o código.

O ambiente Talisman permite popular uma base de software com as informações existentes num arquivo de um **arquivo de importação**. As regras que ditam como este arquivo deve ser construído podem ser escritas na forma de uma gramática. A partir desta

gramática foi construído o domínio Talisman. Este domínio é utilizado por Draco-PUC para transformar o código fonte C++ para arquivos de importação Talisman.

2.3 A integração dos ambientes CASE

O processo de geração da documentação começa com a análise do código fonte utilizando duas gramáticas (ver figura a baixo), uma para analisar o código C++ e outra para analisar os comentários marcados segundo as recomendações contidas em [Staa95]. Uma vez que um determinado módulo foi analisado por Draco-PUC, é aplicado um conjunto de transformações criando o arquivo de importação Talisman para aquele módulo. Este processo é repetido para cada módulo do sistema.

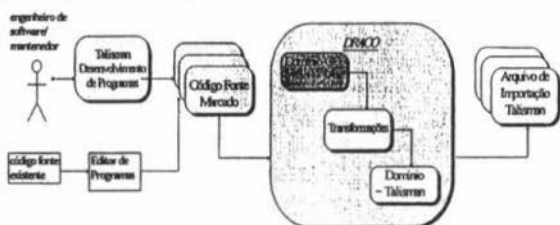


Figura 3 - A primeira parte do processo de geração de documentação

A segunda parte do processo utiliza Talisman, importando cada um dos arquivos gerados por Draco-PUC, ver figura a seguir:



Figura 4 - A segunda parte do processo de geração de documentação

Uma vez que todos os arquivos foram importados, os validadores Talisman são executados, verificando a consistência entre os dados na base de software. Estes validadores verificam se os fragmentos de texto obrigatórios foram preenchidos e validam as referências cruzadas. Um *log* é criado com as inconsistências detectadas. Uma vez que a base de software tenha sido aceita, o conteúdo da base é linearizado produzindo os documentos HTML.

2.4 O Sistema de hipertexto

No hipertexto gerado por **Documentu**, esquematizado na figura a seguir, todos os *links* são criados com as informações conseguidas através da análise do código fonte. Ou seja, o desenvolvedor da aplicação³ não precisa gerar *links*. Já as informações de detalhamento acerca de cada um dos elementos da linguagem (classes ou métodos, por exemplo) provém dos fragmentos de comentários identificados pelos marcadores de especificação existentes no código fonte.

³ A criação de *links* explícitos pode ser feita utilizando marcadores HTML nos comentários do código fonte.

