

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Ricardo Robalinho Kawase

**Um framework para a construção de interfaces
para manipulação direta de conjuntos**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para
obtenção do título de Mestre pelo Programa de Pós-
Graduação em Informática da PUC-Rio.

Orientador: Daniel Schwabe

Rio de Janeiro
Setembro de 2006



Ricardo Robalinho Kawase

**Um framework para a construção de Interfaces
para manipulação direta de Conjuntos**

Dissertação apresentada como requisito parcial para obtenção do título de Mestre pelo Programa de Pós-Graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Daniel Schwabe
Orientador
PUC-Rio

Prof. Bruno Feijó
PUC-Rio

Profa. Simone Diniz Junqueira Barbosa
PUC-Rio

Prof. José Eugenio Leal
Coordenador Setorial do Centro
Técnico Científico - PUC-Rio

Rio de Janeiro, 4 de setembro de 2006

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Ricardo Robalinho Kawase

Graduou-se em Engenharia de Computação na PUC-Rio em 2003. Atuou como programador em empresas desenvolvendo projetos para Web. Possui interesse acadêmico e profissional nas áreas de Hipertexto e Multimídia, Interação Humano-Computador e Computação Gráfica.

Ficha Catalográfica

Kawase, Ricardo Robalinho

Um framework para a construção de interfaces para manipulação direta de conjuntos / Ricardo Robalinho Kawase ; orientador: Daniel Schwabe. – 2006.

78 f. : il. ; 30 cm

Dissertação (Mestrado em Informática)– Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2006.

Inclui bibliografia

1. Informática – Teses. 2. Conjuntos. 3. Navegação por Conjuntos. 4. Modelo de Informação. 5. Navegação Facetada. 6. Framework. 7. Programação por Usuário Final. 8. Manipulação Direta. I. Schwabe, Daniel. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Este trabalho é dedicado à minha família,
pela compreensão e pelo apoio,
e ao meu orientador,
pela motivação.

Agradecimentos

À PUC-Rio e ao departamento de informática pela oportunidade.

Ao meu orientador, Daniel Schwabe, pela motivação e confiança.

Aos amigos que cursaram o mestrado comigo pelas ótimas conversas e trocas de idéias.

A todos os professores, funcionários do Departamento de Informática pelo apoio dado quando precisei.

A Deborah pela paciência e ajuda nos processos burocráticos.

Aos meus amigos e familiares pelo apoio nos momentos difíceis.

A minha querida namorada por estar presente me apoiando e motivando a todo o momento.

Aos meus amados pais Jiro e Glória por tudo.

Resumo

Kawase, Ricardo Robalinho; Schwabe, Daniel. **Um framework para construção de interfaces para manipulação direta de conjuntos**. Rio de Janeiro, 4 de setembro de 2006. 78p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Essa dissertação propõe o desenvolvimento de um framework capaz de gerar uma interface para a navegação e a manipulação direta sobre grupos e objetos de um determinado repositório. Baseada no mantra de busca visual de Shneiderman, "Overview first, zoom and filter, then details on demand" (Visão geral primeiro, zoom e filtragem, depois detalhes sob demanda), a zoomable user interface gerada pelo framework, através da utilização da biblioteca Piccolo, possui as características de exibir todos os grupos em uma única tela, filtros e o zoom infinito. O objetivo deste framework é permitir ao desenvolvedor acoplar qualquer biblioteca baseada em grupos e objetos à interface gerada, e acoplar a sua linguagem específica de domínio que descreve as operações sobre os objetos às interações de manipulação direta. Através da rápida implementação de uma classe de objeto customizável e de uma classe de mapeamento, objetos do repositório são mapeados para uma forma correspondente no framework de acordo com o modo de renderização desejado, assim como as operações sobre os objetos definidas na DSL são mapeadas para operações na interface. Com os mapeamentos das operações, fica aberto para o usuário final, através de um arquivo de configuração muito simples, selecionar quais interações de manipulação direta sobre a interface chamam quais operações. A interface final se utiliza dos dados e ações fornecidos pelo desenvolvedor e se comporta conforme escolha do usuário final.

Palavras-chave

Conjuntos; Navegação por Conjuntos; Modelo de Informação; Navegação Facetada; Framework; Programação por Usuário Final; Manipulação Direta; Zoom; ZUI; DSL.

Abstract

Kawase, Ricardo Robalinho; Schwabe, Daniel. **A framework for development of interfaces for direct manipulation on sets**. Rio de Janeiro, September 4, 2006. 78p. MSc. Dissertation - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

This dissertation proposes the development of a framework capable to generate an interface for navigation and direct manipulation on groups and objects of a specific repository. Based on Shneiderman's Visual Information Seeking Mantra, "Overview first, zoom and filter, then details on demand" the zoomable user interface generated by the framework, using Piccolo library, has the characteristics of exhibit all groups in a single canvas, filters and infinite zoom. The objective of this framework it to allow developers to join the generated interface with any library based on groups and objects, and to join the direct manipulation interactions with it's domain-specific language that describes the operations on objects. Through a quick implementation of the custom object class and of the mapping class, repository's objects are mapped to a correspondent format in the framework according to the desired mode of renderization, as well the operations on objects defined in the DSL are mapped to operations on the interface. With the map of the operations, it is up to the user, through a simple configuration file, to select which direct manipulation interactions on the interface call which operation. The final interface uses the data and the actions given by the developer and behaves according to the choices of the end-user.

Keywords

Sets; Set Navigation; Information Model; Facet Navigation; Framework; End-User Programming; Direct Manipulation; Zoom; ZUI; DSL

Sumário

1	Introdução	14
1.1	Motivação	14
1.2	Objetivo	15
1.3	Organização da dissertação	15
2	Fundamentos Teóricos	16
2.1	Visualização de Informação	16
2.2	Facetas	18
2.2.1	Classificação Facetada	18
2.3	Manipulação Direta	20
2.4	Programação por Usuário Final	23
2.5	Frameworks	25
2.6	Modelo de Implementação	27
3	A Biblioteca Piccolo	30
3.1	Visão geral da arquitetura Piccolo	30
3.2	Nós	31
3.2.1	Usando classes existentes	31
3.2.2	Subclasses de classes existentes	31
3.2.3	Composição	31
3.2.4	Nó Customizado	32
3.3	Interações do usuário	33
3.4	Restrições de Layout	33
3.5	Ações	34
3.6	Modelo de Implementação	34
3.6.1	A Classe PNode	35
3.6.2	A Classe PCamera	36
3.6.3	A Classe PLayer	36
3.6.4	A Classe PRoot	36

3.6.5 A Classe PCanvas	36
4 O Framework	37
4.1 Tecnologias Utilizadas	37
4.2 Arquitetura	38
4.2.1 Classes	38
4.2.2 Aplicação	41
4.3 Utilizando o framework	43
4.3.1 O Desenvolvedor	43
4.3.2 O Usuário Final	51
4.3.3 Guia de Uso da Interface	53
4.4 O Resultado	55
5 Ilustração de Uso	59
5.1 O Modelo	59
5.2 Mapeamento de Classes	60
5.2.1 Mapeando Grupos	61
5.2.2 Mapeando Items	63
5.3 Implementando a Classe Map	65
6 Conclusão	69
6.1 Trabalhos Relacionados	69
6.2 Contribuições	73
6.3 Trabalhos Futuros	74
7 Referências bibliográficas	76

Siglas

API	Application Programming Interface
DSL	Domain-Specific Language
EUP	End-User Programming
GDI	Graphics Device Interface
GUI	Graphical User Interface
HTML	HyperText Markup Language
MVC	Model View Controller
OO	Orientação a Objeto
PDA	Personal Digital Assistant
WIMP	Windows Interface Manipulation Program
WWW	World Wide Web
XML	Extensible Markup Language
ZUI	Zoomable User Interfaces

Lista de figuras

Figura 1 – Seleção de Facetas.....	20
Figura 2 – Biblioteca e Framework	26
Figura 3 – Extensão de Bibliotecas e Frameworks	26
Figura 4 – Modelo MVC	28
Figura 5 – Hierarquia de classes do Piccolo.....	34
Figura 6 – Estrutura de Execução do Piccolo.....	35
Figura 7 – Arquitetura da Aplicação.	38
Figura 8 – Interface de navegação.....	56
Figura 9 – Interface de navegação - grupos e itens.	56
Figura 10 – Interface de navegação – visualizando item.....	57
Figura 11 – Interface de navegação – Seleccionando filtros da lente.	57
Figura 12 – Interface de navegação com lente de filtragem.....	58
Figura 13 – Diagrama de Classes	59
Figura 14 – Modelo de exibição compacta do item vinho.	63
Figura 15 – Modelo de exibição integral do item vinho.	63
Figura 16 – Niagara - Visão de grupos de notas.....	69
Figura 17 – PhotoMesa - Zoom Out exibindo 114 fotos em 6 grupos.	70
Figura 18 – PhotoMesa - Zoom In em um grupo de fotos.	70
Figura 19 – PhotoMesa - Zoom In em uma única foto.....	71
Figura 20 – MFTV - Visão geral da árvore.....	71

Lista de quadros

Quadro 1 – Exemplo de código para criar um nó customizado.....	33
Quadro 2 – Arquivo XML para persistência de dados.	43
Quadro 3 – Configurando nomes das funções na classe grupo.	44
Quadro 4 – Configurando chamadas das funções na classe grupo.....	45
Quadro 5 – Configurando chamadas das funções na classe grupo.....	45
Quadro 6 – Mapeando operações.	46
Quadro 7 – Configurando o construtor PCustomItem.....	47
Quadro 8 – Configurando os dois estados de renderização.....	47
Quadro 9 – Adicionando atributos.	48
Quadro 10 – Configurando os possíveis filtros.	48
Quadro 11 – Configurando os possíveis operadores dos filtros.	49
Quadro 12 – Testando os filtros.	50
Quadro 13 – Configurando nomes das funções na classe item.	50
Quadro 14 – Configurando chamadas das funções na classe item.....	51
Quadro 15 – Configurando chamadas das funções na classe grupo.....	51
Quadro 16 – Arquivo de Configuração (Ações).....	52
Quadro 17 – Arquivo de Configuração (Filtros).....	53
Quadro 15 – Configurando nomes das funções.....	61
Quadro 16 – Configurando chamadas das funções.	61
Quadro 17 – Configurando chamadas das funções.	62
Quadro 18 – Configurando chamadas de manipulação direta.....	62
Quadro 19 – Adicionando atributos à classe customizada.	64
Quadro 20 – Configurando os possíveis filtros da classe customizada.	64
Quadro 21 – Configurando nomes das funções da classe customizada.	64
Quadro 22 – Configurando chamadas das funções da classe customizada.	65
Quadro 23 – Configurando manipulação direta da classe customizada.	65
Quadro 24 – Mapeamento dos métodos de adicionar e remover itens.....	66
Quadro 25 – Mapeamento dos métodos de adicionar e remover grupos.	67
Quadro 26 – Mapeamento da união de grupos.	68