

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Demetrius Arraes Nunes

**HyperDE - um Framework e Ambiente de Desenvolvimento
dirigido por Ontologias para Aplicações HiperMídia**

PUC-Rio - Certificação Digital Nº 0210503/CA

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, março de 2005



Demetrius Arraes Nunes

HyperDE - um Framework e Ambiente de Desenvolvimento dirigido por Ontologias para Aplicações HiperMídia

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.

Daniel Schwabe
Orientador
PUC-Rio

Arndt von Staa
PUC-Rio

Simone Diniz Junqueira Barbosa
PUC-Rio

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

Rio de Janeiro, 30 de março de 2005

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

Demetrius Arraes Nunes

Graduou-se em Engenharia de Computação na PUC-Rio em 2000. Foi um dos fundadores da Interface Tecnologia de Informação, empresa especializada em software para e-Learning, onde é responsável pela área de tecnologia. Seus interesses incluem engenharia de software, linguagens de programação e tecnologias para Web semântica.

Ficha Catalográfica

Nunes, Demetrius Arraes

HyperDE : um framework e ambiente de desenvolvimento dirigido por ontologias para aplicações hipermídia / Demetrius Arraes Nunes ; orientador: Daniel Schwabe. – Rio de Janeiro : PUC, Departamento de Informática, 2005.

199 f. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui referências bibliográficas.

1. Informática – Teses. 2. Web Semântica. 3. Hipermídia. 4. SHDM. 5. Framework. 6. Ontologias. 7. Arquitetura Orientada a Modelos. 8. Linguagem de Programação. 9. Ambiente de Desenvolvimento. 10. OOHDM. I. Schwabe, Daniel. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Este trabalho é dedicado a meus amigos e sócios da Interface, por me darem as condições necessárias para que este projeto se tornasse realidade.

A meu orientador, por me mostrar os caminhos.

À minha musa inspiradora, por me tornar uma pessoa melhor.

À minha mãe, por seu amor permanente e incondicional.

Agradecimentos

Primeiramente gostaria de agradecer ao meu orientador, Prof. Daniel Schwabe, pela constante fonte de conhecimentos e inspirações e por ter me ensinado a arte de olhar o mundo de grandes altitudes.

Um agradecimento para os membros da banca de professores, Prof. Arndt von Staa, Profa. Simone Diniz Junqueira Barbosa e Prof. Marco Antônio Casanova, que tão prontamente atenderam aos meus pedidos de participarem da avaliação desta dissertação.

Gostaria de agradecer bastante a todos os meus grandes amigos e sócios da Interface, Alexandre Cohen, Leon Kleinberger Salim, Luis Eduardo Elerati, Rafael Saavedra, Welton Dannel Souza, que fizeram o hercúleo esforço de “segurar as pontas” na empresa enquanto eu me ausentei nos meses dedicados a dissertação. Baba, Leon, Du, Rafa e Welton: este trabalho também é de vocês!

Para toda minha família, em especial à minha mãe, Josana Terêsa Arraes Nunes, por todo amor, carinho, amizade e compreensão e a meu pai, Claudio Rocha Nunes, por meu primeiro Apple II aos 8 anos de idade e o início de minha paixão pelos bits. À minha cachorra “Clara”, companhia mais freqüente nas muitas madrugadas atravessadas em frente ao computador.

À Fernanda, por sempre me mostrar o lado bom das coisas da vida.

Finalmente, um obrigado a todos os membros da comunidade Ruby e Ruby on Rails, por todo suporte e amigabilidade diante das dúvidas que surgiram ao longo do caminho e que me ajudaram a abrir os olhos para o mundo do software livre. Em especial a David Heinemeier Hansson, criador e incansável evangelista do Rails, e Yukihiro Matsumoto (“Matz”), criador de uma linguagem de programação tão poderosa quanto divertida como o Ruby.

Resumo

Nunes, Demetrius. **HyperDE - um Framework e Ambiente de Desenvolvimento dirigido por Ontologias para Aplicações HiperMídia.** Rio de Janeiro, 2005. 199p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

O HyperDE, apresentado nesta dissertação, é a combinação de um framework no padrão *Model-View-Controller* e um ambiente de desenvolvimento visual para a construção de protótipos de aplicações hiperMídia, modeladas através dos métodos OOHDM ou SHDM. Como framework MVC, o HyperDE fornece componentes reutilizáveis e extensíveis para as camadas de modelo, visão e controle, especificados como ontologias em RDFS. Como ambiente de desenvolvimento visual, o HyperDE fornece, através de sua interface gráfica e ferramentas auxiliares, uma forma interativa e dinâmica de construir e prototipar uma aplicação hiperMídia, com a possibilidade de visualizar imediatamente o resultado de cada passo do processo de desenvolvimento. A arquitetura de desenvolvimento promovida pelo ambiente é orientada a modelos, onde a definição dos modelos navegacionais efetivamente gera a implementação da aplicação. Utilizando um modelo de dados baseado em RDF e RDFS, os modelos navegacionais produzidos no HyperDE podem ser utilizados como ontologias, fazendo-se uso de tecnologias e linguagens da Web Semântica. Além disso, a utilização de uma linguagem de programação dinâmica permite que o HyperDE construa dinamicamente linguagens específicas de domínio para cada aplicação desenvolvida, o que resulta em um modelo de programação mais conciso e natural.

Palavras-chave

Aplicações HiperMídia; Web Semântica; Ambiente de Desenvolvimento; Arquitetura Orientada a Modelos; OOHDM; SHDM; Ruby; Framework; Ontologias; Web Services

Abstract

Nunes, Demetrius. **HyperDE - a Framework and Development Environment driven by Ontologies for Hypermedia Applications**. Rio de Janeiro, 2005. 199p. MSc. Dissertation - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

HyperDE, presented in this dissertation, is a combination of a Model-View-Controller framework and a visual development environment for building prototype applications modeled through the OOHDM and SHDM methods. As a MVC framework, HyperDE provides the reusable and extensible components for the model, view and control layers, specified as ontologies in RDFS. As a visual development environment, HyperDE provides, through its graphical interface and support tools, an interactive and dynamic way for building and prototyping a hypermedia application, making it possible to visualize the results on each step of the development process. The development architecture promoted by the environment is model-driven, in which the definition of the navigational models effectively generate the application implementation. Using data models based on RDF and RDFS, the navigational models produced in HyperDE can be used as ontologies, through semantic web languages and technologies. Also, the use of a dynamic programming language makes it possible for HyperDE to dynamically generate domain-specific languages for each application built, which results in a programming model much more concise and natural.

Keywords

Hypermedia Applications; Semantic Web; Development Environment; Model Driven Architecture; OOHDM; SHDM; Ruby; Framework; Ontologies; Web Services

Sumário

1	Introdução	14
1.1.	Motivação	14
1.2.	Objetivos	16
1.3.	Organização da Dissertação	17
2	Conceitos	20
2.1.	Web Semântica e Ontologias	20
2.2.	RDF, RDF(s) e OWL	21
2.3.	Arquitetura Orientada a Modelos	23
2.4.	Linguagens de Programação Dinâmicas e Interpretadas	24
2.5.	Linguagens Específicas de Domínio e Ontologias	25
2.6.	Padrão de Projeto e Frameworks MVC	26
2.7.	Web Services, Protocolo SOAP e Arquitetura REST	28
2.8.	Arquitetura Orientada a Serviços	30
3	OOHDM e SHDM	32
3.1.	OOHDM	32
3.2.	SHDM	33
3.3.	Fases	33
3.3.1.	Levantamento de Requisitos	34
3.3.2.	Projeto Conceitual	34
3.3.3.	Projeto Navegacional	35
3.3.4.	Projeto de Interface Abstrata	35
3.3.5.	Implementação	36
3.4.	Modelagem Navegacional	37
3.4.1.	Classes Navegacionais	38
3.4.2.	Atributos Navegacionais	38
3.4.3.	Operações	39
3.4.4.	Elos	39

3.4.5. Contextos Navegacionais	40
3.4.6. Classes em Contexto	41
3.4.7. Estruturas de Acesso (Índices)	42
3.4.8. Âncoras	43
3.4.9. Landmarks	44
3.4.10. Nós	44
3.4.11. Navegação Facetada	44
4 Ambiente HyperDE	47
4.1. Framework MVC	48
4.1.1. Camada de Modelo Navegacional	49
4.1.2. Camada de Controle	50
4.1.3. Camada de Visão	51
4.2. Ambiente de Desenvolvimento e Prototipação	52
4.2.1. Interfaces e Ferramentas do Ambiente	53
4.3. Arquitetura de Implementação	54
4.3.1. Linguagem Ruby	54
4.3.2. Framework Rails	59
4.3.3. Base de dados Sesame	60
4.4. Modelagem Navegacional	61
4.4.1. Classes navegacionais, atributos, operações e elos	62
4.4.1.1. Metaclassa NavClass – classes navegacionais	62
4.4.1.2. Metaclassa NavAttribute – atributos de classes navegacionais	64
4.4.1.3. Metaclassa NavOperation – operações de classes navegacionais	69
4.4.1.4. Metaclassa Link – definição de elos	70
4.4.2. Contextos Navegacionais	71
4.4.3. Estruturas de Acesso: Índices	73
4.4.4. <i>Landmarks</i> – pontos de acesso globais	78
4.4.5. Nós – os objetos da aplicação	79
4.5. Controladores	84
4.5.1. CrudControllers – controladores de retaguarda	85
4.5.1.1. ContextController	87
4.5.1.2. IndexController	88

4.5.1.3. LandmarkController	88
4.5.1.4. LinkController	89
4.5.1.5. NavClassController	89
4.5.1.6. NodeController	91
4.5.1.7. ViewController	92
4.5.2. NavigationController – controlador de execução da aplicação	92
4.5.2.1. Ação “index” – ponto de entrada da aplicação	93
4.5.2.2. Ação “show_index” – estruturas de acesso	94
4.5.2.3. Ação “context” – navegação contextual	94
4.6. Modelagem de Interface	97
4.6.1. Visões Genéricas (componentes)	98
4.6.2. Visões de Índice	100
4.6.3. Visões de Classe em Contexto	101
4.6.4. Biblioteca de funções auxiliares para interfaces	102
4.6.4.1. Função “breadcrumb”	104
4.6.4.2. Função “landmarks”	106
4.6.4.3. Funções “context_previous” e “context_next”	107
4.6.4.4. Função “context_params”	108
4.6.4.5. Função “context_index”	109
4.6.4.6. Funções “node_attr” e “node_attributes”	110
4.6.4.7. Funções “node_op” e “node_operations”	113
4.6.4.8. Funções “index” e “indexes”	115
4.6.4.9. Função “component”	117
4.7. Interfaces e Ferramentas do Ambiente	118
4.7.1. Interface Web	119
4.7.1.1. Contextos	119
4.7.1.2. Índices	121
4.7.1.3. Classes Navegacionais	129
4.7.1.4. Elos	136
4.7.1.5. Landmarks	138
4.7.1.6. Visões	140
4.7.1.7. Nós	144
4.7.1.8. Execução da Aplicação	148

4.7.2. Interface de Console	150
4.7.3. Ferramenta para Importação/Exportação de Modelos Navegacionais	152
5 Exemplo: modelo navegacional para conteúdo educacional	154
5.1. Esquema de Classes Navegacionais	156
5.2. Esquema de Contextos Navegacionais	159
5.3. Operações	162
5.4. Visões	165
5.5. Implementação	168
6 Conclusão	170
6.1. Trabalhos Relacionados	170
6.2. Contribuições	172
6.3. Trabalhos Futuros	178
7 Referências	180
Apêndice A – Diferenças entre o metamodelo do HyperDE e o metamodelo SHDM	185
Apêndice B – Sugestão de Workflow de uso do HyperDE	192
Apêndice C – Linguagem Específica de Domínio (DSL – <i>Domain Specific Language</i>) do HyperDE	194
Apêndice D – Informações Técnicas, Ferramentas, Instalação e Requisitos do HyperDE	197

Lista de figuras

Figura 1 - “A onda da Web Semântica”: visão de Berners-Lee sobre a evolução de tecnologias relacionadas a Web Semântica, seus níveis de expressividade e adoção.	15
Figura 2 - Exemplo de declaração RDF na forma de grafo orientado e utilizando XML	22
Figura 3 - Arquitetura dirigida a modelos (MDA), segundo a visão do OMG	23
Figura 4 - Componentes do padrão de projeto MVC	27
Figura 5 - Exemplo de UID como apresentado por [Leite 01]	34
Figura 6 - Exemplo de ADV como apresentado em [Leite 01]	36
Figura 7 - Exemplo de aplicação OOHDM implementada em plataforma ASP.NET	37
Figura 8 - Exemplo de elo definido a partir de um conjunto de relações como apresentado em [Szundy 04].	40
Figura 14 – Relações entre as metaclasses que formam uma estrutura de acesso	74
Figura 15 – Modelo do <i>framework</i> para definição de estruturas de acesso (índices)	75
Figura 17 – Relação das classes “Node”, “NodeLink” e “NodeAttribute” como superclasses das classes navegacionais geradas a partir das metaclasses do framework.	80
Figura 18 – Controladores de retaguarda e suas ações e as classes que eles manipulam	87
Figura 19 – Modelo de visões customizadas do HyperDE	97
Figura 20 – Resultado de um template utilizando as funções auxiliares de interface	104
Figura 22 - Resultado de uso da função “index”	117
Figura 23 - Tela de listagem de contextos navegacionais	120
Figura 24 - Tela para edição de um contexto navegacional	121
Figura 25 - Tela de listagem de índices da aplicação	122
Figura 26 - Tela de edição de um índice baseado em consulta	123
Figura 27 - Tela de edição para índice derivado de contexto	124
Figura 28 - Tela de edição para um atributo de índice do tipo âncora para índice	125
Figura 29 - Tela para edição de um índice de tipo âncora para contexto	126
Figura 30 - Tela para edição de atributo de índice (âncora para contexto) em índice baseado em consulta	127
Figura 31 - Tela para edição de um atributo de índice de tipo simples	128
Figura 32 - Tela de listagem das classes navegacionais	129
Figura 33 - Tela de edição dos dados básicos de uma classe navegacional	130
Figura 34 - Tela de listagem dos atributos, elos e operações de uma classe navegacional	130

Figura 35 - Tela para edição de um atributo de tipo simples	131
Figura 36 - Tela para edição de um atributo de classe navegacional do tipo índice	132
Figura 37 - Tela de edição de atributo de classe naveg. de tipo âncora para contexto	133
Figura 38 - Tela de edição de atributo de classe naveg. do tipo âncora para índice	134
Figura 39 - Tela de edição de um atributo de classe navegacional de tipo computado	135
Figura 40 - Tela de edição de operação de classe navegacional	136
Figura 41 - Tela de listagem de elos da aplicação	137
Figura 42 - Tela de edição de elo	137
Figura 43 - Tela de listagem de landmarks	138
Figura 44 - Tela de edição de landmark do tipo âncora para índice	139
Figura 45 - Tela de edição de um landmark do tipo âncora para contexto	140
Figura 46 - Tela de listagem das visões customizadas da aplicação	141
Figura 47 - Tela para edição de uma visão customizada genérica	142
Figura 48 - Tela de edição de visão customizada para classe em contexto	143
Figura 49 - Tela de edição de uma visão customizada para índice	144
Figura 50 - Tela de listagem de nós de uma classe navegacional escolhida	145
Figura 52 - Tela de edição de nó apresentando as listas de valores dos elos	147
Figura 53 - Tela de adição de valores de elos a um nó	148
Figura 54 - Tela de exibição de um índice da aplicação	149
Figura 55 - Tela de exibição de um nó em contexto navegacional da aplicação	150
Figura 56 - Tela de uma sessão de uso da interface de console interativo	151
Figura 58 - Esquema de contextos navegacionais da primeira aplicação exemplo	155
Figura 59 - Esquema de classes navegacionais para o modelo educacional da aplicação de exemplo	157
Figura 60 - Esquema de contextos, índices e landmarks para o modelo educacional da aplicação de exemplo	160
Figura 61 - Esquema de classes navegacionais exibindo as operações de cada classe	162
Figura 62 - Exemplos de arquiteturas físicas distribuídas possíveis para uso do ambiente	176