

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Rafael Ferreira Rodrigues

**Ambiente Declarativo para Sistemas que
Implementem o GEM**

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: Luiz Fernando Gomes Soares

Rio de Janeiro, agosto de 2007



Rafael Ferreira Rodrigues

Ambiente Declarativo para Sistemas que Implementem o GEM

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

Prof. Luiz Fernando Gomes Soares

Orientador
Departamento de Informática - PUC-Rio

Prof. Renato Cerqueira

Departamento de Informática - PUC-Rio

Prof. Rogério Ferreira Rodrigues

Departamento de Informática - PUC-Rio

Prof. José Eugenio Leal

Coordenador Setorial do Centro
Técnico Científico – PUC-Rio

Rio de Janeiro, 20 de agosto de 2007

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

Rafael Ferreira Rodrigues

Graduado em Engenharia de Computação pelo Instituto Militar de Engenharia (IME) em 2004. Atualmente, integra o grupo de pesquisadores do Laboratório TeleMídia da PUC-Rio, desenvolvendo pesquisas do GINGA-NCL.

Ficha Catalográfica

Rodrigues, Rafael Ferreira

Ambiente Declarativo para Sistemas que Implementem o GEM / Rafael Ferreira Rodrigues ; orientador: Luiz Fernando Gomes Soares. – Rio de Janeiro : PUC-Rio, Departamento de Informática, 2008.

101 f. : il. ; 30 cm

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

Inclui referências bibliográficas

1. Informática – Teses. 2. Gem. 3. NCL. 4. Java 5. Ginga 6. TV Interativa. I. Soares, Luiz Fernando Gomes. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Este trabalho é dedicado: a toda minha família
e amigos, em especial meus pais, Jorge
e Denize, minha irmã Roberta
e minha namorada Juliana.

Agradecimentos

Gostaria de agradecer primeiramente ao meu orientador Prof. Luiz Fernando pela confiança e pela forma de trabalhar tão inspiradora.

A todos os integrantes do Laboratório TeleMídia que contribuíram direta ou indiretamente para a realização deste trabalho. Em especial agradeço ao Márcio, ao Carlão pela ajuda com as revisões. Preciso também agradecer ao Rogério cuja pesquisa contribuiu diretamente para a elaboração desta dissertação.

Aos meus amigos, e irmãos, Eduardo, Thiaguinho, Simão, Vinicius.

Agradeço a toda a minha família pelo carinho e apoio incondicional. Em especial a minha mãe Denize, minha irmã Roberta e a meu pai Jorge.

Agradeço à minha namorada Jujuba por ter sempre sido compreensiva.

Aos membros da banca pelos preciosos comentários e revisões.

Agradeço a todos os professores e funcionários do Departamento de Informativa da PUC-Rio.

Por fim, gostaria de agradecer ao CNPq, à CAPES, à FINEP e à PUC-Rio pelo apoio financeiro fornecido ao longo deste mestrado.

Resumo

Rodrigues, Rafael Ferreira. **Ambiente Declarativo para sistemas que implementem o GEM.** Rio de Janeiro, 2004. 101p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A existência de vários ambientes procedurais definidos para *middlewares* de Sistemas de TV Digital terrestre levou à criação de um framework conhecido como *Globally Executable MHP* (GEM). Esse padrão visa a harmonização de tais ambientes permitindo a execução global das aplicações. Nesse contexto, este trabalho descreve a construção de um ambiente de apresentação declarativo utilizando a API fornecida pelo GEM de forma a permitir a execução global do conteúdo declarativo produzido para o Sistema Brasileiro de TV Digital

Palavras-chave

GEM, NCL, JAVA, Ginga, TV Digital Interativa

Abstract

Rodrigues, Rafael Ferreira. **Declarative Environment for Systems Implementing GEM** Rio de Janeiro, 2004. 101p. Master Thesis - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The several procedural environment proposals for terrestrial Digital TV Systems led to the middleware framework recommendation known as Globally Executable MHP (GEM). This standard aims at the harmonization of such environments allowing the global execution of procedural applications but neglecting the declarative ones. In this context, this work describes the integration of the Ginga declarative environment using the API supplied by GEM and allowing the global execution of declarative contents produced for the Brazilian System of Digital TV (Sistema Brasileiro de TV Digital).

Key words

GEM, NCL, JAVA, Ginga, Interactive TV, Digital TV

Sumário

1	Introdução	13
1.1.	Motivação.....	14
1.1.1.	Ambiente Declarativo x Ambiente Procedural	17
1.2.	Objetivos	18
1.3.	Organização da Dissertação.....	19
2	Conceitos Preliminares	21
2.1.	<i>Middlewares</i> de TV Digital	21
2.1.1.	O <i>Middleware</i> Europeu	23
2.1.1.1.	Tipos de aplicações MHP	24
2.1.1.2.	O uso de <i>Plug-ins</i>	26
2.1.1.3.	O uso do armazenamento de aplicações	28
2.1.2.	O <i>Middleware</i> Americano.....	29
2.1.3.	O <i>Middleware</i> Japonês	31
2.1.4.	O <i>Middleware</i> Brasileiro.....	32
2.2.	Definição do padrão GEM.....	35
2.2.1.	Políticas de segurança para execução de aplicativos no GEM	36
2.2.2.	Sinalização de Aplicações no GEM	37
2.3.	O Blue-Ray Disc	39
2.4.	Ambientes de Execução JAVA para TV	40
2.4.1.	O Modelo de programação <i>XLET</i>	43
3	Trabalhos Relacionados	47
3.1.	O MHP Processando Documentos XHTML	48
3.2.	Ambiente declarativo para TV Digital – UTH	49
3.3.	Ambiente declarativo para TV Digital – ICECREAM.....	50
4	O Fomatador NCL.....	52
4.1.	O Formatador NCL Java.....	52
4.2.	O GEM processando um documento NCL	55
4.3.	A Arquitetura de Implantação	57

4.3.1. Estrutura do Sistema Baseada em Componentes.....	57
4.4. Adaptações e Otimizações no Formatador NCL	58
4.4.1. A Pré-conversão de Documentos NCL e o Módulo Gerenciador de Documentos	58
4.4.2. Adaptações do Módulo do Núcleo do Formatador	60
4.4.3. Adaptações do Módulo Gerenciador de Adaptadores para Exibidores e os Adaptadores Criados.....	63
4.4.4. Adaptações do Módulo Gerenciador de Leiaute.....	64
5 O Fomatador NCL <i>Xlet</i>	66
5.1. Os Componentes do Formatador NCL <i>Xlet</i>	66
5.1.1. Gerenciador de Leiaute	67
5.1.2. Gerenciador de Documentos	68
5.1.3. Núcleo do Formatador	69
5.1.4. Gerenciador de Adaptador para Exibidores.....	70
5.1.5. Adaptadores para Exibidores.....	70
5.2. Implementação dos Componentes para Sistemas GEM	71
5.2.1. O Carregamento Dinâmico	72
5.2.2. O Módulo de Implantação.....	73
5.2.3. O Processo de Implantação do Formatador	76
5.2.4. Persistência do Formatador NCL <i>Xlet</i> em Receptores MHP	78
5.3. Testes.....	79
6 Conclusões	83
6.1.Trabalhos Futuros.....	84
7 Referências Bibliográficas	86
Apêndice A Assinatura Digital de um <i>Xlet</i>	91
Apêndice B Sinalização de Aplicações.....	94
Apêndice C A Linguagem NCL.....	98

Lista de Figuras

Figura 1 – O GEM e os demais padrões.	16
Figura 2 – Arquitetura de um sistema de TV Digital.	22
Figura 3 - Perfis MHP.	24
Figura 4 - Arquitetura básica do MHP.....	25
Figura 5 – Opções de implementação de plug-ins.	27
Figura 6 – A interface Plugin.....	28
Figura 7 – Arquitetura do <i>middleware</i> Ginga.	34
Figura 8 – O processo de identificação da AIT.....	38
Figura 9 – Arquiteturas de plataformas JAVA.....	41
Figura 10 – Exemplo de ambiente de execução Java.	42
Figura 11 – Protocolo entre as entidades presentes no modelo e programação <i>Xlet</i>	44
Figura 12 - Interface do <i>Xlet</i>	44
Figura 13 - Máquina de estados do ciclo de vida de um <i>Xlet</i>	45
Figura 14 – Interface do <i>XletContext</i>	45
Figura 15 – Arquitetura da proposta do transcodificador.....	48
Figura 16 – Arquitetura do Formatador NCL.	53
Figura 17 – Pré-processamento do documento NCL.	56
Figura 18 – Processamento do documento NCL no receptor.....	56
Figura 19 – Modelo de uma apresentação NCL orientada a sincronização por contexto.	60
Figura 20 – Digrama de classe da interface <i>IFormatter</i>	62
Figura 21 – Diagrama da interface <i>NCLEditingCommandListener</i>	63
Figura 22 – Diagrama de classes do modelo de leiaute do Formatador NCL.....	65
Figura 23 – Duas implementações do componente Gerenciador de Leiaute	67
Figura 24 – Componente Gerenciador de Documentos.	68
Figura 25 – Duas implementações do componente do núcleo do formatador.....	69

Figura 26 – Duas implementações do componente do núcleo do Formatador	69
Figura 27 – Duas implementações do componente Gerenciador de Exibidores	70
Figura 28 – Duas implementações de adaptadores para exibidores.....	70
Figura 29 – Classe GingaXlet.	73
Figura 30 – Classe IGingaXletConfiguration.....	74
Figura 31 – Interface IGingaXletDeployer.....	74
Figura 32 – A classe StreamEventHandler e suas associações.	75
Figura 33 – Diagrama de seqüência do processo de preparação do Formatador <i>Xlet</i>	77
Figura 34 – Diagrama de seqüência do processo de inicialização do Formatador <i>Xlet</i>	78
Figura 35 – Digrama de componentes do Sistema testado.....	80
Figura 36 – Arquitetura de execução distribuída do Formatador NCL.	85
Figura 37 – Exemplo de arquivo PRF.....	92
Figura 38 – Estrutura de diretórios de um <i>Xlet</i>	93
Figura 39 – Exemplo de documento NCL 2.0.....	101

Lista de Tabelas

Tabela 1 – Componentes do formatador GingaXlet	80
Tabela 2 – Resultados dos testes de Serialização versus conversão.....	81
Tabela 3 – Plataformas que implementam o CDC.	84
Tabela 4 – Exemplo resumido de uma tabela AIT.....	95
Tabela 5 – Parâmetros adicionais da AIT para sinalização do Formatador <i>Xlet</i>	96