

Marcio Ferreira Moreno

**Conciliando Flexibilidade e Eficiência no
Desenvolvimento do Ambiente
Declarativo Ginga-NCL.**

TESE DE DOUTORADO

DEPARTAMENTO DE INFORMÁTICA

Programa de Pós-Graduação em Informática

Rio de Janeiro
Agosto de 2010



Marcio Ferreira Moreno

**Conciliando Flexibilidade e Eficiência no Desenvolvimento
do Ambiente Declarativo Ginga-NCL**

Tese de Doutorado

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

Orientador: Luiz Fernando Gomes Soares

Rio de Janeiro, 16 de agosto de 2010



Marcio Ferreira Moreno

**Conciliando Flexibilidade e Eficiência no Desenvolvimento
do Ambiente Declarativo Ginga-NCL**

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

Prof. Luiz Fernando Gomes Soares
Orientador
PUC-Rio

Prof. Renato Fontoura de Gusmão Cerqueira
PUC-Rio

Prof.^a Noemi de La Rocque Rodriguez
PUC-Rio

Prof.^a Thais Vasconcelos Batista
UFRN

Prof. Pablo Cesar
CWI

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

Rio de Janeiro, 16 de agosto de 2010

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

Marcio Ferreira Moreno

Graduado em Ciência da Computação pela UFJF, onde desenvolveu projetos na área de redes programáveis e aprendizado colaborativo. Obteve título de Mestre em Informática em 2006 pela PUC-Rio, concentrando seus estudos em middlewares para sistemas de TV Digital. Desde 2004 integra o grupo de pesquisadores do Lab. TeleMídia da PUC-Rio, onde participou de diversos projetos em redes de computadores e aplicações multimídia distribuídas. Atualmente, é pesquisador associado do Laboratório TeleMídia e contribui nas especificações do Sistema Brasileiro de TV Digital, bem como na implementação de referência do middleware Ginga-NCL.

Ficha Catalográfica

Moreno, Marcio Ferreira

Conciliando Flexibilidade e Eficiência no Desenvolvimento do Ambiente Declarativo do Middleware Ginga: Implementação de Referência Ginga-NCL / Marcio Ferreira Moreno ; orientador: Luiz Fernando Gomes Soares. – 2010.

155 f. : il. ; 30 cm

Tese (Doutorado em Informática)–Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2010.

Inclui bibliografia.

1. Informática - Teses. 2. Ginga-NCL. 3. NCL. 4. Middleware. 5. TV Digital. 6. Sincronismo de Mídias. 7. DSM-CC. 8. Desenvolvimento Orientado a Componentes. 9. Múltiplos Dispositivos. 10. Resiliência. 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 Suzana, minha amada e perfeita
esposa, essência da alegria e
felicidade.*

*Aos meus pais, Jasmina e
Wanderley, os verdadeiros Doutores
da minha vida.*

*E aos meus irmãos, Patrícia e
Marcelo, sempre presentes em
momentos precisos.*

Agradecimentos

Agradeço a Deus, acima de tudo.

Agradeço também ao meu orientador, Professor Luiz Fernando Gomes Soares, pelo aprendizado, pelas brilhantes idéias e discussões, pelos esforços na orientação, pela pesquisa e por sua amizade.

Em especial, por compreenderem minha ausência durante esses quatro anos de trabalho, agradeço a minha esposa, Suzana, aos meus pais, Jasmina e Wanderley, meus sogros Antônio Borba e Ângela, meus irmãos, Patrícia e Marcelo, meus cunhados Anginha, Rodrigo, Toni e Lorenza, minhas afilhadas Maria Clara e Luisa, a minha sobrinha Sofia e aos meus padres Rogério e Paulinha.

Agradeço também aos amigos do TeleMídia: Alexandra, Álvaro, Bruno, Carlos, Carlos Eduardo, Eduardo, Felipe Bittencourt, Felipe Nagato, Felipe Nogueira, Francisco, Gabriel, Guilherme, Henrique, Heron, José Geraldo, Jordan, Laiola, Luciana Lima, Luciana Rosa, Marcelo, Paulinha, Rafael Rodrigues, Rafael Savignon, Tadeu, Ricardo, Rogério, Rogerinho, Romualdo, Sérgio, Silvinha, Suzana, Vinicius e Vitor. Igualmente, agradeço aos professores e funcionários da PUC-Rio, pela qualidade do ensino e qualidade que trazem a esta universidade.

Agradeço ainda aos membros da banca, pelos valiosos comentários, sugestões e revisões.

Finalmente, agradeço ao CNPq, a PUC-Rio e ao Laboratório TeleMídia pela estrutura, infra-estrutura e todo tipo de apoio, inclusive financeiro, fundamentais à realização deste trabalho.

Resumo

Moreno, Marcio Ferreira; Soares, Luiz F. G. **Conciliando Flexibilidade e Eficiência no Desenvolvimento do Ambiente Declarativo Ginga-NCL.** Rio de Janeiro, 2010. 155p. Tese de Doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Um sistema de TV Digital (TVD) é definido por um conjunto de especificações que determinam as tecnologias envolvidas na codificação, transmissão, recepção e apresentação de conteúdos, incluindo a especificação da aplicação (programa não-linear), seus vários objetos de mídia e metadados relacionados. Nesse cenário, o suporte a aplicações é realizado por uma camada intermediária de software, ou middleware, posicionada, no ambiente de recepção, entre o código das aplicações e a infra-estrutura de execução (plataforma de hardware e sistema operacional). O projeto e implementação de um middleware para receptores de sistemas de TVD trazem uma série de desafios. Entre eles estão: a eficiência na utilização de recursos, usualmente escassos no dispositivo receptor; o suporte à evolução dinâmica das funcionalidades do middleware; o suporte à recuperação de falhas em tempo de execução; os mecanismos para gerência de localização de recursos, permitindo o uso da mesma sintaxe de autoria em ambientes distintos de recepção; o suporte a edição ao vivo de programas não-lineares, ou seja, das aplicações; a definição de uma infra-estrutura de transporte assíncrono de aplicações interativas e comandos de controle e; o controle do ciclo de vida das aplicações interativas, permitindo que aplicações possam ser iniciadas, pausadas e retomadas em qualquer ponto de sua duração, sem perder o histórico de sua evolução. As soluções da maioria dessas questões presentes nos sistemas existentes apresentam algumas limitações importantes, e em alguns casos nem mesmo existem, sendo o problema apenas contornado. O presente trabalho propõe soluções alternativas para as questões levantadas, e as incorpora na especificação do middleware declarativo Ginga-NCL e em sua implementação de referência. O middleware Ginga-NCL e sua linguagem declarativa NCL foram adotados pelo SBTVD-T em 2007. No início de 2009, Ginga-NCL e NCL se tornaram parte dos padrões ISDB-T_B e parte da recomendação ITU-R BT 1699. Ainda no início de

2009, Ginga-NCL e NCL tornaram-se a Recomendação ITU-T H.761 para serviços IPTV.

Palavras-chave

Ginga-NCL; NCL; Middleware; TV Digital; Sincronismo de Mídias; DSM-CC; Desenvolvimento Orientado a Componentes; Resiliência; NPT.

Abstract

Moreno, Marcio Ferreira; Soares, Luiz F. G. (Advisor). **Bringing Together Flexibility and Efficiency in the Development of the Ginga-NCL declarative environment.** Rio de Janeiro, 2010. 155p. DSc. Thesis – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Digital TV (DTV) systems are defined by a set of specifications that establish the technologies involved in the content encoding, transmission, reception and presentation, including the specification of applications (non-linear programs), their various related media objects and metadata. In this scenario, support to applications is accomplished through an intermediary software layer, or middleware, positioned, in the receiving environment, between the application code and the execution infrastructure (hardware platform and operating system). The middleware design and implementation bring a number of challenging issues. Among them are: efficient resource management, since resources are usually scarce in DTV receiver devices; support to dynamic evolution of the middleware functionalities; support to fault recovery at runtime; the mechanisms for resource location management, allowing using the same syntax used in the authoring environment the different receiver environments; support to live editing of non-linear programs (i.e. applications); the infrastructure definition for the asynchronous transport of interactive applications and control commands; and the life cycle control of interactive applications, allowing applications to be started, paused and resumed at any point in their life duration, without losing their evolution history. Most of these issues are addressed in the existing systems, however with important limitations; some of them are not even addressed, being only treated with workaround tricks. This work proposes alternative solutions to the mentioned issues and incorporates these solutions in the Ginga-NCL declarative middleware specification and in its reference implementation. Ginga-NCL and its declarative NCL language were adopted by SBTVD-T in 2007. In early 2009, Ginga-NCL and NCL have become part of the ISDB-T_B standard and part of the ITU-R BT 1699 Recommendation. Even in early 2009, Ginga-NCL and NCL have become the ITU-T H.761 Recommendation for IPTV services.

Keywords

Ginga-NCL; NCL; Middleware; Digital TV; Media Synchronism; DSM-CC; Component-based Development; Resilience; NPT.

Sumário

1 Introdução	17
1.1. Objetivos e Contribuições	19
1.2. Motivação	22
1.3. Organização da Tese	23
2 Trabalhos Relacionados	25
2.1. Gerência de Recursos e Evolução Dinâmica	25
2.2. Plano de Recuperação	26
2.3. Comandos de Edição	27
2.4. Identificação de Recursos	29
2.4.1. Identificação de Recursos do Carrossel de Objetos	30
2.4.2. Identificação de Fluxos Elementares	35
2.4.3. Identificação de Bases Temporais	37
2.5. Ciclo de Vida das Aplicações NCL	41
2.6. Sistemas de Transporte	43
3 Gerência de Recursos e Evolução Dinâmica	46
3.1. Arquitetura	46
3.1.1. Componentização do Ginga-CC	48
3.1.2. Componentização da Máquina de Apresentação	50
3.2. Implementação Ginga e Modelo de Componentes	52
3.3. Avaliação por Medições	63
4 Plano de Recuperação	67
4.1. Arquitetura	67
4.2. Implementação	71
5 Comandos de Edição NCL	77
5.1. Bases Privadas e Comandos de Edição	77

5.2. Implementação	81
6 Identificando Recursos nas Aplicações NCL	84
6.1. Carrossel de Objetos	84
6.2. Fluxos Elementares	86
6.3. Bases Temporais	88
7 Ciclo de Vida das Aplicações NCL	94
8 Sistemas de Transporte	101
8.1. Comandos de Edição e Aplicações no Ginga-NCL	101
8.1.1. Descritor de Evento	101
8.1.2. Mapa-de-Eventos	104
8.1.3. Arquivos de Dados e Metadados	105
8.2. Sistema de Transporte	108
8.2.1. Transporte de Descritores de Evento	108
8.2.2. Transporte de Metadados	109
8.2.3. Transporte Conjunto das Estruturas	111
9 Revisitando os Trabalhos Relacionados	118
9.1. Gerência de Recursos e Evolução Dinâmica	118
9.2. Plano de Recuperação	119
9.3. Comandos de Edição	120
9.4. Identificação de Recursos	122
9.4.1. Identificação de Recursos do Carrossel de Objetos	122
9.4.2. Identificação de Fluxos Elementares	123
9.4.3. Identificação de Bases Temporais	124
9.5. Ciclo de Vida das Aplicações NCL	125
9.6. Sistemas de Transporte	125
10 Conclusões	127
11 Referências bibliográficas	133

Anexo I – Conceitos MPEG-2	138
Fluxo de Transporte MPEG-2	138
Eventos de Sincronismo DSM-CC	139
Carrossel de Objetos DSM-CC	140
Anexo II – Comandos de Edição para Entidades NCL	143
Anexo III – Avaliação da Arquitetura por Medições	146
Anexo IV – Avaliação do Plano de Recuperação por Medições	154

Lista de figuras

Figura 1 – Arquitetura Ginga.	47
Figura 2 – Diagrama de Componentes do Gerente de Componentes Ginga-NCL.	56
Figura 3 – Quantidade de Código Alocado na Memória.	65
Figura 4 – Uso CPU.	66
Figura 5 – Arquitetura Plano de Recuperação Ginga-NCL.	69
Figura 6 – Diagrama de Componentes para Recuperação Proativa.	72
Figura 7 – Diagrama de Componentes para Recuperação Reativa.	74
Figura 8 – Diagrama de Componentes para Tratamento dos Comandos de Edição.	83
Figura 9 – Exemplo de Conteúdos NPT em uma Grade de Programação.	90
Figura 10 – Sistema de Arquivos da Aplicação <i>primeiroJoao</i> .	104
Figura 11 – Carrossel de dados DSM-CC.	112
Figura 12 – Carrossel de objetos para Figura 10.	114
Figura 13 – O <i>primeiroJoao</i> em Seções MPEG-2.	117
Figura 14 – Exemplo de Carrossel de Objetos gerado a partir de um sistema de arquivos.	141
Figura 15 – Quantidade de Código Alocado na Memória para a Apresentação da Aplicação Representativa do Conjunto 1.	150
Figura 16 – Uso CPU para a Apresentação da Aplicação Representativa do Conjunto 1.	151
Figura 17 – Quantidade de Código Alocado na Memória para a Apresentação da Aplicação Representativa do Conjunto 2.	151
Figura 18 – Uso CPU para a Apresentação da Aplicação Representativa do Conjunto 2.	152
Figura 19 – Quantidade de Código Alocado na Memória para	

a Apresentação da Aplicação Representativa do Conjunto 3.	152
Figura 20 – Uso CPU para a Apresentação da Aplicação Representativa do Conjunto 3.	153

Lista de quadros

Quadro 1 – DTD XML Descrição da Arquitetura Ginga-NCL	58
Quadro 2 – Aplicação NCL que Demanda Evolução Dinâmica.	60
Quadro 3 – Descrição Remota de Componentes.	61
Quadro 4 – Trecho de Aplicação NCL com Identificadores de Fluxos Elementares.	88
Quadro 5 – Aplicação NCL com Associação a Bases Temporais.	92
Quadro 6 – Schema XML NCLMetadataFile.	107
Quadro 7 – Arquivo de metadados do exemplo primeiroJoao.	108

Lista de tabelas

Tabela 1 – TV Locators especificados pelo padrão DVB (ETSI, 2010)	36
Tabela 2 – Comandos para controle do ciclo de vida de aplicações DVB-HTML (ETSI, 2010)	42
Tabela 3 – Descritor de evento para comandos de edição NCL.	75
Tabela 4 – Comandos de edição para operações sobre bases privadas.	78
Tabela 5 – Comandos de edição para manipulação de documentos NCL.	81
Tabela 6 – Comandos de edição addNode.	81
Tabela 7 – URLs para Identificação de Fluxos Elementares	87
Tabela 8 – Controle do Ciclo de Vida das Aplicações NCL por meio de tabelas AIT.	97
Tabela 9 – Exemplo de Transmissão de Estruturas para Controle do Ciclo de Vida das Aplicações.	99
Tabela 10 – Descritor de evento para comandos de edição NCL.	102
Tabela 11 – Descritor de evento para o comando addDocument.	104
Tabela 12 – Lista de identificadores de eventos definidos pela estrutura de mapa.	105
Tabela 13 – Representação Comparativa entre Trabalhos para Gerência de Recursos e Evolução Dinâmica	119
Tabela 14 – Representação Comparativa entre Trabalhos de Recuperação de Falhas	120
Tabela 15 – Comandos de edição de Entidades NCL.	145
Tabela 16 – Hipótese dos Tipos de Objetos de Mídias Especificados em Aplicações que possuem Relação Semântica com programas de TV.	147
Tabela 17 – Tempo de Recuperação de Objetos para Injeção de Falha SIGKILL.	155