

**Renato Figueiró Maia**

**Um Framework para  
Adaptação Dinâmica de  
Sistemas Baseados em  
Componentes Distribuídos**

**DISSERTAÇÃO DE MESTRADO**

**DEPARTAMENTO DE INFORMÁTICA**

**Programa de Pós-graduação em**

**Informática**

Rio de Janeiro

Março de 2004



**Renato Figueiró Maia**

**Um Framework para Adaptação Dinâmica  
de Sistemas Baseados em Componentes  
Distribuídos**

**Dissertação de Mestrado**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática da PUC-Rio

Orientador: Prof. Renato Fontoura de Gusmão Cerqueira

Rio de Janeiro  
Março de 2004



**Renato Figueiró Maia**

**Um Framework para Adaptação Dinâmica  
de Sistemas Baseados em Componentes  
Distribuídos**

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

**Prof. Renato Fontoura de Gusmão Cerqueira**

Orientador

Departamento de Informática — PUC-Rio

**Prof. Noemi de La Rocque Rodriguez**

Departamento de Informática — PUC-Rio

**Prof. Roberto Ierusalimschy**

Departamento de Informática — PUC-Rio

**Prof. Alexandre Sztajnberg**

Departamento de Informática e Ciências da Computação  
— UERJ

**Prof. José Eugenio Leal**

Coordenador Setorial do Centro Técnico Científico —  
PUC-Rio

Rio de Janeiro, 30 de Março de 2004

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

**Renato Figueiró Maia**

Graduou-se em Bacharelado em Ciência da Computação na UFPA (Universidade Federal do Pará) em 2001.

Ficha Catalográfica

Maia, Renato

Um *Framework* para Adaptação Dinâmica de Sistemas Baseados em Componentes Distribuídos/ Renato Figueiró Maia; orientador: Renato Fontoura de Gusmão Cerqueira. — Rio de Janeiro : PUC-Rio, Departamento de Informática, 2004.

v., 104 f: il. ; 29,7 cm

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

Inclui referências bibliográficas.

1. Informática – Dissertações. 2. Sistemas Distribuídos. 3. Componentes de Software. 4. Adaptação Dinâmica I. Cerqueira, Renato. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

À minha mãe, minha tia Graça e meus irmãos.

## Agradecimentos

À minha mãe Maria da Conceição Figueiró Maia, e minha tia Maria das Graças Ferreira Figueiró, quem considero como minha segunda mãe pelo apoio contínuo durante toda minha vida e aos meus irmãos Jakson Figueiró Maia, Rafael Figueiró Maia e Darcy Borges Maia Júnior, que sempre me serviram de fonte de inspiração.

Ao meu orientador Renato Fontoura de Gusmão Cerqueira pelo apoio e incentivo dados durante todo esse trabalho.

À CAPES e à PUC-Rio pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

## Resumo

Maia, Renato; Cerqueira, Renato. **Um *Framework* para Adaptação Dinâmica de Sistemas Baseados em Componentes Distribuídos**. Rio de Janeiro, 2004. 104p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A adaptação dinâmica de aplicações distribuídas tem se tornado um recurso cada vez mais essencial na construção de sistemas de computação. Isso é justificado especialmente pelo avanço da tecnologia, que tem permitido a automação de tarefas complexas em domínios de aplicação cada vez menos tolerantes à suspensão de serviços. Nesta dissertação é proposto o *LuaOrb Adaptation Framework*, que utiliza os recursos da linguagem Lua na adaptação dinâmica de sistemas baseados em componentes do Modelo de Componentes de CORBA (CCM - *CORBA Component Model*). Através desse *framework* é possível utilizar as abstrações de *papéis* e *protocolos* para realizar adaptações criando novas interações entre os componentes do sistema, assim como reconfigurar dinamicamente os componentes CCM. Devido a limitações do modelo CCM, é proposta uma adaptação desse modelo para a linguagem Lua, de onde surge o conceito de *contêiner dinâmico*, que permite a construção de componentes dinamicamente adaptáveis através de alterações na estrutura e implementação desses componentes. O contêiner dinâmico permite que essas alterações sejam feitas em níveis diferentes, ou seja, no nível de uma única instância ou implementação de componente, assim como em todas as instâncias de um determinado componente.

## Palavras-chave

Adaptação dinâmica; componentes de software; sistemas distribuídos; reflexão computacional; configuração dinâmica; papéis e protocolos; CCM; CORBA

## Abstract

Maia, Renato; Cerqueira, Renato. **A Framework for Dynamic Adaptation of Distributed Component-Based Systems**. Rio de Janeiro, 2004. 104p. MSc. Dissertation — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Dynamic adaptation of distributed applications has become an essential feature in development of computer systems, mainly justified by nowadays technology, which enables complex tasks to be performed by computers in application domains less suited for service interruption. This dissertation proposes the LuaOrb Adaptation Framework, which uses features of the programming language Lua to dynamically adapt systems based on the CORBA Component Model (CCM). This framework uses abstractions like *roles* and *protocols* to adapt systems by creating new interactions between systems components, as well as provides features for dynamic reconfigurations of CCM components. Due to limitations of CCM, an adaptation of this model to Lua concepts is proposed, resulting in the definition of *dynamic containers*, which enable development of dynamically adaptable components by changes on component structure and implementation. Dynamic containers allows adaptations to be done on different levels, namely on the level of a single component instance or implementation, as well as on all instances of a given component.

## Keywords

Dynamic adaptability; software components; distributed systems; computational reflection; dynamic configuration; protocol and roles; CCM; CORBA



## Sumário

|       |   |    |
|-------|---|----|
| 1     | Introdução  | 11 |
| 1.1   | Grau de Flexibilização                                      | 11 |
| 1.2   | Abstrações de Programação                                   | 12 |
| 1.3   | Adaptação Dinâmica  | 13 |
| 1.4   | Um <i>Framework</i> para Evolução Dinâmica de Aplicações    | 14 |
| 1.5   | Estrutura do Texto  | 15 |
| 2     | Componentes de Software                                     | 16 |
| 2.1   | Limitações da Arquitetura CORBA                             | 18 |
| 2.2   | O Modelo de Componentes de CORBA                            | 19 |
| 2.2.1 | Declaração de Componentes                                   | 19 |
| 2.2.2 | Eventos   | 21 |
| 2.2.3 | Portas  | 22 |
| 2.2.4 | <i>Homes</i> de Componentes                                 | 28 |
| 2.2.5 | Contêiners  | 29 |
| 2.2.6 | Implantação de Componentes                                  | 34 |
| 3     | Adaptação Dinâmica  | 35 |
| 3.1   | Reflexão Computacional                                      | 35 |
| 3.2   | Adaptação em Ponto Grande                                   | 36 |
| 3.2.1 | Padrões de Projeto  | 37 |
| 3.2.2 | Sistemas Baseados em Componentes                            | 37 |
| 3.3   | Adaptação em Ponto Pequeno                                  | 39 |
| 3.3.1 | Nível da Linguagem de Programação                           | 39 |
| 3.3.2 | Nível Arquitetural  | 40 |
| 3.3.3 | Nível dos Componentes                                       | 40 |
| 4     | LuaCCM: Um Modelo de Componentes Dinâmico                   | 42 |
| 4.1   | Base de Implementação                                       | 43 |
| 4.1.1 | A Linguagem Lua   | 43 |
| 4.1.2 | LuaOrb: O Mapeamento de Lua para Arquitetura CORBA          | 44 |
| 4.1.3 | LOOP: Um Modelo Dinâmico de Programação Orientada a Objetos | 44 |
| 4.2   | Contêiners Lua  | 47 |
| 4.2.1 | Estrutura dos Contêiners Lua                                | 48 |
| 4.2.2 | Interfaces do Contêiner                                     | 50 |

|       |                                      |           |
|-------|--------------------------------------|-----------|
| 4.2.3 | Gerenciamento de Portas              | 51        |
| 4.2.4 | Obtenção de Executores               | 52        |
| 4.2.5 | Ativação de Executores               | 52        |
| 4.3   | Implantação de Componentes           | 53        |
| 4.4   | Componentes Dinâmicos                | 57        |
| 4.4.1 | Formas de Adaptação                  | 59        |
| 4.4.2 | Níveis de Adaptação                  | 61        |
| 4.4.3 | Interfaces de Adaptação              | 63        |
| 5     | Ferramentas para Adaptação Dinâmica  | <b>66</b> |
| 5.1   | Papéis e Protocolos                  | 67        |
| 5.2   | Manipuladores                        | 69        |
| 6     | Exemplos de Uso                      | <b>73</b> |
| 6.1   | Sincronização de Fluxo               | 75        |
| 6.2   | Depuração Distribuída                | 78        |
| 6.3   | Replicação Passiva                   | 81        |
| 7     | Conclusões                           | <b>86</b> |
| 7.1   | Trabalhos Relacionados               | 88        |
| 7.2   | Trabalhos Futuros                    | 90        |
| 8     | Referências Bibliográficas           | <b>93</b> |
| A     | Arquitetura CORBA                    | <b>99</b> |
| A.1   | Arquitetura de Gerência de Objetos   | 99        |
| A.2   | Estrutura do ORB                     | 100       |
| A.3   | Linguagem de Definição de Interfaces | 101       |
| A.4   | <i>Stubs</i> e Esqueletos            | 102       |
| A.5   | Repositório de Interfaces            | 103       |
| A.6   | Chamadas Dinâmicas                   | 103       |

## Lista de Figuras

|      |  |     |
|------|--|-----|
| 2.1  | Estrutura de componentes CCM.  | 20  |
| 2.2  | Um exemplo de definição de componente em IDL 3.0.                            | 20  |
| 2.3  | Interfaces do contêiner CCM.   | 33  |
| 4.1  | Exemplo de classe LOOP implementando uma pilha.                              | 46  |
| 4.2  | Estrutura do contêiner Lua.  | 49  |
| 4.3  | Arquitetura de gerência de definições de componentes do LuaCCM.              | 50  |
| 4.4  | Declaração de um componente repetidor e seu componente de teste.             | 54  |
| 4.5  | Exemplo de descritor de componente LuaCCM.                                   | 56  |
| 4.6  | Exemplo de pacote de componente LuaCCM.                                      | 57  |
| 4.7  | Exemplo de implantação de componentes LuaCCM.                                | 58  |
| 4.8  | Módulo de interfaces do LuaCCM.  | 64  |
| 5.1  | Exemplo de definição de papel no LOAF.                                       | 68  |
| 6.1  | Aplicação de fluxo de eventos  | 73  |
| 6.2  | Definição dos componentes da aplicação de fluxo de eventos em IDL 3.0        | 74  |
| 6.3  | Roteiro de implantação dos componentes da aplicação de exemplo               | 75  |
| 6.4  | Interfaces dos papéis de sincronização de fluxo e depuração distribuída.     | 76  |
| 6.5  | Papel de análise de fluxo de evento  | 77  |
| 6.6  | Papel de regulação de fluxo de evento  | 78  |
| 6.7  | Protocolo de sincronização de fluxo de evento                                | 79  |
| 6.8  | Aplicação de fluxo de eventos com sincronização aplicada por papéis.         | 79  |
| 6.9  | Papel de inspeção de componentes   | 80  |
| 6.10 | Exemplo de papel de inserção de ponto de parada                              | 81  |
| 6.11 | Parte da implementação do protocolo de depuração distribuída.                | 82  |
| 6.12 | Aplicação de fluxo de eventos com depuração distribuída aplicada por papéis. | 83  |
| 6.13 | Papel de replicação passiva.   | 83  |
| 6.14 | Exemplo de utilização do papel de replicação passiva.                        | 84  |
| 6.15 | Aplicação de fluxo de eventos com replicação passiva aplicada por papéis.    | 84  |
| A.1  | Arquitetura OMA.   | 100 |
| A.2  | Funcionamento dos <i>stubs</i> e esqueletos da arquitetura CORBA.            | 103 |