

**Hélcio Bezerra de Mello**

**Proxies Inteligentes —  
Monitoração e Adaptação  
Automáticas**

**DISSERTAÇÃO DE MESTRADO**

**DEPARTAMENTO DE INFORMÁTICA**

**Programa de Pós-graduação em**

**Informática**

Rio de Janeiro  
Setembro de 2004



**Hélcio Bezerra de Mello**

**Proxies Inteligentes — Monitoração e  
Adaptação Automáticas**

**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

Orientadora: Profa. Noemi de La Rocque Rodriguez

Rio de Janeiro  
Setembro de 2004



**Hélcio Bezerra de Mello**

**Proxies Inteligentes — Monitoração e  
Adaptação Automáticas**

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.

**Profa. Noemi de La Rocque Rodriguez**

Orientadora

Departamento de Informática — PUC-Rio

**Prof. Renato Fontoura de Gusmão Cerqueira**

PUC-Rio

**Prof. Sérgio Colcher**

PUC-Rio

**Profa. Thaís Vasconcelos Batista**

UFRN

**Prof. José Eugênio Leal**

Coordenador Setorial do Centro Técnico Científico —

PUC-Rio

Rio de Janeiro, 10 de Setembro de 2004

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

**Hélcio Bezerra de Mello**

Graduou-se em Engenharia de Computação na Universidade Federal do Espírito Santo.

Ficha Catalográfica

Mello, Hélcio Bezerra de

*Proxies Inteligentes — Monitoração e Adaptação Automáticas/ Hélcio Bezerra de Mello; orientadora: Noemi de La Rocque Rodriguez. — Rio de Janeiro : PUC-Rio, Departamento de Informática, 2004.*

v., 55 f: il. ; 30 cm

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

Inclui referências bibliográficas.

1. Informática – Teses. 2. Adaptação automática. 3. CORBA. 4. Qualidade de Serviço (QoS). 5. Monitoração. 6. *Proxy* Inteligente. I. Rodriguez, Noemi de La Rocque. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

## Agradecimentos

Agradeço primeiramente a Deus, por todas as auroras e crepúsculos, passados e vindouros, pela saúde, e por tudo o que ganhei ou conquistei na vida; e à minha família, pelo amor e compreensão a todo momento.

À professora Noemi Rodriguez, pela valiosa orientação nesta dissertação e pela paciência nos momentos difíceis, nos quais me depositou confiança incondicional. Ao professor Renato Cerqueira pelas idéias inspiradoras e ao pessoal do Tecgraf pelo suporte com as ferramentas. À PUC e ao seu excelente corpo docente, com o qual fiz um mestrado de ótima qualidade.

Aos grandes colegas, cuja convivência alivia a saudade dos amigos deixados para trás em Vitória, e que me proporcionaram vários momentos de alegria e distração. Faço questão de mencionar também os funcionários da universidade, tanto os da limpeza, que me possibilitaram usufruir de um ambiente de trabalho limpo e organizado, quanto as secretárias, sempre eficientes.

Finalmente, ao CNPq pelo apoio financeiro sem o qual meu mestrado dificilmente teria sido possível.

## Resumo

Mello, Hécio Bezerra de; Rodriguez, Noemi de La Rocque. ***Proxies Inteligentes — Monitoração e Adaptação Automáticas***. Rio de Janeiro, 2004. 55p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

No contexto de aplicações distribuídas, a necessidade de se adaptar a mudanças no ambiente de execução tem se tornado cada vez mais comum. Diversos trabalhos abordam a reconfiguração dinâmica de clientes e servidores em resposta a tais mudanças, inclusive na área da provisão de qualidade de serviço (QoS).

Esta dissertação explora o uso da reflexividade em uma arquitetura popular de *middleware* (CORBA) e emprega ferramentas adicionais para o desenvolvimento de um proxy inteligente. Um dos pontos principais desse proxy é sua simplicidade de uso, pois requer poucos parâmetros para ser instanciado e reage a eventos externos automaticamente.

Esse comportamento é obtido pela combinação da flexibilidade do *binding* LuaOrb com a conveniência das bibliotecas *LuaTrading* e *LuaMonitor*; a especificação das propriedades relevantes para a adaptação dinâmica é feita através de descritores simples e de fácil reutilização. Finalmente, este trabalho oferece mecanismos para *upload* e *download* de stubs especializados, com o objetivo de executar procedimentos de adaptação mais específicos. Com o objetivo de demonstrar uma possível aplicação para o proxy inteligente, apresentamos seu uso em um jogo simplificado para adaptá-lo automaticamente a eventos de escassez de recursos simulados.

## Palavras-chave

Adaptação Automática; CORBA; Qualidade de Serviço (QoS); Monitoração; Proxy Inteligente

## Abstract

Mello, Hécio Bezerra de; Rodriguez, Noemi de La Rocque. **Smart Proxies — Automatic Monitoring and Adaptation**. Rio de Janeiro, 2004. 55p. MSc. Dissertation — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

In the context of distributed applications, the need for adapting to changes in the execution environment is growing steadily. Several works deal with dynamic reconfiguration of clients and servers in response to such changes, including situations where provision of Quality of Service (QoS) is concerned.

This thesis proposes the use of reflexivity in a popular middleware architecture (CORBA) and other tools to develop a smart proxy. One of its main points is usage simplicity, for the proxy requires few parameters to be instantiated and reacts to external events automatically.

That behavior is achieved by combining the flexibility of the LuaOrb binding and the convenience of the LuaTrading and LuaMonitor libraries; the statement of properties relevant to the dynamic adaptation is accomplished by simple and easy-to-reuse descriptors. Finally, this work offers mechanisms to upload and download specialized stubs as to carry out more specific adaptation procedures.

In order to demonstrate a possible application of the smart proxy, we present a simple game that employs it to automatically adapt to simulated resource shortage events.

## Keywords

Automatic Adaptation; CORBA; Quality of Service (QoS); Monitoring; Smart Proxy

# Sumário

1	Introdução	10
2	Ferramentas Utilizadas	12
2.1	Lua	12
2.2	LuaOrb	13
2.3	LuaTrading	14
2.4	LuaMonitor	16
3	<i>LuaProxy</i>	18
3.1	SQDL — <i>Simple QoS Description Language</i>	18
3.1.1	Atributos	19
3.1.2	Mecanismo de Herança	20
3.1.3	Parâmetros de QoS	21
3.2	Adaptação Automática	23
3.2.1	Seleção de Servidores	24
3.2.2	Tratamento de Eventos	25
3.3	Usando o <i>LuaProxy</i>	26
3.4	Observações	27
4	Stubs Dinâmicos	29
4.1	<i>Download</i> de Stubs	30
4.1.1	Interagindo com o LuaProxy	31
4.2	<i>Upload</i> de Stubs	33
4.2.1	Ustubs Temporários	35
4.2.2	Ustubs Persistentes	36
4.2.3	Comparando os Ustubs	40
4.3	Gerenciador de Interfaces	41
4.4	Observações	42
5	Exemplo de Aplicação	43
5.1	Procedimento de Adaptação	43
5.2	Roteiro de Testes	47
6	Conclusão	49
6.1	Trabalhos Relacionados	50
6.2	Trabalhos Futuros	51
7	Referências Bibliográficas	54



## Lista de Figuras

2.1	Criando um tipo de serviço com o LuaTrading.	15
2.2	Criando um tipo de serviço sem o LuaTrading.	15
3.1	Criação de um descritor de QoS	19
3.2	Efeito indesejado do algoritmo de busca.	21
3.3	Herança de tabelas de parâmetros de QoS.	22
3.4	Definindo uma expressão de qualidade para o retardo.	22
3.5	Implementação da expressão de qualidade <i>default</i> .	23
3.6	Ajustando a expressão de qualidade <i>default</i> para o retardo.	23
3.7	Descritor para exemplo de string de preferência.	25
3.8	Exemplo de chamada de construtor do LuaProxy.	27
4.1	IDL para disponibilizar <i>downloadable stubs</i> .	30
4.2	Proxy encapsulado por um dstub.	32
4.3	Exemplo de uso do dstub, lado do servidor.	32
4.4	Exemplo de uso do dstub, lado do cliente.	32
4.5	IDL para <i>upload</i> de stubs.	34
4.6	Interface <i>Matrix</i> .	35
4.7	Instalando um ustub temporário	36
4.8	Instalando um ustub temporário com <i>callback</i>	37
4.9	Servidor antes e depois da instalação do ustub.	38
4.10	Exemplo de extensão de interfaces.	38
4.11	Usando o objeto estendido.	39
5.1	IDL do LuaEmpire	44
5.2	Uso do LuaProxy	44
5.3	Comunicação cliente-servidor no LuaEmpire.	44
5.4	Adaptação ao evento <i>bandwidth_decrease</i>	45
5.5	Adicionando suporte à compactação	45
5.6	Arquitetura modificada do LuaEmpire.	46
5.7	Adaptação ao evento <i>delay_increase</i>	46
5.8	Método <i>update</i> do objeto de callback.	47