

Luiz Marques Afonso

Um Estudo Sobre  
Contratos em Sistemas de  
Componentes de Software

DISSERTAÇÃO DE MESTRADO

DEPARTAMENTO DE INFORMÁTICA

Programa de Pós-graduação em

Informática

Rio de Janeiro  
Setembro de 2008

**Luiz Marques Afonso**

**Um Estudo Sobre Contratos em  
Sistemas de Componentes de Software**

**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  
Setembro de 2008

**Luiz Marques Afonso**

## **Um Estudo Sobre Contratos em Sistemas de Componentes de Software**

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. Edward Hermann Haeusler**  
Departamento de Informática — PUC-Rio

**Prof. Jean-Pierre Briot**  
LIP6 — Université Paris 6

**Prof. José Eugênio Leal**  
Coordenador Setorial do Centro Técnico Científico —  
PUC-Rio

Rio de Janeiro, 10 de Setembro de 2008

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

## Luiz Marques Afonso

Graduado em Informática pela Universidade Federal do Rio de Janeiro em 1996. Desde então, atuou principalmente nas áreas de Desenvolvimento de Software e Segurança da Informação. Atualmente, é Gerente de Desenvolvimento da empresa 3Elos Segurança em TI.

### Ficha Catalográfica

Afonso, Luiz Marques

Um Estudo Sobre Contratos em Sistemas de Componentes de Software/ Luiz Marques Afonso; orientador: Renato Fontoura de Gusmão Cerqueira. — 2008. 132 f.: il. ; 30 cm

Dissertação (Mestrado em Informática) - Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2008.

Inclui bibliografia.

1. Informática – Teses. 2. Componentes de software. 3. Programação por contratos. 4. CORBA. 5. Sistemas Distribuídos I. Cerqueira, Renato Fontoura de Gusmão. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

## Agradecimentos

Agradeço à minha família, em especial à minha mãe, pelo apoio e carinho em todos os momentos.

À Elen, pelo carinho, dedicação e paciência durante o período do Mestrado, além das sugestões sobre as figuras.

Ao Prof. Renato Cerqueira, pelo apoio e pela orientação no desenvolvimento deste trabalho, além das discussões técnicas sempre interessantes.

Aos amigos André Marins, Eduardo Fonseca e Ricardo Leal, pelos trabalhos desenvolvidos em conjunto, pelos momentos de descontração e, principalmente, pela amizade estabelecida no Mestrado.

Aos amigos da 3Elos, pelo apoio nas minhas freqüentes ausências do ambiente de trabalho.

E, finalmente, agradeço ao amigo Renato Maia pela resposta sempre rápida e precisa nas dúvidas em relação ao uso do OiL.

## Resumo

Afonso, Luiz Marques; Cerqueira, Renato Fontoura de Gusmão. **Um Estudo Sobre Contratos em Sistemas de Componentes de Software**. Rio de Janeiro, 2008. 132p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A programação por contratos é uma das técnicas voltadas para a melhoria da qualidade de software, sendo baseada no aumento do formalismo da especificação das interfaces. No contexto de componentes de software distribuídos, o uso de contratos envolve alguns desafios que o diferenciam do seu uso mais tradicional. O objetivo deste trabalho é a avaliação do uso de contratos no desenvolvimento de sistemas distribuídos baseados em componentes de software, identificando as abordagens existentes e analisando as vantagens e desvantagens do seu uso. São também avaliadas características como robustez, desempenho, flexibilidade, facilidade de uso e limitações. Como estudo de caso, foi desenvolvido um subsistema de suporte a contratos sobre um *middleware* CORBA implementado em Lua, servindo de base para experimentos realizados durante a pesquisa.

## Palavras-chave

Componentes de software, programação por contratos, CORBA, Sistemas Distribuídos

## Abstract

Afonso, Luiz Marques; Cerqueira, Renato Fontoura de Gusmão. **A Study About Contracts in Software Component Systems.** Rio de Janeiro, 2008. 132p. MSc. Dissertation — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Contract-based programming is one of the techniques used to improve the quality of software by enhancing the formalism of interface specifications. In the context of distributed software components, the use of contracts presents new challenges that make it different from its traditional use. This work intends to evaluate the use of contracts in the development of component-based distributed systems, identifying the current approaches and analyzing its advantages and disadvantages. It also covers topics like robustness, performance, flexibility, ease of use and limitations. As a case study, a contract subsystem was developed over a CORBA middleware using Lua, serving as the basis for experiments in our study.

## Keywords

Software Components, Contract Programming, CORBA, Distributed Systems

# Sumário

1	Introdução	13
1.1	Objetivos	16
1.2	Contribuições	16
1.3	Estrutura do Documento	17
2	Contratos	19
2.1	Fundamentos	19
2.2	Classificação em Níveis	21
2.3	Características	24
2.3.1	Especificação	25
2.3.2	Verificação	28
2.3.3	Notificação	29
2.3.4	Concorrência	30
2.3.5	Orientação a Objetos	32
2.3.6	Componentes	35
2.3.7	Suporte nas linguagens	35
2.3.8	Testes	37
2.3.9	Princípios para a especificação	38
3	Trabalhos Relacionados	43
3.1	Tamago	43
3.2	Contract-aware CORBA Container	44
3.3	CoCoMod / CoCoGen	47
3.4	CIPV	48
3.5	Considerações Finais	49
4	Estudo de Caso: Contratos em Lua	50
4.1	Características	50
4.1.1	Suporte a objetos Lua/OiL e Componentes SCS	51
4.1.2	Linguagem de Contratos	53
4.1.3	Especificação de Contratos	54
4.1.4	Quantificadores Lógicos	56
4.1.5	Transição de Estados	57
4.1.6	Sincronização de Operações	59
4.1.7	Avaliação de Contratos no Cliente	60
4.1.8	Notificação de Violações	61
4.1.9	Herança de Interfaces	62
4.1.10	Hierarquia de Contratos	63
4.1.11	Gerenciamento remoto de contratos	64
4.1.12	Configuração	64
4.2	Arquitetura	65
4.2.1	Interceptação de Chamadas	65
4.2.2	Estrutura de Módulos	67

4.2.3	Funcionamento	68
4.3	Implementação	70
4.3.1	Avaliação de assertivas	70
4.3.2	Tratamento de PRE	71
4.3.3	Quantificadores lógicos	72
4.3.4	Sincronização de chamadas	75
4.4	Exemplos	76
4.4.1	<i>Stack</i>	76
4.4.2	<i>List</i>	80
4.5	Desempenho	84
4.6	Limitações	85
5	Experimentos	<b>91</b>
5.1	SCS - Componente Básico	91
5.1.1	<i>IComponent</i>	92
5.1.2	<i>IReceptacles</i>	94
5.1.3	<i>IMetalInterface</i>	100
5.2	SCS - <i>Container</i>	101
5.2.1	<i>ComponentLoader</i>	102
5.2.2	<i>ComponentCollection</i>	103
5.3	SCS - <i>ExecutionNode</i>	105
5.4	SCS - <i>EventService</i>	105
5.5	<i>PingPong</i>	109
5.6	<i>Produtores e Consumidores</i>	110
5.7	<i>Echo</i>	113
5.8	Conclusões	117
6	Conclusões	<b>118</b>
6.1	Avaliação do Estudo	118
6.2	Trabalhos Futuros	121
7	Referências Bibliográficas	<b>124</b>
A	Interfaces do SCS	<b>128</b>

## Lista de Figuras

2.1	Classificação de contratos em níveis	22
2.2	Consultas e Comandos	38
4.1	Componente SCS	52
4.2	Interceptação de chamadas via <i>proxy</i>	66
4.3	Interceptação de chamadas de servidor no OiL	67
4.4	Relacionamento entre a aplicação, LUC e o OiL	68
4.5	Seqüência de operações para detecção de faceta do SCS	69
4.6	Implementação de quantificador com <i>thread pool</i>	74
4.7	Implementação de quantificador com chamada <i>deferred</i>	74
4.8	Execução do servidor <i>stack</i>	78
4.9	Execução do cliente <i>stack</i>	78
5.1	Funcionamento do canal de eventos	107
5.2	Componentes <i>PingPong</i>	110
5.3	Comparativo de desempenho de contrato com FORALL	116

## Lista de Tabelas

4.1	Extensões da linguagem de contratos	53
4.2	Extensões da linguagem de contratos para componentes SCS	54
4.3	Campos de descrição das facetas na tabela <i>FACETS</i>	54
4.4	Campos de descrição dos receptáculos da tabela <i>RECEPTACLES</i>	54
4.5	Chaves reservadas na definição de um contrato de interface	56
4.6	Conteúdo da tabela <i>__state</i>	58
4.7	Funções de um interceptador no OiL	66
4.8	Módulos do sistema	67
4.9	Arquivos do exemplo <i>Stack</i>	76
4.10	Testes de desempenho com a interface <i>List</i>	84
5.1	Tempo médio dos testes e seu desvio padrão	115
5.2	Tempos relativos à execução sem contratos	116

## Lista de Códigos

2.1	Exemplo de contrato em Eiffel . . . . .	26
3.1	Exemplo de contrato no Tamago . . . . .	45
3.2	IDL do componente do exemplo . . . . .	46
3.3	Exemplo de contrato de componente em OCL . . . . .	47
3.4	Exemplo de contrato do CIPV . . . . .	49
4.1	Exemplo de contrato no LUC . . . . .	55
4.2	Exemplo de transições de estado em contrato . . . . .	58
4.3	Exemplo de contrato de sincronização . . . . .	60
4.4	Interface de notificações do LUC . . . . .	62
4.5	Interface de gerenciamento de contratos . . . . .	64
4.6	Interface <i>Stack</i> . . . . .	77
4.7	Arquivo <i>stack_interceptor.lua</i> . . . . .	77
4.8	Arquivo <i>stack_server.lua</i> . . . . .	77
4.9	Arquivo <i>stack_client.lua</i> . . . . .	78
4.10	Contrato para a interface <i>Stack</i> . . . . .	79
4.11	Interface <i>List</i> . . . . .	80
4.12	Implementação da interface <i>List</i> . . . . .	81
4.13	Operação <i>List::count()</i> . . . . .	82
4.14	Operação <i>List::preceded_by()</i> . . . . .	82
4.15	Operação <i>List::item(i)</i> . . . . .	82
4.16	Operação <i>List::is_equal()</i> . . . . .	83
4.17	Operação <i>List::sublist()</i> . . . . .	83
5.1	Contrato da interface <i>IComponent</i> . . . . .	93
5.2	Contrato da operação <i>IReceptacles::connect()</i> . . . . .	95
5.3	Contrato da operação <i>IReceptacles::disconnect()</i> . . . . .	98
5.4	Contrato da operação <i>IReceptacles::getConnections()</i> . . . . .	99
5.5	Contrato da operação <i>IReceptacles::getConnection()</i> . . . . .	99
5.6	Contrato de sincronização da interface <i>IReceptacles</i> . . . . .	100
5.7	Contrato da interface <i>IMetaInterface</i> . . . . .	101
5.8	Contrato da interface <i>ComponentLoader</i> . . . . .	102
5.9	Contrato da interface <i>ComponentCollection</i> . . . . .	104
5.10	Contrato da interface <i>ExecutionNode</i> . . . . .	106
5.11	Contrato do componente <i>EventManager</i> . . . . .	108
5.12	Contrato do componente <i>EventChannel</i> . . . . .	109
5.13	Interface da faceta <i>PingPong</i> . . . . .	109

5.14	Contrato do componente <i>PingPong</i> . . . . .	111
5.15	Interface <i>Buffer</i> . . . . .	111
5.16	Contrato da interface <i>Buffer</i> . . . . .	112
5.17	Implementação das operações <code>Buffer::get()</code> e <code>Buffer::put()</code> . .	113
5.18	Interfaces do exemplo <i>Echo</i> . . . . .	114
5.19	Contrato da interface <i>Echo</i> . . . . .	115
A.1	Arquivo <code>scs.idl</code> . . . . .	128
A.2	Arquivo <code>deployment.idl</code> . . . . .	130
A.3	Arquivo <code>events.idl</code> . . . . .	132