

Hugo Marques de Castro Saldanha

**Utilizando Anotações em
Linguagens Orientadas a Objetos
para Suporte à Programação
Orientada a Componentes**

DISSERTAÇÃO DE MESTRADO

DEPARTAMENTO DE INFORMÁTICA
Programa de Pós-graduação em Informática

Rio de Janeiro
Agosto de 2010



Hugo Marques de Castro Saldanha

**Utilizando Anotações em Linguagens
Orientadas a Objetos para Suporte à
Programação Orientada a Componentes**

Dissertação de Mestrado

Dissertação apresentada ao Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico Científico da PUC-Rio como requisito parcial para obtenção do grau de Mestre em Informática.

Orientador : Prof. Renato Fontoura de Gusmão Cerqueira
Co-Orientador: Prof. Maria Julia Dias de Lima

Rio de Janeiro
Agosto de 2010



Hugo Marques de Castro Saldanha

**Utilizando Anotações em Linguagens
Orientadas a Objetos para Suporte à
Programação Orientada a Componentes**

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

Prof. Renato Fontoura de Gusmão Cerqueira

Orientador

Departamento de Informática — PUC-Rio

Prof. Maria Julia Dias de Lima

Co-Orientador

Departamento de Informática — PUC-Rio

Prof. Noemi de La Rocque Rodriguez

Departamento de Informática — PUC-Rio

Prof. Alessandro Garcia

Departamento de Informática — PUC-Rio

Prof. José Eugenio Leal

Coordenador Setorial do Centro Técnico Científico — PUC-Rio

Rio de Janeiro, 27 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.

Hugo Marques de Castro Saldanha

Graduou-se em Bacharelado em Informática pela PUC-Rio.

Ficha Catalográfica

Saldanha, Hugo Marques de Castro

Utilizando Anotações em Linguagens Orientadas a Objetos para Suporte à Programação Orientada a Componentes / Hugo Marques de Castro Saldanha; orientador: Renato Fontoura de Gusmão Cerqueira; co-orientador: Maria Julia Dias de Lima. — Rio de Janeiro : PUC-Rio, Departamento de Informática, 2010.

v., 74 f: il. (col.); 29,7 cm

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

Inclui referências bibliográficas.

1. Informática – Tese. 2. Componentes. 3. Programação baseada em componentes. 4. Middleware. 5. Sistemas distribuídos. 6. Anotações. 7. Programação orientada a atributos. I. Cerqueira, Renato Fontoura de Gusmão. II. Lima, Maria Julia Dias de. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

CDD: 004

Agradecimentos

Aos meus pais, por sempre me apoiarem em todos os momentos.

Aos meus orientadores Renato Cerqueira e Maria Julia, sempre atenciosos e acessíveis.

Aos meus gerentes Carlos Cassino e Leonardo Barros que me deram todo o apoio e flexibilidade para o desenvolvimento do meu trabalho de mestrado.

Aos colegas de trabalho Carlos Eduardo, Amadeu Barbosa, Maurício Rosas e Tomás Gorham pela atenção e prontidão, me ajudando sempre que necessário.

Aos meus amigos Juliana Jansen e Rafael Martinelli que muito me ajudaram com dicas, conselhos e apoio moral.

À Pontifícia Universidade Católica do Rio de Janeiro e ao Tecgraf pela base e infra-estrutura necessárias para o desenvolvimento do meu trabalho.

À CAPES, pela bolsa recebida durante dois anos de Mestrado.

Resumo

Saldanha, Hugo Marques de Castro; Cerqueira, Renato Fontoura de Gusmão; Lima, Maria Julia Dias de. **Utilizando Anotações em Linguagens Orientadas a Objetos para Suporte à Programação Orientada a Componentes**. Rio de Janeiro, 2010. 74p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Em sistemas distribuídos baseados em componentes de software, o uso de linguagens de programação orientadas a objetos é bastante comum para definir, através de *frameworks*, interfaces de programação para construção e uso de componentes. No entanto, o que se percebe na maioria dos modelos de programação que seguem essa abordagem, é a utilização de construções das próprias linguagens orientadas a objeto, como classes e interfaces, para definir uma interface de programação que siga um paradigma orientado a componentes. Como consequência, o código fonte mistura aspectos da funcionalidade do componente com os mecanismos de implementação específicos do modelo de programação, o que dificulta a reutilização deste componente em outros frameworks, além de incluir uma complexidade extra no código. Recentemente, observamos uma tendência à adição de metadados às implementações dos componentes, utilizando marcações específicas no código fonte. Estes metadados provêm as informações necessárias para que alguma ferramenta, seja baseada em geração de código ou em mecanismos de reflexão computacional da própria linguagem de programação, realize a integração da implementação do componente com a infra-estrutura de suporte do modelo de componentes. Essa técnica é denominada, por alguns autores, de Programação Orientada a Atributos. Linguagens como Java e C# já oferecem suporte nativo a esta técnica através das *Anotações*. O objetivo desta dissertação é investigar a adoção da técnica de programação orientada a atributos juntamente com uma linguagem orientada a objetos para construção de aplicações baseadas em componentes. Como parte do estudo, foi desenvolvido um novo mecanismo de programação baseado em atributos para a versão Java do middleware SCS.

Palavras-chave

Componentes; Programação baseada em componentes; Middleware; Sistemas distribuídos; Anotações; Programação orientada a atributos;

Abstract

Saldanha, Hugo Marques de Castro; Cerqueira, Renato Fontoura de Gusmão; Lima, Maria Julia Dias de. **Using Annotations in Object Oriented Languages to Support Component Oriented Programming**. Rio de Janeiro, 2010. 74p. MSc. Dissertation — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

In component-based distributed systems, the use of object-oriented programming languages is very common to define, through frameworks, programming interfaces for building and using components. However, most programming models that follow this approach, use the own object-oriented language concepts, such as classes and interfaces, to define a programming interface that follows a component-oriented paradigm. As a consequence, the source code mixes component functionality aspects with the programming model specific implementation mechanisms, what prevents the reuse of this component in other frameworks, and moreover, includes an extra complexity in the source code. Recently, a tendency to the addition of meta-data to the component implementation has been observed, by the use of specific markings on the source code. These meta-data provide the necessary information for tools, based on code generation or based on reflection mechanisms, to perform the integration of the component implementation with the component model's infrastructure support. Some authors call this technique as Attribute-Oriented Programming. Languages as Java and C# already provide native support to this technique through *Annotations*. The goal of this dissertation is investigate the adoption of the attribute-oriented programming technique with object-oriented languages to build components based applications. As part of the research, we developed a new programming mechanism based on attributes to the Java version of the middleware SCS.

Keywords

Component; Component-based programming; Middleware; Distributed systems; Annotations; Attribute-oriented programming;

Sumário

1	Introdução	11
2	A Infra-Estrutura SCS	14
2.1	Modelo de Componentes	14
2.2	Modelo de Programação	17
2.3	Construindo uma Aplicação SCS	19
2.3.1	Definição (Arquivo IDL)	19
2.3.2	Implementação	20
2.3.3	Implantação	22
3	ASCS	24
3.1	Modelo de Componentes Estendido	24
3.1.1	Contêiner Estendido	25
3.1.2	LifeCycle	26
3.1.3	Properties	27
3.2	Modelo de Programação	27
3.2.1	@FacetSet	27
3.2.2	@Facet	28
3.2.3	@Receptacle	28
3.2.4	@LifeCycle	30
3.2.5	@Property	32
3.2.6	@MetaInfo	32
3.3	Validação	34
3.4	Construindo uma Aplicação ASCS	36
3.5	Implementação do ASCS	37
4	Avaliação	39
4.1	Exemplos	39
4.1.1	RemoteLogger	39
4.1.2	Chat	42
4.2	Análise dos Modelos de Programação	45
4.2.1	Avaliação Qualitativa (CDN)	45
4.2.2	Avaliando Outras Abordagens	50
4.2.3	Quantidade de Linhas de Código	51
4.3	Análise de Desempenho	51
5	Trabalhos Relacionados	53
5.1	SCA	53
5.2	Fraclet	55
5.3	AOKell	57
5.4	CompJava	57
5.5	Comparando com o ASCS	59
6	Conclusões e Trabalhos Futuros	61

7	Referências Bibliográficas	63
A	Apêndice	66

Lista de figuras

2.1	Componente SCS	15
2.2	Conexão de Componentes SCS	15
2.3	Conexões SCS	16
2.4	Principais Entidades e Relacionamentos do Modelo SCS	16
3.1	Modelo de Componentes ASCS	25
3.2	Validação em tempo de compilação	35
4.1	Componentes RemoteLogger	40
4.2	Componentes Chat	42
4.3	Chat	45
5.1	Modelo de Componentes SCA	54
5.2	Modelo de Componentes Fraclet	56
5.3	AOKell	58
5.4	CompJava	59

Lista de tabelas

4.1	Quantidade de Linhas de Código	51
4.2	Desempenho HelloPrinter SCS	52
5.1	Comparando anotações	60