



Hugo Roenick

Um Modelo de Componentes de Software com Suporte a Múltiplas Versões

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

Rio de Janeiro
Março de 2011



Hugo Roenick

Um Modelo de Componentes de Software com Suporte a Múltiplas Versões

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. Noemi de La Rocque Rodriguez

Departamento de Informática — PUC-Rio

Prof. Alessandro Fabricio Garcia

Departamento de Informática — PUC-Rio

Prof. José Eugenio Leal

Coordenador Setorial do Centro Técnico Científico — PUC-Rio

Rio de Janeiro, 04 de Março de 2011

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

Hugo Roenick

Graduou-se em Engenharia da Computação pela PUC-Rio. É pesquisador do laboratório Tecnologia em Computação Gráfica (Tecgraf) da PUC-Rio desde 2006, onde trabalhou no desenvolvimento de algumas aplicações na área de GIS e Sistemas Distribuídos.

Ficha Catalográfica

Roenick, Hugo

Um modelo de componentes de software com suporte a múltiplas versões / Hugo Roenick; orientador: Renato Fontoura de Gusmão Cerqueira. — Rio de Janeiro : PUC-Rio, Departamento de Informática, 2011.

v., 66 f: il. ; 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. Múltiplas versões. 7. Extensibilidade independente. 8. Evolução de software. I. Cerqueira, Renato Fontoura de Gusmão. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Agradecimentos

Primeiramente agradeço a Deus por tudo o que Ele é e por estar conosco todos os dias e em todos os instantes. Como compreender esse amor??

Agradeço a minha noiva pela amiga e companheira que é.

Agradeço a minha família, e meus amigos por me apoiarem e incentivarem durante toda a minha vida. Vocês muito me ensinam e sou grato à Deus pela vida de todos vocês.

Aos companheiros de estudos e trabalhos, Victor, Amadeu, Cadu, Maia, Capitão e Gusta, obrigado pelo companheirismo, sugestões, críticas, auxílios, e risadas.

Ao Renato Cerqueira e Carlos Cassino, obrigado por confiarem no meu trabalho e me proporcionarem esta oportunidade.

Agradeço a PUC-Rio pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

Obrigado.

Resumo

Roenick, Hugo; Cerqueira, Renato Fontoura de Gusmão. **Um Modelo de Componentes de Software com Suporte a Múltiplas Versões**. Rio de Janeiro, 2011. 66p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Diversos modelos de componentes de software para sistemas distribuídos já foram propostos pela indústria e pela academia, tais como FRACTAL, CCM, COM, OPENCOM, LUACCM, e SCS. Uma das principais vantagens comumente atribuídas à abordagem de desenvolvimento baseado em componentes de software é o melhor suporte à extensibilidade independente (*independent extensibility*). Entretanto, a gerência de múltiplas versões de componentes ainda é um grande desafio, especialmente quando não é possível atualizar todos os componentes do sistema ao mesmo tempo e passa a ser necessária a coexistência, em um mesmo sistema, de diferentes versões de um mesmo componente. Neste trabalho, procuramos identificar os pontos fundamentais para oferecer suporte a múltiplas versões de interfaces de componentes de software e propor um modelo que ofereça esse suporte. Como avaliação da proposta, iremos estender o sistema de componentes SCS para dar suporte ao modelo proposto. Com base no histórico de evolução de diferentes componentes SCS utilizados em uma aplicação real, realizaremos experimentos com a nova versão do SCS para verificar a eficácia do modelo proposto.

Palavras-chave

Componentes; Programação baseada em componentes; Middleware; Sistemas distribuídos; Múltiplas versões; Extensibilidade independente; Evolução de software;

Abstract

Roenick, Hugo; Cerqueira, Renato Fontoura de Gusmão (Advisor). **A Software Component Model with Support for Multiple Versions**. Rio de Janeiro, 2011. 66p. MSc Dissertation — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Several software component models for distributed systems have been proposed by the industry and academy, such as FRACTAL, CCM, COM, OPENCOM, LUACCM, and SCS. One of the greatest advantages of component-based development is the better support for *independent extensibility*. However, managing multiple versions of components is still a challenge, especially when it's not possible to update all the system's components at the same time. Because of that, different versions of the same component interface shall be required to coexist in the same system. In this work, we try to identify the key points to support multiple versions of component interfaces and propose a model that offers this support. To evaluate the proposed model, we will extend the SCS component system to support it. Based on the evolution historic of different SCS components used in a real application, we will conduct experiments with the new version of SCS to verify the effectiveness of the proposed model.

Keywords

Component; Component-based programming; Middleware; Distributed systems; Multiple versions; Independent extensibility; Software Evolution;

Sumário

1	Introdução	9
1.1	Estrutura do Documento	11
2	Trabalhos Relacionados	12
2.1	Modelos de Componentes	12
2.2	ICE	15
2.3	UPSTART	16
2.4	Considerações Finais	18
3	O Modelo de Componentes Proposto	22
3.1	Requisitos do Suporte a Múltiplas Versões	22
3.2	O Modelo Estendido	24
3.3	Cenários de Uso	27
3.4	SCS-MV	30
3.5	Considerações Finais	35
4	Estudo de Caso	38
4.1	Mudanças entre as versões do OPENBUS	39
4.2	Modificações necessárias	41
4.3	Modificações sugeridas para a aplicação	47
4.4	Considerações finais	48
5	Conclusões	52
6	Referências Bibliográficas	54
A	IDL das Interfaces do SCS-MV	58
A.1	Interface <i>IFacet</i>	58
A.2	Interface <i>IComponent</i>	59
A.3	Interface <i>IReceptacles</i>	60
A.4	Interface <i>IMetalInterface</i>	61
B	O OPENBUS	62
C	Infraestrutura de Atualização Dinâmica do UPSTART	64

Lista de figuras

2.1	Modelo de Componente do CCM	13
2.2	Exemplo de Componentes SCS	14
2.3	Componentes SCS conectados	15
2.4	Objeto ICE	16
2.5	Encadeamento dos Objetos de Simulação	18
2.6	Abordagens mais comuns para prover versionamento de interfaces com o modelo tradicional de componentes de software.	19
2.7	Exemplo da extrapolação das várias versões de serviços e a analogia se as versões estivessem encapsuladas (modularizadas) de acordo com o serviço prestado	20
3.1	Definição dos relacionamentos entre os conceitos do novo modelo SCS.	25
3.2	Modelo de Componente Estendido	25
3.3	Casos possíveis de componentes cliente/servidor com uso ou não de múltiplas versões.	28
4.1	Arquitetura do OPENBUS na versão 1.4 utilizando o modelo original do SCS.	50
4.2	Arquitetura do OPENBUS na versão 1.5 utilizando o modelo original do SCS.	50
4.3	Arquitetura do OPENBUS na versão 1.5 utilizando o modelo do SCS com suporte a múltiplas versões.	51
B.1	Arquitetura do OPENBUS	62
C.1	A infra-estrutura de atualização. Os componentes se comunicam através de uma rede (linha em negrito); setas indicam direção de chamadas de métodos (possivelmente remotas). O nó está executando uma instância de classe na versão i e SOs para as versões $i-1$, $i+1$ e $i+2$.	65