



UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

Modularização e Extensão do InteropFrame

Felipe Oliveira Carvalho
Rodrigo Losano Fontes Calheiros

São Cristóvão – SE

2014

Felipe Oliveira Carvalho
Rodrigo Losano Fontes Calheiros

Modularização e Extensão do InteropFrame

Pré- Projeto do Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação da Universidade Federal de Sergipe.

Orientador: Prof. Dr. Tarcísio da Rocha

São Cristóvão – SE
2014

Modularização e Extensão do InteropFrame

Felipe Oliveira Carvalho
Rodrigo Losano Fontes Calheiros

Pré- Projeto do Trabalho de Conclusão de
Curso apresentado como requisito parcial
para obtenção do título de Bacharel em
Sistemas de Informação da Universidade
Federal de Sergipe.

Aprovado em: ____ / ____ / ____.

Banca Examinadora:

Prof. Dr. Tarcísio da Rocha (Orientador)
Universidade Federal de Sergipe

Prof. xxxxxx xxxxxxxx xxxxxxxx
Universidade Federal de Sergipe

Prof. xxxxxx xxxxxxxx xxxxxxxx
Universidade Federal de Sergipe

São Cristóvão – SE
2014

Resumo

O desenvolvimento de Sistemas Distribuídos tem se tornado uma tarefa complexa. Uma técnica que tornou-se amplamente utilizada no desenvolvimento desses sistemas é a Engenharia de Software Baseada em Componentes (ESBC), o que resultou no surgimento de diversos modelos de componentes. Um desafio que surge do desenvolvimento baseado em componentes é o da interoperabilidade entre partes heterogêneas desenvolvidas em diferentes modelos de componentes. Em geral, o problema da interoperabilidade entre partes heterogêneas de um sistema é tratado pelo uso de *middlewares*. Sistemas de *middleware* são capazes de promover integração e abstrair do desenvolvedor detalhes dessa integração.

Dentro desse contexto foi criado o InteropFrame, um *middleware* que lida com a interoperabilidade entre sistemas distribuídos desenvolvidos nos modelos de componentes OpenCOM e Fractal, além de tratar de detalhes de comunicação remota utilizando RMI e *Web Services SOAP*. O InteropFrame é uma solução desenvolvida em Java puro e é extensível para o suporte a novos modelos de componentes e também de comunicação remota. Porém, por ser desenvolvido em Java puro, esse suporte fica dificultado, uma vez que não são estabelecidas interfaces modulares para a extensibilidade. Além dessa problemática, o InteropFrame também possui limitações na comunicação remota interna.

Este trabalho propõe a modularização do InteropFrame utilizando o modelo OSGi, de forma a reorganizar sua arquitetura em *plug-ins* para os modelos de componentes suportados e também para os modelos de comunicação remota. Visando confirmar a proposta de modularização, será adicionado o suporte ao modelo OSGi para tornar-se interoperável com os modelos já suportados pelo InteropFrame. Além da modularização, serão feitas melhorias internas na comunicação remota do InteropFrame.

Palavras-chave: Modelos de Componentes, Interoperabilidade, Modularização, Sistemas Distribuídos.

Sumário

	Sumário	ii
	1 INTRODUÇÃO	iii
1.1	Motivação	iii
1.2	Objetivos do Trabalho	iv
1.2.1	Objetivo Geral	iv
1.2.2	Objetivos Específicos	iv
1.3	Contribuições deste Trabalho	iv
1.4	Organização do Trabalho	v

Capítulo 1

Introdução

Esta introdução tem por objetivo apresentar: a motivação para este trabalho de conclusão de curso; os objetivos gerais e os específicos deste trabalho; as principais contribuições; e a estrutura e organização deste documento.

1.1 Motivação

A necessidade de soluções em sistemas distribuídos complexos vem substituindo a visão de sistemas distribuídos homogêneos onde aplicações de domínio específico são desenvolvidas usando plataformas e *middleware* projetados especificamente para esse domínio. Soluções tecnológicas independentes têm sido interconectadas para criar estruturas ainda mais ricas, os chamados sistemas de sistemas (SoS). Um dos principais desafios dessas interconexões é a questão da interoperabilidade: a habilidade desses sistemas se conectarem, de trocarem dados e de se comunicarem (??), (??), (??).

Uma técnica que tem sido amplamente utilizada nos últimos anos no desenvolvimento de plataformas de sistemas distribuídos é a ESBC (Engenharia de Software Baseada em Componentes) (??). Como consequência do sucesso no uso de componentes de software, vários modelos de componentes diferentes emergiram. OpenCOM (??), Fractal (??), Spring (??), EJB (??), OSGi (??), CCM (??) e SCA (??) são alguns exemplos destes modelos.

Diversas soluções têm sido criadas para lidar com a interoperabilidade entre sistemas distribuídos desenvolvidos em diferentes modelos de componentes. Uma dessas soluções é o objeto de estudo deste trabalho: O InteropFrame (??).

O InteropFrame é um *middleware* extensível desenvolvido em Java que trata a questão da interoperabilidade entre sistemas desenvolvidos nos modelos de componentes OpenCOM ou Fractal através da geração automática de *proxies* para a comunicação remota.

Estes *proxies* atuam como representantes locais a um sistema que repassam as chamadas de métodos para outros computadores dentro de uma rede, seja ela local ou a própria internet. No InteropFrame, as chamadas de métodos podem ser repassadas pela rede utilizando o mecanismo Java RMI (*Remote Method Invocation*) ou *Web Services* do tipo SOAP. Com isso o InteropFrame é capaz de fazer o *binding* (ligação remota) entre partes distribuídas de modo que o utilizador não precise se preocupar com detalhes de comunicação remota nem de aspectos de interoperabilidade entre os modelos de componentes suportados. O InteropFrame é extensível do ponto de vista do desenvolvedor que deseje implementar o suporte a novos modelos de componentes e de *binding* através do desenvolvimento de *plug-ins* à parte.

Embora o InteropFrame funcione para os modelos de componentes propostos, ele possui algumas limitações. Uma dessas limitações é que o InteropFrame não é desenvolvido numa plataforma de componentes específica, sendo totalmente desenvolvido em Java. Esse fato acaba limitando o processo de extensibilidade do *middleware*. Outra limitação é a quantidade de modelos de componentes suportados – apenas Fractal e OpenCOM.

1.2 Objetivos do Trabalho

1.2.1 Objetivo Geral

Superar limitações do InteropFrame e estender o suporte a novos modelos de componentes.

1.2.2 Objetivos Específicos

- Estudo do InteropFrame;
- Fazer a portabilidade do InteropFrame para o modelo de componentes OSGi como forma de modularizá-lo em *plug-ins*. Dessa forma o suporte à extensibilidade ficará facilitado;
- Implementar o suporte ao OSGi como um modelo de componentes interoperável dentro do InteropFrame a fim de comprovar a extensibilidade do *middleware*;
- Avaliar impactos no desempenho do *middleware* devido à introdução do OSGi;
- Realizar melhorias no processo interno de comunicação remota do InteropFrame.

1.3 Contribuições deste Trabalho

As principais contribuições deste trabalho são:

- Evolução do InteropFrame.
- Disponibilização do código-fonte como forma de possibilitar estudos futuros à respeito da interoperabilidade entre sistemas distribuídos desenvolvidos em diferentes modelos de componentes.

1.4 Organização do Trabalho

Os capítulos subsequentes deste trabalho serão organizados da seguinte forma:

- Capítulo 2 - Apresentação dos conceitos básicos necessários para o entendimento do tema.
- Capítulo 3 - Descrição do InteropFrame e apresentação da solução proposta.
- Capítulo 4 - Considerações finais.

Referências

ALLIANCE, O. *The OSGi Architecture*. Disponível em: <<http://www.osgi.org/Technology/WhatIsOSGi>>. Acesso em: 22 maio 2014.

BLAIR, G. et al. Interoperability in Complex Distributed Systems. In: BERNARDO, M.; ISSARNY, V. (Ed.). *Formal Methods for Eternal Networked Software Systems*. Springer Berlin / Heidelberg, 2011, (Lecture Notes in Computer Science, v. 6659). p. 1–26. ISBN 978-3-642-21454-7. 10.1007/978-3-642-21455-4_1. Disponível em: <http://dx.doi.org/10.1007/978-3-642-21455-4_1>.

BROMBERG, Y. et al. Bridging the interoperability gap: overcoming combined application and middleware heterogeneity. *12th IFIP/ACM/USENIX International Middleware Conference*, Springer, p. 390–409, 2011.

BRUNETON, E. et al. The fractal component model and its support in java: Experiences with auto-adaptive and reconfigurable systems. *Softw. Pract. Exper.*, John Wiley & Sons, Inc., New York, NY, USA, v. 36, n. 11-12, p. 1257–1284, set. 2006. ISSN 0038-0644. Disponível em: <<http://dx.doi.org/10.1002/spe.v36:11/12>>.

COULSON, G. et al. A generic component model for building systems software. *ACM Trans. Comput. Syst.*, ACM, New York, NY, USA, v. 26, n. 1, p. 1:1–1:42, mar. 2008. ISSN 0734-2071. Disponível em: <<http://doi.acm.org/10.1145/1328671.1328672>>.

DEMICHIEL, L.; KEITH, M. *JSR 220: Enterprise JavaBeans*. 3.0. ed. Santa Clara, 2007.

INVERARDI, P.; ISSARNY, V.; SPALAZZESE, R. A theory of mediators for eternal connectors. *Leveraging Applications of Formal Methods, Verification, and Validation*, Springer, p. 236–250, 2010.

JOHNSON, R. et al. *The spring framework - reference documentation*. 2.0.6. ed. [S.l.], 2007.

NASCIMENTO, S. C. do.

Um Framework Extensível para Interoperabilidade Dinâmica entre Componentes Distribuídos — Universidade Federal de Sergipe, Aracaju, SE, Brasil, 2013.

OASIS Open. *Service component architecture (SCA) assembly model specification*. 1.1. ed. [S.l.], 2011.

OMG. *OMG CORBA component model (CCM) specification*. 3.0. ed. Needham, 2002.

ROUVOY, R.; MERLE, P. Leveraging component-based software engineering with Fraclet. *Annals of Telecommunications*, Springer Paris, v. 64, p. 65–79, 2009. ISSN 0003-4347. 10.1007/s12243-008-0072-z. Disponível em: <<http://dx.doi.org/10.1007/s12243-008-0072-z>>.