**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 [blair2011interoperability], [inverardi2010theory], [bromberg2011bridging].

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) [rouvoy2009leveraging]. Como consequência do sucesso no uso de componentes de *software*, vários modelos de componentes diferentes emergiram. OpenCOM~\cite{coulson2008generic}, Fractal~\cite{bruneton2006fractal}, Spring~\cite{spring2007}, EJB~\cite{ejb2005}, OSGi~\cite{alliance2013osgi}, CCM~\cite{ccm2002} e SCA~\cite{sca2007} 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 (Nascimento, 2013).

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*. (Pendendo fontes). Outra limitação é a quantidade de modelos de componentes suportados – apenas Fractal e OpenCOM.

1.2 Objetivos

Objetivo Geral:

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

Objetivos específicos:

* Fazer a portabilidade do InteropFrame para o modelo de componentes OSGI como forma de separa-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.
* Avaliar impactos no desempenho do *middleware* devido à introdução do OSGI;
* Desenvolver o configurador distribuído do InteropFrame com *Web service* RESTful.

1.3 Contribuições deste trabalho de conclusão de curso

As principais contribuições deste trabalho de conclusão de curso 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 deste documento

Cap 1 – Introdução

Cap 2 – Conceitos básicos necessários para o entendimento do tema

Cap 3 – Descrição do InteropFrame

Cap 4 – Solução proposta

Cap 5 – Avaliação das mudanças

Cap 6 – Conclusões