6
Conclusões

Propusemos neste trabalho uma extensão para o modelo de componentes do middleware SCS, que adiciona o suporte à comunicação através de fluxos de dados. Para atingir esse objetivo foram realizados estudos das caracteristicas dessa forma de comunicação e das aplicações que as utilizam. Além disso, as abordagens utilizadas nos principais trabalhos relacionados foram analisadas, gerando um conjunto de ideias que auxiliaram no desenvolvimento da solução apresentada.

A extensão desenvolvida proporciona, aos desenvolvedores de componentes, abstrações de conectores de fluxo de dados como método de comunicação alternativo à chamada de procedimentos remotos. Essas abstrações oferecem mecanismos padronizados de estabelecimento de conexão e controle de transmissão, que incrementam a expressividade do middleware SCS, permitindo o desenvolvimento de aplicações que possuam tais requisitos de comunicação de forma relativamente mais simples.

A partir do levantamento de diretivas, requisitos e funcionalidades, um protótipo dessa extensão foi implementado utilizando a versão Java do middleware SCS. Esse protótipo foi utilizado para testar e validar as decisões arquiteturais, resultando na solução apresentada no capítulo 4. O processo de pesquisa e desenvolvimento desse protótipo ajudou constantemente na reavaliação de diversas partes da arquitetura e no aprendizado de diversos conceitos, como por exemplo, conectores de software e arquitetura de servidores.

Para solucionar o problema de comunicação por fluxo de dados, foram adicionados dois novos conectores chamados *ISource* e *ISink*, responsáveis por transmitir e receber grandes quantidade de dados. Uma nova faceta *IStreamComponent* também foi adicionada ao modelo para oferecer operações de identificação e introspecção a esses novos conectores. A arquitetura interna desses novos conectores foi projetada para suportar diferentes mecanismos de transporte através de *drivers* de conexão.

Para avaliar o reuso de bibliotecas de transporte existentes e sua compatibilidade com a arquitetura interna dos conectores, foi desenvolvido um *driver* de conexão baseado no protocolo *FTC* [37]. Por reaproveitar diversos recur-

sos e operações fornecidas pela biblioteca desse protocolo, a implementação resultante se mostrou tão simples quanto a do *driver* TCP. Um resultado que consideramos positivo.

Para solucionar o problema de implantação dinâmica, a infraestrutura de execução desenvolvida por *Augusto* [29] foi adaptada para fazer proveito dos novos conectores e utilizada em conjunto com a infraestrutura de implantação desenvolvida por *Barbosa* [17]. Com isso, uma aplicação de execução de fluxo de algoritmos foi concebida para validar as soluções e avaliar aspectos como completude das funcionalidades e dificuldade de utilização. A avaliação foi positiva, porém ajudou a identificar limitações da extensão proposta.

As limitações identificadas foram: a unidirecionalidade da transmissão de dados, subaproveitando mecanismos de transporte que possuem tal recurso; a falta de suporte a múltiplas fontes nos conectores consumidores, dificultando a utilização de mecanismos de transporte que façam utilização desse recurso; e, por fim, a falta de mecanismos padronizados para descoberta em tempo de execução, de funcionalidades disponíveis nas implementações dos *drivers* de conexão e dos componentes. Entretanto, nenhuma dessas limitações foram consideradas impeditivas para a utilização da extensão proposta no cenário que motivou o desenvolvimento deste trabalho.

Uma avaliação de desempenho foi realizada para investigar a sobrecarga introduzida pela arquitetura desenvolvida e validar a implementação do protótipo. Os resultados obtidos não apresentaram sobrecargas significativas, e quando comparados aos resultados de outros mecanismos de transmissão, se mostraram bastante adequados. Com isso, a implementação da extensão proposta obteve uma avaliação positiva em termos de desempenho.

As principais contribuições obtidas como resultado deste trabalho foram: a extensão supracitada do modelo de componentes SCS detalhada no capítulo 4, que fornece um mecanismo de comunicação alternativo para componentes SCS; e a experiência adquirida no desenvolvimento desse projeto, relatada através das decisões e dificuldades encontradas mencionados durante os capítulos 4 e 5, que podem ser úteis para projetos futuros.

Por fim, alguns trabalhos futuros podem ser desenvolvidos a partir do conhecimento adquirido com este trabalho, como por exemplo:

1. Investigação da influência da arquitetura proposta na Qualidade de Serviço fornecida e, com isso, identificar a viabilidade da utilização desta arquitetura em sistemas que possuam requisitos de qualidade específicos.

2. Implementação de conectores de fluxo de dados bidirecionais que aproveitem os recursos oferecidos pelo mecanismos de transporte utilizados,

evitando a necessidade de estabelecer novas conexões.

3. Investigação da utilização da arquitetura e do sistema de tipos proposto em sistemas de distribuição de conteúdo multimídia auto-organizáveis, similares ao desenvolvido por *Rafaelsen et al.* [23].

4. Avaliação da arquitetura em sistemas reais, investigando detalhadamente a escalabilidade da solução proposta em sistemas com múltiplos consumidores.

5. Avaliação qualitativa de quanto a arquitetura e os mecanismos padronizados de estabelecimento de conexão e controle de transmissão auxiliam no desenvolvimento de sistemas distribuídos com requisitos de conectores de fluxo de dados.