

6

Conclusões

Neste trabalho, fizemos um estudo de *middlewares* adaptáveis que utilizam as mais diversas técnicas para realizar essa adaptação. Para melhor entender a diferença entre eles, dividimos os *middlewares* estudados em categorias, usando três dimensões diferentes: o momento em que é feita a adaptação, onde ela ocorre e como ela é efetuada.

O *middleware* utilizado como base nesse trabalho, o OiL, já era usado anteriormente para pesquisas na área de adaptação e de discussões sobre técnicas de implementações de *middlewares*. Usando idéias retiradas dos *middlewares* estudados, propusemos uma arquitetura de componentes para o OiL, de forma a possibilitar adaptações nas suas camadas internas. Através da implementação dessa arquitetura, conseguimos construir um sistema que, com configurações simples no momento da sua inicialização, pode fornecer funcionalidades diferentes ao desenvolvedor da aplicação que o utiliza.

Com a nova arquitetura, criamos exemplos que utilizam essa capacidade de configuração e fizemos a comparação do desempenho dessa arquitetura com a implementação do OiL anterior. Assim, pudemos confirmar que mesmo com a inserção dessa capacidade adaptativa, o *middleware* não sofreu um impacto significativo no seu desempenho.

Em suma, podemos observar duas contribuições principais deste trabalho. A primeira é a obtenção, com o OiL componentizado, de um ambiente para a realização de experimentos de técnicas de implementação de *middlewares*, como as apresentadas no capítulo 2. A segunda contribuição é a nova arquitetura do OiL, usando componentes, que acrescenta ao *middleware* a possibilidade de novas pesquisas na área de configuração, pois fornece uma separação dos blocos que formam a infra-estrutura em unidades mais bem definidas do ponto de vista funcional, e na área de adaptação dinâmica, por usar um modelo de componentes que oferece mecanismos simples para a modificação da conexão entre os blocos que formam o OiL.

6.1

Trabalhos futuros

No processo de separação de interesses para a formação dos blocos que compõem o *middleware* no capítulo 4, percebemos que o interesse ligado ao escalonador não pôde ser encapsulado de maneira satisfatória, como proposto pelo modelo. As modificações necessárias em outros componentes para a inserção ou remoção do escalonador podem ser consideradas como um aspecto transversal ao resto do *middleware*. Uma revisão do modelo apresentado, tendo como foco a separação do escalonador de outros blocos, bem como um estudo sobre técnicas utilizadas em outros *middlewares* para resolver esse problema podem ser considerados como estudos futuros interessantes.

O modelo de componentes do capítulo 3 cria a possibilidade de adaptações dinâmicas no próprio *middleware*. Embora esse aspecto não tenha sido explorado neste trabalho, já que apenas o tipo de modelo de componentes mais simples foi utilizado, um trabalho futuro possível é a experimentação com tipos de modelo de componentes mais complexos. Com a adição de interceptadores nas conexões entre os componentes do sistema, a capacidade de inserir ou remover implementações de facetas ou receptáculos, e a possibilidade de mudanças entre as conexões dos componentes em tempo de execução, podemos examinar o que essas possibilidades acrescentariam em um *middleware* adaptável.

A inserção de mudanças em tempo de execução, no entanto, cria alguns problemas na manutenção de um estado consistente do *middleware*. Para isso, um estudo possível é a implementação de mecanismos similares ao *LifeCycleController* do Fractal ou à interface *ILifeCycle* do OpenORB, para garantir que o sistema possa reconfigurar as conexões entre os componentes e ao mesmo tempo conservar o estado de execução com um mínimo de trabalho para o desenvolvedor da aplicação.

Outro estudo possível é a comparação da implementação anterior do OiL com a sua versão componentizada, através de métricas de *software* (58) (59). As métricas serviriam para avaliar se a separação de interesses foi feita com sucesso e se o acoplamento entre os blocos do *middleware* diminuiu.

Para validação da arquitetura proposta, propomos também a repetição de estudos feitos na arquitetura anterior do OiL, como por exemplo a pesquisa em travessia de *firewalls* apresentada em (54). Esperamos que a arquitetura apresentada neste trabalho facilite a descoberta de que pontos devem ser modificados, já que há um melhor agrupamento das funcionalidades do *middleware*. Além disso, a inclusão de novos protocolos de *middlewares* diferentes, tais como o ICE (60), pode auxiliar a comprovar se a separação de interesses em

componentes e a nova arquitetura se comportam de maneira satisfatória.