

7 Conclusões

Este trabalho descreveu um grupo de extensões criadas para definir uma nova maneira de descrever e selecionar componentes no *middleware* Kaluana. O objetivo destas extensões é desenvolver um mecanismo de seleção de componentes através de um contrato de reconfiguração, com a noção de equivalência entre suas interfaces, como solução para a evolução e a reconfiguração, em tempo de execução, de aplicações baseadas em modelos de componentes orientados a serviços.

Este novo mecanismo de seleção foi criado de modo a tentar evitar que exista incompatibilidade de componentes com a aplicação em processo de reconfiguração.

Para tal, foi desenvolvido um contrato de reconfiguração que agrega informações sobre os serviços, as dependências e os requisitos não funcionais do componente. Através das restrições computacionais tornou-se explícito o comportamento de um componente de acordo com o contexto de execução encontrado no dispositivo.

A implementação foi validada através da implementação de uma aplicação exemplo que utiliza-se da capacidade de adaptação e implantação dinâmica oferecida pelo *middleware* Kaluana e demonstra a simplicidade na criação de aplicações adaptáveis. O estudo de caso implementado, tem um mecanismo de adaptação baseado na localização do dispositivo, considerando informações de contexto, tais como nível de energia, tipo de conexão sem fio, etc.

Além disso, foi explicitado um exemplo real de uso das extensões propostas neste trabalho para o *middleware* Kaluana, através do uso dos conceitos definidos para a implementação do *middleware* Kamaiurá.

São propostos como trabalhos futuros, outros tipos de extensões e melhorias na implementação do *middleware* para torná-lo mais abrangente e flexível.

7.1. Limitações

A pesar das extensões do *middleware* Kaluana apresentadas neste trabalho criarem uma flexibilidade maior na definição e seleção de componentes para aplicações dinamicamente adaptáveis, o projeto possui algumas limitações em sua implementação, como:

- Não existe uma flexibilização da implementação das restrições computacionais. Para que um desenvolvedor queira realizar uma extensão do *middleware* Kaluana criando restrições computacionais próprias, é necessário ter acesso ao código do próprio *middleware*. Não existe um ponto de conexão do código para que o desenvolvedor possa anexar restrições de maneira dinâmica.
- Não é possível descobrir servidores remotos em tempo de execução, ficando restrito ao desenvolvedor do *middleware* o cadastramento de novos servidores de maneira estática.
- Mesmo que tenha sido feita, para este trabalho, a integração do *middleware* Kaluana com a plataforma Mobilis, o registro de consumidores de informações de contexto no CMS é feito sem a possibilidade de inserção de novos interesses a diferentes provedores de contexto em tempo de execução.
- A pesar de existir alguma flexibilização do mecanismo de avaliação dos componentes, não é possível componentizar esta parte do código de maneira a anexar avaliações mais complexas para a escolha de um componente. Por exemplo, não é possível definir uma ontologia para a avaliação dos componentes candidatos a instanciação.

7.2. Trabalhos futuros

Com base nas restrições de projeto citadas na seção 7.1, são propostas algumas extensões para o *middleware* que criariam uma maior flexibilização do código criado.

7.2.1.

Uso de ontologias para descrição e seleção de componentes

Uma ontologia necessariamente implica ou incorpora algum tipo de visão do mundo de acordo com um dado domínio. Esta visão do mundo é concebida, em sua maioria, como um grupo de conceitos (i.e. entidades, atributos, processos, etc), suas definições e suas relações [73].

Através da flexibilização da descrição e seleção de componentes, o Kaluana pode usar uma ontologia para fornecer um vocabulário compartilhado, que pode ser usado para modelar o domínio dos componentes.

Por exemplo, uma ontologia poderia ser usada para definir o tipo dos objetos e/ou conceitos que existem no repositório de componentes, assim como suas propriedades e relações, criando uma nova maneira de descrever o comportamento de cada componente.

Esta nova maneira de descrever os componentes pode ser anexada como um serviço, que seria criado por um desenvolvedor que não participasse da implementação do Kaluana ou mesmo ser criada como um novo módulo do *middleware*.

7.2.2.

Flexibilização do uso de interesses ao contexto

Para criar uma maneira mais flexível de registrar interesse a provedores de contexto, o Kaluana poderia criar um serviço que provesse uma possibilidade de adicionar interesses em informações de contexto novas em tempo de execução.

Isto deve ser realizado em conjunto com um serviço para acrescentar novos tipos de restrições computacionais em tempo de execução, ou estas novas informações de contexto simplesmente não serão aproveitadas pelo *middleware* na avaliação de um componente candidato a instanciação. Por exemplo, não adianta adicionar dinamicamente um interesse no nível de ruído captado por um dispositivo se não for possível que um componente restrinja sua execução de acordo com o nível de ruído do ambiente.

7.2.3. Flexibilização da descoberta de servidores remotos

Assim como na integração do Kaluana com o CMS, a flexibilização da descoberta de servidores remotos de componentes pode ser criada a partir da integração do *middleware*, com um serviço específico para buscar e cadastrar novos servidores de componentes.

Este serviço deve ser implementado em um servidor externo ao dispositivo visto que a tarefa de *data mining*, necessária para a descoberta dos servidores, seria pesada demais para ser executada no dispositivo. Esta tarefa consumiria energia e largura de banda desnecessariamente, consumindo os recursos do dispositivo, necessários para execução das aplicações do usuário.