

1 Introdução

A computação móvel cria novos paradigmas para o desenvolvimento de *software*. Tradicionalmente, o suporte à adaptabilidade de *software* em ambientes móveis é fornecido por *middlewares* que dão base para adaptação dinâmica de aplicações. Isto é feito para que o desenvolvedor da aplicação não precise se preocupar com o desenvolvimento de mecanismos e estratégias de adaptação específicas para sua aplicação.

Desta maneira o desenvolvedor das aplicações não precisa lidar diretamente com questões como chamadas ao sistema operacional ou conexões de diferentes interfaces de rede. A implementação de código específico para adaptar uma aplicação em tempo de execução, a tornaria extremamente complexa e faria com que sua manutenção fosse custosa e difícil. Existem diversos exemplos de *middlewares* criados para dar apoio a adaptações dinâmicas, como [1-4].

Através da ciência ao contexto, uma arquitetura de *middleware* possui a capacidade de notar mudanças no seu ambiente de execução e disparar adaptações dinâmicas de um *software* em um ambiente móvel. Estas adaptações podem representar uma alteração na conexão de rede sem fio de um dispositivo, mudança da localização de um usuário ou mesmo o nível de ruído de um ambiente.

O desenvolvimento de *software* adaptativo pode ser facilitado usando o paradigma, ou modelo, de componentes de software, que permite customizar a composição de serviços em uma arquitetura de *middleware*. Estes modelos permitem explicitar a integração de novos componentes ao sistema e/ou a substituição por outros. Abordagens baseadas em componentes, combinadas com mecanismos de reflexão são adotadas em diversos sistemas, como [5-8].

1.1. Motivação e contribuição

Com o intuito de simplificar o processo de desenvolvimento e a capacidade de adaptação dinâmica em aplicações móveis, o *middleware* Kaluana [12] define

um modelo de componentes orientado a serviços que permite a composição e implantação de componentes em tempo de execução. As aplicações que utilizam o Kaluana constituem uma composição de componentes, cujo desenvolvimento é independente do desenvolvimento da aplicação que os utiliza. Com essa separação entre o desenvolvimento de componentes e da aplicação, o processo geral de desenvolvimento de aplicações torna-se mais simples e ágil.

Esta dissertação apresenta uma extensão para o *middleware* Kaluana com um novo mecanismo de seleção de componentes para a implantação ou substituição dinâmica de componentes. Esse mecanismo de seleção leva em conta um contrato de reconfiguração, com a noção de equivalência entre suas interfaces. Esta nova seleção de componentes propõe uma melhoria na composição das aplicações, realizada através de uma seleção de componentes mais adequados ao contexto de execução do dispositivo no momento da reconfiguração.

Para que este objetivo pudesse ser alcançado, esse trabalho realizou as seguintes contribuições:

1. Introduziu-se a noção de restrições computacionais de um componente, permitindo que o programador do componente possa descrever os requisitos mínimos necessários para a execução do componente em um dispositivo móvel. Estas restrições definem o comportamento dos componentes de acordo com o contexto de execução do dispositivo. Assim, o desenvolvedor pode acrescentar ao componente, uma descrição precisa das restrições computacionais para sua implantação. Por exemplo, ele pode informar que o componente só executa com, no mínimo, uma conectividade 3G ou $X\%$ de energia residual.
2. Definiu-se o conceito de contrato de reconfiguração, onde são descritos os pontos de conexão de um componente com uma aplicação ou com outros componentes dos quais ele dependa, junto com suas políticas de adaptação. O novo Kaluana realiza a busca de um componente de acordo com este contrato, buscando minimizar a possibilidade de erros por incompatibilidade de implementações e manter o QoS de execução de uma aplicação.
3. Alterou-se a seleção de componentes do Kaluana original para que seja considerado o contexto de execução do dispositivo na avaliação do componente mais adequado a ser implantado no momento da

reconfiguração de uma aplicação. Desta forma, o *middleware* pode selecionar componentes usando outros critérios de avaliação definidos no algoritmo de seleção. Por exemplo: os componentes *A* e *B* podem possuir o serviço requisitado pela aplicação, tornando-se componentes candidatos. O *middleware* irá então avaliar seus contratos de reconfiguração e, de acordo com o contexto de execução, escolher o mais adequado para a implantação naquele momento.

1.2. Estrutura da dissertação

Este trabalho está organizado em sete capítulos.

O capítulo 2 expõe os fundamentos conceituais para o entendimento do mecanismo, cujo conhecimento é necessário para melhor compreensão desta dissertação.

O capítulo 3 descreve, de forma resumida, o Kaluana original, explicando os conceitos básicos do *middleware* e seu funcionamento.

O capítulo 4 apresenta um levantamento de trabalhos que usam diferentes técnicas implantação dinâmica de componentes, e faz uma comparação com o trabalho desta dissertação.

O capítulo 5 apresenta, em mais detalhes, a implementação das extensões do Kaluana citadas na seção 1.1.

O capítulo 6 apresenta uma série de experimentos realizados para exemplificar e avaliar a implementação realizada para este trabalho.

Por fim, o capítulo 7 apresenta conclusões sobre o estudo desenvolvido nesta dissertação, limitações da implementação e sugestões para trabalhos futuros.