



Allan Alves Valeriano

**Um mecanismo de seleção de componentes para o
middleware Kaluana usando a noção de contratos
de reconfiguração**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para
obtenção do título de Mestre pelo Programa de Pós-
Graduação em Informática da PUC-Rio.

Orientador: Prof. Markus Endler

Rio de Janeiro
Abril de 2011



Allan Alves Valeriano

**Um mecanismo de seleção de componentes para o
middleware Kaluana usando a noção de contratos
de reconfiguração**

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico e Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Markus Endler

Orientador

Departamento de Informática - PUC-Rio

Prof.^a Noemi de La Rocque Rodriguez

Departamento de Informática – PUC-Rio

Prof. Renato Fontoura de Gusmão Cerqueira

Departamento de Informática - PUC-Rio

Prof. José Eugenio Leal

Coordenador Setorial do Centro Técnico Científico – PUC-Rio

Rio de Janeiro, 8 de abril de 2011

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Allan Alves Valeriano

Engenheiro de Computação formado pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) em 2007. Atualmente, integra o grupo de pesquisadores do LAC (Laboratory for Advanced Collaboration) da PUC-Rio, desenvolvendo pesquisa na área de Sistemas Distribuídos.

Ficha Catalográfica

Valeriano, Allan Alves

Um mecanismo de seleção de componentes para o middleware Kaluana usando a noção de contratos de reconfiguração/ Allan Alves Valeriano; orientador: Markus Endler.

87 f. : il. ; 30 cm

Dissertação (Mestrado em Informática)–Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2009.

Inclui bibliografia

1. Informática – Teses. 2. Adaptação dinâmica. 3. Contrato de reconfiguração. 4. Componentes de software. 5. Middleware. I. Endler, Markus. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

CDD:004

À minha amada esposa, Rafaela.

Agradecimentos

À Juliana Aquino, por toda sua paciência e apoio durante o desenvolvimento deste trabalho, sem os quais não seria possível sua conclusão.

Ao Professor Markus Endler, por sua compreensão e orientação durante todo o mestrado.

Ao Nima por fornecer o mapa detalhado do campus da PUC-Rio para ser usado na aplicação exemplo deste trabalho.

Ao CNPq, à PUC–Rio, ao Departamento de Informática e ao Tecgraf, pelos auxílios financeiros concedidos, sem os quais este trabalho não poderia ter sido realizado.

Resumo

Valeriano, Allan Alves; Endler, Markus. **Um mecanismo de seleção de componentes para o middleware Kaluana usando a noção de contratos de reconfiguração.** Rio de Janeiro, 2011. 87 p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A computação móvel cria a necessidade de que as aplicações sejam adaptáveis de acordo com o contexto do usuário. Demandas específicas do usuário, assim como alterações no contexto computacional requerem que aplicações móveis clientes devam ser capazes de se adaptar dinamicamente para se adequar ao novo cenário de execução. Estas adaptações precisam ser apropriadas e devem manter a qualidade de serviço evitando falhas ou degradação do desempenho da aplicação. Este trabalho propõe uma extensão do *middleware* de componentes Kaluana, que provê um mecanismo de seleção de componentes para aplicações adaptáveis a partir de contratos de reconfiguração. Esta seleção é feita com a noção de equivalência entre suas interfaces públicas e considera as restrições de execução dos componentes candidatos a serem usados na adaptação de acordo com o contexto computacional do dispositivo para a avaliação dos componentes candidatos a instanciação. Esta seleção visa manter a compatibilidade dos novos componentes com os componentes já implantados bem como com o contexto de execução, ou seja, o estado corrente dos recursos no dispositivo. Com a noção de equivalência entre as interfaces de especificação de componentes, a aplicação tem a possibilidade de selecionar componentes através da interface do serviço requisitado sem a necessidade de conhecer seu nome ou alguma característica específica, evitando assim que uma amarração seja criada com uma determinada implementação.

Palavras-chave

Adaptação dinâmica; Kaluana; Componentes de software; Middleware.

Abstract

Valeriano, Allan Alves, Endler, Markus (Advisor). **A mechanism of component selection with the notion of reconfiguration contracts.** Rio de Janeiro, 2011. 87 p. MSc. Dissertation - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Mobile computing creates the need for applications to be adaptable according to the user's context. Specific user demands as well as changes in the computational context the mobile applications require clients to be able to adapt dynamically to suit the new scenario of execution. These adjustments should be appropriate and should maintain the quality of service avoiding failures or preventing degradation of application performance. This thesis proposes an extension of Kaluana middleware which provides a mechanism for selection of components for adaptive applications based on the notion of reconfigurations contracts. This selection is done also based on the notion of equivalence between their public interfaces and should consider the execution restrictions of the candidate components to be used in the adaptation in accord to the device's execution context to be used to evaluate the candidate components to the instantiation. This selection aims to maintain the compatibility of new components with the components already used as well as the execution context, i.e. the current status of the device's resources. Due to the notion of equivalence between the interface specifications of components, the application should be able to request a component through the interface of the requested service, avoiding the need to know the component's name or any other specific feature that creates a tie with the given implementation.

Keywords

Dynamic adaptation; Kaluana; Software components; middleware.

Sumário

| | | |
|---------|-----------------------------------|----|
| 1 | Introdução | 12 |
| 1.1. | Motivação e contribuição | 12 |
| 1.2. | Estrutura da dissertação | 14 |
| 2 | Fundamentos | 15 |
| 2.1. | Componentes | 15 |
| 2.2. | Ciência ao contexto | 16 |
| 2.3. | Descrição de interfaces | 17 |
| 2.4. | Compatibilidade de tipos | 19 |
| 2.5. | Descrição de requisitos | 20 |
| 2.6. | Políticas de adaptação | 21 |
| 2.7. | Adaptação dinâmica | 22 |
| 2.8. | Arquiteturas orientadas a serviço | 25 |
| 2.9. | Gerenciamento de serviços | 27 |
| 2.10. | Mobilis | 28 |
| 2.10.1. | Shared Data Manager | 29 |
| 2.10.2. | Context Management Service | 30 |
| 3 | Kaluana | 31 |
| 3.1. | Arquitetura | 31 |
| 3.2. | Ciclo de vida | 33 |
| 4 | Trabalhos Relacionados | 36 |
| 4.1. | CASA | 36 |
| 4.2. | IAM | 38 |
| 4.3. | Chisel | 43 |
| 4.4. | DynamicTAO | 45 |

| | |
|--|----|
| 4.5. Síntese dos trabalhos | 48 |
| 5 Mecanismo de seleção de componentes | 50 |
| 5.1. Restrições computacionais para a execução de um componente | 51 |
| 5.2. Contrato de reconfiguração de um componente | 54 |
| 5.3. Integração do Kaluana com a plataforma Mobilis | 55 |
| 5.4. Registro de componentes no Kaluana | 57 |
| 5.5. Requisição de um componente por uma aplicação | 58 |
| 5.6. Equivalência de componentes | 60 |
| 5.7. <i>Download</i> e instalação de componentes, a partir de um servidor remoto | 61 |
| 6 Resultados experimentais | 64 |
| 6.1. Testes de desempenho | 64 |
| 6.1.1. Busca de componentes | 65 |
| 6.1.2. Requisições múltiplas | 66 |
| 6.2. Aplicação exemplo | 67 |
| 6.3. Casos de uso | 69 |
| 6.3.1. Ator percorre uma área não demarcada | 69 |
| 6.3.2. Ator entra em uma área demarcada | 70 |
| 6.4. Middleware Kamaiurá | 73 |
| 6.5. <i>Guideline</i> de criação de uma aplicação dinâmica usando o Kaluana | 76 |
| 7 Conclusões | 79 |
| 7.1. Limitações | 80 |
| 7.2. Trabalhos futuros | 80 |
| 7.2.1. Uso de ontologias para descrição e seleção de componentes | 81 |
| 7.2.2. Flexibilização do uso de interesses ao contexto | 81 |
| 7.2.3. Flexibilização da descoberta de servidores remotos | 82 |
| 8 Referências Bibliográficas | 83 |

Lista de figuras

| | |
|---|----|
| Figura 1. Representação da composição de dois componentes | 16 |
| Figura 2: Classificação dos tipos de adaptação dinâmica, segundo [12].. | 24 |
| Figura 3. Arquitetura em camadas, da plataforma Mobilis | 29 |
| Figura 4. Arquitetura do <i>middleware</i> Kaluana, segundo [12] | 32 |
| Figura 5. Métodos de reconfiguração de componentes [10]. | 37 |
| Figura 6. Arquitetura global do IAM [58] | 38 |
| Figura 7. Exemplo de um metaobjeto no IAM, usando o <i>RAM MetaObject protocol</i> [58]..... | 39 |
| Figura 8. Camada que trata de requisitos não funcionais, no IAM [58]..... | 40 |
| Figura 9. Organização do <i>framework</i> de observação de recursos no IAM [58]..... | 41 |
| Figura 10. Políticas de adaptação no IAM [58]..... | 42 |
| Figura 11. Linguagem definição de políticas de reconfiguração no Chisel [11]..... | 43 |
| Figura 12. Mecanismo de adaptação do Chisel [11]..... | 45 |
| Figura 13 Mecanismo de especialização do DynamicTAO [65]..... | 46 |
| Figura 14 Arquitetura do DynamicTAO [65] | 47 |
| Figura 15. Definição do contrato de reconfiguração de um componente.. | 55 |
| Figura 16. Registro do consumidor de informações de contexto no Kaluana | 56 |
| Figura 17. Exemplo de registro de um componente no Kaluana..... | 57 |
| Figura 18. Declaração de um componente no manifesto do Android..... | 59 |
| Figura 19. Requisição de um componente por nome..... | 59 |
| Figura 20. Requisição de um componente pelo nome de um serviço necessário | 59 |
| Figura 21 Instalação de um componente novo no dispositivo | 62 |
| Figura 22 Usuário escolhe se deseja usar a aplicação instalada ou não.. | 62 |
| Figura 23 Requisição de componentes sequencialmente | 66 |

| | |
|---|----|
| Figura 24 Requisição paralela de componentes | 67 |
| Figura 25. Componente default, <i>GoogleMapComponent</i> | 68 |
| Figura 26. Requisição de carregamento de um novo componente..... | 69 |
| Figura 27. Mapa detalhado da PUC-Rio | 72 |
| Figura 28. Arquitetura do Kamaiurá sobre o <i>middleware</i> Kaluana..... | 74 |
| Figura 29 Realizando o <i>bind</i> de uma aplicação no serviço de gerenciamento de componentes do Kaluana..... | 77 |
| Figura 30. Registro do <i>receiver</i> de notificação do carregamento dos componentes requisitados. | 78 |