

MIDDLEWARE REFLEXIVO: ESTUDO E IMPLEMENTAÇÃO

VAZ, Frederico Forzani¹; COSTA, Fábio Moreira²

Palavras-chave: Middleware Reflexivo, Computação Móvel, Adaptação Dinâmica, Meta-Objeto.

1. INTRODUÇÃO (justificativa e objetivos)

Uma plataforma de middleware é um software adicional colocado entre as aplicações e o sistema de rede, oferecendo um alto nível de abstração. Assim, tais aplicações não fazem uso direto da interface de programação oferecida pelos Sistemas Operacionais. Do ponto de vista do programador, tem-se acesso apenas às Interfaces de Programação de Aplicação, ou, simplesmente, APIs (*Application Programming Interface*). Estas, por sua vez, irão interagir com o sistema operacional nativo.

Em um ambiente distribuído móvel, uma plataforma de middleware deve considerar a heterogeneidade de tais dispositivos, como Celulares, PalmTops, PocketPCs etc. As diferenças como sistemas operacionais, interfaces com o usuário, poder de processamento e armazenamento de dados, largura de banda, mobilidade etc, devem ser relevadas.

Na computação móvel, o alto grau de mobilidade dos dispositivos móveis pode alterar suas configurações com uma certa frequência. Logo, uma plataforma de middleware para este tipo de ambiente deve ser capaz de reconfigurar-se durante tais modificações, provendo adaptação dinâmica no middleware ao invés de nas aplicações. Para oferecer meios de acesso e alteração da configuração da própria plataforma, uma estratégia seria explicitamente representar suas estruturas e comportamentos: Reflexão Computacional [3].

Este resumo descreve o desenvolvimento da parte reflexiva na plataforma de middleware Meta-ORB4Java [2]. Para assim, dar melhor suporte às necessidades de aplicações distribuídas, sobretudo aquelas que fazem uso de multimídia distribuída e computação móvel.

2. MATERIAL E MÉTODOS (metodologia)

Meta-ORB4Java [2] consiste em um protótipo de plataforma de middleware desenvolvido no contexto de pesquisa do orientador. Consiste em uma plataforma middleware baseada em componentes implementada na linguagem Java, herdando o modelo de middleware especificado em [1].

Como a funcionalidade principal da plataforma será o suporte às aplicações que fazem uso de multimídia distribuída e computação móvel, tem-se preferência pela linguagem de programação Java, devido sua portabilidade de execução em diferentes sistemas. Além de considerar o fato que a plataforma deverá executar em dispositivos móveis, o que implica que as APIs (*Application Programming Interface*) utilizadas no desenvolvimento da plataforma seguirão as limitações de J2ME. O padrão CLDC 1.0 e MIDP 1.0 foi adotado para atender uma maior escala de dispositivos.

3. RESULTADOS E DISCUSSÃO

A plataforma Meta-ORB4Java [2] está estruturada como se segue:

Componentes: componentes são unidades para o encapsulamento de estado e funcionalidade.

Interfaces: a interface define quais serviços estão disponíveis entre um componente e um cliente.

Bindings: os *bindings* conectam duas interfaces de componentes distintos possibilitando sua interação. Existem *bindings* locais, implícitos e explícitos, podendo ser primitivos ou compostos.

Repositório de Tipos: fornece as meta-informações utilizadas para instanciar as estruturas da plataforma.

Servidor de Nomes: fornece nomes únicos aos elementos fundamentais da plataforma, viabilizando localizá-los.

Fábricas: responsáveis pela criação das estruturas da plataforma, existindo duas fábricas: de Componentes e de *Bindings*.

O objetivo deste projeto foi especificado como a construção dos mecanismos responsáveis pela reflexão e adaptação dinâmica na plataforma Meta-ORB4Java [2], para assim torná-la dinamicamente reconfigurável. A estratégia utilizada consiste em implementar os conceitos da Reflexão Computacional [3], ou seja, representando e manipulando as estruturas internas da plataforma. Cria-se, então, o conceito de níveis: o nível base e o meta-nível.

O nível base consiste das funcionalidades técnicas de uma plataforma de middleware, tais como a localização e a comunicação entre objetos remotos. O meta-nível, considerado o objetivo principal deste projeto, corresponde a uma representação explícita destas funcionalidades, que oferece ao programador uma forma de acesso à implementação interna da plataforma, tanto para inspecionar, quanto para alterar (adaptação dinâmica).

Os objetos que povoam o meta-nível são implementados como componentes, assim como as fábricas, possuindo interfaces e classe de implementação dos métodos. É importante citar que o nível-meta está causalmente conectado ao nível-base, implicando que modificações em um destes dois níveis implicarão em modificações correspondentes no outro nível.

Para diminuir a complexidade do meta-nível e seguindo o modelo de reflexão proposto em [1], os meta-componentes são divididos de acordo com o interesse de atuação de cada um, constituindo vários modelos de meta-espço, como citado abaixo:

Interface: permite acesso às estruturas individuais das interfaces dos componentes e *bindings*.

Encapsulamento: permite acesso a uma visão externa dos componentes e *bindings*.

Arquitetura: permite acesso direto às estruturas internas dos componentes e *bindings*, possibilitando alterações.

Interceptação: controla a política de acesso às interfaces dos componentes.

A funcionalidade destes meta-componentes (modelos de meta-espço) está em criar um mecanismo seguro e confiável de acesso ao nível base da plataforma. Para tanto, estes devem estar ligados causalmente aos objetos que estão representando, para assim retornar informações válidas e atuais, além de reconfigurar o objeto em questão de acordo com as alterações do ambiente de execução, caracterizando a adaptação dinâmica.

Os objetos Interface, Encapsulamento e Arquitetura estão implementados, enquanto o objeto Interceptação está em fase de projeto, para garantir o melhor local para inseri-lo dentro da estrutura da Meta-ORB4Java [2]. As aplicações de teste estão sendo implementadas de acordo com evolução de tais objetos, assim como a versão da plataforma destinada a dispositivos móveis.

4. CONCLUSÕES

Sabe-se que existe um crescimento do número dos dispositivos móveis (Celulares, PDAs, PalmTops etc). Com os mesmos, aumenta-se também o grau de heterogeneidade que tais dispositivos apresentam entre si. Portanto, uma infraestrutura capaz de homogeneizar o desenvolvimento de aplicativos, nesse tipo de ambiente móvel, torna-se necessária. Neste ambiente, as aplicações exigem funcionalidades de adaptação dinâmica, que devem ser encontradas em tal infraestrutura: plataforma de middleware.

Este Resumo Expandido descreve a implementação da parte reflexiva da plataforma de middleware Meta-ORB4Java [2], para dá-la a capacidade de adaptação dinâmica. Para garantir tal adaptação, a plataforma mantém uma representação e meios de modificar suas estruturas e comportamento internos, para assim prover alterações na própria plataforma, ao invés de nas aplicações.

5. REFERÊNCIAS BIBLIOGRÁFICAS

[1] COSTA, FÁBIO M.. Combinig meta-Information Management and Reflection in an Architecture for Configurable and Reconfigurable Middleware. PhD thesis, University of Lancaster, 2001.

[2] COSTA, F. M. and SANTOS, B. S. Structuring reflective middleware using meta-information management: The meta-orb approach and prototypes. Journal of the Brazilian Computer Society. Brasil, 10(1):43–58, 2004.

[3] PATIE MAES. Concepts and experiments in computational reflection. In ACM Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA'87), Orlando, FL USA, 1987. American Computer Machinery, ACM Press.

FONTE DE FINANCIAMENTO – CNPq/ITI-A

¹ Voluntário de iniciação científica. Instituto de Informática, frederico@inf.ufg.br

² Orientador. Instituto de Informática/UFG, fmc@inf.ufg.br