
UM SERVIÇO DE MOBILIDADE PARA AGENTES BASEADOS EM PLATAFORMAS CORBA

A MOBILITY SERVICE FOR AGENTS BASED ON CORBA PLATFORMS

O. FALSARELLA¹, C. MENDEZ², I. FONTES³,
P. SILVA³, W. LOYOLLA^{1,3}, M. MENDES¹

ABSTRACT

The number of applications that use mobile agent technology and of systems that support it are constantly increasing. Some applications send mobile software agents across the network to perform a set of tasks. The mobile agents, representing the applications, have autonomy to migrate from a node to another to complete their tasks. The main advantages of the agent to be executed at a remote node are: reduction of communication cost, asynchronous processing, flexibility and load balancing. The execution of an agent during its process of migration requires a supporting environment with installed services at each network node. This paper describes in detail the mobility service for agents. This service is being developed in the context of MAGNA - Mobile AGeNt Architecture and PAGE - Prototyping an AGent Environment projects.

Keywords: Mobile Agents, Agent Platforms, Distributed Agent Environment, Mobility Service, CORBA Platforms.

RESUMO

O número de aplicações que usam tecnologia de agentes móveis e sistemas que os suportam está em constante crescimento. Algumas aplicações enviam agentes de software através da rede para executar um determinado conjunto de tarefas. Agentes móveis, representando as aplicações, têm autonomia necessária para migrar de um nó para outro a fim de completar suas tarefas. As principais vantagens de um agente executar em um nó remoto são: redução do custo de comunicação, possibilidade de processamento assíncrono, flexibilidade e divisão da carga de processamento. A execução de um agente, durante seu processo de migração, requer um ambiente de suporte com serviços instalados em cada nó de rede. Este artigo descreve em detalhe o serviço de mobilidade de agentes. O serviço de mobilidade está sendo desenvolvido no contexto dos projetos MAGNA - Mobile AGeNt Architecture e PAGE - Prototyping an AGent Environment.

Palavras-chaves: Agentes Móveis, Plataformas de Agentes, Ambientes de Agentes Distribuídos, Serviço de Mobilidade, Plataforma CORBA.

(1) Pontifícia Universidade Católica de Campinas (PUCCAMP) Campinas Brasil.

(2) Universidade de Magallanes, Punta Arenas, Chile.

(3) UNESP-FET-Bauru, Brasil, {orandi, loyolla, mendes}@zeus.puccamp.br
{carias, ivo,pss}@dca.fee.unicamp.br

1. INTRODUÇÃO

A tecnologia de agentes móveis vem evoluindo para um novo paradigma que se adequa a uma ampla variedade de domínios de problemas. O número de projetos e novas aplicações usando este paradigma está constantemente aumentando, chamando a atenção da comunidade de pesquisadores para estudá-lo e propor soluções para vários problemas detectados.

Geralmente um agente pode ser uma entidade física ou lógica que é responsável pela execução de um conjunto de tarefas, delegadas a ele por outras entidades (aplicações, outros agentes, usuários humanos, etc.). Os agentes representam estas entidades enquanto executam suas tarefas (eles agem a favor do usuário)[15].

Um dos aspectos dos agentes que é bastante discutido por vários autores [3], [6], [10], [9] é sua capacidade de poder se deslocar na rede, migrando de um nó para outro, a fim de executar sua tarefa. Neste contexto, um agente que está sendo executado em um determinado nó decide autonomamente migrar para um outro nó ou retornar para o seu nó de origem, trazendo consigo o resultado obtido durante sua execução. Agentes com a capacidade e autonomia de migração formam uma classe especial de agentes denominados agentes móveis.

Significantes benefícios e vantagens podem justificar o uso de agentes móveis em certas aplicações, tais como:

- Redução do custo de comunicação [12] - dependendo da quantidade de informações a serem transmitidas pelo agente entre dois nós da rede, pode ser mais eficiente em termos de custo de comunicação enviar um agente para onde estas estão localizadas. Desta forma, o agente pode filtrar e selecionar somente as informações relevantes e transferi-las para o seu nó de origem.;
- Processamento assíncrono [12] - um agente pode ser enviado através da rede, continuando suas tarefas em outros nós. Enquanto o agente se encontra fora de seu nó de origem não é necessário que este nó permaneça em operação. A computação móvel é um exemplo de aplicação que pode utilizar sobremaneira agentes empregando o processamento assíncrono;

- Divisão da carga de processamento - computadores com baixa capacidade de processamento podem ser usados eficientemente para armazenar e processar dados que são filtrados e enviados por agentes móveis a partir de grandes servidores de informações;
- Flexibilidade - Agentes móveis se constituem em uma alternativa para o tradicional modelo cliente-servidor. [5] sugere que o modelo cliente-servidor e o paradigma de agentes móveis podem ser complementares.

Os agentes móveis podem ser utilizados em diferentes domínios de aplicação, por exemplo: comércio eletrônico e empresas virtuais.

O comércio eletrônico envolve um sistema híbrido no qual pessoas e computadores trabalham juntos para obter certos resultados comerciais comuns. Uma transação genérica em comércio eletrônico é um conjunto finito de interações entre membros com diferentes papéis [4].

Agentes móveis podem ser úteis em todas as fases de uma transação como por exemplo:

- agentes podem visitar através da rede empresas e pesquisar por potenciais fornecedores, de acordo com regras e condições pré-estabelecidas por um comprador;
- agentes podem se deslocar para servidores de fornecedores e negociar com eles termos e condições de entrega de produtos e formas de pagamento;
- agentes podem prover pagamento eletrônico viajando pela rede.

As empresas virtuais representam uma aliança temporária de companhias formadas para compartilhar custos com o objetivo de oferecer produtos e serviços competitivos a custos reduzidos. Este conceito de empresa independe do porte organizacional e das fronteiras técnicas e geográficas [11].

Agentes móveis podem ser úteis no contexto das empresas virtuais, auxiliando em vários tipos de tarefa, tais como:

- pesquisar por potenciais parceiros na rede perguntando por companhias que podem cumprir algum requerimento de produção ou executar algum serviço;

- monitorar o status de serviços e recursos que estão sendo executados remotamente para poder reprogramar local ou globalmente fluxos de trabalho;
- viajar para locais remotos e fazer requisições de produtos e serviços oferecidos por parceiros;
- negociar e mediar disputas, pesquisando informações ou serviços sobre assuntos específicos para auxiliar discussões.

Diferentes tipos de serviços devem existir com o objetivo de formar uma infraestrutura computacional necessária para suportar os agentes móveis [15], [5]. O principal objetivo deste artigo é descrever um dos serviços, o serviço de mobilidade, responsável pelo transporte de agentes móveis de um local para outro remoto. A descrição está baseada nos Serviços CORBA definidos pelo Object Management Group [13], [14].

Este artigo apresenta na seção 2 um breve resumo das principais definições e conceitos da tecnologia de agentes móveis. Na seção 3, uma Arquitetura do Ambiente de Suporte a Agentes Móveis que está sendo considerado pelo projeto Magna/Page é apresentada. O DAE (Ambiente de Agentes Distribuído) será utilizado como uma plataforma estruturada que proverá serviços, incluindo o serviço de mobilidade. Na seção 4 é descrito, em detalhe, o serviço de mobilidade e na seção 5 são discutidos os aspectos relativos ao desenvolvimento de um protótipo inicial. Finalmente na seção 6, são apresentadas as conclusões do trabalho.

2. MOBILIDADE DE AGENTES - DEFINIÇÕES

Embora não exista uma definição de consenso entre os pesquisadores [16], [7], [2], [1] sobre exatamente o que é um agente, no nosso ponto de vista uma entidade computacional de software só pode ser considerado um agente quando ela desenvolve suas capacidades com o auxílio de uma plataforma de agentes.

Uma plataforma de agentes é uma infraestrutura de serviços instalados em um nó de rede que suporta todo ciclo de vida de um agente.

[8] define um conjunto de capacidades que um agente pode possuir. Dentre elas, duas estão diretamente relacionadas com o contexto de migração:

- *mobilidade* ou poder de migração que um agente pode ter;
- *autonomia* ou poder de auto-controle de um agente.

Uma das propriedades mais atrativas de um agente é a sua capacidade de se mover de um lugar para outro autonomamente [5], [8], [3], [12]. Diferentes tipos ou gradações de mobilidade podem existir:

- *Estático*: agente que só será executado na própria plataforma em que foi originalmente criado;
- *Execução Remota*: agente que é enviado para um nó remoto para executar suas tarefas. Este agente inicia sua execução no local remoto sempre a partir do seu ponto inicial;
- *Migração (com estado)*: agente que pode parar sua execução em um nó, migrar para qualquer outro nó e continuar sua execução do ponto onde parou antes da migração. Para isso ele deve levar consigo seu estado. Entende-se por estado do agente todas as informações necessárias (valores de variáveis, próxima instrução a ser executada, valores de parâmetros, etc.) para que o mesmo possa dar continuidade à sua tarefa no novo nó.

Nos casos de agentes estáticos ou de execução remota, o agente uma vez instalado ou transportado para o nó específico, reside neste até terminar a sua execução. Em ambos os casos, não existe a necessidade de transportar estado.

No caso de migração, existe um processo de movimentação de um nó da rede para outro permitindo-se que o agente execute sua tarefa progressivamente. O mais importante é que neste processo, quando o agente termina sua tarefa em um nó, ele autonomamente decide se deve ou não migrar para outro nó.

3. ARQUITETURA DO AMBIENTE DE SUPORTE A AGENTES MÓVEIS

A figura 1 apresenta a arquitetura do ambiente de suporte a agentes móveis. O Ambiente de Agentes Distribuído (DAE) representa a plataforma do projeto Magna/Page [15], [5], na qual, agentes móveis construídos de acordo com os princípios da estrutura do Magna/Page têm seus ciclos de vida (*lifecicle*), i.e., criação, operação, gerenciamento, etc.

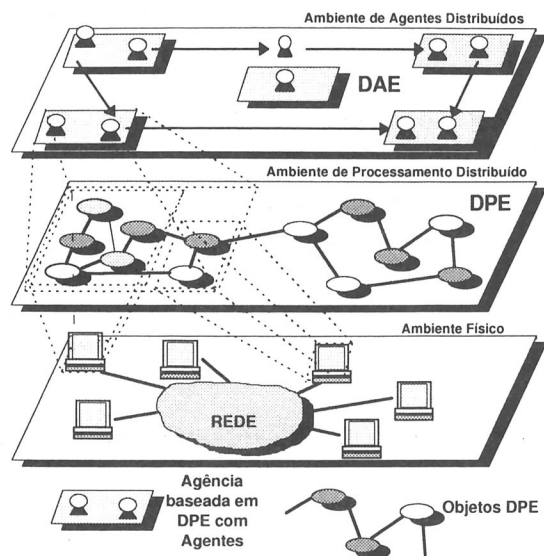


Figura 1 - Ambiente de Agentes Distribuídos

O DAE conta com as funcionalidades proporcionadas pelo Ambiente de Processamento Distribuído (DPE).

O DPE deve ser um ambiente de processamento distribuído compatível com CORBA ou TINA-C e proporciona a transparência para as interações entre clientes e servidores, e um conjunto de serviços genéricos que suportam aplicações distribuídas, tais como: *serviço de nomes (naming service)*, *serviço de persistência (persistence service)*, *serviço de repositório (repository service)*, *serviço de transações (transaction service)*, entre outros.

Portanto, o DAE é uma aplicação distribuída baseada nos serviços oferecidos pelo DPE, i.e., um sistema de software complexo constituído de múltiplos componentes de software que interagem entre si e são suportados por um DPE.

Dentro desta arquitetura, os componentes distribuídos do DAE são denominados *agências*, referindo-se deste modo, ao fato de que seu propósito primário é abrigar e suportar *agentes*. Um ambiente distribuído de agentes típico consistirá de um grande número de agências separadas, que podem ser agrupadas em *domínios* seguindo-se alguma forma de agrupamento, como por exemplo: *propriedade, política de segurança, conectividade direta* ou *requerimentos comuns*.

No ambiente físico (hardware e sistema operacional nativo), uma agência é mapeada diretamente para um nó de rede. Contudo, as agências podem ser expandidas em um conjunto de nós, cada

um deles suportando parcial ou totalmente os serviços providos pelo DAE. As agências oferecem uma interface comum para os agentes, escondendo desta forma a localização exata dos seus serviços.

4. SUPORTE A MOBILIDADE NO MAGNA/PAGE

O propósito é construir uma plataforma para dar suporte à mobilidade de agentes baseada nos serviços da OMG OMA (*Object Management Group - Object Management Architecture*) [14]. A arquitetura OMA já tem serviços especificados (e.g., *Life Cycle, Externalization, Naming, etc.*) que podem ser utilizados. Ela suporta um modelo Cliente/Servidor usando especificações IDL (*Interface Definition Language*), que permitem que as partes Cliente e Servidor sejam programadas em diferentes linguagens de forma a interagirem transparentemente. Além disso, oferece uma arquitetura aberta e orientada a objetos.

Conseqüentemente, um agente móvel será um objeto CORBA. Portanto, ele pode aceitar invocações ORB, principalmente a invocação de *start*, quando o agente deve ser iniciado em uma nova agência.

O serviço de mobilidade descrito neste texto proporciona *migração*. Deste modo a função do serviço de mobilidade é transportar um agente de uma agência de origem para outra de destino. A semântica associada com este movimento é tal que no momento em que o agente deseja ir para uma outra agência, ele suspende a sua execução na agência de origem e solicita o seu transporte para a agência de destino, onde continua sua execução, reiniciando a partir do ponto em que parou. Este processo requer que o *estado* do agente seja salvo, exatamente no momento em que este solicita ser transportado. Para suportar este processo o serviço de mobilidade necessita de outros serviços:

- *Serviço de Persistência de Agente*, para armazenar os dados e o estado de execução do agente;
- *Serviço de Segurança de Agente*, para realizar as ações necessárias de segurança antes, durante e após o transporte;
- *Serviço de Repositório de Implementação de Agentes*, para armazenar o código do agente;
- *Serviço de Instanciação de Agentes*, para instanciar o agente na agência destino e colocá-lo em execução.

Para suportar a execução remota o serviço de mobilidade necessita dos mesmos serviços que a migração, exceto o Serviço de Persistência, uma vez que não é necessário que agente salve seus dados e seu estado de execução, para retomá-lo novamente.

DISCUSSÃO

Uma característica típica associada com a mobilidade é a heterogeneidade (de hardware e software) dos lugares para os quais os agentes irão e serão executados. Isto impõe dois problemas básicos: Como executar agentes em máquinas heterogêneas? Como salvar o próximo estado a ser executado (estado de execução) quando o agente decide migrar, de tal forma que o agente possa continuar sua execução a partir do ponto que parou?

O primeiro problema pode ser resolvido usando linguagens baseadas em máquinas virtuais, que são portáteis entre diferentes plataformas (e.g. Java bytecode).

O segundo problema pode ser resolvido representando o estado de execução em um arquivo de estados, i.e., o agente executa a partir do início quando é reiniciado e decide como proceder seu trabalho consultando os valores das variáveis existentes neste arquivo. Estes valores são mantidos persistentes durante a migração. Esta solução introduz uma restrição para a programação do agente, uma vez que o programador tem que se preocupar com a estrutura do código, para poder recuperar os estados anteriores.

ARQUITETURA DO SERVIÇO DE MOBILIDADE

A arquitetura geral do serviço de mobilidade é apresentada na figura 2 onde são mostrados os componentes do Serviço de Mobilidade de Agentes, as relações entre eles e outros serviços que são necessários para suportar a migração de agentes.

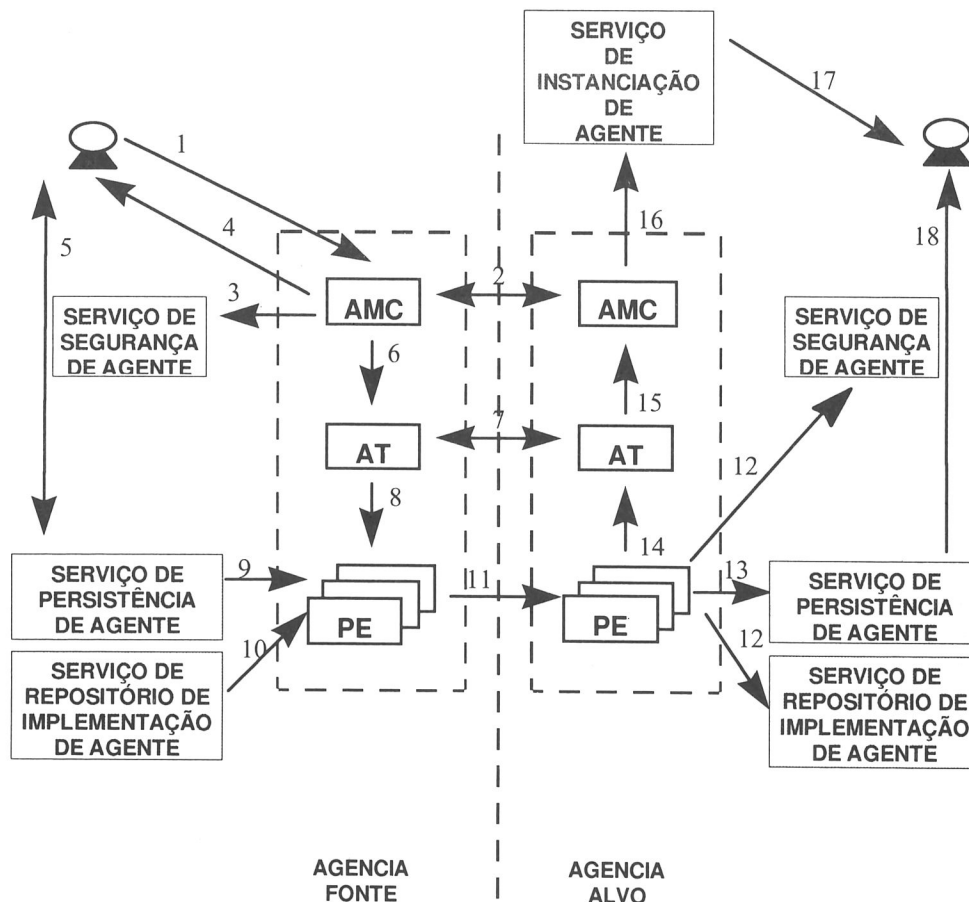


Figura 2 - Arquitetura Geral do Serviço de Mobilidade de Agentes

O Serviço de Mobilidade de Agentes proposto considera três componentes principais: *Coordenador de Mobilidade de Agente* (AMC), *Transporte de Agente* (AT) e *Entidade Protocolo* (PE), que serão explicados a seguir:

a) AMC (Coordenador de Mobilidade de Agente)

Este componente proporciona uma interface ao agente, através da qual o agente pode fazer um pedido para ser transportado. O pedido de transporte é feito pela invocação de uma operação *go*. Esta operação tem o nome da agência destino e os parâmetros identificadores do agente. A partir da solicitação, o AMC coordena os passos necessários para atingir o objetivo, ou seja, o AMC negocia com o seu parceiro da agência destino as condições para enviar o agente e controla as interações entre os módulos locais e serviços (na agência de origem) para transportar o agente. O AMC também solicita ao agente para externalizar seu estado e ao Serviço de Segurança de Agente para que sejam criadas as credenciais do agente. Na agência destino o AMC utiliza as credenciais para realizar o processo de autenticação. Quando o AMC recebe o agente, e após autenticá-lo, passa-o para o Serviço de Instanciação de Agente.

b) AT (Transporte de Agente)

Após o agente estar com seu estado persistente e empacotado, ele está pronto para ser transportado para a agência destino. O AT proporciona a funcionalidade para efetivamente transportá-lo para a agência desejada. Este componente faz com que o processo de transmissão e recepção seja transparente para o componente AMC. Parceiros ATs podem negociar o protocolo de transporte específico a ser usado (vários meios de transporte entre agências podem ser suportados). A negociação entre ATs parceiros é feita através de interações CORBA. No nó remoto, o AT diz ao AMC que o agente chegou e o envia para a autenticação.

c) PE (Entidade de Protocolo)

Vários componentes PE diferentes podem ser suportados pela plataforma. Cada componente PE proporciona ao AT acesso a um meio específico de transporte (FTP, HTTP, E-MAIL, ORB, etc.), porém

através de uma interface comum. Um componente PE, usando os serviços de Persistência de Agente e Repositório de Implementação de Agente, usa o estado do agente e o seu código para empacotá-lo e o envia com as informações de controle (tal como a linguagem na qual o agente foi programado, suas credenciais e sua identificação). Na agência destino o PE realiza o processo inverso. Componentes PE parceiros executam o protocolo escolhido para transmitir o agente.

Para suportar o ciclo completo de migração, o serviço de mobilidade necessita dos serviços: Instanciação de Agente, Persistência de Agente, Repositório de Implementação de Agente e Segurança de Agente, os quais possuem as seguintes funcionalidades:

a) Serviço de Instanciação de Agente

Quando o AMC recebe um agente do AT, o AMC usa este serviço para instanciar o agente. O serviço de Instanciação de Agente seleciona a máquina virtual específica baseada na linguagem na qual o agente foi programado (declarado pela informação de controle anexada ao agente). Após ser instanciado o agente está pronto para ser executado.

b) Serviço de Persistência de Agente

Antes da transmissão, a execução do agente e os dados de estado necessitam ser feitos persistentes (armazenados em um meio não volátil) na agência de origem. Na agência destino, o estado do agente precisa ser internalizado (o estado de execução será recuperado antes do agente iniciar sua execução no novo nó). O serviço de Externalização que engloba externalização e internalização [14], especificado pela OMG OMA, satisfaz estes requisitos.

c) Serviço de Repositório de Implementação de Agente

O Serviço de Repositório de Implementação de Agente oferece funcionalidades para armazenar e recuperar os arquivos de código associados com o agente. Os PEs usam este serviço para recuperar códigos dos agentes a serem transmitidos e para armazenar os agentes recebidos. Este serviço está

fortemente associado com o Repositório de Implementação do ORB, um local onde o ORB armazena e ativa os objetos de implementação.

d) Serviço de Segurança de Agente

O Serviço de Segurança de Agentes apresenta as funcionalidades necessárias à autenticação de agentes, tais como a geração de credenciais e a correspondente autenticação. Este serviço também inclui, dentre outras funções, as de verificação da existência de vírus nos agentes que chegam a uma agência, bem como realiza o processo de criptografia de parte do agente quando solicitado.

Para explicar melhor as relações entre os componentes e os serviços apresentamos uma sequência descritiva do processo de mobilidade. Assumimos que todos os passos são executados com sucesso. A sequência de passos é representada pelos números na figura 2:

1. O agente decide se mover e invoca a operação *go* oferecida pelo AMC.
2. O AMC comunica-se com o seu parceiro, para obter permissão para que o agente possa ser transportado para agência destino.
3. O AMC solicita ao serviço de segurança de agentes a geração de credencias para o agente.
4. O AMC solicita o processo de externalização do agente.
5. O Agente externaliza seu estado.
6. O AMC ordena ao AT para transmitir o agente.
7. O AT negocia com seu parceiro um protocolo específico a ser usado.
8. O AT seleciona o PE correspondente ao protocolo de transporte negociado.
9. O PE selecionado prepara o pacote a ser enviado. Primeiramente são utilizados os dados de estado do agente armazenados pelo Serviço de Persistência de Agente.
10. A seguir, o PE solicita ao Serviço de Repositório de Implementação de Agente o código do agente, e codifica o pacote juntamente com o estado do agente, as informações de controle (tais como, identificação do agente e a linguagem na qual o agente foi programado) e as credenciais geradas pelo Serviço de Segurança de Agente.
11. O agente (pacote) é transmitido entre os PEs, usando o protocolo de transporte que o PE suporta (escolhido pelos ATs).
12. Na agência destino, o PE recebe o pacote e o decodifica. Após a autenticação das credenciais do agente, a parte que corresponde ao código do agente é armazenada pelo Serviço de Repositório de Implementação de Agente. Neste ponto, o código do agente se torna disponível e pode ser executado posteriormente.
13. O PE armazena a parte correspondente ao estado do agente através do Serviço de Persistência de Agente.
14. O PE termina seu trabalho notificando o AT sobre a chegada de um agente passando-lhe sua identificação.
15. O AT notifica o AMC que um agente chegou.
16. O AMC solicita ao Serviço de Instanciação de Agente que instancie o agente.
17. O serviço de Instanciação de Agente instancia o agente identificado de acordo com a linguagem na qual o mesmo foi programado.
18. O agente inicializa o seu estado e continua sua tarefa na agência destino.

Estes serviços proporcionam a funcionalidade mínima para o suporte à migração de agentes.

A migração de um agente, entretanto, envolve a interação com um conjunto de outros serviços da plataforma que não foram considerados acima. Alguns destes serviços são Policy Service, Query Service, Trader Service, Management Service, Communication Service.

ESTRUTURA DO AGENTE

Normalmente, quando um agente está realizando sua tarefa ele atua como um cliente acessando

outros objetos CORBA, mas quando um agente necessita se mover em um ambiente DPE ele precisa ser um objeto CORBA, uma vez que a agência o acessará através de interfaces IDL que o agente deve possuir.

A partir da descrição acima (sequência de passos apresentados na figura 2) vemos que o agente precisa ser externalizado e iniciado pela agência. No momento estas são as únicas operações que a agência precisa ter para acessar o agente. Portanto, o agente deve ter interfaces necessárias destes serviços a serem ativadas pela agência.

Considerando que a externalização é um serviço CORBA, este proporciona a interface *Streamable* que deve ser incorporada ao objeto (o agente em nosso caso) a ser externalizado.

Iniciar um agente significa que este necessita uma interface que proporcione esta operação. Chamamos este serviço de Controle de Execução de Agente, cuja interface oferece dois métodos: *start* e *stop*. Não necessitamos das operações de *suspend* e *resume* no caso da migração uma vez que para o nosso propósito este mecanismo já se encontra situado no próprio código do agente.

Obviamente, podemos incorporar outros serviços ao agente, especialmente serviços CORBA tal como *Lifecycle* para controlar a existência do agente, um serviço de *Property* para classificar os agentes, etc.

5. IMPLEMENTAÇÃO

O desenvolvimento de um protótipo para a plataforma se encontra em fase de codificação e testes, tendo sido especificadas as interfaces IDLs correspondentes aos diversos módulos que compreendem os serviços que a compõe. O sistema está sendo desenvolvido sobre uma plataforma ORBIX (CORBA) e linguagem de programação C++. Como primeira etapa, já foram realizados os testes para o serviço de execução remota e encontra-se em fase preliminar a implementação do serviço de migração e dos demais serviços básicos de suporte. Também estão sendo convertidos para o ambiente *Java/OrbixWeb* os módulos já implementados.

6. CONCLUSÕES

Nosso propósito foi demonstrar a estrutura de um agente e como ele se relaciona com outros serviços existentes em uma plataforma de agentes com o objetivo principal de dar suporte à mobilidade. Com relação a maneira proposta de se estruturar o serviço de mobilidade de agentes, pode-se dizer que esta é uma das alternativas. É evidente que outras possibilidades podem ser consideradas, por exemplo, mandar o agente sem aviso prévio, transportar agentes sem negociação do protocolo de transmissão, etc.

Nossa abordagem impõe uma restrição na estrutura do código do agente, uma vez que para que haja transparência na obtenção do seu estado de execução é necessário que sejam feitas modificações no ambiente de execução para o qual o agente foi criado (e.g. no caso de Java seria necessário mudar o compilador ou a máquina virtual). Estas modificações podem introduzir problemas de manutenção da plataforma devido, por exemplo, o lançamento de novas versões de compiladores ou máquinas virtuais.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Coen, M. 'The Sodabot Slide Show, <http://www.ai.mit.edu/people/sodabot/slideshow/P001/htm>
- [2] URL http://www.ee.mcgill.ca:80/~belmarc/agent_faq.html
- [3] IBM Intelligent Agents Home Page, 'The Role of Intelligent Agent in the Information Infrastructure', <http://www.raleigh.ibm.com/iag/iagpct2.htm>.
- [4] Klein, S. and Langenohl, T, Electronic Markets: An Introduction, In Proc. of Information and Communication Technologies in Tourism, Springer Verlag, Wien, 1994.
- [5] Krause, S., 'MAGNA - A DPE-based Platform for Mobile Agents in Electronic Service Markets', ISADS '97, Berlin, April 1997 (Submitted paper).
- [6] Lange, D. B., 'Agent Transfer Protocol ATP/O.1 Draft', <http://www.trl.ibm.co.jp/aglets>.
- [7] Lingau, A. and Drobnik, O., 'An Infrastructure for Mobile Agent: Requirements and Architecture',

- <ftp://ftp.tm.informatik.uni-frankfurt.de/orlando.ps>, 1995
- [8] M.Mendes et.all , 'Agents skills and their roles in mobile computing and personal communications', IFIP 96, Camberra, September 1996 (accepted paper).
- [9] M. mendes et.all, 'Architectoral Considerations About Open Distributed Agent Support Platforms', ISADS '97, Berlin, April, 1997 (accepted invited paper).
- [10] Project Mole, <http://www.informatik.uni-stuttgart.de/ipvr/vs/alt/projekte/mole.html>
- [11] National Industrial Information Infrastructure Protocols Consortium, Reference Architecture: Concepts and Guidelines, Stamford, CT, USA, 1996.
- [12] Nwana, H.S. 'Software Agents: An Overview', <http://www.cs.umbc.edu/agents/introduction/ao>
- [13] Object Management Group, Common Facilities RequestforProposal3, available at <http://www.omg.org>.
- [14] Object Management Group, Corba Services Specification, available at <http://www.omg.org>.
- [15]PAGE Project - Prototyping an Agent Platform Environment, Projeto de Cooperação entre Instituto de Informática da PUCCAMP e GMD/FOKUS/ Technical University Berlin, Germany.
- [16] Smith, D. C.; Cypher, A.; Spoher, J.; KIDSIM: 'Programming Agents Without a Programming Language', Communications of the ACM, 37(7):55-67,1994

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS
INSTITUTO DE INFORMÁTICA**

CHAMADA DE TRABALHOS

REVISTA DO INSTITUTO DE INFORMÁTICA

ARTIGOS TÉCNICOS - CIENTÍFICOS - OPINIÕES TENDÊNCIAS

MAIORES INFORMAÇÕES E SOLICITAÇÃO DO FORMATO PARA
APRESENTAÇÃO DE TRABALHOS

informat@zeus.puccamp.br

ou

Correspondência: Rodovia D. Pedro I km 136
Caixa Postal 317 CEP 13020-904 Campinas SP.
BRAZIL

Att. Marilda dos Reis Gomes

Tel.: (019) 754-7094 Srtª Marilda
(019) 754-7195 Prof. Frank Behrens

Revista do Instituto de Informática

Publicação Semestral do Instituto de Informática

PUCCAMP

NORMAS AOS COLABORADORES

- 1 - Serão aceitos trabalhos técnicos, científicos, tendências e opiniões;
- 2 - Os artigos deverão conter, sequencialmente:
 - Título em português;
 - Título em inglês;
 - Nome(s) do(s) autor(es) (colocados por extenso, o último nome maiúsculo, e seguidos de*, para especificações profissionais do(s) autor(es) no rodapé da primeira página);
 - Abstract (máximo de 200 palavras);
 - Resumo;
 - Introdução (precedida do número 1);
 - Corpo do artigo (itens numerados sequencialmente a partir do número 2);
 - Conclusões;
 - Referências Bibliográficas (conforme utilizado nesta edição);

Desenhos, gráficos e fotografias serão denominados Figuras, numerados sequencialmente (algarismos arábicos) e constantes do corpo do trabalho. As tabelas serão denominadas Tabelas, numeradas sequencialmente (algarismos arábicos) e constantes do corpo do trabalho;
- 3 - Os trabalhos, digitados com, no máximo, 30 000 caracteres (aproximadamente dez páginas), deverão ser elaborados em WinWord 2.0 ou WinWord 6.0 (acompanhados de folha de estilos);
- 4 - Os trabalhos poderão ser apresentados em três cópias impressas, devendo constar a identificação do(s) autor(es) em uma folha a parte, para permitir a avaliação dos mesmos pelo Conselho Editorial. Os trabalhos, uma vez aceitos, deverão ser encaminhados como descrito no item 3.;
- 5 - Os trabalhos podem ser enviados em disquete ou pela rede, anexos a uma mensagem indicando o editor de textos usado;
- 6 - Os trabalhos serão publicados após pareceres favoráveis de membros do Conselho Editorial da Revista;
- 7 - Quaisquer outros esclarecimentos poderão ser feitos pelo Conselho Consultivo da Revista;

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS

Grão-Chanceler: *D. Gilberto Pereira Lopes*

Magnífico Reitor: *Prof. Gilberto Luiz M. Selber*

Vice-Reitor para Assuntos Administrativos: *Prof. Alberto Martins*

Vice-Reitor para Assuntos Acadêmicos: *Pe. José Benedito A. David*

Diretora do Instituto de Informática: *Profª Angela M. Engelbrecht*

Vice-Diretor do Instituto: *Prof. José Oscar Fontanini de Carvalho*

