Utilizando o Paradigma de Programação Remota em Aplicações Multimídias Distribuídas *

Marcelo Gonçalves Rubinstein
rubi@gta.ufrj.br

Otto Carlos Muniz Bandeira Duarte
otto@gta.ufrj.br

Grupo de Teleinformática e Automação (GTA)
Universidade Federal do Rio de Janeiro
COPPE/EE - Programa de Engenharia Elétrica/Departamento de Eletrônica
Caixa Postal 68504 - CEP 21945-970 - Rio de Janeiro - RJ - Brasil
FAX: + 55 21 2906626

Resumo

Com o crescimento da Internet, a grande quantidade de informação disponível pelo mundo, o custo de movimentação de dados e a pequena experiência de muitos usuários em computação têm sido motivo de preocupação por parte da comunidade de redes. Uma possível solução para estes problemas consiste na utilização de agentes. De modo a possibilitar a mobilidade e a interação entre um agente e seu hospedeiro, uma infra-estrutura de agentes é necessária. Esta infra-estrutura deve permitir flexibilidade em relação à implementação de agentes, aos acessos de agentes a hospedeiros e às comunicações entre agentes. Este trabalho descreve características básicas e atributos de agentes, apresenta os agentes móveis e suas principais aplicações e descreve componentes importantes de uma infra-estrutura de agentes, destacando as infra-estruturas existentes.

Abstract

With the Internet's growth, the great quantity of information available in the world, the data moving cost, and the little experience in computation of many users have been reason of preoccupation by the network community. A possible solution for these problems may be agent utilization. In order to allow mobility, and support for agents on a particular host, an agent infrastructure is necessary. This infrastructure may permit flexibility concerning agent implementation, agent host access, and agent communication. This work describes basic agent characteristics and attributes, presents mobile agents and their main applications, and describes important components of an agent infrastructure, pointing out existing infrastructures.

*Este trabalho foi realizado com recursos da UFRJ, FUJB, PROTEM-CC, CNPq, CAPES e COFECUB.

1 Introdução

Com o extraordinário desenvolvimento da rede mundial de computadores, a Internet, três fatos principais têm sido motivo de grande preocupação por parte da comunidade de redes e também de desenvolvedores de aplicativos: a quantidade de informação disponível pelo mundo, o custo de movimentação de dados e a pequena experiência em computação de muitos usuários. A quantidade de informação disponível em diferentes sítios por todo o mundo não tem sido acompanhada pela capacidade de um usuário de procurar, selecionar e recuperar estas informações. Além disso, a movimentação de grandes quantidades de dados através da rede possui um custo significativo. Por último, o número de usuários com pouco conhecimento em computação tem crescido e deverá continuar a crescer de forma muito rápida.

Uma possível solução para os problemas citados anteriormente consiste na utilização de agentes. Os agentes podem mover-se para o lugar onde os dados estão realmente armazenados e, utilizando inteligência, selecionar as informações que o usuário necessita. Com isso, a quantidade total de dados das mais diversas fontes não é transferida, economizando-se banda passante, tempo e dinheiro. Os agentes também facilitam a vida dos usuários com pequeno conhecimento de computação, funcionando como interfaces com o usuário "espertas" que podem detectar e resolver problemas e aprender as preferências e hábitos do usuário, automatizando ações do cotidiano.

O modelo cliente-servidor tem sido muito utilizado em sistemas distribuídos. Neste modelo, os sistemas são divididos em duas componentes (cliente e servidor) que assumem papéis fixos. Um servidor provê um conjunto de serviços enquanto que um cliente requisita estes serviços. Na maioria dos casos, o cliente e o servidor são executados em máquinas diferentes, o que significa que a interação cliente-servidor requer comunicação através da rede e que os eforços e custos de desenvolvimento e manutenção de clientes em diversas plataformas são aumentados. A chamada remota de procedimentos (*Remote Procedure Call* - RPC) é o mecanismo mais utilizado para a comunicação entre clientes e servidores e é baseada em conexões síncronas entre um cliente e um servidor, onde o cliente mantém o estado completo do processo até que receba a resposta do servidor. Isto leva a uma utilização ineficiente dos recursos do computador e da rede e aumenta os custos da rede.

O modelo de programação remota, uma nova forma de interação entre clientes e servidores, é normalmente utilizado por agentes no lugar do RPC para melhorar o desempenho, através da eliminação do tráfego intermediário na rede. Neste modelo, um agente representando um cliente pode ser transportado para um hospedeiro remoto a fim de interagir com o(s) servidor(es) deste hospedeiro, realizando suas tarefas localmente. Existem duas vantagens principais em se utilizar este modelo. Uma vantagem é que somente resultados selecionados de cálculos ou pedidos complexos serão enviados, diminuindo o tráfego na rede pois não haveria necessidade de transmitir comandos, todos os dados e resultados intermediários. Outra vantagem é que como um agente sabe como acessar os dados que necessita utilizar no hospedeiro, para realizar sua tarefa, o hospedeiro pode ser simplificado pois não necessita necessariamente reconhecer uma linguagem complexa de pedidos ou saber como realizar cálculos complexos.

O estudo de agentes engloba várias áreas tais como sistemas distribuídos, recuperação de informações, comércio eletrônico e inteligência artificial [1] e com isso, existem muitas definições para agentes. Para pesquisadores de Inteligência Artificial (IA), agentes são programas de computadores que simulam uma relação humana, fazendo algo que outra pessoa poderia fazer por você [2]. Para a área de Sistemas Distribuídos/Engenharia de Software, agentes são um meio de superar as incompatibilidades de interfaces com o usuário, formatos de dados e protocolos em uma rede de computadores [2]. Para os pesquisadores do Grupo Telematik, da Universidade Johann Wolfgang Goethe, um agente é um programa que ajuda um usuário a realizar alguma tarefa (ou um conjunto de tarefas), possivelmente através da manutenção de um estado e da comunicação com seu dono, outros agentes ou seu ambiente em geral [2]. Os agentes podem realizar trabalhos rotineiros para usuários ou ajudá-los em tarefas complicadas [1]. Estes também podem "mediar" programas incompatíveis e então gerar novas soluções modulares e orientadas ao problema. Para Gilbert e Janca [3], agentes são programas que ajudam pessoas e agem em interesse destas. Basicamente, a definição de agentes a ser utilizada neste trabalho consiste em programas que ajudam um usuário a realizar tarefas, agindo em interesse desse usuário.

Vários sistemas baseados em agentes existem hoje em dia, como o TACOMA (*Tromsϕ And COrnell Moving Agents*) [4], o Agent Tcl [5] e o Telescript [6]. No projeto TACOMA, foi desenvolvido um protótipo de plataforma de execução com um mecanismo flexível para a movimentação e a distribuição de agentes em redes de computadores. O Agent Tcl é um sistema de agentes transportáveis que dá suporte a várias linguagens e mecanismos de transporte, provê migração e comunicação transparentes e provê segurança efetiva e tolerância a falhas. O Telescript é uma linguagem orientada a objetos que contém facilidades para a construção de aplicações baseadas em agentes. Esforços de padronização também estão sendo realizados pela FIPA (*Foundation for Intelligent Physical Agents*) [7] e pela Agent Society [8]. A FIPA é uma associação internacional de companhias e organizações criada para produzir especificações de tecnologias genéricas de agentes. Esta associação desenvolveu uma arquitetura, uma lista de aplicações, uma lista de requisitos e uma lista de tecnologias cujas padronizações são necessárias para tornar as aplicações disponíveis. A Agent Society é uma organização internacional industrial e profissional estabelecida para ajudar no desenvolvimento de tecnologias de agentes. Suas atividades são facilitar a transferência de colaboração e tecnologia através da coleta e troca de informações de desenvolvimento de agentes e desenvolver interoperabilidade entre agentes e padrões abertos de agentes.

Este trabalho está organizado da seguinte maneira: na seção 2 são apresentados os princípios básicos dos agentes, a seção 3 apresenta os agentes móveis, a seção 4 descreve uma infra-estrutura de agentes e na seção 5 são apresentadas as considerações finais.

2 Princípios Básicos

As características básicas de agentes e seus atributos, que os diferenciam de programas em geral, são apresentados a seguir.

2.1 Características de Agentes

Os agentes são encarregados de realizar tarefas e podem operar sobre requisitos fornecidos. Para executar estas tarefas, os agentes podem utilizar quatro capacidades: agência, inteligência, mobilidade e interface com o usuário [3].

A **agência** é o grau de autonomia e autoridade investida no agente, medida através de sua interatividade. Existem três níveis de agência: interatividade com usuário, interatividade com aplicação e interatividade com agente. A interatividade com usuário indica um agente auto-suficiente que interage somente com o usuário. A interatividade com aplicação indica interação com o usuário e com as aplicações. A interatividade com agente indica interação com o usuário, as aplicações e outros agentes.

A **inteligência** é o grau de raciocínio e comportamento aprendido: a habilidade do agente em aceitar as metas do usuário e realizar a tarefa a ele conferida. Existem quatro níveis de inteligência: preferência, raciocínio, planejamento e aprendizado. No mínimo, deve existir alguma declaração de preferências, talvez na forma de regras, com uma máquina de inferência ou algum outro mecanismo de raciocínio para agir sobre estas preferências. Níveis maiores de inteligência incluem um modelo do usuário ou alguma outra forma de entender e raciocinar a respeito do que o usuário quer que seja feito. Além disso, o agente pode planejar os meios de alcançar este objetivo. Níveis mais altos de inteligência podem ser obtidos através de sistemas que aprendem e se adaptam ao seu ambiente, ambos em termos dos objetivos do usuário e dos recursos disponíveis ao agente.

A **mobilidade** corresponde ao grau de movimento dos agentes através da rede. Existem dois níveis de mobilidade: fixa e móvel. Um agente pode ser fixo, residindo na máquina cliente (por exemplo, para gerenciar uma interface com o usuário) ou instanciado no servidor. Um agente pode ser móvel: criado em uma máquina e enviado para outra para execução em um ambiente seguro. Um agente móvel pode ser transportado de uma máquina para outra no meio de sua execução e carregar com ele dados acumulados sobre o seu estado.

A **interface com o usuário** é utilizada quando existe uma interação direta do usuário com o agente.

2.2 Atributos de Agentes

A idéia de delegação separa agentes de programas em geral, pois a delegação passa algum controle do iniciador para o agente [3]. Os agentes são incorporados a aplicações como correio eletrônico, gerenciamento de rede e ferramentas de computadores pessoais. Estes diferem-se de qualquer outro tipo de programas devido às seguintes características [3]:

- delegação: o usuário encarrega o agente de lidar com uma parte ou com toda a ação;
- personalização: o usuário determina como o agente interage. Em muitos casos, o agente aprende sobre o usuário e adapta suas ações de acordo;
- sociabilidade: o agente é capaz de interagir com outros agentes de maneira similar à comunicação entre pessoas. Isto inclui alguns graus de "dar e receber", flexibilidade e comportamento orientado a metas;
- predição: o usuário tem uma expectativa razoável dos resultados;
- mobilidade: a habilidade de ir normalmente para a rede para realizar a tarefa a ele delegada;
- custo real: os benefícios obtidos pelo usuário (tempo, informação, filtragem etc.) devem ser maiores que o custo (monetário, tempo, reconfecção etc.);
- habilidade: o agente tem sua própria destreza. Um agente simples pode ser capaz de somente executar um comando simples que não contém ambigüidade, por exemplo: "ligue o computador quando eu receber uma mensagem" ou, por outro lado, um agente pode ser capaz de efetivamente lidar com ambigüidade, como no comando "ache a melhor informação sobre o Rio de Janeiro";
- requisitos: podem ser simples como "ache-me um terno, mas não compre-o" ou complexos como "vá para as fontes apropriadas de informação, pois o acesso a uma fonte de dados é cobrado". Alguns serviços de informação, por exemplo, permitem ao usuário estabelecer a quantidade máxima de dinheiro a ser gasta em qualquer busca.

3 Agentes Móveis

Agentes móveis, também chamados itinerantes, são agentes criados para executar suas tarefas, para seus usuários, em diferentes computadores da rede. Estes agentes podem viajar para agências na quais irão apresentar suas credenciais e obter acesso a serviços e dados gerenciados pelas agências. As agências podem servir como "corretores" ou "casamenteiros", agrupando agentes com interesses similares e metas compatíveis e provendo um ponto de encontro no qual os agentes possam interagir de modo seguro [2]. A mobilidade de agentes conduz a uma extensão do modelo cliente-servidor: clientes enviam partes deles mesmos para o servidor (ou servidores) para execução [9].

Várias vantagens podem justificar a utilização de agentes móveis em certas aplicações [10]:

- redução do custo: caso uma aplicação exija uma grande quantidade de informações a ser veiculada pela rede, pode ser mais econômico enviar um agente que execute a tarefa onde estão localizadas estas informações. Desta forma, o agente pode filtrar e selecionar somente as informações relevantes e transferí-las para a sua origem;

- processamento assíncrono: 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 o nó permaneça em operação. Este processamento assíncrono pode ser utilizado na computação móvel;
- divisão de processamento: computadores com pequena capacidade de processamento podem ser utilizados de maneira eficiente para armazenar e processar dados que são filtrados e enviados por agentes móveis a partir de grandes servidores de informações;
- flexibilidade: agentes móveis se contituem como uma alternativa para o modelo cliente-servidor, podendo também complementá-lo.

3.1 Aplicações de Agentes Móveis

Os agentes móveis podem ser úteis em várias aplicações. Hoje em dia podem ser utilizados principalmente em recuperação de informações, gerenciamento de rede, comércio eletrônico e computação móvel [2].

A **recuperação de informações** na rede pode ser suportada de modo mais eficiente se um agente representando um pedido puder se mover para o lugar onde os dados estão realmente armazenados, ao invés de transmitir todos os dados através da rede para analisar e subseqüentemente descartar a maior parte da transmissão. Isto funciona bem para "pedidos que não podem ser antecipados", isto é, o implementador de um sistema de banco de dados geralmente não pode adivinhar tudo que os usuários podem querer obter e prover código para fazer a análise relevante. Portanto, se os usuários têm que escrever seus próprios programas especiais de recuperação, uma abordagem baseada em agentes pode economizar bastante tráfego na rede. Isto é ainda mais evidente quando se leva em conta que os agentes podem se mover para outras fontes de informação. Técnicas como roteamento semântico - despachar um pedido de acordo com o lugar onde este tem mais chances de ser respondido, ao invés de utilizar alguma informação de endereços - podem ser usadas para aumentar a utilidade do sistema e a facilidade de utilização.

No **gerenciamento de redes** grandes, compreendendo centenas ou milhares de computadores, as operações de monitoração e detecção de falhas são complexas e envolvem grandes quantidades de "relatórios" (*logs*). Não é possível pré-fabricar programas de diagnóstico para todas as eventualidades, porém seria possível utilizar agentes móveis para vigiar o sistema, detectar possíveis pontos de problemas ou gargalos de desempenho e chamar a atenção de gerentes de redes. O potencial de agentes móveis para gerenciamento de redes é de interesse das companhias de telecomunicações, pois seus usuários estão preocupados com a qualidade de serviço da rede. Com a utilização de agentes, uma distribuição dinâmica e uma melhor descentralização da "inteligência" do gerenciamento de redes de telecomunicações podem ser realizadas [11].

O **comércio eletrônico** na Internet está se tornando uma realidade, e, como padrões para pagamento eletrônico estão disponíveis, propriedades comerciais acessíveis pela rede irão crescer de maneira rápida. Os agentes móveis podem ajudar a localizar as ofertas mais baratas, fazer negócios ou até mesmo concluir transações comerciais de interesse de seus donos.

Os computadores portáteis estão se tornando menores e mais poderosos, fazendo crescer a demanda por **computação móvel**. No entanto, o acesso sem fio para uma infra-estrutura fixa de informação tende a continuar lento e enfadonho devido às restrições nas transmissões via rádio. Além disso, para minimizar o consumo de energia e os custos de transmissão, os usuários não vão querer estar ativos enquanto algum pedido complicado é levado em conta pelos recursos computacionais fixos. Para solucionar estes problemas, os usuários podem submeter agentes móveis que contém seus pedidos, desconectar em seguida e esperar que os agentes depositem seus resultados prontos para serem pegos mais tarde. Ainda para esta aplicação, a tecnologia de agentes pode ser utilizada para modelar como os dados deverão ser transmitidos para o usuário móvel, de modo a respeitar as características da conexão [12].

Obviamente, nenhuma destas aplicações requer necessariamente a utilização de agentes móveis, pois a maioria poderia utilizar programas estacionários e RPC. Contudo, isto só poderia ser feito ao custo de aumentar a carga do sistema (e da rede) e, possivelmente, de inconveniência para os usuários [2]. Harrison et al. [13] afirmam que enquanto as vantagens individuais dos agentes não representam uma motivação para suas utilizações, a criação de uma arquitetura de agentes de grande aceitação facilita um grande número de serviços e aplicações em redes.

4 Infra-estrutura de Agentes

Os agentes devem interagir com seus hospedeiros para utilizar seus serviços e negociar serviços com outros agentes [2]. Os agentes móveis devem ainda podem se mover em redes de computadores. O ambiente que permite estas ações é chamado **infra-estrutura de agentes** e a entidade que provê suporte para os agentes em um hospedeiro particular é chamada servidor de agentes [2]. Além destes serviços básicos, uma infra-estrutura de agentes deve também prover segurança a fim de proteger o hospedeiro (e os agentes) de acessos não autorizados e espionagem (seção 4.6). As características básicas de uma infra-estrutura de agentes descritas a seguir estão baseadas na proposta de Lingnau et al. [1, 2].

4.1 Modelo de Arquitetura

A base do modelo de arquitetura utilizado para uma infra-estrutura de agentes (figura 1) é o **servidor de agentes** [1]. Este é um programa (como um servidor de correio eletrônico ou um servidor de FTP) que é executado em cada computador que será acessível ao agentes e está encarregado dos agentes que estão sendo executados neste computador [1]. Suas tarefas incluem aceitar agentes, criar o ambiente de execução apropriado, supervisionar as execuções dos agentes (no meio tempo respondendo pedidos sobre seus estados) e finalizá-las quando necessário. O servidor de agentes também deve organizar a transferência de agentes para outros hospedeiros, gerenciar comunicações entre agentes e entre agentes e seus donos, e autenticar e controlar o acesso para todas as operações de agentes.

Para cada agente sendo executado em um servidor, existe um **ambiente em tempo de execução** dedicado. O ambiente é a interface entre o agente e seu hospedeiro e disponibiliza recursos ao agente de modo controlado (seção 4.5).

O usuário interage com a infra-estrutura de agentes através de um **cliente**. O cliente deixará o usuário submeter um agente para execução, descobrir seu estado, parar ou chamar novamente um agente e realizar outras operações, se necessário. O cliente não necessita de uma conexão permanente com o resto da infra-estrutura de agentes, devido ao processamento assíncrono de agentes. O cliente pode, por exemplo, estar em um computador móvel que se comunica com a infra-estrutura fixa via um enlace de rádio.

A principal linha de projeto para uma infra-estrutura de agentes é permitir flexibilidade em relação à implementação de agentes, acessos de agentes ao(s) sistema(s) hospedeiro(s) e comunicações entre agentes. Por exemplo, não é bom restringir a linguagem de implementação de agentes (seção 4.2). Uma infra-estrutura geral para agentes móveis deve prover mecanismos, não policiamento, para ser aceita de um modo geral.

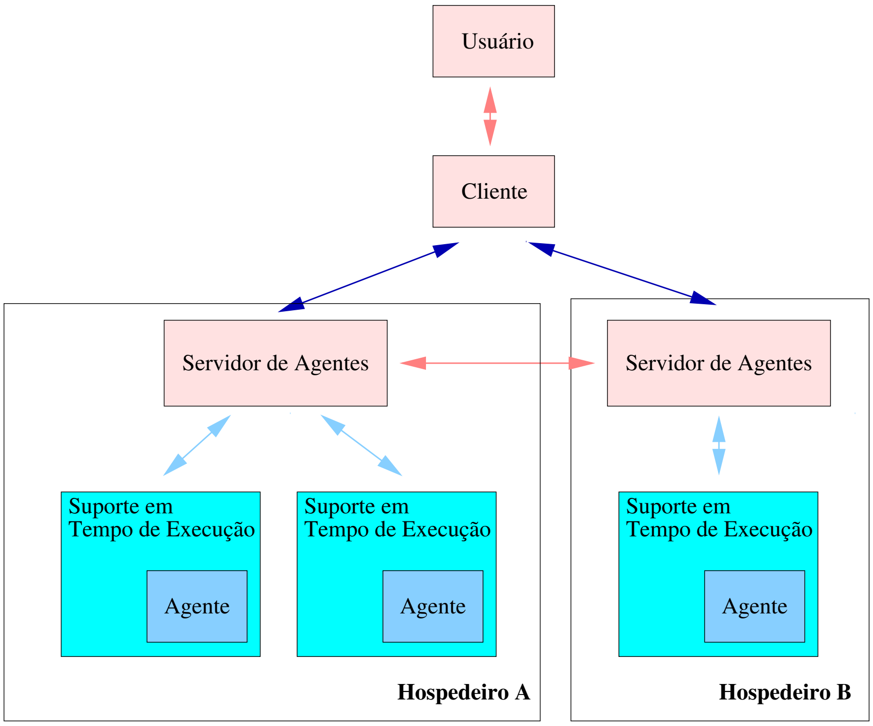

Figura 1: Modelo de Arquitetura.

4.2 Estrutura de Agentes

Os agentes consistem de código, estado e atributos [1]. O **código** de um agente móvel é um programa que define o comportamento do agente. Obviamente um código de um agente pode ser escrito em qualquer linguagem, porém um agente deve poder ser executado de uma mesma maneira em qualquer hospedeiro para o qual possa se mover [2]. Para isto, as linguagens que são interpretadas diretamente ou compiladas para uma linguagem intermediária baseada em interpretador que possa ser facilmente transportada e executada sem recompilação são as mais adequadas. Como exemplos, tem-se: Tcl, Perl e Java.

Uma infra-estrutura aberta para agentes móveis não deve forçar os usuários a utilizar uma linguagem fixa, mas encorajá-los a utilizar aquela mais apropriada para as aplicações dos usuários, mantendo uma base comum de interoperabilidade para que os agentes escritos em diferentes linguagens possam se comunicar e trabalhar juntos [2].

O TACOMA [4] permite que os agentes sejam escritos em Tcl e C, um agente Agent Tcl [5] deve ser escrito em Tcl e o Telescript [6] requer a utilização da linguagem de mesmo nome.

Além do código, um agente carrega um **estado** que permite retomar o trabalho de onde parou, após mudar para outro hospedeiro. No caso mais simples, as variáveis globais dos agentes devem ser preservadas, assim como a noção de a partir de onde o código do agente deve ser executado. Isto pode ser suportado através de uma ramificação com vários caminhos no começo do código do agente, para pular para o lugar correto de acordo com os valores de algumas variáveis globais. Outra solução seria codificar uma representação da pilha de execução do agente (completa, com chamadas de subrotinas e variáveis locais) como uma parte do estado. Em algumas linguagens, seria possível representar um estado do agente como parte do seu código durante a transmissão, por exemplo através da colocação no início de uma seqüência apropriada de associações ao próprio código.

No TACOMA, um programador precisa capturar explicitamente o estado e no Agent Tcl e no Telescript, o estado do agente é implicitamente transferido com o agente.

A terceira parte de um agente móvel consiste de **atributos** que descrevem o agente, seus requisitos e sua história para a infra-estrutura. Isto inclui dados como um identificador único de agente, o dono do agente como um endereço destinatário para resultados intermediários, mensagens de erro ou reclamações quanto a um mau comportamento de um agente, o tempo e o lugar de origem e a história do movimento. Os atributos também devem impor limitações na mobilidade do agente, tais como datas de expiração, "fique dentro do domínio `ufrj.br`" e outras. Os agentes também podem ter que expressar seus requisitos de recursos, tais como a quantidade de memória necessária, limitações (por exemplo "R$ 1,00 deixado para pagar por serviços de recuperação de informações") e requisitos para o ambiente no qual necessitam executar suas tarefas (como "acesse um banco de dados bibliográficos de acordo com os padrões do `GTA`"). Um agente deve carregar informação de autenticação que deixa a infra-estrutura verificar as credenciais de qualquer usuário que queira parar ou chamar novamente o agente ou verificar seu progresso, assim como checar quaisquer senhas ou chaves que o agente necessite para obter acesso a informações restritas, se necessário. Parte destes atributos pode ser acessível ao agente, porém o agente não pode ser capaz de modificar seus atributos [1].

4.3 Transporte de Agentes

Uma das características principais de agentes móveis é seu poder de se mover de um hospedeiro para outro. A diferença entre este tipo de mobilidade e a encontrada em, por exemplo, sistemas de balanceamento de carga que permitem a realocação de processos é que o movimento acontece de acordo com um pedido explícito do agente ao invés de ocorrer de acordo com alguma decisão do sistema operacional (sem o agente notar) [2]. Isto é, a linguagem de implementação do agente inclui uma sentença ou chamada de biblioteca da forma `MOVA-PARA hospedeiro`. Quando esta chamada é executada, o agente é suspenso, codificado para transmissão, transmitido para o novo hospedeiro, decodificado e instalado para retomar a execução neste novo local.

Uma questão importante é como um agente escolhe um hospedeiro para o qual irá se mover. No caso mais simples, um agente móvel pode carregar consigo um itinerário de lugares, visitando um após o outro enquanto realiza seu trabalho. É mais interessante permitir que o agente possa mudar seu destinatário de acordo com as descobertas que fez durante as explorações anteriores. Por exemplo, supondo que durante uma pesquisa bibliográfica em algum endereço um agente nota um grande número de artigos escritos pelos pesquisadores da Universidade Federal do Rio de Janeiro, este agente deve tentar se mover para lá, para obter localmente informações mais interessantes.

Outra possibilidade seria um servidor de agentes avisar um agente sobre lugares interessantes em sua vizinhança (não necessariamente geográfica); no exemplo anterior, outras coleções de informações bibliográficas. Se cada servidor conhecer seus servidores vizinhos, um agente será capaz de atravessar um subconjunto interessante de todos os servidores [1]. Esta informação pode ser gerada de acordo com a origem ou o propósito do agente em questão [1]. A vantagem deste esquema é que nenhum servidor precisa de informação completa sobre todos os possíveis destinatários de agentes móveis. Isto facilita a construção de *firewalls* ou domínios limitados que restringem o movimento de agentes.

Roteamento semântico também pode ser utilizado. Por exemplo, um agente cuja meta seja fazer a assinatura de uma determinada publicação para seu dono poderia ser transparentemente roteado pela infra-estrutura para a editora apropriada. Parece existir um mercado para "agentes de diretórios", onde o primeiro movimento de um agente seria para um lugar onde um agente de diretórios poderia apontar um lugar apropriado ao agente.

Mecanismos de transporte são necessários para transmitir um agente de um hospedeiro a outro hospedeiro. Para isto, podem ser utilizados padrões como TCP/IP, HTTP (*HyperText*

Transfer Protocol), X.25 e correio eletrônico [14]. Na infra-estrutura do Grupo Telematik [2], os agentes são encapsulados utilizando o padrão MIME (*Multiporpose Internet Mail Extensions*) da Internet e transmitidos por HTTP. O Telescript pode utilizar TCP/IP, X.25 ou correio eletrônico e o Agent Tcl e o TACOMA utilizam TCP/IP.

4.4 Comunicação entre Agentes

Para realizarem trabalhos que tenham alguma utilidade, os agentes devem poder se comunicar com o sistema hospedeiro, com outros agentes e com seus donos. Um modelo básico de comunicação entre agentes explora a noção de um **espaço de informações**, uma entidade mantida localmente em cada servidor de agentes pela infra-estrutura em interesse dos agentes que estão a seu cargo. O espaço de informações pode manter triplas da forma (chave, acesso, valor), onde a chave é um nome único de uma tripla, o acesso é uma lista de controle de acessos que permite o acesso à tripla de um agente, grupos de agentes ou todos os agentes, e o valor é um conteúdo de tamanho arbitrário (figura 2).

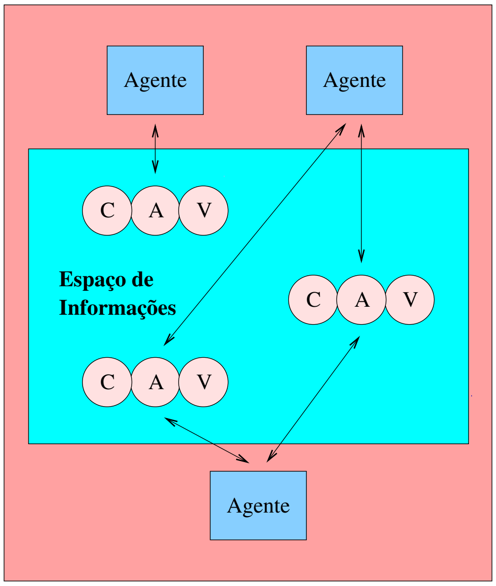

Figura 2: Espaço de Informações.

As operações suportadas pelo espaço de informações incluem escrever uma tripla e lê-la de modo destrutivo (é removida do espaço de informações após ser lida) ou de modo não destrutivo. Todas estas operações são assumidas atômicas para evitar condições de competição. Esquemas de comunicação mais avançados como RPC síncrono ou assíncrono, multiponto ou difusão podem ser construídos em cima destas primitivas. Operações adicionais que seriam úteis incluem a habilidade de um agente registrar interesse em mudanças em uma tripla específica do espaço de informações ou em inclusões ao espaço de informações como um todo. Isto iria evitar que agentes tivessem variáveis de espera ocupada ou varredura para detectar mudanças, o que aumentaria a eficiência total do sistema.

Na linguagem Telescript, um "lugar" é uma unidade dentro de uma rede de computadores onde serviços são oferecidos. O Telescript tem dois mecanismos de comunicação entre agentes: encontro, para interação no mesmo lugar, e conexões, para interações em diferentes lugares. O Agent Tcl utiliza o TCP/IP para a comunicação entre os agentes. O TACOMA contém um mecanismo de encontro em uma mesma localidade para a comunicação entre agentes.

O principal propósito do espaço de informações é permitir a comunicação entre agentes, porém o espaço também pode ser utilizado para implementar a comunicação entre os agentes e a infra-estrutura de agentes e entre os agentes e o sistema hospedeiro. Por exemplo, o servidor de agentes pode utilizar o espaço de informações para tornar disponíveis dados de interesse geral para os agentes, como por exemplo uma lista dos agentes sendo executados, no momento, no hospedeiro.

Seria óbvio generalizar este modelo de comunicação para incluir o acesso a um espaço de informações de um servidor de outro lugar da rede. Este "espaço de informações global" não seria, contudo, arbitrariamente escalável sem perder sua eficiência. Além disso, dado que os agentes podem se mover facilmente entre hospedeiros, não é totalmente claro se a extensão do modelo para comunicações entre agentes irá realmente solucionar o problema [2].

A existência do espaço de informações não impede que os agentes utilizem outros tipos de comunicação. Se dados devem ser transmitidos em alta velocidade, pode ser permitido aos agentes utilizar conexões dedicadas ou memória compartilhada.

Finalmente, os agentes devem poder se comunicar com seus donos e vice-versa. O esquema mais simples de fazer isto é deixar o agente enviar correios eletrônicos para o endereço do dono dado por um atributo do agente. Este modo de comunicação assíncrono se encaixa bem ao fato que o dono de um agente pode estar desconectado ou utilizando um terminal móvel e, portanto, não estaria conectado quando a mensagem do agente chegasse. Se um agente necessita de comunicação síncrona, por exemplo, para pedir confirmação para alguma ação, ele terá que ser suspenso até que a resposta seja recebida [2].

No Telescript, a comunicação de um agente com um usuário pode ser feita através de interfaces gráficas, correio eletrônico ou *paging*. O Agent Tcl utiliza o Agent Tk para adicionar interfaces gráficas com o usuário, enquanto que o TACOMA também utiliza interfaces gráficas que podem ser escritas em Tcl/Tk ou C.

4.5 Suporte em Tempo de Execução

A infra-estrutura de agentes deve prover suporte em tempo de execução para os agentes por várias razões [2]:

- como os agentes necessitam tirar vantagem da infra-estrutura para serviços de mobilidade e comunicação, operações apropriadas devem estar disponíveis na linguagem de programação do agente. Isto inclui operações que deixem o agente se mover para outro servidor, acessar o espaço de informações e comunicar resultados intermediários ou fazer perguntas ao seu dono;

- os agentes não são supostos fazer má utilização do acesso ao sistema hospedeiro, por exemplo, enviando sua senha para algum "quebrador de senhas" ou formatando o disco rígido do sistema. A melhor maneira de forçar este tipo de segurança é desabilitar este tipo de operação em uma primeira instância ou controlá-la bem de perto, de modo que os argumentos de operações não confiáveis sejam cuidadosamente examinados e as operações rejeitadas, se parecerem suspeitas;
- necessidade de funcionalidades orientadas ao problema do agente. Por exemplo, um agente cujo objetivo consiste em realizar uma pesquisa bibliográfica poderia requisitar acesso a uma ferramenta de procura utilizando texto, ferramenta que seria provida pelos fornecedores de banco de dados para que o agente não precisasse carregá-la consigo.

No caso mais simples, o ambiente em tempo de execução para os agentes pode ser implementado de maneira específica para a linguagem, suportando cada linguagem de implementação de agentes através de um pacote em tempo de execução escrito naquela linguagem. Eventualmente, um conjunto de operações padrões pode surgir, os quais poderiam estar disponíveis utilizando serviços independentes da linguagem, como CORBA (*Common Object Request Broker Architecture*).

4.6 Segurança

Talvez o tópico mais importante para a completa aceitação de sistemas baseados em agentes seja a segurança. Em um sistema de redes de computadores, nem o agente nem as máquinas são "confiáveis" [15]. Muitas pessoas associam agentes movéis às ameaças impostas por programas maliciosos como o vírus e os cavalos de Tróia. Um agente pode atacar uma máquina e ganhar acesso a recursos locais [15]. Deve-se evitar que um servidor adultere os agentes a seu cargo ou mesmo olhe o código de um agente que implementa, por exemplo, uma estratégia de negociação de um competidor. Dada a tecnologia dos sistemas operacionais atuais, não existe uma maneira prática de evitar que um atacante determinado faça uma re-engenharia no código do agente enquanto for executado em um sistema sob controle do atacante. A única resposta a este dilema, por enquanto, parece ser a utilização de premissas neutras operadas por um terceiro parceiro confiável, desinteressado, que serve como um ponto de encontro para agentes móveis que queiram realizar negociações, leilões ou eventos similares. Algumas funções de segurança que podem ser utilizadas são: autenticação de um transmissor de agente, verificação de autorização de um transmissor para propósitos de execução, controle de acesso a recursos, relatórios de ações e verificação de integridade [14].

Os agentes podem ser protegidos de espionagem e manipulação durante um movimento por métodos adequados de criptografia. Dado o esquema apresentado anteriormente, quando o servidor de agentes em um hospedeiro conhece os servidores em sua vizinhança, é fácil suportar criptografia com chave pública para agentes sendo transmitidos, pois parte do conhecimento do servidor sobre sua vizinhança pode consistir de chaves públicas apropriadas para serem usadas para os vários destinatários. Como o gerenciamento de chaves pode ser considerado como parte do gerenciamento da infra-estrutura do agente local, os agentes não têm que lidar com criptografia.

O Telescript verifica a autoridade de um agente quando necessário. O Agent Tcl provê uma segurança rudimentar na qual qualquer agente, mensagem ou conexão que não vier de uma "máquina aprovada" é ignorada. O administador do sistema especifica o conjunto de máquinas aprovadas. O TACOMA provê mecanismos de segurança através de um agente *firewall* que atualmente só relata em um disco a interação de um agente com o *firewall* quando da execução da instrução `encontro`.

5 Considerações Finais

As principais aplicações de agentes móveis são em recuperação de informações, gerenciamento de redes, comércio eletrônico e computação móvel. Nenhuma destas aplicações requer necessariamente a utilização de agentes móveis, porém caso isto não seja feito, haverá um aumento substancial na carga do sistema (e da rede) e, possivelmente, existirá inconveniência para os usuários. Além disso, a criação de uma arquitetura de agentes de grande utilização facilitará um grande número de serviços e aplicações em redes.

O desenvolvimento de uma infra-estrutura que sirva de base para a comunicação entre os agentes e os hospedeiros e para a mobilidade dos agentes é de vital importância. Várias infra-estruturas estão sendo propostas e implementadas, cada qual com características diferentes.

Como continuação deste trabalho, será desenvolvida uma infra-estrutura de agentes móveis, visando a implementação de aplicações multimídias distribuídas nas quais se aplica o modelo de programação remota.

Referências

[1] A. Lingnau, O. Drobnik e P. Dömel, "An HTTP-based infrastructure for mobile agents", in *Fourth International WWW Conference*, Boston/Massachusets, EUA, dezembro 1995.

[2] A. Lingnau e O. Drobnik, "An infrastructure for mobile agents: requirements and architecture". Fachbereich Informatik (Telematik), Johann Wolfgang Goethe-Universität, Frankfurt, Alemanha, disponível em ftp://ftp.tm.informatik.uni-frankfurt.de/agents/13dis-paper.ps.gz.

[3] D. Gilbert e P. Janca, "IBM intelligent agents". IBM Corporation, EUA, disponível em http://www.networking.ibm.com/iag/iagwp1.html.

[4] D. Johansen, R. van Renesse e F. B. Schneider, "An introduction to the TACOMA distributed system", tech. rep., University of Tromsϕ/Cornell University, Tromsϕ/Ithaca, Noruega/EUA, junho 1995.

[5] R. S. Gray, "Agent Tcl: a transportable agent system", in *CIKM Workshop on Intelligent Information Agents, Fouth International Conference on Information and Knowledge Management (CIKM 95)*, Baltimore/Maryland, EUA, 1995.

[6] J. White, "Mobile agents white paper", tech. rep., General Magic, 1996.

[7] L. Chiariglione, "Helping agent technologies get across to the market place". Foundation for Intelligent Physical Agents (FIPA), disponível em http://drogo.cselt.stet.it/fipa/general_agents.htm.

[8] "The Agent Society". Disponível em http://www.agent.org, 1997.

[9] D. Chess, B. Grosof, C. Harrison, D. Levine, C. Parris e G. Tsudik, "Itinerant agents for mobile computing", tech. rep., IBM T. J. Watson Research Center/IBM Zurich Research Center, Nova Iorque/Rueschlikom, EUA/Suíça, março 1995.

[10] H. S. Nwana, "Software agents: an overview", *Knowledge Engineering Review*, vol. 11, no. 3, pp. 1–40, setembro 1996.

[11] T. Magedanz e T. Eckardt, "Mobile software agents: a new paradigm for telecommunications management", in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, Kyoto, Japão, abril 1996.

[12] "Intelligent agents". IBM Corporation, IBM Intelligent Agent Center of Competence, EUA, disponível em http://www.networking.ibm.com/iag/iaghome.html, 1996.

[13] C. G. Harrison, D. M. Chess e A. Kershenbaum, "Mobile agents: are they a good idea?", tech. rep., IBM T. J. Watson Research Center, Nova Iorque, EUA, março 1995.

[14] P. C. de Oliveira e E. Cardozo, "Mobile agent-based systems: an alternative paradigm for distributed systems development", in *XV Simpósio Brasileiro de Redes de Computadores (SBRC 97)*, São Carlos, SP, Brasil, pp. 508–524, maio 1997.

[15] M. Abdalla, W. Cirne, L. Franklin e A. Tabbara, "Security issues in agent based computing", in *XV Simpósio Brasileiro de Redes de Computadores (SBRC 97)*, São Carlos, SP, Brasil, pp. 231–244, maio 1997.