

Uma arquitetura para integrar sociedades de agentes na Internet

Sílvio César Cazella¹
scc@inf.ufrgs.br

Luis Otávio Campos Alvares²
alvares@inf.ufrgs.br

Universidade Federal do Rio Grande do Sul
Centro de Pós-Graduação em Ciência da Computação (UFRGS)
Campus do Vale - Bloco IV
Av. Bento Gonçalves, 9500 - Bairro Agronomia
Caixa Postal 15064
Cep 91501-970, Porto Alegre, RS
Brasil

Palavras-Chave: Inteligência Artificial, Inteligência Artificial Distribuída, Sistemas Multiagentes, Internet, Arquitetura para sociedades de agentes.

Resumo

O grande salto tecnológico ocorrido nos últimos decênios em áreas como a informática e as telecomunicações, começa a causar uma verdadeira revolução. Com o advento da Internet, a possibilidade de substituir a presença física de pessoas por assistentes inteligentes é uma realidade que começa a tomar forma.

O objetivo deste trabalho é apresentar como deve ser feita a interação entre agentes, nas sociedades dentro da Internet. Para tanto, é apresentada uma proposta de arquitetura para integrar sociedades de agentes na Internet, de forma a possibilitar: a localização, a conexão, a identificação de agentes (nomes de agentes, função e endereços destes na rede) sem que ocorra a repetição de nomes na sociedade, a entrada e saída destes da sociedade, a troca de mensagens e a procura de agentes com determinadas características.

Abstract

Technological breakthroughs which occurred in the last decades in areas such as Computer Science and Telecommunications have started to cause an undeniable social revolution. With the Internet, the possibility of replacing the physical presence of people by intelligent assistants or agents is now becoming a reality.

This paper aims at defining how the interaction among agents in the Internet should be achieved. An architecture to coordinate the interaction of agents in the Internet is presented, so as to make it possible, among other things, the localization of agents, the connection among agents, the identification of agents (their names, function, and addresses in the net), the way in and the way out of agents of the society, the exchange of messages among agents and the search for agents with specific characteristics.

¹ Mestre em Ciências da Computação pela Universidade Federal do Rio Grande do Sul (Brasil). Áreas de Interesse: Inteligência Artificial, Inteligência Artificial Distribuída, Sistemas Multiagentes.

² Doutor em Informática pela Universidade Joseph Fourier (Grenoble, França). Áreas de Interesse: Inteligência Artificial, Inteligência Artificial Distribuída, Sistemas Multiagentes.

1 Introdução

Com o advento da Internet, a possibilidade de substituir a presença física pela eletrônica no trabalho, realmente começou a tomar forma. A Internet possibilitou através de seus recursos, que os mais variados profissionais pudessem comunicar-se a longas distâncias e consultar informações provenientes dos mais variados endereços eletrônicos através do mundo.

Muito do tempo útil de trabalho humano que acaba sendo perdido com tarefas simples, poderia ser realizado por assistentes inteligentes junto a Internet, assistentes que teriam a finalidade de auxiliar o usuário da rede. Cada assistente inteligente constitui-se em um agente modular e autônomo, e encontra seu embasamento teórico na Inteligência Artificial e na Inteligência Artificial Distribuída respectivamente.

Os sistemas de Inteligência Artificial (IA) tradicionais concentravam sua atenção em um único agente, o qual seria provido de alguma espécie de inteligência, e sozinho seria especialista na realização de uma tarefa específica.

Atualmente o enfoque tende a ser outro, fala-se em Inteligência Artificial Distribuída. Nesta subárea da IA o enfoque recai sobre o aspecto de integração de agentes providos de alguma inteligência. São especialistas na atividade que desempenham, não trabalham mais isolados, mas de forma cooperativa tentam resolver um problema da melhor forma possível, caracterizando uma área da Inteligência Artificial Distribuída, a área dos Sistemas Multiagentes [DEM 93].

Os agentes são entidades autônomas, e encontram-se, geralmente, imersos em uma sociedade. Conceitualmente são entidades capazes de reagir e/ou agir a favor de algo ou alguém, influenciando seu ambiente. Este ambiente necessariamente inclui outros agentes. Interagir com este ambiente, ou mais especificamente com outros agentes, é o elemento chave de um Sistema Multiagente.

Os agentes podem atuar em nome de alguém em particular, isto é, eles podem tomar algumas atitudes as quais representam de maneira apropriada os interesses de outros; portanto os agentes também devem ser robustos e capazes de manipular com segurança informações privadas. Os agentes são entidades altamente interativas - eles despendem muito de seu tempo comunicando-se com outros agentes e com seres humanos, são participantes altamente ativos em seu mundo computacional, isto é, eles reagem a algo e causam mudanças em todo o estado do sistema.

O propósito destes assistentes inteligentes (agentes), é o de auxiliar o usuário na realização de tarefas específicas dentro da Internet. Estes assistentes realizam tarefas como: vasculhar a rede na busca de informações e endereços eletrônicos de interesse do usuário, coordenação de tarefas entre um grupo de pessoas, troca de informações entre os agentes que auxiliam os componentes de um grupo de trabalho, etc [SEA 95, ETZ 95, AEK 95, HED 95].

Os agentes apresentam, todas as características relatadas acima; mas se estes podem interagir entre si, convivendo em uma chamada sociedade de agentes, como deve ser a arquitetura de interação entre estes elementos? De que maneira um agente consegue estabelecer um contato direto com o agente da sociedade de seu interesse? E se reportarmos estas sociedades para o imenso ambiente da Internet, como deve ser o procedimento para que estes agentes encontrem-se neste emaranhado de redes?

O objetivo deste trabalho é apresentar uma arquitetura que possibilite implementar agentes inteligentes no ambiente da Internet, possibilitando que haja a interação entre estes agentes, nas mais diversas sociedades. Em outras palavras, o objetivo é definir uma arquitetura de agentes que permita a implementação de Sistema Multiagentes na Internet.

2 Problemas Analisados

Abaixo serão listados três problemas principais que foram analisados, para definir a arquitetura que será apresentada:

Problema número 1: A arquitetura proposta tem de ser independente do tipo de sociedade em questão, ou seja, ela deve apresentar um funcionamento que venha a suprir as necessidades de integração entre os agentes, independente do fato da sociedade ser aberta ou fechada. A figura 2.1 ilustra estes tipos de sociedade de agente:

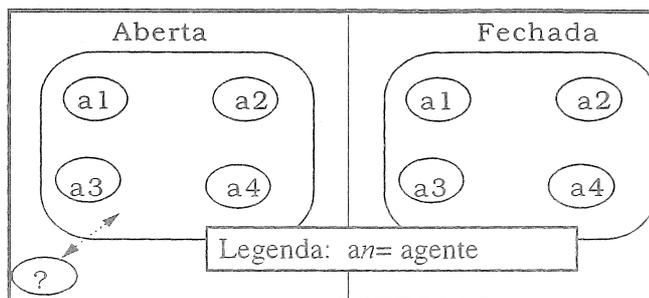


FIGURA 2.1 - Tipo de Sociedade

O problema que foi analisado sob o ponto de vista do tipo de sociedade pode ser descrito da seguinte maneira: a sociedade em questão é uma sociedade aberta, ou seja, a entrada e saída de indivíduos é permitida, ou esta sociedade apresenta-se sem alteração, ou seja, a migração de seus integrantes é nula. Este problema deve ser analisado principalmente no momento de propor a integração entre os agentes. A questão seria - como todos ficarão sabendo da entrada, ou saída de membros da sociedade?

Problema número 2: Um outro problema que foi avaliado, é o fato da necessidade ou não da existência de um agente centralizador, como figura integradora de uma sociedade de agentes. A figura 2.2 ilustra este fato.

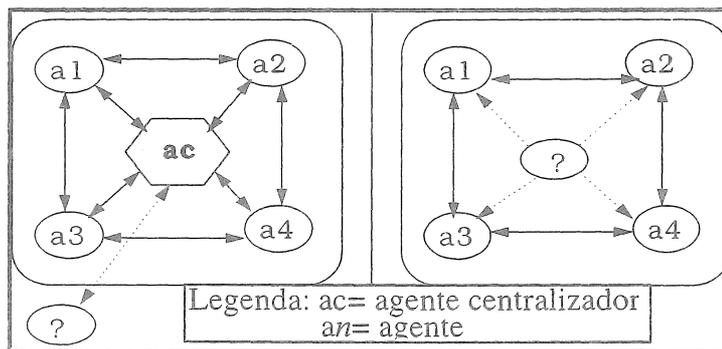


FIGURA 2.2 - Necessidade do Agente Centralizador de Endereços

É interessante que nesta sociedade de agentes exista um agente centralizador (ac) dos endereços de máquina de cada agente? Como deve ser a apresentação destes indivíduos dentro da sociedade? Duas soluções iniciais podem ser apresentadas:

1) Existe um agente central na sociedade, e este tem a função de atualizar as informações gerais da sociedade, ou seja, caso um "agente x" chegue e passe a integrar a sociedade, este deve primeiro dirigir-se ao agente centralizador e informar seus dados (função desempenhada, nome, endereço de IP, etc.). Logo após da apresentação ter sido feita, o agente centralizador deve informar a entrada do novo membro a sociedade. O mesmo deve ocorrer no caso da

retirada, temporária ou não, de um agente da sociedade (neste caso a retirada do agente deve ocorrer de maneira consciente, ou seja, ele sai por vontade própria e não por uma queda de sua rede de trabalho, por exemplo).

2) Não existe a figura do agente central. Neste caso, se um "agente x" quiser entrar em uma sociedade de agentes, este deve dirigir-se a cada membro integrante da sociedade (para encontrar os membros da sociedade, entende-se que os agentes terão de consultar um arquivo comum, onde todos os agentes deixam seus dados escritos). Esta solução é bastante viável, mas vai depender da quantidade de indivíduos que vierem a integrar esta sociedade.

No caso da primeira solução proposta, temos como vantagem uma estrutura central, que ficaria localizada em uma determinada máquina da rede, e caso algum agente não saiba como entrar em contato com outro membro da sociedade, entraria em contato com este agente centralizador, e solicitaria o endereço de máquina do agente desejado.

A desvantagem desta solução seria o fato de toda a sociedade ficar, até certo ponto, dependente de um agente centralizador fixo em uma máquina, ou seja, caso (por exemplo) a máquina que está hospedando este agente, venha a apresentar algum tipo de pane (perder o contato com a rede), toda a sociedade de agentes seria de certa maneira atingida, pois os agentes não saberiam quais os membros que entraram ou deixaram esta sociedade de agentes, e nem saberiam como acessar uns aos outros.

No caso da segunda solução apresentada, pode-se destacar a vantagem da não existência de um agente centralizador das informações, ou seja, não existe a dependência dos demais agentes em relação às informações prestadas pelo agente centralizador. Por outro lado, esta segunda solução deve ser avaliada de maneira ponderada, pois ela propõe que a cada entrada ou saída de agente de uma sociedade, este informe, este fato para todos os membros da sociedade.

Esta ação de manter uma comunicação com todos os membros, pode ser saudável no caso de termos poucos membros integrando esta sociedade, mas por outro lado, caso a sociedade seja integrada por muitos membros e/ou seja muito dinâmica, pode acabar ocorrendo um congestionamento na comunicação da rede, o que pode acarretar uma queda de seu desempenho geral.

Problema número 3: O último problema levantado, para ser levado em consideração na definição de uma arquitetura para agentes, foi a existência dos IP dinâmicos de máquina. O agente de IP dinâmico deve apresentar-se ao agente central (ac) ou a toda a sociedade, e como trabalhar com as trocas de IP? Como resolver o problema dos IP dinâmicos?

Na Internet muitas máquinas não possuem um endereço de IP fixo, ou seja, cada vez que entram na rede, são designados endereços diversificados para identificar aquela máquina naquele determinado momento. Ao desconectar-se da rede o endereço é perdido, e ao reconectar-se novo endereço é fornecido.

Este é o caso das máquinas de usuários que conectam-se a servidores que disponibilizam os serviços da Internet. Por estarem dependendo deste servidor para conectar-se, é ele que designa a cada interação um endereço para a máquina do usuário.

Neste caso temos um problema bem mais complexo, pois temos os agentes nos mesmos locais fixos, mas não podemos utilizar sempre o mesmo endereço para achá-los, ou seja, não adianta manter um arquivo que armazene o trinômio "agente, porta, endereço de máquina", pois o último item se modificará constantemente.

3 Arquitetura proposta

Considerando os problemas analisados anteriormente, e tendo sido feitas implementações (utilizando a linguagem Java) de agentes utilizando as facilidades da internet, propõe-se uma

arquitetura para coordenar a interação de agentes, baseada no uso de um agente central por domínio da Internet [CAZ 97].

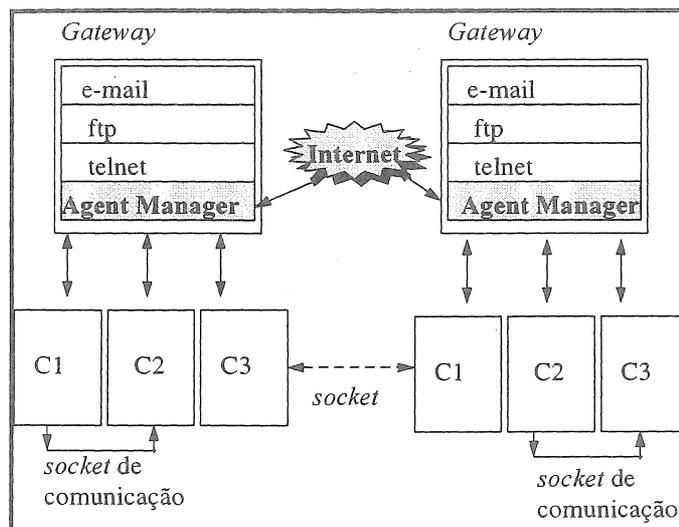


FIGURA 3.1 - Visão geral da Arquitetura de Agentes para a INTERNET

A figura 3.1 apresenta uma visão geral desta arquitetura. Na figura pode-se identificar duas sub-redes (com seus computadores locais designados por c1, c2 e c3) conectados aos seus respectivos gateways, comunicando-se através da Internet .

As principais características desta arquitetura são:

1) Na arquitetura existe um agente central, por domínio, que tem como função armazenar (em um arquivo local) e fornecer (quando solicitado) os endereços dos agentes que compõem a sociedade e suas descrições. Este agente é denominado de "Agent Manager", constitui-se em um processo que está sempre em execução e em estado de *listening*.

2) Para haver uma maior qualidade no serviço prestado por este agente central, ele é fisicamente colocado na máquina *gateway* da rede, ou seja, ele passa a figurar como mais um serviço (ou aplicativo) fornecido por esta máquina.

Cada um dos serviços suportado por um *gateway* (e-mail, FTP, Telnet, HTTP) faz uso de uma porta lógica específica. No caso, este agente (*Agent Manager*) utiliza uma porta lógica própria (dedicada) para que os agentes façam conexão (via *socket*³) com ele.

Colocar este *Agent Manager* na máquina *gateway* traz facilidades várias do tipo:

- se um agente desejar encontrar o agente central basta reportar-se a máquina *gateway* da rede e conectar-se a porta correta;

- para haver conexão entre agentes da mesma sociedade, basta o agente pedir o endereço do outro agente desejado para o *Agent Manager* e realizar a conexão via *socket*, não sendo necessária a participação do *Agent Manager* nesta comunicação;

- uma vez fornecido este endereço, o agente local o guarda em um arquivo local em sua máquina, para caso, haja nova necessidade de conexão, ele não precise mais pedir o endereço para o *Agente Manager*.

³ Um socket é um dos pontos finais de uma comunicação entre dois programas que estão sendo executados em uma determinada rede. Para cada serviço oferecido pela Internet é especificado uma determinada porta lógica (ex: e-mail = 25, telnet=13), para que nosso micro entre em contato com um determinado serviço de um servidor, precisamos estabelecer um *socket* de comunicação com esta máquina através da porta desejada.

3.1 Estrutura dos agentes

Na figura do item anterior (figura 3.1) verificou-se como seria a integração entre os agentes da sociedade e o agente central. A seguir, será apresentada uma visão dos agentes dentro de uma das máquinas que compõem a rede de comunicação (conforme figura 3.2).

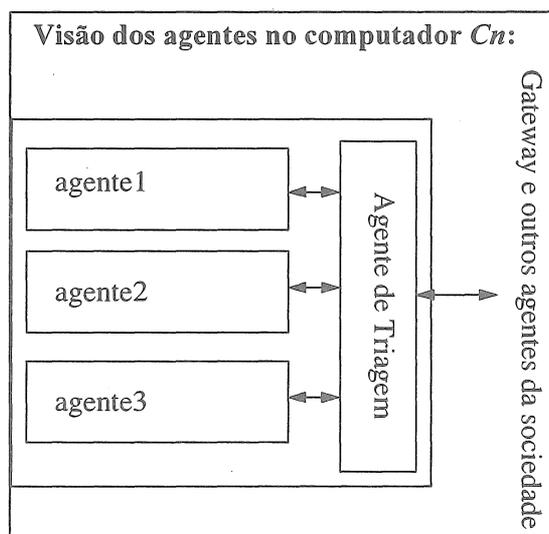


FIGURA 3.2 - Agentes em uma máquina local

Nesta arquitetura, cada máquina pode alocar um ou mais agentes específicos, independente de função⁴ ou origem deste. Em cada máquina da rede tem-se uma entidade específica denominada de Agente de Triagem, que é um processo que está sempre em execução e em estado de *listening*, em uma porta lógica específica e desempenha duas funções de fundamental importância:

1) Este Agente de Triagem, é responsável por distinguir por mecanismos internos, para qual agente alocado na sua máquina é destinada a informação que está chegando via *socket*. Para tanto, ele possui localmente um arquivo com todos os dados de identificação dos agentes que estão alocados na sua máquina.

2) Como já foi citado, este agente constitui um processo que está constantemente sendo executado, "escutando" através de uma porta lógica, ou seja, na verdade ele é a porta de comunicação do mundo dos agentes de uma determinada máquina com o mundo externo e interno.

Como, na realidade, somente o Agente de Triagem terá acesso a uma porta lógica, é ele que deve receber o que vem de fora (mundo externo) e entregar ao agente correto dentro de sua máquina. O mesmo deve ocorrer com a comunicação de dentro para fora, para um agente interno à máquina mandar alguma informação para o mundo externo, ele tem de fazer uso dos recursos de comunicação oferecidos pelo Agente de Triagem, ou seja, pedir para este entregar a informação.

No caso de necessidade de comunicação entre os agentes de uma mesma máquina, também será necessária a participação do Agente de Triagem, ou seja, o agente interessado em enviar alguma informação para outro agente dentro da máquina, deverá solicitar os serviços do Agente de Triagem. Pode-se pensar no Agente de Triagem, como sendo uma *classe*

⁴ Nesta arquitetura está sendo abordado o aspecto de integração e comunicação entre os agentes que compõem uma sociedade de agentes, não é relevante neste momento a função desempenhada pelo agente como entidade funcional.

responsável por toda triagem da comunicação realizada entre a máquina e a rede, e dentro da própria máquina.

Não se deve pensar que esta centralização da comunicação no Agente de Triagem, seja uma forma de tirar a flexibilidade da arquitetura proposta, ela é necessária devido a alguns aspectos:

- Toda a segurança para garantir a integridade da máquina, será feita pelo Agente de Triagem, ou seja, ele será uma espécie de gargalo, onde passa toda a informação, e cabe a ele verificar se a informação não está burlando a segurança da máquina.

- Seria inviável propor uma arquitetura onde todos os agentes ficassem em execução constante na máquina, apenas aguardando o momento de agirem, pois desta forma a performance da CPU, estaria altamente comprometida. A arquitetura propõe que apenas o Agente de Triagem fique em execução, e no momento em que for necessária a ativação de um determinado agente, este deverá ser instanciado pelo Agente de Triagem, para que possa agir.

3.2 Identificação do agente

A identificação de um agente deve ser formada pelo E-mail do usuário + Função do agente + número de ordem.

Como por exemplo: scc@inf.ufrgs.br|agendal01.

O e-mail do construtor do agente é necessário, por três motivos:

- para que no caso de termos mais de um agente de funções semelhantes funcionando na mesma máquina (por exemplo, várias agendas de diferentes usuários), seja possível identificar para qual destes agentes a informação é destinada;

- este nome é importante, para que os roteadores saibam em qual domínio de rede encontram-se os agentes, ou seja para onde deve-se encaminhar o fluxo de dados;

- caso um determinado agente1 tente estabelecer conexão com um outro agente2, e após um certo número de tentativas esta conexão não seja possível, fica definido que o agente1 poderá mandar um e-mail padronizado informando ao responsável pelo agente2, que algum tipo de irregularidade está ocorrendo com o agente.

3.3 Criação de agentes

Para criar um agente o usuário deve programá-lo utilizando sua linguagem preferida e cadastrá-lo junto ao *Agent Manager* e ao Agente de Triagem de sua máquina. O programa de cadastramento irá inserir os dados deste novo agente, no do arquivo de agentes da máquina local, e depois estabelecerá uma comunicação via socket com o *Agent Manager* da rede e informará os dados deste novo agente, incluindo o endereço IP da máquina na Internet que está hospedando o agente em questão, e também qual o número da porta de comunicação disponível para o estabelecimento de sockets de comunicação, como por exemplo: 143.002.345.122|4321. Pode-se observar que para os agentes que estiverem na mesma máquina, o campo referente a porta de comunicação será o mesmo, no caso o número de porta lógica alocado para o Agente de Triagem. Feito isto o agente estará pronto para fazer parte da sociedade.

3.4 Comunicação entre diversos agentes

A comunicação entre os agentes desta sociedade dar-se-á da seguinte maneira: um agente1 deseja estabelecer comunicação com o agente2 de uma determinada máquina, mas desconhece o endereço atual do mesmo. O agente1 entra em contato com o *Agent Manager* e pede o endereço atual do agente2.

De posse desta informação é firmada a comunicação via socket. Sendo que esta informação sobre o endereço do agente2 é armazenada em um arquivo local de endereços de

agentes na máquina do agente1. Este arquivo armazena informações sobre (por exemplo) os últimos dez endereços de agentes acessados e pode ser consultado por todos os agentes que estão localizados na máquina do agente1. A idéia é a de evitar que um agente que está constantemente comunicando-se com um outro, tenha de a cada interação com este, ter de solicitar ao *Agent Manager* o endereço do agente.

Quando for ser firmada a comunicação com o agente de outra máquina, primeiramente o endereço deste agente é buscado neste arquivo comum, caso não seja encontrado, então é estabelecida a comunicação com o *Agent Manager* para obter esta informação.

Quando o agente for estabelecer comunicação, deve ser fornecido, por parâmetros de protocolo para o Agente de Triagem, o nome, a função, o número da máquina e número da porta do agente desejado.

Caso um agente tente entrar em contato com outro agente, e o endereço deste já encontre-se gravado no arquivo comum, a comunicação é estabelecida de forma direta (sem consulta ao *Agent Manager*). Caso o estabelecimento da comunicação demore algum tempo, o agente deve parar de tentar a conexão e verificar com o *Agent Manager* se não houve alterações de dados daquele agente. Caso haja alterações, os dados devem ser atualizados no arquivo comum. Se mesmo de posse de todos os dados não for possível firmar a comunicação, o agente deve mandar um e-mail informando o problema para o responsável pelo agente.

3.5 Saída de agente da sociedade

Caso o usuário queira retirar o seu agente da sociedade (desativá-lo), este deverá executar um programa, para pedir que este agente seja retirado da sociedade. Este programa pedirá que seja fornecido o e-mail do usuário, a função desempenhada pelo agente e o número da máquina. De posse destas informações o programa irá eliminar o nome do agente do arquivo de agentes da máquina em questão, e depois fará uma conexão com o *Agent Manager* e pedirá que seja retirado do arquivo de agentes da sociedade o nome deste agente.

No caso daqueles agentes que trabalham com um IP dinâmico, a saída destes agentes será tratada de duas maneiras:

1) Caso o agente apenas seja desconectado da rede, mas irá retornar na próxima vez que a sua máquina for conectada a Internet, o agente irá informar este fato para o *Agent Manager*. O *Agent Manager* estando a par deste fato, irá verificar em seu arquivo local de cadastro de agentes, qual o registro deste arquivo que está ocupado pelos dados deste agente. Feito isto, os dados do agente não serão deletados, mas será colocado um *flag* junto ao registro, informando que está temporariamente indisponível. O agente que informou sua saída da rede, receberá do *Agent Manager*, o número correspondente ao seu registro.

No momento de retorno deste agente para a sociedade, ele não necessitará ser recadastrado, apenas fará uma conexão via *socket* com seu *Agent Manager*, e informará seu número de registro dentro do arquivo local de endereços de agentes e o seu novo IP. O *Agent Manager*, de posse destas informações, irá reativar o registro do agente.

2) Caso o agente seja desativado pelo usuário, o procedimento será feito de maneira idêntica ao de qualquer outro agente.

3.6 Exemplo de utilização da arquitetura

Neste tópico, será apresentado um exemplo do uso da arquitetura. Para a exemplificação será utilizado o esquema apresentado na figura 3.3 abaixo:

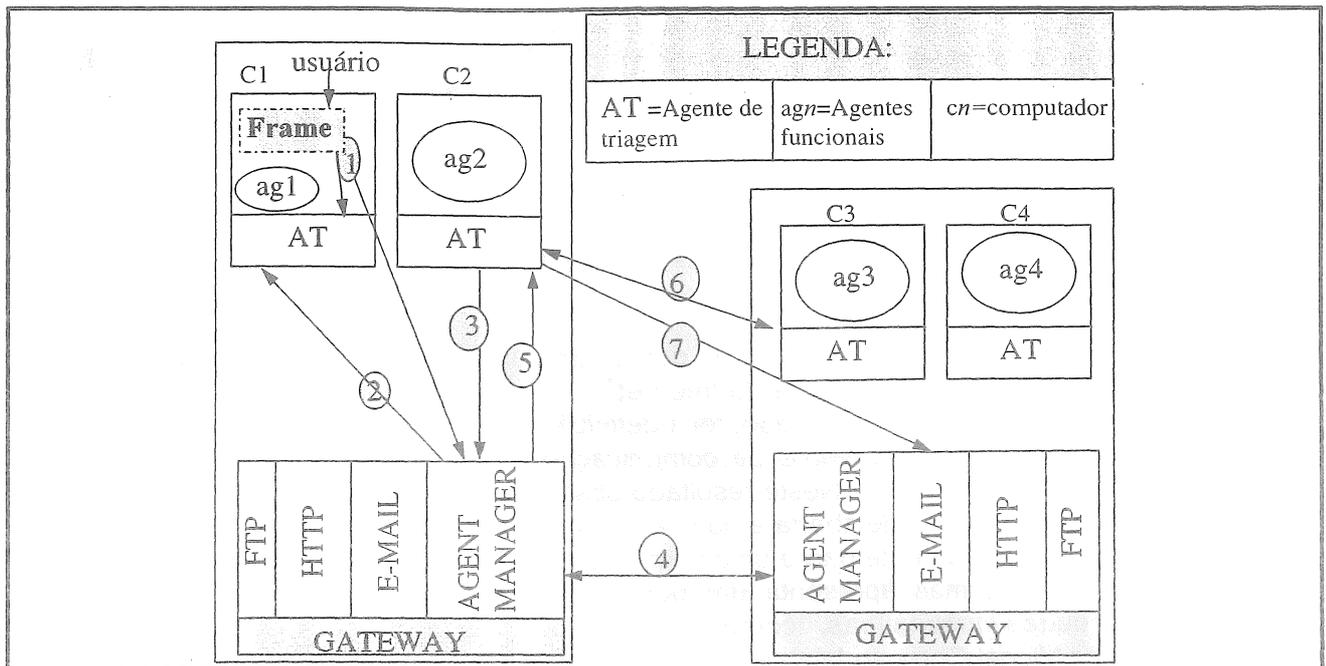


FIGURA 3.3 - Exemplo de utilização da Arquitetura proposta

1) Na primeira etapa do esquema, será suposto que o agente1 (ag1) ainda não é parte integrante da sociedade. Sendo assim, este deve apresentar-se para o *Agent Manager* da sua rede, e para o *Agente de Triagem* de sua máquina, para ser cadastrado.

Nesta etapa cabe ao proprietário executar o frame de cadastramento e passar as informações sobre o novo agente. No esquema, esta etapa está sendo representada pelo item número 1.

Depois do programa de cadastramento ter informado ao *Agente de Triagem* e ao *Agent Manager* da rede, da presença deste novo agente, o próprio *Agent Manager* cria um *socket* de comunicação com o novo agente para verificar se ele está devidamente cadastrado e respondendo (representado no esquema pelo item 2).

2) Na segunda etapa do exemplo, supõe-se que o agente2 (ag2) e o agente3 (ag3) desempenham a função de agentes agenda. O usuário do agente2 pediu para que fosse marcada uma reunião com o agente3.

Para poder marcar a reunião com o agente desejado, primeiramente o agente2 procura o endereço deste agente no arquivo local de endereços de agentes de sua máquina (função realizada na verdade pelo *Agente de Triagem*). Não o encontrando, entra em contato, via *socket*, com o *Agent Manager* da rede e solicita o endereço do agente desejado (item 3 do esquema). Neste exemplo, apresenta-se a comunicação entre os agentes como se fossem diretas, pois fica transparente a participação do *Agente de Triagem* nesta atividade.

Como neste caso o agente desejado não faz parte da sub-rede do *Agent Manager*, este terá de estabelecer um *socket* de comunicação com a máquina que possui o domínio referenciado no nome do agente e entrar em contato com o *Agent Manager* desta outra rede (item 4 do esquema), e finalmente solicitar o endereço desejado. De posse deste endereço, ele é passado para o agente2 (item 5), e este agora pode entrar em contato com o agente3.

Deve-se observar que este acesso a outro *Agent Manager* ocorreu pelo fato do agente em questão não estar na mesma rede do agente2.

3) Na terceira etapa (item 6) o agente2 estabelece um *socket* de comunicação, com o agente3 (no esquema a conexão é apresentada de maneira direta ag2 → ag3, os detalhes de

conexão através das camadas da rede, não são abordados, pois fica transparente ao usuário), sendo feita a partir deste momento toda a interação exigida por um agente agenda. Após o estabelecimento da conexão, e depois de toda a negociação ter terminado, a conexão é desfeita e o endereço do agente3 é armazenado no arquivo local de endereços da máquina do agente2.

Supondo que mesmo de posse do endereço completo do agente3, o agente2 não tenha conseguido estabelecer conexão. Depois de algumas tentativas frustradas, o agente2 mandará um e-mail informando o ocorrido para o usuário responsável por este agente (item 7).

Conclusões

O objetivo geral deste trabalho, foi o de definir uma arquitetura genérica que facilitasse a implementação de agentes no ambiente da Internet.

Resultado de todo o estudo realizado, foi a definição de uma arquitetura, que é robusta, no aspecto de facilitar o estabelecimento de comunicação entre agentes que estão no mesmo domínio ou em domínios distintos. Neste resultado observa-se a definição de uma arquitetura, a qual não trata de uma sociedade aberta e nem fechada, mas sim de uma sociedade mista.

Esta arquitetura certamente não é única e possivelmente não seja a melhor para algumas classes de aplicações, mas apresenta um ponto que é de grande peso a seu favor, se comparada a outras arquiteturas, como a proposta pelo sistema SodaBot [COE 94], desenvolvido por Michel Coen no laboratório de Inteligência Artificial do MIT e o sistema JAT [JAT 96] (*Java Agent Template*) desenvolvido por Rob Frost da Universidade de Stanford. Esta vantagem, esta no fato de que nesta arquitetura proposta, existe a possibilidade de utilizar toda a estrutura já definida para a Internet, como é o caso de utilizar-se das máquinas gateways das redes e dos domínios já existentes.

Bibliografia

- [AEK 95] AEKEN, F.Van et al. A Multi-Agent Approach for Mediation Support on the net. In: DIMAS, 1., 1995, Cracóvia. *Proceedings...* [S.l.:s.n.], 1995.
- [CAZ 97] CAZELLA, Sívio César. *Uma arquitetura para coordenar a interação de agentes na Internet*. Porto Alegre: CPGCC da UFRGS, 1997. Dissertação de mestrado.
- [COE 94] COEN, Michel H. *SodaBot: A software Agent Environment and Construction System*. Disponível por FTP anonymous em publications.ai.mit.edu. Arquivo aitr-1493.ps.Z. Massachusetts Institute of Technology, 1994. (23 jul. 1996).
- [DEM 93] DEMAZEAU, Yves. Distributed Artificial Intelligence & Multi-Agent Systems. In: Simpósio Brasileiro de Inteligência Artificial, 10., Porto Alegre. *ANAIS...* Porto Alegre : SBC, 1993.
- [ETZ 95] ETZIONI, Oren & WELD, Daniel S. Intelligent Agents on the Internet: Fact, Fiction, and Forecast. *IEEE Expert*. New York, v. 10, n. 4, p. 44-49, Aug 1995.
- [HED 95] HEDBERG, Sara Reese. Intelligent Agents: The First Harvest of softbots looks promising. *IEEE Expert*. New York, v. 10, n. 4, p. 06-09, Aug 1995.
- [JAT 96] *Java Agent Tamplate*. Disponível em <http://cdr.stanford.edu/ABE/JavaAgent.html> (18/08/96).
- [SEA 95] SEARS, Jay Allen. Harnessing the World Wide Web. *IEEE Expert*. New York, v. 10, n. 4, p. 42-43, Aug 1995.