

GESTÃO DE SERVIÇOS COMPUTACIONAIS NUMA REDE DE COMPUTADORES ATRAVÉS DE UM AGENTE INTELIGENTE

Jose Elias da Silva Justo (IFF)

sjjusto@gmail.com

Annabell del Real Tamariz (UNF)

sjjusto@gmail.com



O objetivo deste trabalho é desenvolver um protótipo baseado em um agente de software, que auxilie na gestão de serviços numa rede de computadores, monitorando os serviços prestados e intervindo reativamente, reiniciando quaisquer dos serviços com problemas. O protótipo deverá analisar sua base de conhecimento, periodicamente, para verificar serviços que tiveram que ser reiniciados constantemente e relatar tal situação, para que a equipe de administradores possa agir proativamente e reconfigurá-los antes que sofram alguma parada permanente.

Palavras-chaves: gestão, redes de computadores, agente inteligente

1. Introdução

Com a evolução da Tecnologia da Informação (TI) os *hardwares* computacionais adquiriram um grande poder de processamento de dados e os *softwares*, funcionalidades inimagináveis há três décadas, possibilitando a criação e popularização do computador pessoal (PC), a princípio nas médias e pequenas empresas e depois em escritórios e casas. Porém com a globalização e a internet, o volume de dados a serem processados cresceu mais rápido que o poder computacional, surgindo à necessidade de interligar os computadores em rede para processar e compartilhar tais dados (KUROSE, 2003).

Uma rede de computadores é um conjunto de computadores autônomos interconectados para compartilhar recursos como hardware, software, interação e interatividade (SICA, UCHÔA e SIMEONE, 2003), em que há máquinas que assumem os papéis de clientes, servidores e/ou parceiros.

Administrar redes de computadores não é uma tarefa trivial, administrar redes com cerca de mil e quinhentos computadores clientes, trinta servidores dedicados e mais de dois mil usuários é muito complexo, pois exige trabalho conjunto de grandes equipes de administradores de redes. Monitorar os serviços dessas redes e agir preventivamente para evitar paradas inesperadas e prejuízos é o anseio das instituições, mas com a tecnologia utilizada nos sistemas de monitoramento atuais, apenas mensagens, normalmente *e-mails*, são enviadas para avisar que determinado problema, como por exemplo, o serviço de *proxy* estar parado.

O objetivo deste trabalho é desenvolver um protótipo de um sistema de gestão de serviços em servidores de redes de computadores, baseado em um agente inteligente, que monitore serviços de redes e intervenha reativamente, reiniciando quaisquer dos serviços que deveriam estar ativos no servidor e caso algum destes serviços não seja restabelecido, envie um alerta, por *e-mail*, aos administradores, informando os serviços e os *logs* resultantes da tentativa de iniciá-los, para que a equipe já tenha algumas pistas do que está ocorrendo de errado no servidor. Além dessa atribuição, o sistema deverá analisar sua base de conhecimento, periodicamente, para verificar serviços que tiveram que ser reiniciados constantemente e relatar tal situação, para que a equipe de administradores possa agir proativamente e reconfigurá-los antes que causem alguma parada permanente no sistema.

A motivação para desenvolver este trabalho veio da própria experiência do autor que faz parte da equipe que administra a rede de computadores do Instituto Federal Fluminense (IFF), em que constantemente ocorrem paradas de serviços, sendo que as ferramentas em uso atualmente não possibilitam reiniciá-los, nem apresentam relatórios das possíveis causas da interrupção, apenas sinalizam que algo está errado. Nesse contexto é comum os administradores saberem que algo está errada na rede somente quando os usuários reclamam que não conseguem realizar alguma tarefa, o que prejudica o bom funcionamento das atividades administrativas e acadêmicas da instituição.

2. Pesquisa relacionada

O São identificadas oito áreas de aplicação onde agora ou no futuro próximo a tecnologia de agente é ou será usada. Neste trabalho propõe-se um agente para trabalhar na área definida como Sistemas e Administração de Redes (BIANCHINI et al. , 2003). Esta é uma das mais novas áreas de aplicação que usa tecnologia de agente inteligente. A evolução da computação para cliente/servidor intensificou a complexidade de sistemas que são administrados, especialmente em redes LANs (*Local Area Network* ou redes locais), e como redes de

serviços centralizados estão em plena expansão, essa complexidade aumenta progressivamente. Os usuários, principalmente os operadores e administradores de sistemas, necessitam de uma forma simplificada de gerenciar tais redes. Arquiteturas de agentes são aplicadas nos sistemas e na área de administração de rede há algum tempo, mas estes são geralmente, apenas agentes móveis, desprovidos de inteligência. Porém, podem ser usados agentes inteligentes para melhorar a performance e eficiência dos softwares de administração de redes de computadores. Por exemplo, eles podem ajudar a filtrar e a executar ações automáticas a um nível mais alto de abstração e podem até mesmo serem usados para descobrir e reagir a padrões de comportamento de sistema.

Segundo a *Foundation for Intelligent Physical Agents* - FAPI (1997), um agente inteligente fisicamente constituído deve apresentar as seguintes características:

- autonomia: operar sem intervenção direta de humanos e adapta-se ao ambiente;
- habilidade social: interagir com outros agentes ou humanos;
- reatividade: perceber o ambiente onde está e responder em tempo hábil a mudanças ocorridas nele;
- pró-atividade: comportamento direcionado a um objetivo, tomando a iniciativa de ação;
- mobilidade: locomover-se para outros ambientes;
- continuidade temporal: funcionar continuamente.

No entanto, posto que o ciberespaço e as redes de computadores são formados através da interligação entre computadores, os modelos de agentes que mais se adequam ao seu estudo são os de agentes de *software*.

No estágio atual em que se encontra o desenvolvimento dos modelos de inteligência aplicados à construção de agentes de *software*, as abordagens mais bem sucedidas estão orientadas à solução de problemas com contornos muito bem delimitados, tais como os que ocorrem durante a execução de atividades rotineiras, complexas e colaborativas nas áreas de engenharia, gerência e manufatura (STANFORD KSL, 1998). Outras aplicações práticas de agentes inteligentes têm sido a construção de assistentes de comunicação pessoal (JANSON, 1998), capazes de cuidar de vários detalhes como organizar correio eletrônico (SEGAL e KEPHART, 1999), realizar filtragem colaborativa de informações (STARR, ACKERMAN e PAZZANI, 1996), organizar agendas de compromissos, gerenciar reserva de hotéis e passagens aéreas.

3. Especificações do agente inteligente

Segundo Russel e Norvig (2004), um agente é qualquer coisa ou ser que é capaz de perceber o ambiente em que está inserido, por meio de sensores e, se necessário, agir sobre o mesmo com seus atuadores.

Um agente racional executa ações acertadas, consideradas corretas, boas para realizar as mudanças necessárias no ambiente, obtendo sucesso. Para definir o que é obter sucesso e então completar o conceito de racionalidade, há o que se chama de medida de desempenho, que mede o sucesso do comportamento do agente para saber se fez tudo certo, averiguando se os estados provocados pelas ações do agente, pelos quais o ambiente passa, é a sequência desejável, assim, então, o agente funcionou bem, obteve sucesso, foi racional (RUSSEL e NORVIG, 2004).

O agente em estudo agirá reativamente. As mudanças ocorrem no ambiente, ele as percebe e então reage adaptando-se ao ambiente ou atuando neste para que retorne às condições ideais de funcionamento (CARBONELL e HOOD, 1986).

Um agente racional deve selecionar uma ação que se espera venha a maximizar sua medida de desempenho para cada seqüência de percepções possível, dada à evidência fornecida pela mesma e por um conhecimento interno do agente, caso exista. Tal agente é considerado então como agente inteligente (LUGER, 2004). A seguir apresentamos o ambiente onde o agente irá trabalhar.

3.1. Ambiente

O agente atua numa pequena rede de computadores, localizada na sala administrativa do Departamento de Tecnologia da Informação do IFF, composta de 1 computador servidor e 2 clientes, denominados aqui *cliente1* e *cliente2*. O servidor e o *cliente1* possuem sistema operacional *Linux*, o *cliente2* possui sistema operacional *Windows XP*.

São executados no servidor e serão avaliados os seguintes serviços: *samba*, *proxy* e *web*. O primeiro é um servidor de arquivos, o segundo é serviço de *proxy* e *cache* para acesso a internet e o último, é um serviço de páginas *web*, implementado através do servidor *apache*.

3.2. Função do agente

O agente que será apresentado neste artigo tem a função de auxiliar na administração dos serviços de um computador servidor numa pequena rede de computadores, localizada no IFF, em Campos dos Goytacazes.

3.3. Objetivos do agente

Ao ser inserido no ambiente o agente deverá fazer continuamente uma coleta de dados para descobrir se houve mudanças nele, por exemplo, uma máquina que foi desligada, um serviço que parou de funcionar ou não está funcionando adequadamente, etc. Os passos seguidos pelo agente para descobrir e monitorar o ambiente são: primeiro, ele descobre quantas e quais das duas máquinas clientes da rede estão ligadas e ativas, sendo que o computador servidor tem que obrigatoriamente estar ligado para que o agente esteja funcionando, pois o mesmo é instanciado e fica rodando no servidor. O agente administra os serviços que estão disponíveis no servidor.

O segundo passo é mapear os serviços ativos no servidor e conferir, com o seu conhecimento prévio, definido na base de conhecimento, para saber se todos os serviços que deveriam estar ativos estão e se há algum serviço novo em execução.

Na terceira etapa, o agente já conhece o seu ambiente e começa a monitorá-lo através dos seus sensores para perceber possíveis mudanças, como por exemplo, serviços que estavam ativos e agora, estão inativos, analisar as causas das mesmas e tentar restaurar as condições ideais de funcionamento do ambiente, através dos seus atuadores. Caso, após várias tentativas, não consiga, enviará um alerta, via e-mail para a pessoa que tem a função de administrador da rede, com o relatório do que está acontecendo no ambiente e o que tentou fazer para restabelecê-lo. O administrador analisará o problema, resolverá e informará ao agente, via base de conhecimento, qual foi o problema e como o solucionou, para que o mesmo aprenda.

Segundo De Franceschi, Barreto e Roisenberg, M. (2000), um agente inteligente é autônomo quando consegue se adaptar às mudanças que ocorrem no ambiente. O agente projetado neste trabalho além de racional é autônomo, o que significa que caso o ambiente que ele tem conhecimento prévio, informado pelo projetista seja modificado, como por exemplo, acrescentando-se mais computadores clientes na rede, ele explorará o novo ambiente, aprenderá sobre o mesmo e passará a monitorar as novas máquinas.

3.4. Medidas de desempenho

O agente tem um bom desempenho se a latência entre computadores da LAN for inferior a 1ms (milissegundo) e a da LAN para a internet for até 300ms. Outra medida adotada é a de estar em estado ativo o máximo de serviços do servidor.

A medida de desempenho premia o agente com 1 ponto para cada uma das três medidas apresentadas acima. Para conceder a pontuação o agente verifica se as latências, durante intervalos de tempo, estão dentro dos limites estabelecidos e contabiliza a quantidade de serviços ativos. Esses testes são realizados de duas formas: a primeira é feita apenas em dias úteis, em intervalos cíclicos de cinco minutos, começando às 8h, que é o horário de início de atividades dos usuários da pequena rede de computadores e encerra-se às 18h. Nesse caso tem-se 120 medições, sendo que em cada uma pode-se obter 3 pontos no máximo, somando um total de 360 pontos. Caso o agente alcance um percentual igual ou maior a 95%, considera-se, neste trabalho, que o agente obteve bom comportamento/desempenho e foi racional, pois está gerindo bem os serviços.

A segunda forma de testar a medida de desempenho é das 18h do dia atual até as 8h do dia subsequente, nos dias úteis e por 24h consecutivas, a partir das 8h da manhã nos sábados, domingos e feriados. Os testes são realizados, também a cada cinco minutos, mas, não averigua a latência entre as máquinas da rede local, pois só o servidor está ligado. Os outros testes são realizados normalmente, sendo possível alcançar o máximo de 240 pontos. O percentual para alcançar um bom comportamento é o mesmo.

Caso o agente alcance a faixa de 80 a 94% da pontuação máxima da medida de desempenho, ele estará com um comportamento razoável, sinalizando que atuou de forma aceitável. Abaixo de 80%, o agente está com uma péssima atuação e informará imediatamente ao administrador da rede que é necessário uma análise profunda da sua base de conhecimento, onde estão registrados os testes e os resultados dos mesmos.

Nas duas formas de testar, sempre que o agente não alcançar a porcentagem mínima de sucesso, ele enviará relatório para o administrador da rede, contendo todas as estatísticas dos testes e detalhará os momentos em que não alcançou o máximo de pontuação possível.

Se houver falhas nos equipamentos de rede, como *switch*, cabos, roteador, ou falhas no *hardware* do servidor, o agente terá um mau desempenho, mesmo sem ter culpa alguma, porém neste contexto o desempenho abaixo do mínimo esperado é algo bom, pois a administração da rede será informada do mau desempenho e pelo relatório perceberá que há falhas em equipamentos, o que poderia passar dias despercebidos caso o agente não informasse.

3.5. Sensores e atuadores

Os sensores do agente são comandos do sistema operacional *Linux*, que está instalado no servidor; eles monitoram os serviços da máquina servidora e a conectividade dela com as demais máquinas da LAN e algumas da internet.

Os atuadores são comandos do sistema operacional que gerenciam os serviços disponíveis no servidor, além dos que reiniciam e desligam o computador.

3.6. Estados do ambiente – tabulação da função do agente

Para que o agente fosse desenvolvido neste trabalho, era necessário apresentar quais seriam os estados possíveis do ambiente e as ações correspondentes que o agente teria que executar para que o ambiente estivesse em perfeito funcionamento.

Na Tabela 1 estão especificadas as sequências de percepções que o agente tem que identificar com os sensores e as respectivas ações que deve executar com os atuadores.

Sequência de Percepções	Ação
[Servidor, Desligado]	-- (agente inoperante)
[Servidor, sem acesso à LAN]	Verificar configuração; Reiniciar serviço de rede; Testar comunicação LAN;
[Servidor, sem acesso à WAN (<i>Wide Área Network</i> , no contexto deste trabalho, internet)]	Verificar configuração; Reiniciar serviço de rede; Testar comunicação LAN; Testar equipamento WAN; Testar comunicação WAN;
[Servidor, Serviço de arquivos samba inativo]	Reiniciar serviço de rede; Iniciar serviço;
[Servidor, Serviço de arquivos samba sem aceitar conexões remotas do <i>cliente1</i>]	Verificar porcentagem de ocupação do disco rígido; Verificar configuração; Reiniciar serviço; Verificar <i>firewall</i> do servidor; Verificar <i>firewall</i> do <i>cliente1</i> ;
[Servidor, Serviço de arquivos samba sem aceitar conexões remotas do <i>cliente2</i>]	Verificar porcentagem de ocupação do disco rígido; Verificar configuração; Reiniciar serviço; Verificar <i>firewall</i> do servidor; Verificar <i>firewall</i> do <i>cliente2</i> ;
[Servidor, Serviço de páginas <i>web</i> (apache) inativo]	Reiniciar serviço de rede; Iniciar serviço;
[Servidor, Serviço de páginas <i>web</i> (apache) sem aceitar conexões remotas do <i>cliente1</i>]	Verificar porcentagem de ocupação do disco rígido; Verificar configuração; Reiniciar serviço; Verificar <i>firewall</i> do servidor; Verificar <i>firewall</i> do <i>cliente1</i> ;
[Servidor, Serviço de páginas <i>web</i> (apache) sem aceitar conexões remotas do <i>cliente2</i>]	Verificar porcentagem de ocupação do disco rígido; Verificar configuração; Reiniciar serviço; Verificar <i>firewall</i> do servidor; Verificar <i>firewall</i> do <i>cliente2</i> ;
[Servidor, Serviço de páginas <i>web</i> (apache) sem aceitar conexões remotas da WAN]	Verificar porcentagem de ocupação do disco rígido; Verificar configuração; Reiniciar serviço; Verificar <i>firewall</i> do servidor; Testar equipamento WAN; Testar comunicação WAN;
[Servidor, Serviço de <i>proxy</i> transparente inativo]	Reiniciar serviço de rede; Iniciar serviço;
[Servidor, Serviço de <i>proxy</i> transparente não permitindo a navegação do <i>cliente1</i>]	Verificar porcentagem de ocupação do disco rígido; Verificar configuração; Reiniciar serviço; Verificar <i>firewall</i> do servidor; Verificar <i>firewall</i> do <i>cliente1</i> ; Testar equipamento WAN; Testar comunicação WAN; Testar serviço DNS (<i>Domain Name System</i>) a partir do <i>cliente1</i> ;
[Servidor, Serviço de <i>proxy</i> transparente não permitindo a navegação do <i>cliente2</i>]	Verificar porcentagem de ocupação do disco rígido; Verificar configuração; Reiniciar serviço; Verificar <i>firewall</i> do servidor; Verificar <i>firewall</i> do <i>cliente1</i> ; Testar equipamento WAN; Testar comunicação WAN; Testar serviço DNS a partir do <i>cliente2</i> ;
[<i>Cliente1</i> , Desligado], [Dias: sábado / domingo / feriado]	NoOP (não realizar ação alguma)
[<i>Cliente1</i> , Desligado (ou com rede inativa)], [Dias úteis]	Enviar e-mail informando o fato ao administrador
[<i>Cliente2</i> , Desligado (ou com rede inativa)], [Dias úteis]	Enviar e-mail informando o fato ao administrador
[<i>Cliente2</i> , Desligado], [Dias: sábado / domingo / feriado]	NoOP (não realizar ação alguma)

Tabela 1 – Estados do ambiente e as ações do agente

4. Implementação do agente

Nesta etapa, trabalha-se a transformação do modelo de agente, desenvolvido na seção 3, em um protótipo que implementa algumas características e funções da arquitetura proposta.

4.1. Base de conhecimento

A base de conhecimento do agente está organizada na forma de cálculo de predicados, implementada em *Prolog* (PALAZZO, 1997), uma linguagem de programação muito utilizada em projetos de inteligência artificial.

O agente tem um conhecimento prévio do ambiente, chamado de conhecimento inicial, fornecido pelo projetista e será sempre utilizado como ponto de partida, com informações básicas sobre o ambiente, tais como: quantidade de máquinas na rede, neste caso são 3; endereço IP (*Internet Protocol*) de cada computador; qual computador é o servidor, *cliente1* e *cliente2* e os serviços que devem estar disponíveis no servidor.

A partir da base de conhecimento inicial, o agente verifica o ambiente em busca de mais serviços de rede no servidor, verifica se os clientes estão ligados e se há mais computadores na rede. Conforme o projeto, o agente fará verificações a cada cinco minutos para acompanhar os estados do ambiente e com tais informações, criará uma base de conhecimento dinâmica, diferente da base inicial que é estática, que guarda as informações dos estados do ambiente, dos problemas ocorridos na rede e do processamento realizado pelo agente para tentar resolvê-los. Esta base é também utilizada para gerar os relatórios sobre o funcionamento dos serviços na rede. A Figura 1 ilustra tal base, onde é mostrado o estado atual dos serviços, ativo ou inativo, a data das observações do ambiente, o dia da semana e o horário.

```
situacao(ativo(web),data(23,03,2011,quarta),horario(11,23)).  
situacao(ativo(samba),data(23,03,2011,quarta),horario(11,23)).  
situacao(ativo(proxy),data(23,03,2011,quarta),horario(11,23)).  
situacao(ativo(web),data(23,03,2011,quarta),horario(11,28)).  
situacao(ativo(samba),data(23,03,2011,quarta),horario(11,28)).  
situacao(ativo(proxy),data(23,03,2011,quarta),horario(11,28)).  
situacao(ativo(web),data(23,03,2011,quarta),horario(11,33)).  
situacao(inativo(samba),data(23,03,2011,quarta),horario(11,33)).  
situacao(ativo(proxy),data(23,03,2011,quarta),horario(11,33)).  
situacao(ativo(web),data(24,03,2011,quinta),horario(08,42)).  
situacao(ativo(samba),data(24,03,2011,quinta),horario(08,42)).  
situacao(inativo(proxy),data(24,03,2011,quinta),horario(08,42)).  
situacao(inativo(web),data(24,03,2011,quinta),horario(08,47)).  
situacao(ativo(samba),data(24,03,2011,quinta),horario(08,47)).  
situacao(ativo(proxy),data(24,03,2011,quinta),horario(08,47)).  
situacao(inativo(web),data(25,03,2011,sexta),horario(10,53)).  
situacao(ativo(samba),data(25,03,2011,sexta),horario(10,53)).  
situacao(inativo(proxy),data(25,03,2011,sexta),horario(10,53)).
```

Figura 1 – Base de conhecimento do agente.

A partir da atuação do sensor, como por exemplo, na Figura 1, às 11h e 03min, do dia 25/03/2010, constata-se que os serviços *web* e *proxy* estão inativos, o atuador verifica tal situação, como a mais atual da base de conhecimento e tenta inicializar tais serviços.

4.2. Sensores

Os sensores do agente foram implementados através de comandos do sistema operacional *Linux*, tais como: *ping*, *netstat*, *telnet* e *traceroute*. Na Figura 2 é mostrado um trecho do pseudocódigo do procedimento que implementa o sensor do agente.

```
Procedimento SensorAgente  
1 verificar status serviço web;  
2 verificar status serviço samba;  
3 verificar status serviço proxy;  
4 atualizar base de conhecimento;  
fim SensorAgente;
```

Figura 2 – Pseudocódigo do sensor do agente

4.3. Atuadores

Os atuadores também foram implementados utilizando-se comandos do sistema operacional, como por exemplo: *reboot*, *halt*, *service <nomeServiço> [start/restart]*. O procedimento que implementa os atuadores é ilustrado na Figura 3.

```
Procedimento AtuadorAgente
1 verificar serviços com status de inativo na base de
  conhecimento;
2 servicosInativos ← vetor de serviços inativos;
3 tamVetor ← comprimento(servicosInativos);
4 i ← 0;
5 enquanto (i <= tamVetor) faça
6     se (inicializar(servicosInativos[i]) = falso) então
7         ArquivoRelatorio ← logs inicialização serviço;
8     fim-se;
9 fim-enquanto;
10 enviar e-mail com conteúdo arquivoRelatorio;
fim AtuadorAgente;
```

Figura 3 – Pseudocódigo do atuador do agente

O sensores e o atuadores foram implementados em *shell script* do sistema operacional *Linux*.

5. Conclusões

Neste trabalho apresentou-se o modelo de um protótipo de agente racional que auxilia na gestão de serviços numa rede de computadores. Até o momento conseguiu-se implementar e executar o agente para que identifique serviços inativos, reinicialize-os e forneça relatórios das ações realizadas para os administradores, oferecendo todos os detalhes observados durante um determinado intervalo de tempo, se assim é solicitado, ou periodicamente caso o agente não consiga reativar os serviços inativos.

O agente mostrou-se eficaz dentro dos objetivos traçados, mantendo os serviços em funcionamento e gerando os relatórios, auxiliando na administração da rede. O agente projetado neste trabalho além de racional também é autônomo, pois adapta-se às mudanças que ocorrem no ambiente.

Referências

BIANCHINI, C. B.; ALMEIDA, E. D.; FONTES, D. S. & ANDRADE, R. M. C. *Um Padrão para Gerenciamento de Redes*. In: III Conferência Latino-Americana em Linguagens de Padrões para Programação, Porto de Galinhas. Anais... Porto de Galinhas, 2003.

CARBONELL, J.G. & HOOD, G. *The World Modelers Project: learning in a reactive environment*. In *Machine Learning: A Guide to Current Research*. Kluwer Academic Press, p. 29-34, 1986.

DE FRANCESCHI, A.S.M.; BARRETO, J.M. & ROISENBERG, M. *Desenvolvimento de Agentes Autônomos em Gerência de Redes de Computadores*. In: XVIII Simpósio Brasileiro de Telecomunicações, Gramado. Anais... Gramado, 2000.

FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS - FIPA. www.fipa.org, junho, 1997.

JANSON, S. *Intelligent Software Agents*. www.sics.se/isl/abc/survey.html, setembro, 1998.

KUROSE, J. F. *Redes de Computadores e a Internet: uma nova abordagem*. São Paulo: Addison Wesley,

2003.

LUGER, G. F. *Inteligência Artificial: estruturas e estratégias para a solução de problemas complexos.* Tradução Paulo Martins Engel. 4ª edição – Porto Alegre: Bookmann, 2004.

PALAZZO, L. A. M. *Introdução a Programação Prolog.* Pelotas: UCPEL, 1997.

RUSSEL, S. J. & NORVIG, P. *Inteligência Artificial.* Tradução Publicare Consultoria. 2ª edição – Rio de Janeiro: Elsevier, 2004.

SEGAL, R. & KEPHART, J. *MailCat: an Intelligent Assistant for Organizing E-mail.* In: Proceedings of the Third International Conference on Autonomous Agents. www.research.ibm.com/swiftfile, maio, 1999.

SICA, F. C.; UCHÔA, J. Q. & SIMEONE, L. E. *Administração de Redes Linux.* Lavras, MG: Editora UFLA/FAEPE, 2003.

STANFORD KNOWLEDGE SYSTEMS LABORATORY – KSL. *How Things Work Demonstrations.* www.ksl.stanford.edu/htw/htw-demos.html, setembro, 1999.

STARR, B.; ACKERMAN, M. & PAZZANI, M. *Do-I-Care: A Collaborative Web Agent.* In: Proceedings of the ACM CHI'96 Conference, 1996.