

DESENVOLVENDO AGENTES DE SOFTWARE PARA GERÊNCIA DE REDES UTILIZANDO TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

Analúcia S. M. de Franceschi*

analucia@ieb.ufsc.br

Jorge M. Barreto**

barreto@inf.ufsc.br

Mauro Roisenberg**

mauro@inf.ufsc.br

* Doutoranda do Programa de Pós-Graduação em Engenharia Elétrica

Instituto de Engenharia Biomédica

** Departamento de Informática e Estatística, Laboratório de Conexionismo e Ciências Cognitivas

Universidade Federal de Santa Catarina

RESUMO

O presente trabalho apresenta uma metodologia para desenvolvimento de agentes de software inteligentes para gerência de redes de computadores. Dois tipos de agentes são identificados a partir do enunciado do problema: agentes inteligentes estáticos e agentes inteligentes dinâmicos. Agentes estáticos podem ser implementados utilizando-se heurísticas obtidas através de um especialista ou administrador de redes de computadores. Este conhecimento será utilizado para construir regras de produção ou redes neurais diretas. Ao contrário, para se construir agentes dinâmicos utiliza-se exemplos obtidos da própria rede. É possível utilizar as cinco áreas funcionais propostas pelo modelo de referência OSI para classificar a necessidade de agentes estáticos ou dinâmicos.

Palavras-chave: Agentes de Software, Técnicas de Inteligência Artificial, Redes Neurais, Gerência de Redes.

ABSTRACT

This work presents a methodology to develop intelligent agents for network management. Two distinct kinds of intelligent agents were identified: static or dynamic agents. The first one can be implemented, using heuristics obtained from an expert or the network administrator, through production rules or feed forward neural networks. Using the network examples we can construct dynamic agents. The neural network may be trained to solve a problem using some examples. We divided the network management in the five functional areas proposed by OSI Model Reference. Thus, each area has a different intelligent solution.

Key-words: Software Agents, Artificial Intelligence Techniques, Neural Networks, Network Management.

1 INTRODUÇÃO

Existem várias abordagens na área de Inteligência Artificial (IA), nas quais variam a manipulação do conhecimento, no sentido de como adquiri-lo, armazená-lo e empregá-lo. Classificando a IA quanto ao método de solução de problemas tem-se a IA Simbólica (IAS), a IA

Conexionista (IAC), a IA Evolucionária (IAE) e a IA Híbrida (IAH) BARRETO (2001). Quanto a localização espacial tem-se ainda a IA Monolítica (IAM) e a IA Distribuída (IAD) BARRETO (2001). A IAS possui como ferramenta básica para manipular o conhecimento a lógica, com suas regras de inferência inspiradas nos silogismos enunciados há mais de 2.000 anos por Aristóteles BARRETO (2001). A IAC usando redes neurais artificiais (RNAs), aplica-se a problemas mal definidos, mas que são conhecidos através de exemplos. Entre os campos de aplicações de técnicas conexionistas, das quais destacam-se as RNAs, estão: reconhecimento de padrões; controle de processos industriais; robótica; e também, como opção às técnicas de raciocínio baseado em casos para a resolução de problemas. Na IAE, os mecanismos utilizados são os mesmos encontrados na evolução biológica. É um exemplo de solução de problema bem definido de sobrevivência de uma espécie em ambiente variável. Pode ser encarada como um método de otimização com restrições variáveis e muitas vezes desconhecidas. A IAH reúne vantagens de mais de um tipo de método de abordagem para a resolução de problemas. A IAM é o modo de implementar soluções de IA como uma unidade indivisível. Muitas vezes são sistemas simples sem modularidade, como é o caso de muitos sistemas especialistas que utilizam uma base de conhecimento única e podem funcionar em um computador. O funcionamento da IAD depende de um determinado conjunto de partes (ou módulos), que funcionam de modo relativamente independente, para resolver de modo cooperativo um determinado problema. Tradicionalmente a IAD começou a ser usada empregando o paradigma conhecido como do “quadro negro”. Neste paradigma a comunicação entre cada módulo se faz imitando uma empresa em que todos os funcionários se comunicam por mensagens deixadas em um quadro negro. Todos lêem e podem escrever no quadro, mas não tomam conhecimento de nada mais do que o foi aí escrito. Um modelo mais cooperativo é o de agentes.

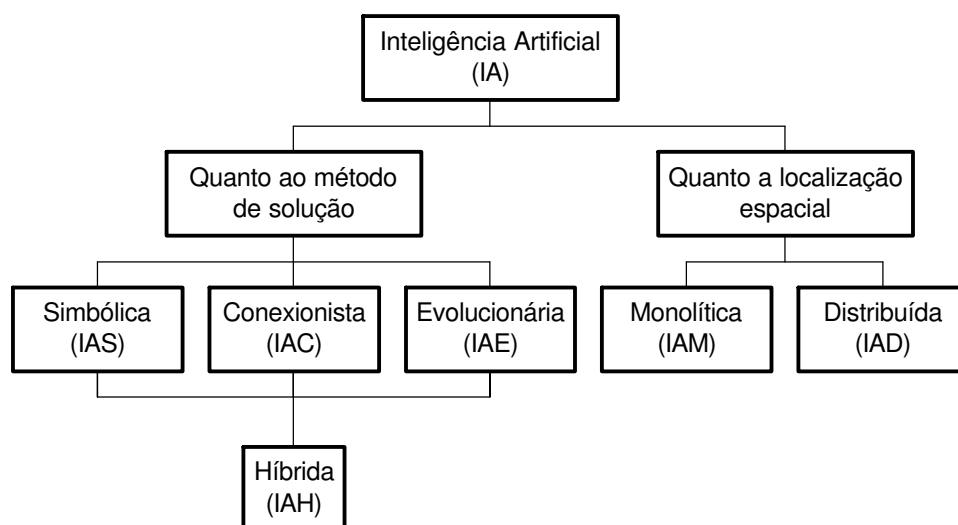


Figura 1: Várias abordagens de IA

O presente trabalho está organizado em seis seções. Uma classificação sobre agentes na área de redes de computadores, considerando a presença ou não de técnicas de IA, é apresentada na Seção 2. A terceira seção descreve as cinco áreas funcionais de gerência de redes e como identificar problemas estáticos ou dinâmicos em cada uma delas. As técnicas de IA que podem ser utilizadas para desenvolver agentes inteligentes estáticos e agentes inteligentes dinâmicos são discutidos nos itens 3.1 e 3.2. A Seção 4 apresenta o sumário e os trabalhos futuros. Seguem os agradecimentos e as referências bibliográficas.

2 CLASSIFICAÇÃO DE AGENTES

Na literatura, é possível encontrar diferentes conceituações para o paradigma de agentes. Em Franklin & Graesser (1996), são citados diversos autores que classificaram os agentes sob diversos aspectos, entre eles destacam-se: estruturas de controle, ambientes (bases de dados, sistemas de arquivos, redes, Internet), linguagens ou aplicações. O autor apresenta uma taxonomia de modo a distinguir agentes de simples programas. No presente trabalho, se segue a classificação baseada no comportamento dos agentes ROISENBERG, BARRETO, & AZEVEDO (1997). Esta conceituação moderna usa conceitos da Teoria de Automata e é adequada para agentes envolvidos em gerência de redes de computadores.

Os agentes de software podem ser encontrados em uma série de aplicações. Em alguns casos são desenvolvidos com o auxílio de técnicas de IA, em outros não. O comportamento passivo foi atribuído àqueles agentes que não possuem autonomia e não utilizam técnicas de IA. Assemelham-se a simples programas e são construídos através de instruções. É o caso de agentes e gerentes de gerência de redes. Os agentes normalmente coletam informações dos objetos gerenciados e repassam aos gerentes. Estes por sua vez realizam operações sobre os agentes instruídos por um administrador de redes humano.

Os agentes de software desenvolvidos com o auxílio de técnicas de IA podem ser ativos ou autônomos. Os agentes ativos são desenvolvidos com o auxílio de heurísticas. Na área de gerência de redes, estas heurísticas são fornecidas pelo administrador da rede. Servem para definir regras de produção ou redes neurais diretas para auxiliar na solução dos problemas, conhecidos como sistemas de raciocínio lógico e fazem parte da IAS. Assemelham-se aos sistemas especialistas e não possuem autonomia.

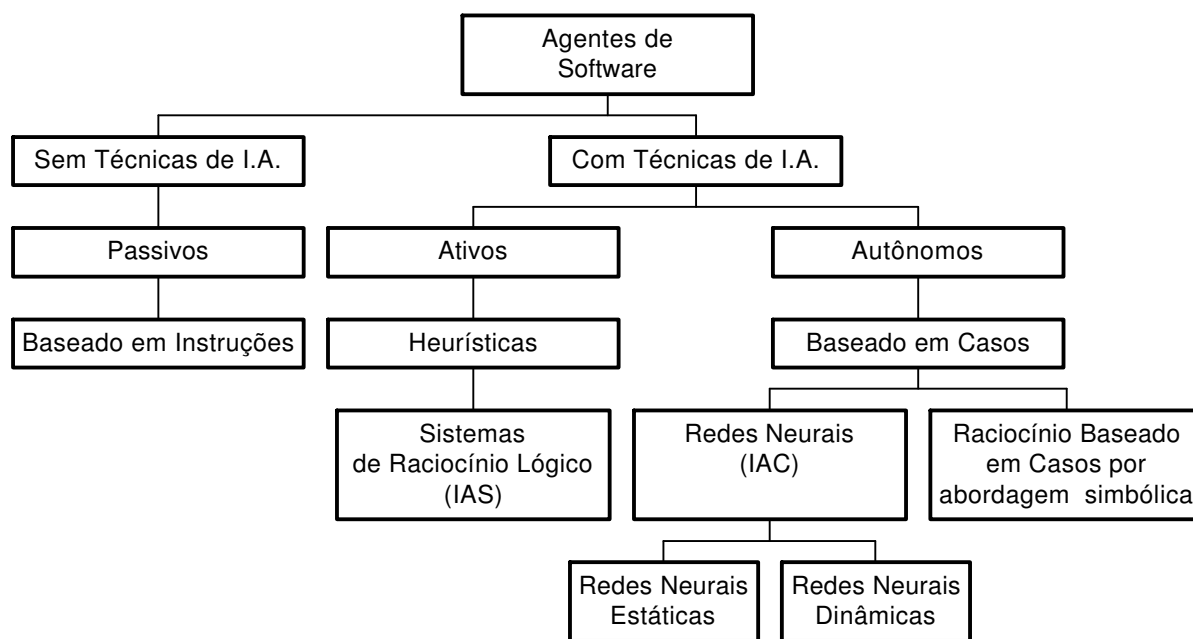


Figura 2: Paradigma de agentes

Existem vários conceitos para agentes autônomos na literatura. Adotar a conceituação proposta por Roisenberg, Barreto & Azevedo (1997) baseada na Teoria de Automata, permite classificar os agentes autônomos em estáticos e dinâmicos. O primeiro passo para desenvolver este tipo de agente é coletar exemplos de como deve ser seu funcionamento. No caso de se desejar implementar o agente por uma rede neural, estes exemplos servirão para treinar a rede. Se o problema for de caráter estático treina-se uma rede neural direta. Caso contrário, deve-se utilizar uma rede neural dinâmica. Estes dois casos serão discutidos com maiores detalhes na próxima seção.

Outra possibilidade é desenvolver um sistema de raciocínio baseado em casos (RBC), utilizando a abordagem simbólica. RBC é um caso particular do raciocínio por analogia, resultado das pesquisas realizadas por Roger Shank da Universidade de Yale AZEVEDO et al. (1991) SIGAKI (1997). Neste caso, tem-se uma coleção de casos resolvidos de um mesmo problema e dado um novo caso, deseja-se saber como resolvê-lo. Seguindo a heurística de desprezar inteligentemente casos, escolhe-se uma distância entre casos e despreza-se todos os que estão mais distantes do exemplo a resolver, retendo-se ao mais próximo. A solução do novo caso será a solução do caso mais próprio. A grande desvantagem deste tipo de sistema é a determinação de um sistema métrico apropriado. Ao contrário, em redes neurais existe uma função que caracteriza o problema conhecido através dos exemplos (BARRETO, 2001).

3 AGENTES INTELIGENTES EM GERÊNCIA DE REDES

Entre as principais dificuldades em se automatizar o processo de gerência de redes destacam-se a complexidade e a manipulação do volume de informações. No sentido de minimizar a complexidade da gerência de redes, a ISO/OSI subdividiu-a em cinco áreas funcionais: gerência de falhas, gerência de configuração, gerência de desempenho, gerência de contabilização e gerência de segurança (BRISA, 1993) (LEINWAND & CONROY, 1996).

A gerência de falhas é o processo de localizar problemas, ou falhas, em uma rede de dados. Envolve as tarefas de descobrir o problema, isolá-lo e solucioná-lo quando possível. Entre as causas mais prováveis para falhas em uma rede estão: erros de projeto e implementação da rede, erros de sobrecarga, distúrbios externos, tempo de vida útil de equipamentos expirado e má implementação de softwares (famosos “bugs”). Uma gerência de falhas bem projetada aumenta a confiabilidade na rede fornecendo ferramentas ao administrador da rede que auxiliem a detectar os problemas e iniciar os procedimentos de recuperação.

A gerência de configuração é o processo que determina e configura os dispositivos que controlam a rede. Roteadores, pontes (“bridges”, em inglês), terminais e servidores são exemplos destes dispositivos.

A gerência de desempenho deve assegurar que a rede tenha capacidade para suportar e acomodar uma certa quantidade de usuários. Ou seja, ela é extremamente necessária para otimizar a qualidade do serviço. Este processo mede o desempenho dos equipamentos e softwares disponíveis através de registros de algumas taxas de medidas. Exemplos destas taxas são vazão (“throughput”), taxas de erros, taxas de utilização e tempo de resposta.

Determinar quais os recursos e a forma que estão sendo utilizados pelos usuários é tarefa da gerência de contabilização. Além disso, este processo auxilia a assegurar que os usuários tenham acesso a quantidade suficiente dos recursos disponíveis. Envolve também, garantir ou remover permissões de acesso à rede.

A gerência de segurança é o processo que controla o acesso às informações disponíveis na rede. Existem informações armazenadas em computadores ligados à rede que são impróprias a

todos os usuários. O conjunto de senhas que permitem o acesso à rede é um exemplo de informações que devem ser bem protegidas. A gerência de segurança permite ainda que o administrador monitore as tentativas de entrada na rede.

Analizou-se cada uma destas áreas considerando-se suas características estáticas ou dinâmicas, e conforme o tipo de comportamentos para solucionar os problemas: reativo ou pró-ativo. A primeira preocupação é definir o problema. Um problema pode ser modelado como o objeto matemático $P = \langle D, R, q \rangle$, consistindo de dois conjuntos não vazios, D os dados e R os resultados possíveis e de uma relação binária $q \subset D \times R$, a condição que caracteriza uma solução satisfatória, associando a cada elemento do conjunto de dados a solução única desejada. O problema pode ser representado matematicamente por uma função. Resolver o problema será então encontrar um modo de implementar esta função ou de aproximá-la com o conhecimento que se dispõe.

Após a definição do problema é necessário definir o comportamento de gerência que será adotado: reativo quando as ações de gerência são realizadas após o aparecimento de algum problema; ou, pró-ativo no caso de se adotar um gerenciamento com ações preventivas. Outra questão, é determinar se a solução do problema deve ter características estáticas ou dinâmicas. Estáticas quando não existe o conceito de estado, e dinâmicas quando forem identificados trocas de estados na solução do problema.

Com base na metodologia apresentada anteriormente, indica-se as seguintes aplicações para as cinco áreas funcionais de gerência de redes definidas pelo modelo de referência OSI (MOTOROLA CODEX, 1993).

Área Funcional	Comportamento da Gerência		Características	
	Reativa	Pró-ativa	Estáticas	Dinâmicas
Falhas				
Desempenho				
Configuração				
Contabilização				
Segurança				

Tabela 1: Caracterização das áreas funcionais de gerência de redes

A gerência de falhas pode assumir os dois tipos de comportamento. Reativo no caso de falhas impossíveis de se prevenir, como é o caso das falhas provocadas pela ação do ambiente ou

provocadas pela má qualidade de peças e equipamentos utilizados. Ou poderá ser pró-ativo no caso de falhas relacionadas ao desempenho da rede, como por exemplo a sobrecarga da rede. Estes comportamentos poderão ter problemas de caráter estático ou dinâmico.

A gerência de desempenho deve ser pró-ativa em todos os seus aspectos. É uma forma preventiva de garantir a qualidade do serviço oferecido aos usuários. A gerência de configuração normalmente é reativa com características dinâmicas. O dinamismo é identificado pelo crescimento do número de equipamentos e de usuários que altera o estado da rede. A Gerência de Contabilização é estática e reativa. A gerência de segurança pode ser tanto reativa quanto pró-ativa. Contém características dinâmicas, uma delas é o controle de acesso que é diretamente ligado ao número de usuários. Com esta interpretação é possível definir se o problema a ser resolvido necessita de agentes autônomos estáticos ou dinâmicos.

3.1 AGENTES INTELIGENTES ESTÁTICOS

Para desenvolver Agentes Inteligentes Estáticos (AIE) busca-se normalmente heurísticas para a solução do problema. Ou seja, uma forma declarativa de resolver uma função, apresentando as propriedades que devem ser satisfeitas para solucionar o problema. Normalmente, extrai-se o conhecimento com o auxílio de um especialista, neste caso, será o administrador de redes. Sua tarefa é indicar quais os sintomas para diagnosticar determinados problemas, e sugerir as possíveis soluções. Assim, tem-se o conjunto dos dados, o conjunto das possíveis soluções e a relação entre eles de forma bem definida. Estas informações serão utilizadas para treinar uma rede neural direta (paradigma conexionista) ou poderão ser traduzidas para regras de produção (paradigma simbólico).

As redes neurais diretas são estáticas, não possuem ciclos e são representadas por camadas. Normalmente, possuem camadas de entrada, intermediária e de saída. Além disso são as mais populares, principalmente por existirem métodos de aprendizado fáceis de usar, o mais conhecido é o da retropropagação (“backpropagation”, em inglês) (BARRETO, 2001).

Koch BOGONIKOLOS, FRAGOUDIS & LIKOTHANASSIS (1999) apresenta agentes autônomos para gerência de redes utilizando regras de produção e uma pequena biblioteca para programação de uma rede neural direta.

3.2 AGENTES INTELIGENTES DINÂMICOS

Agentes Inteligentes Dinâmicos (AID) podem ser desenvolvidos com o auxílio de exemplos. Através de exemplos tem-se uma representação indireta do problema, ou seja, apenas a definição do problema é conhecida para um subconjunto de dados possíveis. Problemas dinâmicos, só podem ser “bem resolvidos” se forem utilizadas redes neurais artificiais com características dinâmicas.

Para incluir dinamismo a uma solução conexionista, deve-se:

- Inserir linhas de atrasos após a camada de entrada em redes neurais diretas;
- Utilizar redes com ciclos e neurônios dinâmicos: redes recorrentes ou redes de Hopfield;

Deseja-se então, conhecer os elementos do conjunto de respostas admissíveis para todos os elementos do conjunto de dados, mesmo aqueles que não estão incluídos na definição da função. Os exemplos, portanto, são utilizados para treinar uma rede neural com algoritmo supervisionado e obter os valores estimados da solução para os outros valores, utilizando a propriedade de generalização. Neste trabalho utiliza-se redes recorrentes ROISENBERG, BARRETO & AZEVEDO (1997). As redes neurais recorrentes ou recursivas são redes com ciclos e neurônios dinâmicos, esta dinâmica é provocada pelo retardo entre a saída do neurônio e sua entrada BARRETO (2001). Através desta topologia de rede neural é possível estimar o conjunto de estados a partir de um padrão de entrada e saída. Estes padrões são fornecidos por uma coleta de exemplos sobre o problema de gerência a ser solucionado.

Este tipo de solução é adotado quando o problema é mal definido, mas é conhecido através de exemplos. É o mesmo caso de técnicas topológicas de raciocínio baseado em casos, só que neste caso existe o problema de encontrar os pesos das diferenças entre os casos. Por exemplo, tempo de resposta longo em uma rede de computadores pode significar vários tipos de falhas. Fazendo uma analogia a sistemas médicos, vários tipos de doenças têm como sintoma febre alta. A maior limitação desta técnica é como diferenciar o peso de cada sintoma para cada doença.

4 SUMÁRIO E TRABALHOS FUTUROS

O presente trabalho apresentou uma metodologia para desenvolvimento de agentes de software inteligentes para gerência de redes. Dois tipos de agentes foram identificados a partir do enunciado do problema. Se o problema possui características estáticas existem heurísticas para solucioná-lo. As heurísticas podem ser extraídas do conhecimento de um especialista, neste caso, o administrador da rede. Caso contrário, utiliza-se exemplos da rede em questão para determinar padrões de entrada e saída. Estes padrões serão fornecidos a uma rede neural recorrente encarregada de estimar as trocas de estado para alcançar a solução do problema.

A implementação dos agentes está sendo desenvolvida em JAVA. O ambiente desta linguagem é naturalmente distribuído e integra de forma natural os aspectos de segurança. A vantagem principal é a possibilidade de definir os agentes de forma independente, não proprietária a nenhuma plataforma de gerência de redes ou de sistema operacional e processador (LEMAY & PERKINS, 1996).

Como trabalhos futuros pretende-se desenvolver classes de objetos para linguagem JAVA para auxiliar no desenvolvimento de agentes inteligentes estáticos e dinâmicos para gerência de redes de computadores. Pretende-se integrar estes objetos às plataformas de desenvolvimento de agentes em JAVA existentes no mercado, tais como o Java Dynamic Management Kit – JDMK (da Sun) ou aos agglets (da IBM).

5 AGRADECIMENTOS

Agradeço ao Conselho Nacional de Pesquisa e Desenvolvimento Tecnológico (CNPq) – Ministério da Ciência e Tecnologia pelo apoio financeiro, através de Bolsa de Formação de Pesquisador I (Doutorado) para a realização do presente trabalho.

6 REFERÊNCIAS BIBLIOGRÁFICAS

AZEVEDO, F. M. de; et al. Two Approaches in Case-Based Connectionist Expert Systems. In: AINN'91, 1991, **Artificial Intelligence and Neural Networks**. Ahnahaim, Calgary,: Acta Press, M. H. Hanza (editor), 1991. p. 13-17.

BARRETO, J. M. **Inteligência Artificial no limiar do Século XXI - Abordagem Híbrida Simbólica, Conexionista e Evolucionária**. 3ª edição ppp Edições, Florianópolis, 2001.

BOGONIKOLOS, N.; FRAGOUDIS, D.; LIKOTHANASSIS, S. "ARCHIMEDES: An Intelligent Agent for Adaptive – Personalized Navigation within a WEB Server". In: **Proceedings of the 32nd Hawaii International Conference on System Sciences**, IEEE Press, 1999.

BRISA. **Gerenciamento de Redes – Uma abordagem de sistemas abertos**. Makron Books do Brasil ed. : São Paulo, 1993.

COVACI, S.; ZHANG T.; BUSSE, I. Java-based Intelligent Mobile Agents for Open System Management. In: **Proceedings of the 9th International Conference on Tools with Artificial Intelligence**, ICTAI'97, IEEE Press, 1997.

FRANKLIN, S.; GRAESSER, A. "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents", In: **Proceedings of the 3rd International Workshop on Agents Theories, Architectures and Languages**, Springer, 1996.

KOCH, F. **Agentes Autônomos para Gerenciamento de Redes de Computadores**. Dissertação de Mestrado (CPGCC, UFSC), Florianópolis, 1997.

KURFESS, F.J.; SHAH, D.P. Monitoring Distributed Process with Intelligent Agents. In: **Proceedings of IEEE Conference and Workshop on Engineering of Computer-Based Systems**, ECBS'98, IEEE Press, 1998.

LEINWAND, A.; CONROY, K. F. **Network Management – A Practical Perspective**. Addison-Wesley, 2nd ed., 1996.

LEMAY, L.; PERKINS, C. L. **Teachyourself JAVA in 21 days**. SamsNet Publishing, 1996.

LESSER, V.R. Cooperative Multiagent Systems: A Personal View of the State of the Art. **IEEE Transactions on Knowledge and Data Engineering**, vol. 11, no. 1, Jan./Fev., 1999. Pp.133-142.

MINSKY, M.L., PAPERT, S. A. **Perceptrons: an introduction to computational geometry**, MIT Press, 1988.

MOTOROLA CODEX. **The Basics Book of OSI and Network Management**. Motorola University Press, Addison-Wesley, 1993.

NISSEN, M.E. "An Intelligent Agent for Web-based Process Redesign". In: **Proceedings of the 32nd Hawaii International Conference on System Sciences**, IEEE Press, 1999.

PINHEIRO, R.; POYLISHER, A.; CALDWELL, H. Mobile Agents Aggregation of Network Management Data. In: **Proceedings of First International Symposium on Agent Systems and Applications – Third International Symposium on Mobile Agents**, 1999.

ROISENBERG, M. **Emergência da Inteligência em Agentes Autônomos através de Modelos Inspirados na Natureza**. Tese de Doutorado (PPGEEL, UFSC), Florianópolis, 1998.

ROISENBERG, M., BARRETO, J. M, de AZEVEDO, F. M. Uma Proposta de Modelização para Agentes Autônomos Baseada na Teoria de Sistemas. In: **Anais do 3º Simpósio Brasileiro de Automação Inteligente**, Vitória, ES, 1997. pp.500-507.

SIGAKI, N. A. **Estudo da Influência de Bases de Casos em Redes IAC (Interactive Activation and Competition) na Implementação de Sistemas Especialistas**. 1997. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Santa Catarina. (orientanda do Fernando)