

# Inteligência Artificial Distribuída em Ambientes Reativos

J.P.A.Prado; R.L.Freitas; I.R.Guilherme; G.S.Nakamiti;  
Grupo de Inteligência Artificial - DEMAC  
Instituto de Geociências e Ciências Exatas - UNESP  
Rua 10, 2527 - CEP 13.500 - Rio Claro - SP

Márcio Rillo  
Laboratório de Sistemas Integráveis  
Divisão de Automação e Inteligência Artificial - EPUSP  
Av. prof. Luciano Gualberto, travessa 3, nº 158 - CEP 05.508 - São Paulo - SP

## Resumo:

A Inteligência Artificial Distribuída é uma área que tem recebido grande atenção dentro da Inteligência Artificial. Nela são combinadas técnicas tradicionais de Inteligência Artificial com técnicas de Processamento Distribuído. Esta combinação, tem como propósito a criação de modelos mais flexíveis para resolução de problemas, onde um único problema possa ser resolvido, de forma cooperativa, por um conjunto de agentes inteligentes. Neste trabalho, serão discutidos alguns dos conceitos fundamentais da Inteligência Artificial Distribuída, e bem como, será proposto um ambiente reativo para testes de sistemas de Inteligência Artificial Distribuída.

## 1. INTRODUÇÃO

Por muitos anos, as pesquisas em Inteligência Artificial foram orientadas para abordagens onde uma única entidade atuava em um ambiente estático. Contudo, esta abordagem mostraram-se inadequadas para diversos problemas do mundo real onde há a necessidade de se prever e modelar as interações de uma entidade inteligente, com um ambiente dinâmico e com as demais entidades que possam existir (3). Na verdade, quando uma entidade inteligente planeja suas atividades, deve ter "em mente" o comportamento das demais entidades existentes, pois estas podem auxiliar ou dificultar a sua atividade.

Unindo conceitos de Inteligência Artificial e de Processamento Distribuído, a **Inteligência Artificial Distribuída (IAD)**, tem como propósito estudar as interações entre entidades inteligentes. De fato, um Sistema de Inteligência Artificial Distribuída pode ser visto como um grupo de entidades inteligentes, chamadas de agentes, que interagem por cooperação, coexistência ou por competição.

Apesar do conceito de agente ser bastante amplo, neste trabalho, o termo agente será empregado segundo a definição proposta por Bond (1): *"agente é um processo computacional com escopo bem definido e com um único local de controle"*.

Segundo Huhns, em (14), as pesquisas em IAD podem ser caracterizadas e entendidas através do seguinte paradigma: existe uma coleção de agentes tentando resolver um único problema. Onde, embora haja um estado e um objetivo global para o problema, cada agente possui apenas uma visão parcial e imprecisa destes. Sendo assim, cada agente procura entender, prever e influenciar o estado global segundo sua visão local dos objetivos do sistema. A comunicação entre os agentes é permitida e é através dela que são trocadas soluções, tarefas, objetivos e estimativas do estado global do problema.

Deste seu primeiro "WorkShop" em 1980, a IAD vem se tornando uma importante área de estudos dentro da Inteligência Artificial. Para melhor poder estudá-la, Huhns (14) divide as pesquisas em Inteligência Artificial Distribuída em duas grandes áreas:

- Pesquisas com **resolução distribuída de problemas**: que estuda como o trabalho para resolver um problema pode ser dividido em um conjunto de nós ou módulos, que dividem e compartilham conhecimento sobre o problema e sobre o estado de solução do mesmo;
- Pesquisas com sistemas **multi-agente**: que procura modelos e técnicas que coordenem as atividades de um grupo de agentes autônomos, de forma que os mesmos trabalhem de forma inteligente e cooperativa.

A seguir serão discutidos alguns dos principais conceitos da IAD e ao final deste documento será descrito um ambiente para o teste de sistemas de IAD, além de uma arquitetura para um sistema multi-agente.

## 2. RESOLUÇÃO DISTRIBUÍDA DE PROBLEMAS

Nos sistema que seguem esta abordagem, os resolvedores formam uma rede. Onde cada nó possui apenas uma visão local do problema que está sendo resolvido.

Em uma abordagem tradicional de Processamento Distribuído, seria necessária uma intensa troca de mensagens entre os nós da rede, na tentativa de suprir os mesmos com as informações necessárias ao processamento e ao controle local de cada nó. Como resultado desta intensa comunicação tem-se uma queda na performance de todo o sistema e um alto nível de sincronismo no processamento dos nós.

Uma forma de se reduzir a taxa de comunicação e o sincronismo entre os nós, é aceitar que os mesmos produzam resultados parciais — ou candidatos a resultados — incompletos, incorretos ou, ainda, inconsistentes com os resultados parciais gerados nos demais nós da rede.

Este tipo de processamento requer um modelo para resolução de problemas, que permita a cooperação entre os nós, de tal forma que os resultados parciais de cada nó possam ser revisados e evoluídos a partir das informações obtidas através da interação com os demais nós.

Um dos mais importantes modelos para a resolução de problemas de forma distribuída, foi proposto por Lesser e Corkill em (17). Por possuir as características descritas acima, este sistema foi denominado **Sistema Funcionalmente Correto e Cooperativo** ou Sistema FA/C<sup>1</sup> — do inglês *Functional Accurate and Cooperative*. Funcionalmente correto por ser capaz de gerar soluções aceitáveis a partir de resultados intermediários, muitos deles incorretos e inconsistentes; cooperativo porque é a cooperação entre os nós que garante a coerência do processo de resolução.

O problema da cooperação entre os agentes, nos sistemas FA/C, tem sido tratado em diversos trabalhos (6,9,16,17 e 18). Contudo, nenhum dos protótipos apresentados operam com um fluxo contínuo de dados. Permanecendo, ainda, em aberto a questão de como estruturar a cooperação em sistemas FA/C, para que estes possam funcionar de forma reativa.

### 3. SISTEMAS MULTI-AGENTE

Em se tratando de sistemas multi-agente, deve-se acrescentar à definição de agente vista anteriormente o fato que: os agentes desta classe de sistemas são capazes de receber informações do ambiente, processar estas informações e conforme o resultado deste processamento atuar no ambiente, modificando-o.

Os agentes em um sistema multi-agente podem trabalhar em um único objetivo, ou em objetivos distintos que interajam entre si. Assim como os nós, nos sistemas de resolução distribuída, os agentes necessitam compartilhar informações sobre o problema e sobre o estado de solução do mesmo, porém diferente dos nós, os agentes precisam "raciocinar" sobre a coordenação de suas atividades — ações — com as dos demais agentes.

Em muitos domínios de aplicação os sistemas multi-agente precisam ter características reativas — como por exemplo, no controle de robôs e de células flexíveis de manufatura. Nestes domínios, a complexidade da tarefa de controle cresce na proporção em que aumentam o número e a variedade de estímulos enviados pelo mundo externo ao sistema.

Para lidar com esta complexidade e responder aos estímulos com a velocidade que o domínio de aplicação impõe, a tarefa de controle não pode ser centralizada (10). A descentralização do controle, contudo, não é fácil de ser implementada. Pois, paradoxalmente ela pode levar a um acréscimo do tempo de solução do problema, uma

---

<sup>1</sup> Devido ao seu uso já consagrado, neste trabalho será utilizado a notação FA/C para se referir aos sistemas Funcionalmente Corretos e Cooperativos.

vez que esta pode comprometer a coerência do processo de resolução. Para evitar este fenômeno é necessário determinar:

- a) Como cada um dos agentes irá usar seu conhecimento, planos, objetivos e habilidades, no processo de resolução;
- b) Como e quando cada agente irá comunicar aos demais agentes seus planos, objetivos e habilidades;
- c) Como cada um dos agentes irá representar internamente as informações recebidas dos demais agentes, bem como as recebidas (coletadas) do ambiente.

Com o intuito de se estudar algumas destas questões, iniciou-se o desenvolvimento do sistema multi-agente, descrito na seção 7.

#### 4. COERÊNCIA E COORDENAÇÃO

Coerência e Coordenação são conceitos largamente usados na análise e na descrição de sistemas de IAD, mas raramente são definidos ou diferenciados (1).

O termo **coerência** deve ser usado para se referir a uma propriedade de todo sistema — ou de alguma região deste —, e a **coordenação**, é uma propriedade da interação entre um conjunto de agentes, que executam tarefas distintas. Deve-se ter em mente que, quando dois ou mais agentes trabalham em tarefas que possuem algum grau de interação, suas atividades devem ser coordenadas para que o sistema funcione coerentemente.

A coerência é particularmente problemática em domínios de aplicação, como os de IAD, onde: cada agente possui apenas uma visão limitada das atividades do sistema, a comunicação entre os agentes também é limitada e não existe um controlador central.

#### 5. COOPERAÇÃO

A cooperação entre pessoas, ou grupos de pessoas, ocorre quando cada uma acredita que pode obter mais benefício cooperando com as demais, do que se estivesse trabalhando isoladamente (5).

Apesar de ser bastante comum, no mundo real, a cooperação é um fenômeno pouco estudado. O advento do computador como um agente em ambientes humanos têm levado à problemas bastante interessantes. Por exemplo, o de como garantir a cooperação em ambientes onde existam pessoas e agentes computacionais interagindo.

A cooperação entre humanos é fácil, uma vez que cada pessoa possui uma compreensão sobre a outra — é possível prever as ações da outra pois ambas tem muitas coisas em

comum. Por outro lado, a interação homem/computador é frequentemente acompanhada de frustrações, isto ocorre porque ambos os agentes possuem uma visão inadequada do porque — e do como — o outro agente atua.

Este problema se agrava ainda mais em se tratando da interação entre agentes computacionais. Diversos pesquisadores de IAD estão preocupados em desenvolver mecanismos que permitam aos agentes, de um sistema de resolução distribuída de problemas, interagirem de forma cooperativa. Alguns destes pesquisadores (5 e 24), tratam a cooperação com sendo um tipo especial de coordenação.

Desde que agentes individuais possam tomar decisões sobre quais ações tomar, cada agente busca atividades que pareçam interessantes, segundo seu ponto de vista local. Cada agente é desta forma "*self-interested*", pois procura sempre maximizar seu resultado local, (6). Este ponto de vista pode variar de um agente para outro e a compatibilidade destes irá determinar o tipo de interação entre os agentes — competição, cooperação, ou simplesmente coexistência.

Para cooperar, um agente deve reconhecer que a cooperação é de seu interesse. Isto é, ele deve ter um conhecimento local que o leve à cooperação, sempre que esta trouxer algum benefício. Quando os agentes são guiados por objetivos, a cooperação pode surgir através do compartilhamento destes. Como os agentes são guiados à atingir seus objetivos, os dois agentes tendem a cooperar. Entretanto, se os agentes tiverem diferentes visões de como chegar ao objetivo pode ocorrer uma competição, comprometendo, assim, a coerência do processo de resolução.

## 6. O MODELO BLACKBOARD

O modelo BlackBoard para resolução de problemas é um modelo oportunístico fortemente estruturado (22). O termo oportunístico refere-se a capacidade do sistema selecionar e aplicar o conhecimento mais adequado — oportuno — para um dado instante no processo de resolução do problema (23).

Além do raciocínio oportunístico e de uma estratégia para a aplicação do conhecimento, o modelo também prescreve a organização do domínio do conhecimento, de todas as entradas do sistema e das soluções finais e parciais do problema. Estas soluções constituem o espaço de solução do problema que, no modelo BlackBoard, é organizado em um ou mais níveis de hierarquia.

Não há uma definição formal para o modelo BlackBoard, mas, de uma forma mais ou menos evidente, os sistemas implementados possuem as características do primeiro sistema BlackBoard desenvolvido: o HEARSAY-II (7 e 15). Este sistema foi desenvolvido para ouvir e entender frases, ditas em inglês, sobre um determinado assunto que estivesse armazenado em sua base de conhecimento.

Esta falta de uma definição formal, somada a uma grande variedade de problemas que foram atacados usando o modelo BlackBoard — interpretação de sinais de sonar (23), geração de planos (11,12 e 21), etc — fez com que os projetistas destes novos sistemas propusessem variações do modelo inicialmente proposto pelo HEARSAY-II.

O que pode ser visto, entretanto, é que mesmo os mais diferentes tipos de BlackBoard possuem módulos com características similares, tais como:

a) As **Fontes de Conhecimento** - FCn ("Knowledge Sources"): os conhecimentos necessários para resolver um problema são divididos e armazenados nas FCn, as quais são totalmente independentes uma das outras. O objetivo das FCn é contribuir com informações que podem levar à solução do problema. Elas coletam dados no BlackBoard e os modificam usando o conhecimento nelas contido, apenas as FCn podem modificar o BlackBoard, sendo que todas as modificações são explícitas e visíveis;

b) A **Estrutura de Dados do BlackBoard**: os dados referentes aos estados do sistema são mantidos em uma base de dados global, o BlackBoard propriamente dito. As FCn podem ler e alterar o BlackBoard na tentativa de levá-lo mais próximo da solução do problema. Toda a comunicação entre as FCn é feita exclusivamente através do BlackBoard.

O termo oportunístico, como foi mencionado anteriormente, é frequentemente associado com os sistemas BlackBoard. O oportunismo pode ser definido como uma estratégia de controle que faz o melhor uso do espaço de solução corrente (4). Ou ainda, como sendo a capacidade do sistema explorar os seus melhores dados, usando-os no método de resolução mais promissor (12).

O modelo BlackBoard, a princípio, não é um modelo para a Inteligência Artificial Distribuída. Contudo, devido a sua modulariedade e ao seu raciocínio oportunístico, o modelo BlackBoard vem sendo usado para modelar os agentes, em inúmeros protótipos desta área (5, 6 e 9).

## 7. O SPIDER-BYTE

O ambiente de chão de fábrica é caracterizado pelo grande número de agentes envolvidos, pela incerteza quanto aos eventos futuros e pela necessidade de respostas em tempo-real. Em se tratando de aplicações de Inteligência Artificial, o termo **tempo-real** implica que o sistema deva ser capaz de gerar uma solução antes que o ambiente tenha oportunidade de mudar substancialmente. Mesmo a melhor solução pode levar ao desastre se o mundo tiver mudado substancialmente no período que separou o desenvolvimento da solução de seu uso na prática.

As abordagens tradicionais de planejamento (8) — são puramente estratégicas, onde primeiro o plano é gerado, para depois ser executado — garantem a coerência nas atividades dos agentes e um certo grau de adaptação do sistema às mudanças do mundo. Contudo, esta separação entre planejar e agir compromete a eficiência destas abordagens em aplicações de tempo-real.

Uma abordagem alternativa é proposta por Chapman (2) onde o sistema é puramente reativo, planejando apenas a próxima ação a executar. Esta abordagem garante a resposta do sistema, às mudanças do mundo, em tempo real. Porém, compromete a coerência nas atividades dos agentes, uma vez que fica praticamente impossível aos agentes coordenarem suas atividades — um agente, para coordenar sua atividade com as de outros agentes, deve ser capaz de antever o comportamento destes (20).

Qualquer modelo ou arquitetura de sistema de Inteligência Artificial que venha a ser proposta para operar no ambiente de chão de fábrica, deve possuir características reativas e características multi-agente. Muitas questões terão que ser resolvidas antes de ser possível construir tal sistema (13), com o intuito de elucidar algumas destas questões, iniciou-se o desenvolvimento de uma ambiente para testes de sistemas multi-agente: o **Spider-Byte**.

### 7.1. O Domínio da Aplicação

Os projetistas de sistemas de Inteligência Artificial contam com um desafio adicional, que é a dificuldade de se testar os novos modelos e arquiteturas em um mundo real. Não apenas pela variedade de componentes — de "hardware" e "software" envolvidos, como também pela complexidade dos mesmos. Um teste, por exemplo, de uma arquitetura de controle em uma célula de manufatura pode ser dificultado, ou até mesmo ser inviabilizado, pela falta de um sistema de visão adequado.

Um "video-game" é de fácil construção e pode possuir as mesmas características do chão de fábrica — no que diz respeito a necessidade de reatividade<sup>2</sup> e a coordenação das atividades de mais de um agente, ou jogador.

O Spider-Byte é um "video-game" onde cada jogador controla uma aranha e tem como objetivo eliminar, em um tempo limitado, um conjunto de moscas que estão em constante movimento pela tela, Figura 1.

Sempre que a última mosca for eliminada uma nova fase do jogo terá início, a cada nova fase o grau de dificuldade aumenta — com o aumento do número de moscas e com o surgimento de obstáculos —, exigindo um melhor desempenho dos jogadores.

<sup>2</sup> Na verdade, em muitos jogos as restrições de tempo são, ainda, mais rígidas que no ambiente de chão de fábrica.

Esta melhoria do desempenho dos jogadores deve ser obtida não apenas através do aumento da eficiência individual de cada jogador, mas também através de uma maior coordenação entre os mesmos. Desta forma, o ambiente dos "video-games" oferece uma

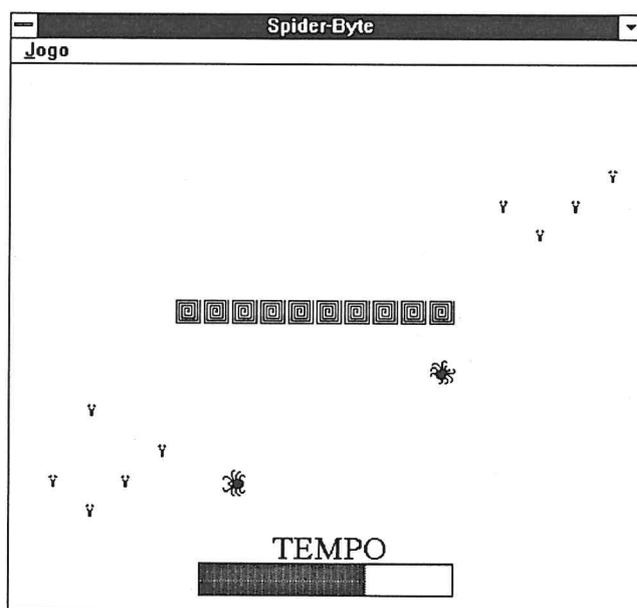


Figura 1: Spider-Byte

vantagem adicional aos projetistas de sistemas de IAD, que é a facilidade de se aferir a qualidade de uma arquitetura através do placar dos jogos.

## 7.2. Uma Primeira Experiência

Após a elaboração do Spider-Byte, iniciou-se o desenvolvimento de um sistema capaz de coordenar as atividades das aranhas no "video-game" — ou seja, jogar o Spider-Byte.

Este sistema foi denominado **Player**. Cada um dos agentes do sistema Player é formado por um BlackBoard do sistema SBBS(21). Cada BlackBoard foi modificado para permitir a comunicação entre os agentes, o fluxo contínuo de informações e para acomodar as restrições de tempo do domínio de aplicação.

Através da observação de jogadores humanos, constatou-se que o planejamento das atividades — de dois ou mais agentes — em um "video-game" como o Spider-Byte, pode ser dividida em três fases:

- a) A fase **estratégica**: onde através de negociação são estabelecidos os sub-objetivos de cada um dos agentes. A maior taxa de comunicação entre os agentes é

observada nesta fase do planejamento. É nesta fase que os jogadores do Spider-Byte determinam qual conjunto de moscas será atacado por cada aranha;

b) A fase **tática**: após estabelecidos os sub-objetivos de cada agente, estes passam a planejar suas ações com o intuito de alcançar seus objetivos. Nesta fase, ocorre um queda na taxa de comunicação entre os agentes, existindo apenas para resolver algum conflito que possa existir entre dois agentes. No Spider-Byte esta fase é caracterizada pela elaboração das rotas das aranhas até as moscas;

c) A fase **reativa**: nesta fase ocorre o refinamento final do plano e a sua execução. Durante esta, o plano vai sofrendo adaptações para atender as mudanças do meio — a aranha pode modificar sua rota para pegar uma mosca que está próxima.

Com o intuito de comportar estas fases, cada BlackBoard do sistema Player foi dividido em dois painéis<sup>3</sup>:

a) Painel Estratégico: que comporta as fases estratégica e tática. — a reunião destas duas fases em um único painel foi possível devido a simplicidade do domínio de aplicação. Este painel é encarregado de gerar um plano de alto nível, — semelhante meta-plano gerado no sistema OPM (12) — que descreve a estratégia adotada pelo agente para alcançar seu objetivo, que também é estabelecido por este painel. Este painel possui um conjunto de FCn cuja função é comunicar-se com os demais agentes.

b) Painel Reativo: este painel é responsável pelo refinamento final e pela execução do plano — através de um conjunto de FCn especiais, capazes de atuar no mundo externo.

Diferente da interpretação de sinais, que é uma tarefa de natureza basicamente analítica, o planejamento é uma tarefa, primariamente, de natureza generativa. Nos planejadores oportunistas (11,12 e 21) o plano é, normalmente, gerado em um processo "top-down" — através de refinamentos sucessivos — e otimizado, de maneira oportunística, em um processo "bottom-up". O sistema Player, ao contrário do demais planejadores BlackBoard, não faz esta otimização do plano. Uma vez que este processo consome um considerável esforço computacional e não faz sentido otimizar um plano, pois este é suscetível a constantes alterações.

No sistema desenvolvido, o processo "bottom-up" ocorre quando o plano precisar ser adaptado às mudanças do ambiente. Embora, a maior parte das adaptações do plano ocorra na fase reativa — no painel reativo do sistema Player —, podem ocorrer mudanças no ambiente que inviabilizem parcialmente ou totalmente o plano. Nestes casos, dependendo do grau de mudança que o ambiente sofreu, os agentes devem retornar, em um processo "bottom-up", às fases tática ou estratégica.

---

<sup>3</sup> Para aumentar a eficiência das buscas no espaço de solução, o BlackBoard pode ser sub-dividido em estruturas menores chamadas painéis (21).

## 8. CONCLUSÃO

A IAD propicia uma nova luz nas ciências cognitivas e na Inteligência Artificial, esta importância é reconhecida por pesquisadores como Nilsson, (19), que considera os estudos em IAD de fundamental importância para o avanço das demais áreas de Inteligência Artificial. Pois, sem estes estudos, provavelmente a Inteligência Artificial estaria fadada a domínios estáticos (3).

Enquanto o sistema Player constitui o início de um estudo, que visa entender melhor alguns dos principais conceitos da IAD, principalmente no que diz respeito à coordenação das atividades em um sistema multi-agente, o ambiente Spider-Byte mostrou a viabilidade de se empregar "video-games" no auxílio ao desenvolvimento de sistemas de IAD que irão atuar em ambientes reativos, como é o caso do ambiente de chão de fábrica.

## 9. REFERÊNCIA BIBLIOGRÁFICAS

- ( 1 ) Bond, A. H.; Gasser, L. *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers, Inc., San Mateo, CA., 1988.
- ( 2 ) Chapman, D. *Vision Instruction and Action*. MIT Press, Cambridge, MA, 1991.
- ( 3 ) Chaib-Draa, B.; Moulin, B.; Mandiau, R.; Millot, P. *Trends in Distributed Artificial Intelligence*. *Artificial Intelligence Review*, 6:35-66, 1992.
- ( 4 ) Craig, L.D. *The BlackBoard Architecture: A Definition and its Implications*. Research Report Nro RR98, University of Warwick, UK, 1987.
- ( 5 ) Durfee, E.H.; Lesser, V.R.; Corkill, D.D. *Coherent Cooperation Among Communicating Problem Solvers*. *IEEE Transactions on Computers*, C-36:1275-1291, 1987.
- ( 6 ) Durfee, E.H.; Lesser, V.R.; Corkill, D.D. *Cooperation Through Communication in a Distributed Problem Solving Network*. In Michael N. Huhns, editor, *Distributed Artificial Intelligence*, pag.29-58, Pitman Publishing/Morgan Kaufmann Publishers, San Mateo, CA, 1987.
- ( 7 ) Erman, L.D.; Lesser, V.R. *A Multi-level Organizational for Problem Solving using Many, Diverse, Cooperating Sources of Knowledge*. *Proceeding of the Fourth International Joint International on Artificial Intelligence*, Vol.2, pp 483-490, 1975.

- ( 8 ) Freitas, R.L. *Planejamento Apoiado em Inteligência Artificial*. Anais do XXV Congresso Nacional de Informática, 1992.
- ( 9 ) Gasser, L.; Braganza, C.; Herman, N. *MACE: A Flexible Testbed for Distributed AI Research*. Michael N. Huhns, editor, Distributed Artificial Intelligence, pag.119-152, Pitman Publishing/Morgan Kaufmann Publishers, San Mateo, CA, 1987.
- ( 10 ) Ginsberg, M.L. *Decision Procedures*. Michael N. Huhns, editor, Distributed Artificial Intelligence, pag.3-27, Pitman Publishing/Morgan Kaufmann Publishers, San Mateo, CA, 1987.
- ( 11 ) Hayes-Roth, B.; Hayes-Roth, F.; Stan, R. *Modelling Planning as an Incremental, Opportunistic Process*. Proceeding of the 6th International Joint Conference on Artificial Intelligence, pp 375-383, 1979.
- ( 12 ) Hayes-Roth, B. *The BlackBoard Architecture: A General Framework For Problem Solving*. Report Nro HPP-83-80, Computer Science Department, Stanford University, 1983.
- ( 13 ) Durfee, E.H. *Towards Intelligent Real-Time Cooperative Systems*. In J. Hendler, editor, Planning in Uncertain, Unpredictable, or Changing Environments, University of Maryland, Technical Research Report TR 90-45, 1990.
- ( 14 ) Huhns, M.N. *Distributed Artificial Intelligence*, Pitman Publishing/Morgan Kaufmann Publishers, San Mateo, CA, 1987.
- ( 15 ) Lesser, V.R.; Fennel, R.D.; Ernam, L.D.; Richard, D.F. *Organization of the HEARSAY-II Speech Understanding System*. IEEE Symposium on Speech Recognition, Contributed Papers, pp 11-21, 1974.
- ( 16 ) Lesser, V.R.; Corkill, D.D. *Distributed Problem Solving*. In Stuart C. Shapiro, editor, Encyclopedia of Artificial Intelligence, pag.245-251, John Wiley and Sons, New York, 1987.
- ( 17 ) Lesser, V.R.; Pavlin, J.; Durfee, E.H. *Aproximate Processing in Real Time Problem Solving*. AI Magazine 9(1):49-61, 1988.
- ( 18 ) Lesser, V.R.; Corkill, D.D. *The Distributed Vehicle Monitoring Testbed: A Tool for Investigating Distributed Problem Solving Networks*. In Robert Englemore and Tony Morgan, editor, Blackboard Systems, pag.353-386, Addison-Wesley Publishing Company, New York, 1988.
- ( 19 ) Nilsson, N.J. *Two Heads are Better than One*. Sigart Newsletter, 43-43, 1980.

( 20 ) Parunak, H.V.D. *Manufacturing Experience with the Contract Net*. In Michael N. Huhns, editor, *Distributed Artificial Intelligence*, pag.285-310, Pitman Publishing/Morgan Kaufmann Publishers, San Mateo, CA, 1987.

( 21 ) Prado, J.P.A. *O Uso da Arquitetura BlackBoard em Planejamento*. Dissertação de mestrado, ICMSC-USP, março, 1991.

( 22 ) Prado, J.P.A. *O Uso da Arquitetura BlackBoard na Automação Industrial*. V CONAI, São Paulo, 1992.

( 23 ) Nii, H.P.; Feignebaum, E.; Anton, J.; Rockmore, J. *Signal-to-Symbol Transformation: HASP/SIAP Case Study*. *AI Magazine* 3:2:23-35, 1982.

( 24 ) Rosenschein, J.S. *Rational Interaction: Cooperation Among Intelligent Agents*. PhD thesis, Computer Science Department, Stanford University, Stanford, CA, 1985.