

SISTEMA WEB DE CONTROLE MULTIAGENTE

IVAN RIZZO GUILHERME, RAFAEL PEDROSANTO

Laboratório de Sistema Inteligente, DEMAC, IGCE, UNESP
CP 176, 13506700, Rio Claro-SP
E-mails: ivan@rc.unesp.br, rafael.pedrosanto@gmail.com

ALEX TEIXEIRA, JOSÉ RICARDO P. MENDES

LIAP, DEP, FEM, UNICAMP
Caixa Postal 6052, 13083-970, Campinas, SP
E-mails: {teixeira, jricardo}@dep.fem.unicamp.br

CARLES SIERRA

DSI, IIIA, CSIC
Campus de la UAB, 08193, Bellaterra, Spain
E-mails: sierra@iiia.csic.es

Abstract— In this work is proposed a software architecture that integrated concepts from multiagent systems, fuzzy control and Web 2.0. The architecture is used to development a Multiagent Web System to control the liquid level inside the VASPS (Vertical Annular Separation and Pumping System). A VASP process simulation and process control, including Web visualization, is showed.

Keywords— Fuzzy Control, Multiagent System, Web 2.0 system

Resumo — Neste trabalho é proposta uma arquitetura de software que integra aspectos provenientes dos sistemas multiagentes, do controle nebuloso e da Web 2.0. A arquitetura é utilizada na construção de um Sistema Multiagente Web para o controle de nível de líquido do sistema VASPS (*Vertical Annular Separation and Pumping System*). Um experimento da simulação e controle do processo no VASPS, e a visualização na Web são apresentados.

Palavras-chave— Controle Nebuloso, Sistemas Multiagentes, Sistemas Web 2.0

1. Introdução

A larga utilização dos controladores, e sensores e a alta conectividade dos equipamentos criam um ambiente industrial onde são gerados grandes volumes de dados e em diferentes formatos. A análise dos dados gerados e a supervisão dos processos nesses ambientes industriais requerem o desenvolvimento de sistemas computacionais extremamente complexos, sendo compostos por vários módulos que interagem entre si. O desenvolvimento desses sistemas de supervisão utilizando as técnicas convencionais de desenvolvimento de software é uma tarefa complexa.

As pesquisas na área de multiagente são atualmente apontadas como uma das mais promissoras técnicas para a solução de problemas complexos na área de Ciência da Computação. Embora a área de multiagente tenha sido inicialmente proposta como área de pesquisa na Inteligência Artificial, seus resultados vêm sendo utilizados na solução de problemas nas diversas áreas da Computação e nas Engenharias (Klush,2001). As pesquisas de metodologias que permitam o desenvolvimento de sistemas multiagentes é um dos principais esforços dos pesquisadores das áreas de Inteligência Artificial e Engenharia de Software. Neste sentido, Jennings (Jennings et. al.,

2003), Weyns (Weyns et. al, 2005), Sierra (Sierra et. al, 2004), Garcia (Garcia, 2004), Silva (Silva, 2004), e outros estão pesquisando métodos e tecnologias baseados nos conceitos de agentes, a serem utilizadas na análise, no projeto e na implementação de sistemas voltados para as diversas áreas das empresas, destacando-se comércio, gerenciamento e de supervisão dos processos industriais. As metodologias baseadas em agentes permitem a construção de componentes flexíveis, autônomos e interativos.

No contexto da automação e controle, as diversas técnicas de desenvolvimento de sistemas multiagentes estão voltadas ao desenvolvimento de sistemas de supervisão e controle de processos discretos. Entretanto, foi constatada a inexistência de metodologias de desenvolvimento software multiagentes para tratar processos de produção contínua, por exemplo, os processos da indústria química e petróleo (Jennings et. al., 2004).

Os sistemas de produção *offshore* são sistemas complexos, de alto custo operacional e dinâmico, que são gerenciados por profissionais qualificados e de diversas áreas. A ocorrência de problemas operacionais ou de anormalidades em processos pode causar a parada total da produção, tornando os custos operacionais frequentemente altos. A predição desses pro-

blemas torna-se uma importante forma de garantir a continuidade da produção e, conseqüentemente, aumentar a segurança operacional e prevenir danos ambientais. As paradas do sistema e manobras para solução de problemas específicos não podem ser executadas de forma deliberada, pois requerem o conhecimento especializado na execução de uma seqüência de passos, de acordo com as normas operacionais das empresas. O início ou reinício da produção também requer que os operadores tenham o conhecimento especializado para executar os passos necessários.

Assim, neste trabalho é apresentada uma arquitetura multiagentes voltada à supervisão e controle. No desenvolvimento da arquitetura proposta foram considerados diversos trabalhos correlatos (Jennings et al., 2004; Bunch et al., 2004; Gabbar et al., 2003; Mařík et al., 2005). Os agentes de controle atuam utilizando controladores nebulosos. Utilizando a arquitetura proposta são apresentados os detalhes dos processos de controle responsáveis pelo controle dos processos de separação em uma planta de produção offshore utilizando o Vertical Annular Separation and Pumping System (VASPS).

2. Arquitetura Multiagente

A arquitetura proposta, apresentada na Figura 1, é voltada a supervisão e controle de processos de produção contínua, sendo composta de três módulos:

- Módulo de Supervisão e Controle Multiagente;
- Módulo de Simulação do Processo;
- Módulo de Monitoramento Web

2.1 Módulo de Supervisão e Controle Multiagente;

O Módulo de Supervisão e Controle proposto é composto de agentes em dois tipos: agentes de Controle e agentes de Supervisão. A descrição dos tipos de agentes é a seguinte:

- *Agentes de Controle*: é composto de agentes que executam o controle dos sistemas e dos componentes da planta de produção. Nos agentes estão codificados, utilizando o conceito da Teoria dos Conjuntos Nebulosos, os conhecimentos dos especialistas no controle da operação dos equipamentos.
- *Agentes de Supervisão*: consiste de agentes que integram conhecimentos e dados provenientes de diferentes repositórios e dos Agentes de Controle visando à solução de problemas de maior complexidade. Os agentes deste nível poderão trocar mensagens com os agentes do nível de controle na solução de eventuais problemas. Nos repositórios estão disponíveis as ontologias referentes às questões de segurança operacional, diagnóstico de problemas dos equipamentos, início e parada da planta de produção e de outras fontes.

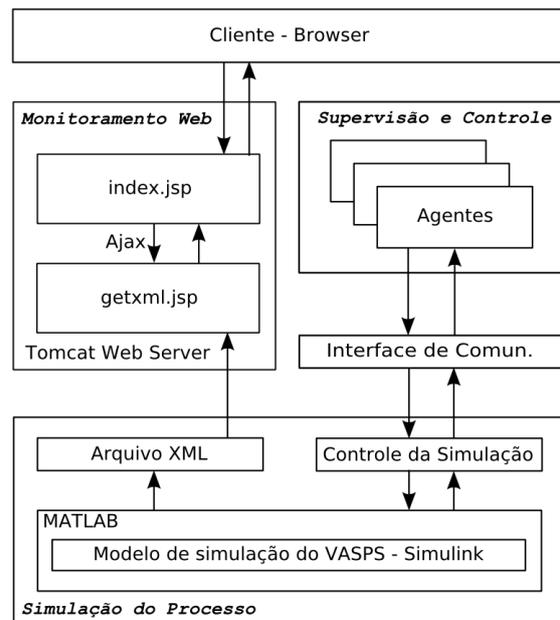


Figura 1: Arquitetura do Sistema

2.2 Módulo de Simulação do Processo

Os agentes de um sistema multiagentes atuam interagindo com o ambiente ou com outros agentes. Neste sentido, de acordo com Weyns (Weyns et al., 2005), o ambiente é um importante aspecto a ser considerado tanto na simulação quanto no projeto das aplicações multiagentes. Ainda de acordo com Weyns (Weyns et al., 2005), duas importantes considerações merecem destaque: o domínio que é simulado deve ser modelado; a infra-estrutura da simulação deve ser projetada. Neste trabalho em decorrência destas duas conclusões, verificou-se a necessidade de desenvolver o modelo do domínio a ser simulado, onde há um relógio único para os agentes e a simulação. Isto visa permitir a simultaneidade das ações dos agentes e da simulação do processo contínuo. Em relação ao projeto da infra-estrutura, foi considerado necessário definir mecanismos para a integração dos agentes ao simulador do processo. Considerando que os agentes são implementados em Java, é, portanto necessária a utilização de uma ferramenta de simulação que tenha mecanismos para a integração com o Java.

Neste sentido, foi executado um estudo visando a definição da ferramenta a ser utilizada na simulação do processo contínuo. A primeira solução analisada foi o SIMILE (Simile, 2005), que consiste de um ambiente que permite a simulação de processos contínuos e discretos, com uma interface gráfica para a construção do modelo a ser simulado. Após o estudo e a realização de alguns testes para implementar o modelo simulado, ocorreram dificuldade na integração do Simile com os agentes então implementados em Java. Em conseqüência desta dificuldade a utilização desta ferramenta foi desconsiderada.

A outra solução avaliada e posteriormente adotada foi o Simulink. O Simulink dispõe de facilidades que permitem a construção visual do processo a ser simulado. A integração dos agentes (Java) com o programa Simulink é feita utilizando o JMatlink (Müller, 2005), como apresentado na Figura 1. O JMatLink é uma API Java que permite executar a partir de programas Java, simulações implementadas no Matlab e no Simulink. No programa JMatlink também há funcionalidades que permitem visualizar os processos simulados na Web (Müller, Waller, 1999).

2.3 Módulo de Monitoramento Web

A Web é atualmente o principal meio para disponibilizar informações. Visando dar maior versatilidade ao sistema proposto e permitir o acesso remoto à simulação do processo, foi definida na arquitetura uma camada que permite o monitoramento do processo simulado por meio da Web. Esta camada é basicamente composta por dois programas criados com a tecnologia JSP (Hall e Brown, 2005) rodando sobre o servidor web Tomcat da Apache. Os dados do Módulo de Simulação do Processo e do Módulo de Monitoramento Web são trocados utilizando padrões de dados definidos em XML. Esta abordagem tem como base a proposta de Müller e Waller (1999), com algumas alterações. Um esquema mostrando como a camada de monitoramento interage com o restante do sistema é mostrado na Figura 1.

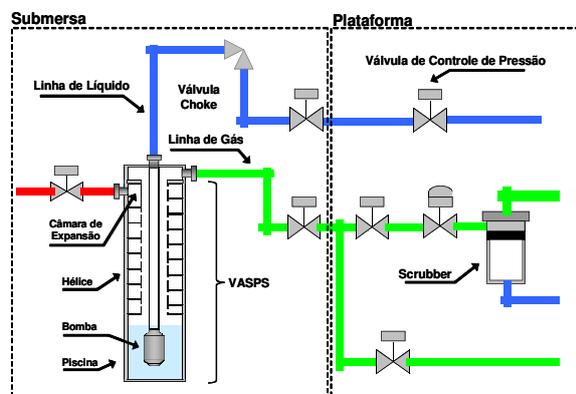


Figura 2: Esquema do VASPS

3. Implementação da Arquitetura

A implementação da arquitetura proposta foi definida visando o controle e supervisão dos processos do sistema de produção de petróleo.

O sistema de separação e bombeio é dividido em duas partes, o VASPS, que é localizado no fundo do mar e um conjunto de válvulas que são localizadas na superfície (Figura 2).

Conforme apresentado na sessão 2, a arquitetura implementada (Figura 1) consiste de três módulos: Módulo de Simulação do Processo, implementado no Matlab e Simulink; o Módulo de Supervisão e Controle, onde foi dado ênfase ao desenvolvimento dos agentes de controle do sistema, que foi implementado

em Java; e, o Módulo de Monitoramento Web que utilizam os conceitos da Web 2.0.

A seguir são apresentados os detalhes da implementação.

3.1 Módulo de Simulação do Processo

O modelo de simulação do VASPS, utilizado no Módulo de Simulação do Processo, é uma adaptação do modelo apresentado em Teixeira (Teixeira *et al.*, 2004), onde o controle do processo foi retirado e incorporado nos agentes do Módulo de Supervisão e Controle.

A integração dos agentes do Módulo de Supervisão e Controle com o processo simulado no Módulo de Simulação do Processo é feita utilizando o JMatlink (Müller, 2005). Para a integração foi necessário: alterar algumas configurações do Simulink; no modelo a ser simulado e também a utilização de recursos que permitem o compartilhamento das variáveis entre os dois ambientes a serem integrados (Java - JMatlink e o Simulink).

3.2 Módulo de Supervisão e Controle

O VASPS recebe o fluxo contínuo de produção de poços de petróleo. Para o seu funcionamento adequado o nível do líquido interno (nível da piscina) do vaso separador deve ser mantido dentro de uma faixa operacional. Para o controle de nível de líquido na piscina do VASPS podem ser utilizadas três estratégias: controle pela bomba, controle pela válvula *choke* e controle simultâneo pela bomba e pela válvula *choke* (Vale *et al.*, 2002, Peixoto *et al.*, 2005).

No controle pela bomba, o nível é controlado automaticamente através da variação da velocidade da bomba, o que é feito alterando-se a frequência de alimentação do motor. A variação da frequência de alimentação do motor é feita por um variador de velocidade (VSD). Nesta estratégia de controle, a abertura da válvula *choke* é ajustada manualmente e mantida constante durante a operação.

No controle pela válvula *choke*, a vazão de saída é alterada variando-se a abertura da válvula automaticamente. Nesta estratégia, a velocidade da bomba é ajustada manualmente e mantida constante durante a operação.

Por fim, na estratégia de controle pela bomba e pela válvula *choke*, ambos são ajustados automaticamente, porém a bomba terá prioridade no controle. Caso a bomba não consiga controlar o nível, a abertura da válvula *choke* é alterada de forma a tentar corrigir tal nível.

Um sistema de controle de nível de líquido eficaz é extremamente importante para o bom funcionamento do sistema, visto que uma ação de controle inadequada pode danificar a bomba, o que resultaria em um elevado prejuízo econômico.

No controle do processo são utilizados dois agentes que atuam como controladores nebulosos. Um agente para o controle do *choke* e um outro para o controle da pressão no VASPS. O conhecimento

utilizado no controle dos processos foi codificado utilizando lógica nebulosa e adquirido junto a especialistas. Para implementar o conhecimento do controle nebuloso nos agentes foi desenvolvido em Java um ambiente de desenvolvimento de Sistemas Nebulosos.

3.2.1 Ambiente de Desenvolvimento de Sistemas Nebulosos-ADSINE

O ambiente desenvolvido consiste das interfaces (Figuras 3 e 4) que permitem a definição do conhecimento e as classes que implementam o cálculo e inferência nebulosa. Através das interfaces o usuário poderá definir as variáveis do sistema, os conjuntos nebulosos e as regras nebulosas.

As funcionalidades disponíveis nesse ambiente são equivalentes àquelas disponíveis no Matlab, incluindo funcionalidades que permitem importar projetos de controladores nebulosos anteriormente desenvolvidos no padrão do Matlab. O conhecimento codificado também pode ser descrito no formato XML.

As classes Java que implementam o cálculo e as regras de inferências são incorporadas aos Agentes de Controle.

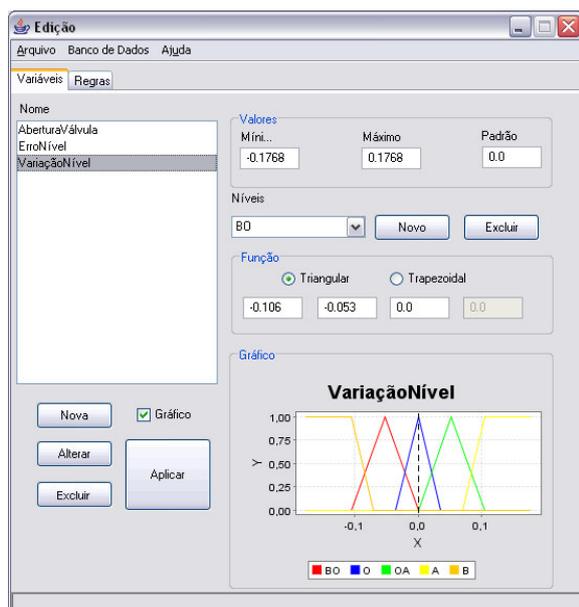


Figura 3: ADSINE: Editor de Funções Nebulosas.

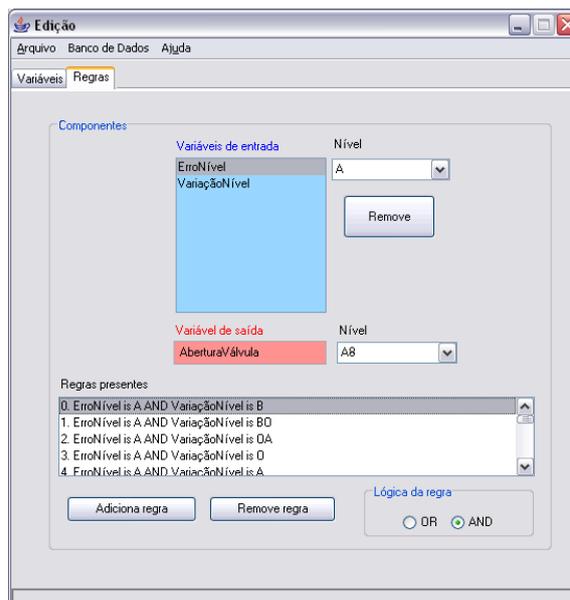


Figura 4: ADSINE: Editor de Regras Nebulosas.

3.2.2 Agentes de Controle

No controle do processo são utilizados dois agentes que operam como controladores nebulosos.

Um agente atua no controle da válvula *choke* e outro no controle da válvula de controle da pressão do sistema, sendo que a rotação da bomba permanece constante ao longo do processo.

Os agentes de controle utilizam o conhecimento expresso em funções e regras nebulosas apresentados nas Figuras 3 e 4, criados com o auxílio da ferramenta ADSINE. Para cada variável de entrada dos controladores foram definidos cinco conjuntos nebulosos, enquanto que para a variável de saída foram definidos nove conjuntos nebulosos.

Durante o funcionamento do sistema, o agente definido para o controle através da Válvula, recupera do processo simulado no Simulink, o valor do nível e efetua o cálculo do erro de nível e a derivada do erro de nível. Após a inferência utilizando as regras nebulosas definidas, este agente tem como saída um sinal de controle que representa a ação a ser efetuada na válvula.

3.3 Módulo de Monitoramento Web

A interface Web foi construída de modo a permitir o monitoramento Web da simulação do VASPS (Figura 1). Assim que a simulação do sistema VASPS é executada, o controlador da simulação registra, em um arquivo no formato XML valores referentes a pontos processo que se deseja monitorar. Para este trabalho, foram monitorados o nível de líquido e a pressão dentro do VASPS.

No servidor Web (Figura 1), sempre que uma requisição é feita através programa *index.jsp* é invocado o: programa *getxml.jsp*. Este é o responsável por resgatar o valor atual das variáveis monitoradas do arquivo XML, que é atualizado a cada passo da simulação.

Para garantir um bom desempenho do sistema de monitoramento, o programa `index.jsp` foi construído utilizando a tecnologia Ajax, dessa forma são realizadas requisições assíncronas ao programa `getxml.jsp` que retornará apenas o mínimo de dados necessários, reduzindo o overhead do servidor e garantindo a velocidade da aplicação. Uma vez que o `index.jsp` recebe os dados requisitados, é feita a atualização da página no *browser* do cliente (Figura 5).

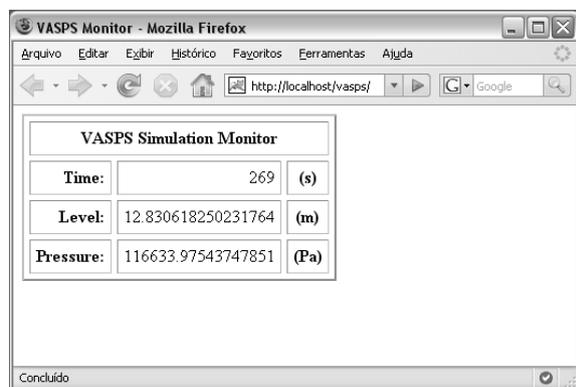


Figura 5: Visualização Web do Processo.

4. Resultados

Para realização dos testes, utilizou-se um sinal de onda quadrada para representar a vazão de entrada de líquido. Isto foi feito para simular uma variação significativa nas condições de entrada, visando garantir que o controlador responde satisfatoriamente a grandes variações nas condições de operação. A vazão mínima simulada foi de $600 \text{ m}^3/\text{d}$, enquanto que a máxima foram de $1030 \text{ m}^3/\text{d}$, com picos de 400s. O tempo total de simulação foi de 5000s. O *set point* de nível adotado foi de 30m, sendo este medido a partir da base do sistema. Os resultados obtidos são apresentados na Figura 6.

Na Figura 6a, pode-se observar a vazão de entrada de líquido no VASPS em função do tempo, o que atua com um distúrbio no sistema tentando desestabilizá-lo. Na Figura 6b, tem-se o nível de líquido no VASPS em função do tempo, onde é possível observar que o tempo de subida (tempo para a variável ir de 10 a 90% do seu valor final) foi de aproximadamente 2200s. O erro estacionário obtido foi de 0,85m, o que mostra que o controlador atende a primeira meta. As Figuras 6c e 6d apresentam os valores no tempo das variáveis de entrada do controlador. A Figura 6e apresenta os valores da variável de saída do controlador, onde podemos ver que apesar da grande variabilidade da vazão de entrada o controlador não requisitou excessivamente a bomba centrífuga, evitando o desgaste excessivo e uma eventual falha prematura.

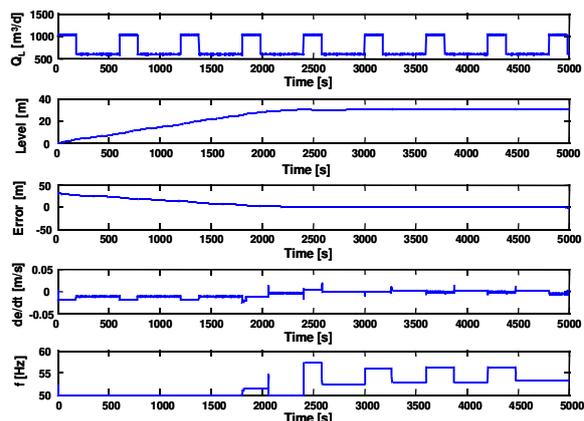


Figura 6: Resultados do teste: **a)** Vazão de entrada de líquido **b)** Nível de líquido **c)** Erro de nível **d)** Derivado do erro **e)** Freqüência de alimentação da bomba.

5. Conclusão

Os resultados das simulações mostram que o agente controlador da bomba desenvolvido é robusto, operando satisfatoriamente mesmo diante de grandes variações das condições operacionais, sendo capaz dar a partida no sistema sem a necessidade de intervenção humana.

A separação dos sistemas de tomada de decisão do controlador permite flexibilizar a adoção de novas estratégias de controle.

O ambiente de desenvolvimento de controladores nebuloso implementado em Java permite reutilizar os projetos implementados utilizando o *Toolbox* do Matlab. O ambiente também foi desenvolvido para permitir o desenvolvimento de aplicações na Web.

Nos próximos passos serão estudadas as diversas estratégias de controle (controle pela bomba ou pressão) a ser implementadas e os mecanismos de negociação entre os agentes. Assim, de acordo com o fluxo da produção poderá ser definida a melhor estratégia de controle a ser utilizada. Na fase seguinte também serão desenvolvidos os agentes responsáveis pela Supervisão dos processos. Esses agentes irão interagir com os agentes de controle na supervisão do sistema. Uma metodologia será utilizada para a definição dos agentes e as regras de comunicação entre eles.

Na parte de monitoramento Web serão estudados os padrões de Interface utilizados no monitoramento de processos visando implementá-los utilizando os recursos assíncronos da Web 2.0.

Agradecimentos

Este trabalho é parcialmente suportado pelo projeto CT-Petro/MCT/CNPq 504863/2004-5 e FAPESP-Processo 05/51632-0.

Referências Bibliográficas

- Benedetti, M., Villa, M. Field Tests on VASPS Separation and Pumping System. OTC 8449, 1997.
- Bunch, Larry; Breedy, Maggie; Bradshaw, Jeffrey M.; Carvalho, Marco; Suri, Niranjana; Uszok, Andrzej; Hansen, Jack; Pechoucek, Michal; Marik, Vladimir; (2004) Software Agents for Process Monitoring and Notification, ACM Symposium on Applied Computing.
- França, F. A., Rosa, E. S., Bannwart, A. C., Moura, L. F. Hydrodynamic Studies on a Cyclonic Separator. OTC 8059, 1996.
- Gabbar, Hossam A.; Shinohara, Shintaro; Shimada, Yukiyasu; Suzuki, Kazuhiko; (2003), Experiment on distributed dynamic simulation for safety design of chemical plants, Simulation Modelling Practice and Theory, 11, 109–123
- Garcia, Alessandro, Objetos e Agentes: Uma Abordagem Orientada a Aspectos. Departamento de Informática, 2004. 298p. Tese de Doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.
- Hall, M.; Brown, L. Core Servlets e JavaServer Pages. 2.ed., Rio de Janeiro: Ciência Moderna, 2005.
- Jennings, Nicholas R., Bussmann, Stefan, Agent-Based Control-Systems. Why They Suited to Engineering Complex Systems. IEEE Control Systems Magazine, June 2003, pp. 61-73
- Jennings, Nicholas R., Bussmann, Stefan, Wooldridge, M. Multiagent Systems for Manufacturing Control, Springer Verlag, 2004, pp. 288.
- Klush, Matthias., Information agent technology for internet: A survey, Data & Knowledge Engineering, 36, (2001), 337-372.
- Müller, S.: 2005. JMatLink library. <http://jmatlink.sourceforge.net/>
- Müller, Stefan; Waller, Heinz; (1999), Efficient Integration of Real-Time Hardware and Web-Based Services into MATLAB, Proceeding of 11th EUROPEAN SIMULATION SYMPOSIUM AND EXHIBITION, Germany
- Peixoto, G. A., Ribeiro, G. A. S., Barros, P. R. A., Meira, M. A. VASPS Prototype in Marimba Field – Workover na Re-Start. SPE 95039, Anais do Latin American and Caribbean Petroleum Engineering, 2005.
- Sierra, Carles; Rodríguez-Aguilar, Juan Antonio; Noriega, Pablo, Esteva, Marc; Arcos, Josep Lluís; Engineering multiagent systems as electronic institutions. European Journal for the Informatics Professional, V(4), August 2004.
- Silva, Viviane Torres. Uma Linguagem de Modelagem para Sistemas Multi-agentes Baseada em um Framework Conceitual para Agentes e Objetos. Rio de Janeiro, 2004. 252p. Tese de Doutorado- Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.
- SIMILE., 2005, <http://simulistics.com/>, Simulink.
- <http://www.mathworks.com/products/simulink/>
- Teixeira, Alex F., Mendes, José Ricardo P., Guilherme, Ivan R., MOROOKA, Celso K., ESTEVAM, Valdir, Um Controlador Fuzzy para o Sistema de Separação e Bombeamento Submarino – VASPS, Anais da Rio Oil & Gas Expo and Conference 2004.
- Vale, O.R. do, Garcia, J. E., Villa, M. VASPS Installation and Operation at Campos Basin. OTC 14003, 2002.
- Vladimír Mařík, Pavel Vrba Ken H. Hall, Francisco P. Maturana, Rockwell Automation Agents for Manufacturing, AAMAS'05, July, 2005, Utrecht, Netherlands.
- Weyns, Danny; Schumacher, Michael; Ricci, Alessandro; Mirko Viroli; Holvoet, Tom; (2005), Environments for Multiagent Systems, The Knowledge Engineering Review, 20: 127-141 Cambridge University Press