O Papel dos Agentes Inteligentes nos Sistemas Tutores Inteligentes

Luciana Bolan Frigo[1], Eliane Pozzebon and Guilherme Bittencourt[2]

Resumo *— A maioria dos Sistemas Tutores Inteligentes (STI's) não apresenta o nível esperado de inteligência. Um dos motivos está no pouco conhecimento sobre o que é e como funciona a inteligência humana. Muitos cursos à distância utilizam sistemas disponíveis comercialmente, mas estes não incorporam as funcionalidades propostas nos STI´s idealizados no meio acadêmico. Os Sistemas Multiagentes (SMA) são utilizados para diminuir a distância entre os sistemas reais e os ideais. Os SMA ajudam na modelagem e organização do sistema, além de, muitas vezes, apontar o melhor caminho a ser seguido pelo modelo pedagógico. Este artigo descreve a utilização de agentes cognitivos num sistema tutor inteligente cujo objetivo é auxiliar no ensino de Fundamentos da Estrutura da Informação para os alunos de Engenharia de Controle e Automação da Universidade Federal de Santa Catarina (UFSC).*

Palavras-chave — Sistemas Multiagentes, Sistema Tutor Inteligente

1. INTRODUÇÃO E MOTIVAÇÃO

A Educação à Distância (EAD) faz uso da Internet, que é um ambiente muito propício para a utilização da tecnologia de agentes inteligentes, por ser complexo e distribuído. A tecnologia de agentes inteligentes é um campo abrangente dentro da Inteligência Artificial, podendo ser aplicada nos mais diversos tipos de problemas. Como um paradigma vem sendo estudado e fundamentado, suas várias definições possibilitam sua utilização em aplicações para simular o comportamento humano.

Os sistemas multiagentes são bastante flexíveis, podem ser formados por diversos e distintos elementos, chamados de agentes. Os agentes podem executar diferentes funções e modificar seu comportamento dinamicamente. Sistemas Multiagentes são capazes de resolver problemas complexos, principalmente aqueles que os sistemas tradicionais não conseguem resolver.

Para auxiliar no ensino-aprendizagem propõe-se um sistema multiagente naturalmente distribuído e cooperativo cujo objetivo é resolver problemas em ambientes dinâmicos como a *Web*. O MathTutor é um sistema distribuído, que utiliza técnicas de Inteligência Artificial Distribuída (IAD), seguindo a abordagem de Sistemas Multiagentes (SMA) e está sendo desenvolvido na UFSC. Através da implementação de agentes, num ambiente de aprendizagem como o MathTutor, é possível fazer o acompanhamento de um estudante num dado domínio. O STI modela o conhecimento do estudante sobre um tópico e a medida que ele realiza determinadas tarefas no sistema, compara este conhecimento com o modelo do especialista do domínio. O sistema pode também adaptar os níveis e estilos de aprendizagem do estudante e apresentar a informação, os testes e as respostas que são mais apropriadas.

Este artigo mostra algumas vantagens da utilização de um sistema multiagente em um STI como o Mathtutor.

O artigo está organizado da seguinte maneira: a seção 2 apresenta uma breve apresentação da Inteligência Artificial Distribuída, enquanto que a seção 3 introduz os sistemas multiagentes. Na seção 4 o modelo dos agentes do MathTutor é descrito e na seção 5 tem-se uma breve explicação sobre aspectos de implementação dos agentes. Na seção 6 alguns trabalhos correlatos. Por fim, na seção 7 apresentam-se as conclusões.

2. INTELIGÊNCIA ARTIFICIAL DISTRIBUÍDA

A Inteligência Artificial Distribuída (IAD) é o ramo da inteligência artificial que está relacionado com a solução cooperativa de problemas em um certo ambiente através de agentes distribuídos. As técnicas desenvolvidas em IAD permitem aplicações em diversos níveis [2]:

1. Desenvolvimento de novas aplicações baseadas em metodologias tradicionais de desenvolvimento de *software*, mas que se beneficiem das idéias da IAD;
2. A expansão das funcionalidades de sistemas existentes através do encapsulamento destas aplicações em plataformas de IAD;
3. Desenvolvimentos de sistemas que incorporem as técnicas de IAD da concepção até sua implementação.

A aplicação da IAD baseia-se na idéia de que a agilidade, flexibilidade, inteligência e desempenho de um sistema podem ser melhorados a medida que ela permite alcançar objetivos como [15]:

- Construção de sistemas descentralizados ao invés de centralizados;
- Obtenção de soluções emergentes (resultado das interações entre agentes e/ou humanos) ao invés de totalmente planejadas;
- Execução concorrente ao invés de seqüencial.

Muitos problemas reais são naturalmente e fisicamente distribuídos, por isso, a necessidade de soluções distribuídas

[1] Luciana Bolan Frigo,, Eliane Pozzebon; Doutorandas Universidade Federal de Santa Catarina, LCMI/DAS, Caixa Postal,Florianópolis, SC, Brazil, {lu, eliane}@das.ufsc.br
Guilherme Bittencourt, Prof. Dr. - Universidade Federal de Santa Catarina, LCMI/DAS, Caixa Postal,Florianópolis, SC, Brazil gb@das.ufsc.br

e a exigência da capacidade de adaptação dos sistemas são alguns dos principais motivos que estimulam o desenvolvimento de sistemas inteligentes distribuídos [6].

A IAD está dividida em: Solução distribuída de problemas (SDP) e Sistemas multiagentes (SMA) [3].

A SDP tem como foco principal o problema. Os principais objetivos são utilizar a capacidade de processamento e a robustez oferecidas pela tecnologia de redes para atacar problemas de natureza distribuída ou muito complexos [2], tendo como exemplo o problema de controle de tráfego aéreo.

Os agentes envolvidos em SDP são programados para cooperar, dividir tarefas e comunicar-se de maneira confiável, entretanto não é simples estabelecer estas propriedades. O foco principal dos SMA (Figura 1) é o estudo das pressuposições básicas sobre agentes que garantam a possibilidade de ação cooperativa em sociedade, ou seja, o foco das pesquisas são os agentes [2].

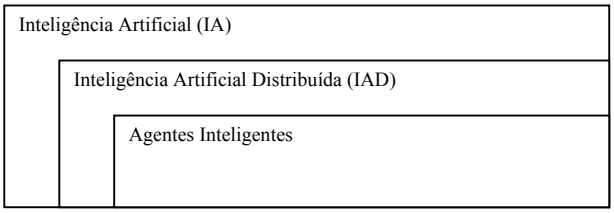

FIGURA 1
ORGANIZAÇÃO

3. SISTEMAS MULTIAGENTES

Sistema multiagente é uma abordagem da IAD cujo foco da pesquisa são os agentes.

Um agente pode ser definido em termos de suas propriedades fundamentais e deve possuir um certo grau de autonomia para raciocinar e tomar decisões por sua própria vontade além de interagir com outros agentes. Por fim, um agente deve possuir um certo grau de independência para resolver um problema, nem que seja uma parte dele.

Não existe uma definição única para o conceito de agente, e isto ocorre porque os autores normalmente ligam a definição ao domínio da aplicação, às formas de cooperação e nos níveis de autonomia. A definição que melhor se adapta ao sistema multiagente utilizado para a construção de um STI é:

"Um agente é um sistema computacional, posicionado em algum ambiente, que é capaz de agir com autonomia flexível visando atingir os objetivos para o qual foi projetado. [12]"

Existem algumas propriedades que devem ser observadas em um agente. São elas [7]:

1. Posicionamento (*situatedness*): o agente recebe sinais de entrada dos seus sensores vindos do ambiente no qual está localizado e pode executar ações contextualizadas que modifiquem o ambiente de alguma forma;
2. Autonomia: o agente deve ter a possibilidade de agir sem a intervenção direta de usuários ou de outros agentes, além de poder controlar totalmente suas ações e seu estado interno;
3. Pró-atividade (*pro-activeness*): os agentes não devem apenas agir em resposta ao seu ambiente, mas devem agir oportunamente por iniciativa própria de acordo com seus objetivos;
4. Sociabilidade: os agentes devem poder interagir, quando apropriado, com outras entidades do ambiente de forma a melhor resolver seus problemas e ajudá-las nas suas atividades;
5. Adaptabilidade (*adaptiveness*): os agentes devem poder mudar o seu comportamento devido a uma experiência anterior;
6. Receptividade (*responsiveness*): os agentes devem poder perceber o seu ambiente e responder adequadamente a mudanças que ocorram nele;
7. Mobilidade: os agentes podem estar aptos a transportar-se de uma máquina para outra.

Existem diversos tipos de agentes, e eles podem ser: de software ou de hardware, estacionários ou móveis, persistentes ou temporários, reativos ou cognitivos [6].

Agentes estacionários são agentes de software que, uma vez lançados em um dado ambiente computacional, não tem a habilidade de se moverem pela rede para outros computadores. Agentes móveis são agentes de software que podem se mover para outros ambientes através da rede e, quando se movem, levam consigo seus estados internos, ou seja, sua representação mais a memória [16].

Agentes persistentes são agentes de software que, uma vez lançados em um dado ambiente computacional, não podem ser excluídos do sistema. Agentes temporários são agentes de software que tem uma vida finita, normalmente de duração igual ao tempo de uma dada tarefa, ou seja, tão logo eles finalizam sua missão eles são excluídos do sistema, normalmente por eles próprios [6].

Uma das classificações mais importante sobre os agentes é quanto a eles serem reativos ou cognitivos. Os agentes reativos são baseados em modelos de organização biológica ou etológica, como por exemplo, as sociedades de formigas ou cupins. Uma formiga sozinha não é considerada uma entidade inteligente, mas o formigueiro sim, pois apresenta comportamento inteligente relacionado à busca e armazenamento de alimentos, além da organização dos berçários. O modelo de funcionamento de um agente reativo é o de estímulo-resposta. Em geral, estes agentes não apresentam memória, não planejam suas ações futuras e não se comunicam com outros agentes, tomando conhecimento das ações dos outros agentes pelas mudanças ocorridas no ambiente. Normalmente existem em grande quantidade no sistema e possuem baixa complexidade [2].

Os agentes cognitivos são baseados em organizações sociais humanas como grupos, hierarquias e mercados. Os agentes possuem uma representação explícita do ambiente e dos outros agentes, dispõem de memória, e por isto são capazes de planejar ações futuras. Agentes cognitivos podem comunicar-se entre si diretamente, isto é, seus sistemas de percepção e de comunicação são distintos, o que não acontece nos reativos. Normalmente estão em pequena quantidade no sistema e são de média ou alta complexidade [2]. Além disso, requerem sofisticados mecanismos de coordenação e protocolos de alto nível para suporte à interação [6].

Os agentes cognitivos aumentam a qualidade do sistema sob o ponto de vista pedagógico porque permitem gerar um sistema mais perceptivo com autonomia, flexibilidade, colaboração e adaptação.

A característica essencial da abordagem SMA é a filosofia de resolução distribuída de problemas, na qual é adotada uma estratégia de dividir para conquistar. A resolução cooperativa distribuída de problemas diz que um problema é dividido em subproblemas e cada um é solucionado separadamente por um agente, cada um destes comunicando ou cooperando entre si quando necessário, com a idéia básica de que a soma dos resultados locais corresponderá à solução do problema geral [12]. Através desta afirmação é possível perceber que a cooperação é um assunto que merece um detalhamento.

Existem dois tipos de cooperação: partilha de informação ou partilha de tarefas [4]. A partilha de informação se dá quando um dado agente dispõe ou produz informações parciais que julga serem úteis a partilhar e as envia para os outros agentes. A partilha de tarefa é efetuada quando um dado agente, ao decompor uma dada tarefa, detecta subtarefas que não pode ou não quer realizar, sendo necessário procurar outros agentes que possam auxiliá-lo.

O projeto de um sistema multiagente deve considerar vários aspectos, como por exemplo, tipo ou classe do problema a ser tratado, níveis de autonomia dos agentes, representação do conhecimento, protocolos de comunicação, definição da organização do sistema, modelagem de dados, modelos de coordenação, cooperação e resolução de conflitos [16].

A organização de um sistema multiagente pode ser classificada como: democrática, federada ou hierárquica. Na organização democrática os agentes não possuem organização, e atuam em graus similares de independência e autonomia, sem hierarquia alguma. Neste caso são necessários procedimentos de coordenação sofisticados para garantir convergência e evitar interações desnecessárias.

Na organização federada existe algum tipo de hierarquia. Há a presença de agentes facilitadores, agentes intermediários entre o cliente e o supervisor. Grupos de agentes da federação tem graus similares de independência e autonomia, e normalmente são bastante heterogêneos. Os procedimentos de coordenação são de média complexidade dado a própria existência do facilitador. Entretanto, as arquiteturas hierárquicas são as mais utilizadas em aplicações industriais, onde o sistema multiagente está estruturado em níveis de hierarquia [16].

Uma das propriedades essenciais de um agente é a sua capacidade de comunicar-se com outros agentes, usuários e sistemas visando atingir seus objetivos, a interação. Durante o planejamento da interação deve-se lembrar que a informação pode ser incompleta, imprecisa e ou prevista, e a qualidade dela varia de acordo com o tipo de agente. A interação pode ser dividida em 4 camadas de complexidade: comunicação, coordenação, cooperação e colaboração [17].

A camada comunicação é básica de qualquer software que precisa interagir. A camada coordenação define as regras de interação considerando as agendas dos agentes de forma a se evitar comportamentos indesejados. A camada cooperação é uma camada encontrada apenas em sistemas onde a cooperação reflete uma estratégia de ação decidida pelo agente, permitindo a negociação. A camada mais refinada é a camada colaboração onde um agente tem capacidade de detectar possíveis objetivos comuns, e de planejar sua agenda com os outros de forma a atingir o objetivo da melhor forma possível, aproveitando ao máximo a partilha de informações.

Existem ainda muitos outros tópicos a serem tratados quando se estuda sistemas multiagentes, como protocolos de comunicação inter-agentes, ambientes de desenvolvimentos, integração e interoperação de sistemas multiagentes, entre outros. Entretanto procurou-se apresentar os conceitos básicos que serão citados na descrição da aplicação.

4. MODELO DOS AGENTES DO MATHTUTOR

MathTutor é um Sistema Tutor Inteligente em desenvolvimento na Universidade Federal de Santa Catarina, que pretende apresentar os principais conceitos de abstração de dados e de procedimentos aos alunos da disciplina de Fundamentos da Estrutura da Informação, aplicada ao curso de Engenharia de Controle e Automação Industrial.

O sistema possui quatro módulos: o módulo do estudante, o módulo do especialista, o módulo pedagógico e o módulo da interface.

O módulo do especialista possui as informações a respeito do conhecimento do conteúdo a ensinar. O módulo do estudante armazena as informações sobre a compreensão do aluno sobre o domínio de conhecimento. O módulo pedagógico contém as regras para a tomada de decisão que permitem determinar o quanto o aluno está aprendendo. O módulo da interface apresenta ao usuário o ambiente de aprendizagem desenvolvido pelo especialista.

MathTutor utiliza a tecnologia dos agentes cognitivos que permitem gerar um sistema mais perceptivo, aumentando a qualidade sob o ponto de vista pedagógico [9].

Os agentes seguem um modelo de cooperação onde ocorre compartilhamento das tarefas a serem resolvidas.

Cada agente fica responsável por uma parte da resolução do problema.

A arquitetura escolhida é a troca de mensagens. Os agentes comunicam-se diretamente, respeitando a hierarquia do sistema, enviando mensagens assíncronas. Nesta configuração pode haver um agente facilitador de comunicação, que no caso do MathTutor é o agente de interface. Neste tipo de arquitetura é fundamental que os agentes saibam os nomes e endereços uns dos outros para não ocorrerem problemas no encaminhamento das mensagens. Este método é bastante eficiente no sentido de obter as mensagens em tempo hábil. Um protocolo de conversação é essencial para que as mensagens sejam trocadas e interpretadas de maneira adequada.

Os agentes são cognitivos, pois possuem um processo explícito de escolha da ação a ser tomada. Baseia-se em mecanismos de processamento simbólico, neste caso, os sistemas de regras. Este tipo de agente têm uma capacidade local de decisão e podem negociar uma informação.

FIGURA 2
ESTRUTURA DOS AGENTES

Quanto a hierarquia, existem os níveis: (a) superior: formado pelo Agente de Interface e (b) inferior: constituído pelos agentes fornecedores de serviços, ou seja, o agente responsável por um determinado domínio como mostra a Figura 2. Entre os agentes fornecedores de serviços não existe hierarquia, todos eles podem conversar, negociar e trocar informações dependendo dos seus interesses.

O MathTutor tem uma arquitetura cooperativa, onde diferentes agentes precisam cooperar para encontrar a solução para o problema de cada aluno [14].

Os componentes vitais para o funcionamento e caracterização do MathTutor como um STI são: a base de conhecimento e os agentes. O MathTutor é formado por quatro agentes, onde dois deles são responsáveis pelo conteúdo teórico e os outros dois pelo prático, segundo a modelagem do domínio.

Para o sucesso do sistema são necessários o compartilhamento de conhecimento e um aprendizado das preferências do usuário.

5. IMPLEMENTAÇÃO DOS AGENTES

Os agentes do MathTutor utilizam uma representação do conhecimento comum. Estas características permitem simplificar o processo de compartilhamento e raciocínio. Os agentes se comunicam a fim de solicitar cooperações entre si, solucionando problemas mais rapidamente. O ambiente que permite a troca de mensagens entre os agentes é o JATLite (Java Agent Template Lite) [11], através da linguagem KQML. O conteúdo destas mensagens são fatos que serão inseridos na máquina JESS (Java Expert System Shell) [8], permitindo que o sistema tome decisões e mude o comportamento. A comunicação do usuário com o sistema ocorre através da interface, que é uma página HTML com conteúdo gerado por Servlets. A comunicação entre os agentes é feita através de performativas. As performativas armazenam ordens implícitas aos agentes.

O JATLite pode ser definido como um conjunto de classes escrito em linguagem Java que provê uma arquitetura básica para a construção de agentes que se comunicam através da Internet. A transmissão das mensagens entre os agentes pode se dar através de um mecanismo de *pooling* onde o agente emissor verifica se o receptor está conectado e envia a mensagem. As conexões entre os agentes são feitas de uma maneira persistente, ou seja, fica ativa até que o agente resolva fechá-la ou um tempo máximo de ociosidade seja atingido (*timeout*). A troca de mensagens entre os agentes é feita conforme a Figura 3.

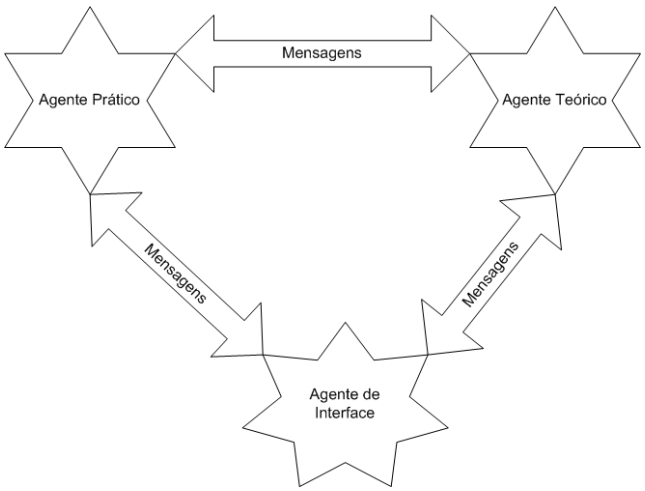

FIGURA 3
TROCA DE MENSAGENS

O JESS é utilizado em diversas aplicações, mas o uso do JESS com a tecnologia dos Applets deixa o sistema muito pesado. Por isso quando a idéia é utilizar aplicações com JESS via navegador, devemos considerar o uso do JESS do lado do servidor, como o que ocorre no caso dos Servlets, dispensando o usuário de carregar grande parte do sistema para sua máquina o que torna a interação com o sistema bastante lenta e entediante do ponto de vista do usuário.

6. TRABALHOS CORRELATOS

A utilização de sistemas multiagentes em STI's cresceu significativamente nos últimos anos. Alguns trabalhos nesta área, semelhantes ao MathTutor, serão apresentados a seguir.

O sistema Multi-agent Co-Operative Environment (MCOE) [10], desenvolvido no Grupo de IA em Educação da Universidade Federal do Rio Grande do Sul, é a implementação de uma arquitetura multiagentes, que combina agentes reativos e cognitivos, e que apresenta uma metáfora de um ecossistema com vários elementos (peixes, plantas e microorganismos) e personagens (turista, governante, etc.) que podem atuar sobre o ecossistema de forma positiva ou negativa, sob o ponto de vista de poluição e controle do equilíbrio ambiental.

O Eletrotutor III [1] implementa um ambiente distribuído de ensino-aprendizagem inteligente baseado em uma arquitetura multiagente. A sociedade é composta por sete agentes. Cada um deles possui uma função específica. As estratégias de ensino consistem na seqüência de conteúdos, de exemplos e de exercícios que serão propostos ao aluno. A avaliação é realizada por uma série de até, no máximo, sete vezes o mesmo tipo de exercícios.

Um outro sistema tutor inteligente é o ELM-ART Lisp Course [4] desenvolvido na Alemanha. É um STI na web para dar suporte ao ensino da linguagem de programação Lisp. O sistema é dividido em subseções associadas com conceitos a serem estudados. Todas as interações do estudante são registradas em um modelo individual autônomo.

7. CONCLUSÕES

Atualmente as pesquisas em STI ou ambientes de ensino-aprendizagem, preocupam-se com a construção de ambientes que possibilitem um aprendizado mais eficiente. A tecnologia dos agentes tornou os STI mais adaptados às necessidades e características individuais de cada estudante. O MathTutor utiliza agentes visando obter um sistema com maior qualidade e flexibilidade seja sob a ótica do estudante ou do especialista.

Construir os agentes e colocá-los em operação de acordo com as funcionalidades propostas é uma tarefa complexa.

A interação entre estudante e tutor tenta se aproximar cada vez mais da interação estudante professor tornando o sistema uma extensão da sala de aula e fazendo com que esta interação ocorra de maneira mais natural possível. Contudo existe uma lacuna em relação a modelagem do estudante, por exemplo, quando o professor percebe uma determinada expressão na face do estudante que o tutor não é capaz de perceber. Por fatores semelhantes a estes, a modelagem do estudante têm sido o foco de pesquisa nos últimos anos na área de STI's.

A busca por novos paradigmas trouxe para a IAD a tecnologia dos agentes, que são utilizados com finalidades distintas em diversas áreas, mas que em muito tem contribuído para o crescimento dos STI.

REFERÊNCIAS

[1] BICA, F., SILVEIRA, R., e VICCARI, R."Eletrotutor III: Uma Abordagem Multiagentes." *IX Simpósio Brasileiro de Informática na Educação/SBIE*, 1998.

[2] BITTENCOURT, G. "Inteligência Artificial Distribuída", Anais: *I Workshop de Computação do ITA*, 1998.

[3] BITTENCOURT, G. *Inteligência Artificial: ferramentas e teorias*, Editora da UFSC, 1998.

[4] DURFEE, E. H. "A Unified Approach to Dynamic Coordination: Planning Actions and Interactions in a Distributed Problem Solving Network", *COINS Technical Report* 87-84, Univ. of Massachusetts at Amherst, 1987.

[5] ELM-ART "Lisp Course" Disponível em: apsymac33.unitrier.de:8080/elm-art/login-e, 2001.

[6] FERBER, J. *Multi-Agent System, An Introduction to Distributed Artificial Intelligence* Addison-Wesley Publishers, 1999.

[7] FRANKLIN, S. E GRAESSE, A. "Is it na agent, or just a program? A taxonomy for autonomous agents" *Proceedings of the Third International Workshop on Agent Theories, Architecture and Languages*, 1996, Springer-Verlag.

[8] FRIEDMAN-HILL, E. "Jess, the Java Expert System Shell". Disponível em: http://herzberg.ca.sandia.gov/jess, 1997.

[9] FRIGO, L.B. e BITTENCOURT, G. "MathTutor: Uma ferramenta de apoio a aprendizagem" *Anais do XXII Congresso da Sociedade Brasileira de Computação* – X WEI, 2002.

[10] GIRAFFA, L. M., VICCARI, R.M. e SELF, J. "Multi-agent based pedagogical games." *Inteligent Tutoring Systems* – ITS, 1998.

[11] JATLite "Java Agent Template Lite" Stanford University, 1997, Disponível em: http://java.stanford.edu/java_agent/html/index2.html,

[12] JENNINGS, N. R. "Cooperation in Industrial Multi-agent Systems", World Scientific, 1994.

[13] JENNINGS, N. R., WOOLDRIDGE, M., KINNY, D. "A Methodology for Agent-Oriented Analysis and Design" *Proc. 3rd Int Conference on Autonomous Agents* (Agents-99) Seattle, WA. 28 ,1999.

[14] MATHNET , Relatório Técnico 1999/2000, LCMI http://www.das.ufsc.br/mathnet/rel-00

[15] PARUNAK, H. V. D. "Applications of Distributed Artificial Intelligence in Industry" *Foundations of Distributed Artificial Intelligence.* Wiley Inter-Science, cap 4, 1994.

[16] RABELO, R. J. " Sistemas Multi-agentes" Disponível em : http://www.das.ufsc.br/~rabelo,2002.

[17] WORTMANN, H. e SZIRBIK, N. "ICT Issues among Collaborative Enterprises: from Rigid to Adaptive Agent-Based Technologies" *International Journal of Production Planning and Control* 12 (5) 452-465, 2001.