AprendEAD: Ambiente para Educação à Distância Apoiado em Agentes

Teresa Cristina B. Dantas, Gustavo E. Soares, Rosa Maria E. M. Costa,
Vera Maria B. Werneck, Maria Clicia S. de Castro

Universidade do Estado do Rio de Janeiro
Dept. Informática e Ciência da Computação
gustavoespose@gmail.com, {rcosta, vera, clicia@ime.uerj.br}

Abstract

The growth of the Internet and the advances of the computer technology are opening new possibilities to explore the Artificial Intelligence techniques to optimize the exchanges between the different actors involved in the teach-learning processes. In particular, the concepts of Intelligent Agents offer new perspectives for the development of more efficient tools to support the Distance Education. The objective of this work is to discuss the main concepts and involved challengers in the creation of a pedagogical tool to support the Distance Education, and to present the AprendEAD system that uses the principles of multiagent systems to promote the interaction between students and professors.

1. Introdução

A disseminação dos computadores pessoais e da rede mundial de comunicação, Internet, tornou possível a transferência de informação em escala global, de qualquer material multimídia e de sua integração aos mais diversos meios de comunicações. A disponibilidade desses vários recursos tecnológicos associados ampliou as suas fronteiras e permeou a área de ensino-aprendizagem.

Atualmente, os computadores pessoais e a Internet são importantes ferramentas no cenário da educação a distância (EAD). Esses recursos oferecem oportunidades de comunicação entre participantes distribuídos em distintas regiões geográficas e de acesso a conteúdos digitais, previamente preparados por professores e tutores. Neste sentido, torna-se premente o apoio de ambientes específicos que facilitem as atividades envolvidas nos processos educacionais à distância.

Os ambientes de apoio à EAD hoje disponíveis exploram modelos e tecnologias que dificultam a manutenção e a automação de atividades. Uma exceção é o ambiente Moodle [18], que se apóia em software livre e possui uma estrutura modular, onde novos componentes podem ser adicionados. Entretanto, o Moodle foi desenvolvido na linguagem PHP, que apresenta limitações por realizar todo o processamento na máquina servidora, diminuindo assim sua eficiência no atendimento síncrono a grandes grupos de alunos.

As técnicas de Inteligência Artificial, em especial, a abordagem de agentes inteligentes, podem ajudar a controlar alguns procedimentos envolvidos na comunicação dos usuários com o ambiente, automatizando o acesso às informações e a distribuição de conteúdos. Estas técnicas aliadas à linguagem JAVA e toda a gama de aplicativos integrados, ampliam a capacidade de acesso a um maior número de alunos, já que o aplicativo roda na máquina do cliente. Outra vantagem da linguagem JAVA é ser Orientada à Objeto, que promove uma maior segurança e regularidade [4] e ainda, pode tratar de eventos de iteração, fornecendo uma ilimitada gama de *applets* dinâmicos, satisfazendo as necessidades de seus diferentes usuários [23].

Neste sentido, o objetivo deste trabalho é experimentar técnicas e ferramentas de modelagem e implementação de Agentes Inteligentes e Sistemas Multi-agentes para a construção de um ambiente de apoio a EAD, utilizando a Internet como canal de comunicação e as tecnologias Java, JSP e *Servlets*.

A organização deste trabalho é a seguinte: na Seção 2 apresentamos os conceitos de agentes inteligentes. Na Seção 3 abordamos a utilização de sistemas Multi-agentes. Na Seção 4 discutimos a Educação à Distância. Na Seção 5, o desenvolvimento do ambiente AprendEAD é apresentado detalhadamente. Finalizamos na Seção 6 com nossas conclusões, ressaltando algumas diretrizes obtidas a partir do desenvolvimento do AprendEAD e tecemos sugestões para trabalhos futuros.

2. Agentes Inteligentes

Na área de Inteligência Artificial Distribuída (IAD) existem diferentes abordagens na solução de problemas. Uma dessas abordagens trata dos processos de busca de soluções através da interação entre entidades em um meio comum. Neste caso, os sistemas multiagentes se destacam como o modelo mais apropriado. Um agente percebe através de seus sensores os estímulos do ambiente em que está inserido. Em seguida, utiliza sua lógica para determinar suas ações. É através dos atuadores, que irão interagir com o meio, onde são realizadas estas ações [3]. Um agente pode ser considerado como uma abstração de software, cujo enfoque é dado pelas ações e tarefas que executa, através de um comportamento autônomo e dotado de inteligência [27]. Assim, um agente inteligente atua promovendo automatização de trabalhos repetitivos, interage com outros agentes ou usuários, notificando, aprendendo e questionando.

Um agente, em geral, é determinado por suas características básicas, tais como: autonomia, adaptação,

comunicabilidade, mobilidade, persistência, pró-atividade ou orientação ao objetivo, reatividade, representatividade, sociabilidade ou cooperatividade [21].

A abordagem de agentes é uma tendência para o desenvolvimento de sistemas e traz como vantagens a modularidade, possibilitando assim, a reutilização de componentes e a facilidade de manutenção.

3. Sistemas Multiagentes

Os sistemas multiagentes são compostos por uma sociedade de agentes autônomos. Eles cooperam de forma coordenada, em busca de objetivos comuns, compartilhando suas informações, conhecimentos, suas habilidades e planos individuais. Dessa forma, eles podem inferir as melhores estratégias para a resolução de problemas, através de uma abordagem global [27].

O paradigma distribuído torna possível realizar execuções mais rápidas e eficientes em virtude das características do processamento paralelo e das metodologias de tolerância a falhas (maior segurança e confiabilidade).

Percebe-se que, em geral, estes sistemas possuem as seguintes características: redução dos custos de tecnologia; distribuição de informação e dados; modularidade; processamento paralelo mais rápido se comparado a sistemas não distribuídos; processamento em múltiplas plataformas; segurança no fluxo de informações; sistemas tolerantes a falhas e confiáveis; solução de problemas complexos; sociabilidade e interoperabilidade (entre agentes, usuários e sistemas).

Atualmente, existem alguns *frameworks* e plataformas específicas para o desenvolvimento de aplicações Multiagentes, dentre os quais destacam-se JADE **(***Java Agent Development Framework)* [11], OAA **(***Open Agent Architecture)* [22] e NetBeans [20].

4. Educação à Distância

Nos últimos anos, observa-se um expressivo crescimento da oferta de cursos a distância (EAD) no Brasil. Em geral, estes cursos adotam modelos de desenvolvimento onde a comunicação, entre alunos e professores, é mediada por recursos didáticos sistematicamente organizados, apresentados através de variados suportes de informação e plataformas eletrônicas, que promovem as interações síncronas ou assíncronas entre os envolvidos no processo educacional.

O processo educacional realizado a distância envolve a articulação de uma série de ações pedagógico-administrativas, onde se destacam a construção do material didático, a estrutura de tutoria, a montagem da infra-estrutura, a gestão do sistema e a avaliação. Neste sentido, é fundamental a utilização de um ambiente de trabalho que disponibilize vários tipos de serviços e apóie a interação e a comunicação dos diferentes atores envolvidos. Assim, para viabilizar a educação a distância com o apoio da *Internet*, são necessários ainda sistemas de mensagens assíncronas ou síncronas, de co-autoria de conteúdos, de apoio a reuniões, de conferências e de controle acadêmico, entre outros.

Diferentes ambientes de apoio a cursos a distância vêm sendo utilizados em cursos no Brasil e no exterior, dentre os quais, destacam-se o AulaNet [1], o LearningSpace [17], o TelEduc [25], o Moodle [18] e o e-PROINFO [6]. Entretanto, a maioria destes sistemas não explora técnicas mais avançadas de inteligência artificial, que poderiam automatizar tarefas que hoje dependem de ações humanas, tais como: controlar a liberação de textos e avaliações das disciplinas para os alunos; enviar relatórios analíticos para o supervisor, controlar o processo de tutoria, ou gerar avisos automáticos.

A seguir, são apresentadas as principais características de um ambiente de apoio a cursos a distância, que explora tecnologias de Inteligência Artificial para tentar amenizar alguns problemas encontrados nos ambientes de apoio à EAD hoje disponíveis.

5. O ambiente APRENDEAD

O ambiente AprendEAD foi concebido de maneira a integrar uma sociedade de agentes que seria responsável pela realização de diferentes tipos de tarefas de coordenação e acadêmicas. Os agentes estão inseridos como assistentes que executam individualmente, um conjunto de tarefas e fornecem apoio tanto ao estudante, quanto ao professor e à administração dos cursos.

Nesta primeira versão não foram desenvolvidas as ferramentas de comunicação síncrona e assíncrona, como *chats* e fóruns, pois queríamos estudar a viabilidade técnica de modelagem e implementação da estrutura básica do sistema. A princípio, consideramos que devido à modularidade do sistema, estas atividades podem ser implementadas sem maiores problemas em um futuro próximo.

Os sistemas multiagentes são considerados capazes de realizar ações flexíveis e muitas vezes, autônomas em um ambiente dinâmico. Sua modelagem demanda um alto nível de abstração na forma de pensar sobre as características e os componentes do sistema, exigindo o apoio de metodologias de modelagem. Este aspecto será abordado na seção a seguir.

5.1. O Desenvolvimento do AprendEAD

O desenvolvimento de sistemas multiagentes exige cuidado especial, pois é necessário considerar os vários objetos, atividades, atores e recursos.

Na literatura da área existem várias metodologias de modelagem orientadas a agentes, tais como TROPOS, GAIA, MAS-CommonKADS, ADELFE, MESSAGE e INGENIAS, cada uma aportando suas especificidades e limitações [8], [24].

Neste caso, optamos por modelar o ambiente utilizando uma abordagem híbrida, que integra a UML-Unified Modeling Language [2] e o MAS-CommonKADS [9], pois assim associamos uma

linguagem padrão de orientação a objetos à uma metodologia oriunda de engenharia de conhecimento.

A UML por ser uma linguagem de representação bem conhecida na área de Engenharia de Software, facilitou a definição dos atores e dos agentes do ambiente. MAS-CommonKADS é uma extensão da metodologia CommonKADS [9] englobando aspectos que são relevantes para sistemas multiagentes. O CommonKADS tornou-se, principalmente na Europa, uma referência no desenvolvimento de Sistemas Baseados em Conhecimento (SBC).

O MAS-CommonKADS se integra bem com a UML, pois possui uma abordagem de Casos-de-uso com representação semelhante à UML o que contribui para seu entendimento e uso. Na fase de identificação dos requisitos, os casos de uso externos, que são semelhantes aos casos de uso do UML são definidos em MAS-CommonKADS.

Para o AprendEAD foi desenvolvido um diagrama de Casos de Uso Externos. Nele é possível identificar três atores que interagem com o ambiente: aluno; professor; e administrador, sendo definidas as atividades que cada um deles pode realizar no ambiente. A definição destes Casos de Uso foi essencial para definir os requisitos iniciais do sistema, contemplando atividades fundamentais para uma ferramenta de apoio à EAD.

O modelo de agentes no MAS-CommonKADS tem o objetivo de descrever os agentes que participam da solução dos problemas e são descritos principalmente nos diagramas de casos de uso interno (extensão do conceito de casos de uso do UML para agentes humanos e de software), nos diagramas de seqüência de mensagens relativas aos Casos de Uso Internos e no Modelo de Agentes e de Distribuição de Tarefas e Agentes.

O modelo de agentes do AprendEAD procurou identificar as habilidades e restrições de cada agente, sendo representado através de uma coleção de casos de uso internos. Um exemplo de um desses Casos de Uso Internos pode ser observado na Figura 1.

Figura 1. Caso de Uso Interno "Fazer Cadastro de um novo aluno"

Os Casos de Uso Internos apresentam as relações dos atores com os agentes e estão sempre relacionados com as atividades desempenhadas pelos agentes.

A Figura 2 apresenta o detalhamento de um Caso de Uso, destacando as atividades desempenhadas pelo Agente Cadastro e seu Modelo de Tarefas.

Fazer Cadastro de um Novo Aluno	
Ator Principal	Aluno
Agente	Agente Cadastro
Resumo	Este caso de uso define o cadastramento do aluno no sistema.
Ações	01. Novo aluno fornece seus dados: nome, endereço, bairro, cidade, estado, CEP, telefone, *e-mail*, *login* e senha. 02. Aluno solicita cadastro após fornecer os dados. 03. Agente valida dados. 04. Agente efetua cadastro. 05. Aluno recebe confirmação do casdastramento.
Fluxo Alternativo	O aluno poderá sair do sistema a qualquer momento.
Modelo de Tarefas - Agente Cadastro	
Plano de Ação	02a. Recebe os dados do aluno. 02b. Verifica os campos vazios, além dos campos CEP, telefone, *login* e senha. 03a. Enquanto houver dados inválidos, solicita novos dados. 04a. Gera número de registro para o aluno. 04b. Trata os dados digitados para inserir no BD e armazena os mesmos. 05a. Envia mensagem de confirmação para o aluno, apresentando os dados armazenados. 05b. Direciona o aluno para página inicial do aluno, repassando seu nome e registro.
Interação	AlunoPagPrincipal.jsp

Figura 2. Descrição do Caso de Uso "Fazer cadastro de um novo aluno"

Para que o professor possa redigir uma aula é necessário que haja interação entre agentes: o agente "cadastro" reconhece a senha do professor pra que o agente "Gerenciador de Curso" apresente a tela para a entrada do conteúdo das aulas.

O sistema automatiza algumas atividades de acordo com a participação do aluno no curso. Por exemplo, se o aluno ficar muito tempo sem acessar o sistema, uma mensagem pode ser disparada para incentivar o aluno a prosseguir na realização das atividades do curso.

A comunicação entre os agentes foi modelada nos diagramas de seqüência de mensagens do MAS-CommonKADS que se assemelham aos Diagramas de seqüência do UML. Estes interagem a partir de uma solicitação de um agente humano ou quando um agente inicia uma atividade pró-ativa. Na Tabela 1 estão relacionados alguns agentes de software e suas tarefas no ambiente.

No ambiente AprendEAD foi utilizado um agente de Interface para promover um enriquecimento da interação com o usuário. O *Peedy* é uma ferramenta disponibilizada gratuitamente pela *Microsoft*. Ela oferece uma figura animada, que pode se movimentar por toda a tela, falar (balões de escrita e voz), sussurrar, pensar, reconhecer comandos vocais, permitindo grande flexibilidade e maior interatividade. O *Peddy* aparece quando se quer chamar a atenção sobre algum procedimento, atividade ou situação. Os estados emocionais do *Peedy* utilizados no ambiente são controlados através da tecnologia *JavaScript*, e alguns exemplos deles estão relacionados na Tabela 2.

Tabela 1. Alguns agentes e suas respectivas tarefas

Agentes	Tarefas
Agente Cadastro	Faz o cadastro de um novo aluno. Apresenta o cadastro do aluno e os campos disponíveis para alteração(ões). Valida os dados alterados, realiza as alterações de cadastro solicitadas pelo aluno e apresenta o cadastro alterado. Verifica e valida a senha do aluno. Faz o cadastro de um novo professor. Apresenta o cadastro do professor e os campos disponíveis para alteração(ões). Valida os dados alterados, realiza as alterações de cadastro solicitadas pelo professor e apresenta o cadastro alterado. Verifica e valida a senha do professor. Verifica e valida a senha do administrador.
Agente Acadêmico	Cadastra curso. Apresenta a página principal do curso solicitado pelo aluno. Apresenta a ementa do curso selecionado. Inicia a aula do curso. Realiza a matrícula de um aluno no curso escolhido. Cancela a matrícula de um aluno no curso selecionado Valida os dados alterados do curso, realiza as alterações de cadastro solicitadas pelo administrador e apresenta o cadastro alterado. Apresenta o boletim do aluno com todos os cursos em que ele está inscrito, suas notas e situações. Armazena as notícias de interesse dos alunos em um arquivo. Apresenta as notícias gravadas no arquivo. Armazena as sugestões digitadas pelos professores e alunos em um arquivo. Apresenta as sugestões gravadas no arquivo.
Agente Gerenciador de Curso/Disciplina	Mostra uma página onde o professor redige a ementa do curso. Apresenta a página principal de um curso escolhido pelo professor. Lista os alunos inscritos em um curso. O professor redige a aula. Determina os pré-requisitos para que as avaliações sejam disponibilizadas para os alunos. O professor determina as condições e mensagens a serem distribuídas. Apresenta o cadastro do curso e os campos disponíveis para alteração(ões).

Tabela 2. Peedy e alguns de seus estados emocionais

Estado Emocional	Representação	Estado Emocional	Representação
Aceno		Alerta	
Anunciar		Atenção	

5.2 Ferramentas Utilizadas na construção do AprendEAD

A construção do AprendEAD foi apoiada por várias ferramentas de livre distribuição e *open source*, citadas a seguir.

- **Netbeans.** A plataforma de desenvolvimento escolhida foi a Netbeans [20], distribuída pela *Sun Microsystems*. Ela possui vários recursos para múltiplas plataformas, que permitem a criação de inúmeras aplicações. No Netbeans é possível criar aplicações *Web* com grande flexibilidade, disponibilizando uma simulação completa de um servidor, conexões com bancos de dados e várias outras tecnologias. Além disso, ela promove uma boa visualização e estruturação do projeto, compilação e depuração do código Java.
- ***Servlets.*** *Servlets* Java compõem a base do desenvolvimento do aplicativo *Web*, usando a linguagem de programação Java. Eles trabalham por meio de um modelo solicitação-resposta, baseado em HTTP. Segundo Kurniawan [16], os *servlets* são módulos que estendem a capacidade de um servidor, permitindo o processamento otimizado de tarefas na máquina servidora, melhorando o envio de respostas ao cliente. Eles facilitam a implementação da comunicação cliente/servidor, além de prover mais segurança aos dados.
- **XML.** Acrônimo de *Extensible Markup Language* [26], é uma linguagem de marcação extensível que permite e facilita o compartilhamento de dados entre diversos sistemas de informações, principalmente para os interligados via Internet. Ela foi utilizada para evitar acessos desnecessários ao banco de dados, no acesso à dados estáticos e no controle e mapeamento dos *servlets*.
- **CSS.** O *Cascading Style Sheets* [5] possibilita definir uma estrutura lógica do texto apresentado em uma página. Ele é muito flexível**,** uma vez que permite que vários *layouts* sejam pré-definidos em arquivos externos. Ou seja, toda a formatação do texto está em documento externo à *homepage*.
- **JSP.** *Java Server Pages* [15] é uma tecnologia que possibilita aos desenvolvedores e projetistas de *Web*, a criação rápida e fácil manutenção de documentos. Como definido por Hall [7], estes documentos (páginas JSPs) são caracterizados por um documento HTML, incrementado com código Java. A tecnologia JSP é uma extensão da tecnologia Java *Servlet*. Note que ao utilizar a linguagem Java, os documentos podem herdar todo o conjunto de suas características, oferecendo portabilidade através de plataformas, *browsers* e servidores, desempenho elevado para múltiplas requisições concorrentes de cliente ao servidor, criação de páginas *Web* dinâmicas, fácil implementação e manutenção do código fonte.
- **Javascript.** Linguagem do tipo *script* [13] que é interpretada. Ela é baseada em Java e permite a inclusão de características e conteúdos dinâmicos e ativos em páginas HTML.
- **Java Beans [12]**. São componentes de software cuja principal funcionalidade é a possibilidade de reutilização em diversas aplicações.
- **MySQL**. É um sistema de gerenciamento de banco de dados [19]. Foi utilizado por ser multiplataforma, *multithread* e multiusuário, além de ser muito popular para aplicações *Web*.
- **JDBC.** A integração da base de conhecimento e os agentes foi realizada através da biblioteca JDBC (*Java Database Connector*). É uma interface que diminui a complexidade na execução e gerenciamento de instruções SQL em ambientes distribuídos [14]. De modo geral, a arquitetura de software para uso do JDBC é simples. Uma aplicação solicita ao JDBC para que seja criada uma conexão, via um *driver JDBC*, a uma base de dados. A partir do estabelecimento da conexão, um serviço de passagem de mensagens permite à aplicação o envio de *queries* (instruções SQL) ao banco de dados, que executa os pedidos da aplicação e envia os resultados.

A escolha dessas tecnologias se fez a partir de um estudo das possibilidades de integração dos diferentes componentes necessários para o desenvolvimento deste ambiente. A tecnologia Java favorece o desenvolvimento de agentes de software e a execução em múltiplas plataformas, principalmente, para o uso de agentes móveis. O uso dos *servlets* introduz a troca de informações entre o servidor e o cliente. O *Java Beans* e o paradigma da orientação a objetos, aplicados à implementação em Java, permitem o reuso e extensibilidade dos componentes.

No AprendEAD identificamos três atores que interagem com o ambiente: aluno; professor; e

administrador. O aluno interage com o sistema participando dos cursos. Professor é o ator responsável pelo conteúdo e avaliações dos cursos. Finalmente, o Administrador é responsável pelo cadastramento dos professores, análise de críticas e sugestões e atualização de notícias.

O acesso ao ambiente é realizado através de páginas *Web* amigáveis. Para isso, é necessário primeiramente que os usuários se cadastrem no sistema e, com isso possua uma chave alfanumérica (*login*) e sua respectiva senha (*password*).

A seguir apresentamos algumas das telas do ambiente AprendEAD. A página inicial (Figura 3) é compartilhada por todos os usuários, apresenta um menu para acesso ao ambiente do aluno, do professor e para cadastro de novos alunos.

A Figura 4 apresenta a página inicial do aluno, com a presença do agente de Interface, que neste caso, poderá dar dicas sobre os cursos disponíveis.

Figura 3. Página Inicial do AprendEAD

Figura 4. Página Inicial do aluno, com a presença do agente de interface

6. Conclusões

Neste trabalho foram discutidas importantes questões associadas ao desenvolvimento de ambientes de apoio à EAD e foi apresentado o processo de desenvolvimento de um ambiente que explora os sistemas multiagente e utiliza ferramentas de livre distribuição.

O protótipo do ambiente AprendAED foi desenvolvido com tecnologias para múltiplas plataformas, como Java, JSP, e *Servlets*. Dessa forma, foi criado um ambiente flexível e apto a solucionar problemas dinâmicos e complexos de acesso e distribuição de conteúdos.

Um dos principais objetivos deste projeto foi estudar a tecnologia de agentes inteligentes, testando algumas técnicas a partir da criação deste ambiente multiagentes. Apesar da complexidade envolvida na modelagem e implementação desses agentes, pudemos observar que os sistemas multiagentes oferecem novas possibilidades para a criação de ambientes computacionais que necessitam de menos intervenções humanas.

A utilização de uma metodologia de modelagem específica para sistemas multiagentes foi fundamental para o entendimento da complexa estrutura que envolve a integração dos agentes inteligentes. O estudo preliminar das diversas propostas encontradas na literatura foi o processo-chave para que concluíssemos que a metodologia MAS-CommonKads tinha muitas similaridades com a UML, o que facilitou o entendimento e a posterior modelagem do ambiente.

A partir desta experiência, destacamos alguns aspectos que podem ser considerados como diretrizes para a construção de ambientes multiagentes de apoio à EAD:

- Explorar tecnologias de software livre, o que aumenta o potencial de utilização da ferramenta e facilita o seu desenvolvimento;
- Considerar a utilização de agentes de interface, que podem intervir quando perceberem algum problema na interação do usuário com o ambiente;
- Realizar estudos preliminares que apontem a metodologia de modelagem mais adequada para o tipo de ambiente e de tarefas a serem disponibilizadas para os usuários;
- Considerar uma completa independência em relação aos domínios da aplicação, para que o ambiente tenha um maior potencial de utilização em diferentes cursos;

Uma das propostas para futuro trabalhos é desenvolver agentes cognitivos, que permitiriam uma interação mais natural com o usuário. Como conseqüência, alterar ou acrescentar características ao agente pedagógico tornando seu comportamento mais próximo do ser humano. Além disso, o ambiente poderia ser integrado com *webcans*, aumentando o nível de interação entre os atores envolvidos no processo de ensino-aprendizagem.

7. Referências

[1] AulaNet, Em: http://www.aulanet.com.br, visitado em maio/2007.

[2] G. Booch, , J. Rumbaugh, I. Jacobson, *UML: Guia do Usuário*, Ed Campus, 2007.

[3] J. Bradshaw, *An Introduction to Software Agents*, Ed Software Agents, MIT Press Massachusetts, 1997.

[4] K. Carlson, "Comparing Web Languages in theory and Practice", Research Reporter INSS690, Bowie State University, 2005, em http://faculty.ed.umuc.edu/~meinkej/inss690/carlson.pdf, visitado em maio/2007.

[5] CCS - Cascading Style Sheets, Em: http://www.w3.org/Style/CSS, visitado em abril/2007.

[6] e-PROINFO, em: http://www.eproinfo.mec.gov.br/fra_eProinfo.php?opcao=1, visitado em abril/2007.

[7] M. Hall, *Core Servlets and Java Server Pages*, Prentice Hall, 2000.

[8] B. Handerson-Sellers, P. Giorgini, *Agent-Oriented Methodologies*, IDEA Group Publishing, USA, 2005.

[9] Iglesias, C.A., Garijo, M., "The Agent Oriented Methodology MAS-CommomKADS", *Agented-Oriented Methodologies*, B. Henderson-Sellers e P. Giorgini (ed.), IDEA Group Publishing, p. 46-78, 2005.

[10] C.A. Iglesias, M. Garijo, J.C. González, J. Velasco, *A Methodological Proposal for Multiagent Systems Development extending CommonKADS,* 1996. Em: http://ksi.cpsc.ucalgary.ca/KAW/KAW96/iglesias/Iglesias.html, visitado em dez/2006.

[11] JADE- Java Agent Development Framework. Em http://jade.tilab.com/, visitado em dez/2006.

[12] Java Beans, em: http://java.sun.com/products/javabeans, visitado em abril/2007.

[13] JavaScript, em: http://javascript.internet.com, visitado em abril/2007.

[14] B. Jepson, *Java Database Programming*. Wiley Computer Publishing, 1997.

[15] JPS - Java Server Pages, Em: http://java.sun.com/products/jsp, visitado em abril/2007.

[16] Kurniawan, B., *Java for the Web with Servlets, JSP, and EJB*. New Riders, 2002.

[17] LearningSpace, em: http://www.lotus.com/home.nsf/welcome/learnspacevisitado em fev/2007.

[18] Moodle, em: http://moodle.org/, visitado em nov/2006.

[19] MySQL, em: http://www.mysql.org, visitado em abril/2007.

[20] NetBeans, em: http://www. netbeans.org, visitado em jan/2007.

[21] Nwana, H. S., *An Introduction to Agent Technology*, Re-Drawn by Mobile Computing, Dept. of IECS, Feng Chua University, R.O.C., 2003.

[22] OAA, em: http://www.ai.sri.com/~oaa, visitado em nov/2006.

[23] A. Sabariz, Jardim, A., Cunha, M., "Uso de PHP e JAVA para aplicações na Educação a Distância", *II Seminário Nacional de Tecnologia para EAD*, Uberlândia, 2002, em: www.ead.ufu.br/tecead_II/anais/pdfs/sabariz.pdf, visitado em 12/2006.

[24] A. Sturm, O.Shehory, "A Framework for Evaluating Agent-Oriented Methodologies", 5th International Workshop on Agent-Oriented Information Systems (AOIS'03), p. 60-67, 2003.

[25] TelEduc, em: http://teleduc.nied.unicamp.br, visitado em jan/2007.

[26] XML - *Extensible Markup Language*, em: http://www.xml.org, visitado em abril/2007.

[27] M. Wooldridge, N. Jennings, *Agent Technology: Foundations, Applications, and Markets*, Springer, 2002.