

Utilização de agentes FIPA em ambientes para Ensino a Distância

Eduardo Rodrigues Gomes¹, Ricardo Azambuja Silveira², Rosa Vicari¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

²DMEC – IFM – Universidade Federal de Pelotas (UFPEL)
Caixa Postal 354 – 96.010-900 – Pelotas – RS - Brasil

ergomes@inf.ufrgs.br, rsilv@ufpel.tche.br, rosa@inf.ufrgs.br

Resumo

Este artigo situa-se na área de Inteligência Artificial aplicada à Educação e enfoca o uso de Sistemas Multiagentes em ambientes de ensino voltados para a Educação à Distância. Propõe uma arquitetura constituída por um conjunto de recursos para a construção de Ambientes Inteligentes Distribuídos de Ensino. Esta arquitetura é baseada numa sociedade composta por agentes autônomos e cooperativo, compatíveis com o padrão FIPA, uma organização que visa padronizar as tecnologias de sistemas multiagentes de forma a permitir a interoperabilidade entre eles. Este artigo apresenta uma proposta de arquitetura de ambiente de aprendizagem baseado em uma sociedade de agentes, que suporta os padrões FIPA.

Palavras-chave: Ambientes Inteligentes de Aprendizagem, Sistemas Multiagentes, FIPA, Inteligência Artificial.

Abstract

This paper deals with Artificial Intelligence applied to Education and describes the use of multiagents approach in distance education learning systems. It proposes a framework to design intelligent distributed learning environments. This framework is based on an autonomous and cooperative agents' society. The agents are FIPA standard compatible. FIPA is an organization which intends to create open standard technologies of multiagent system, in order to promote agent interoperability. This paper presents the proposed architecture of the learning environment based on FIPA compliant agent society.

Key words: Intelligent Learning Environments, Multi-agents Systems, Artificial Intelligence.

Introdução

Desde o início da era da informática, tem-se pensado em utilizar o computador como uma ferramenta educacional. Nesse contexto, a Ciência da Computação, juntamente com a Psicologia e a Educação, têm buscado criar e aperfeiçoar ferramentas computacionais de ensino e, a cada dia, traz novas abordagens para esta área.

A introdução de técnicas de Inteligência Artificial (IA) em ambientes computacionais voltado as à educação propicia a criação de mecanismos para as modelagens do processo de ensino e do estado cognitivo do estudante, o que leva à obtenção de um ambiente adaptado às características de cada aluno. Essa área tem progredido na direção dos denominados Ambientes Inteligentes de Aprendizagem (ILE-*Intelligent Learning Environment*) [5]. Os ILE, influenciados pelo avanço da Ciência Cognitiva, propõem a troca de um paradigma focado no ensino para um paradigma focado na aprendizagem, onde o professor passa a ser visto como um facilitador ao invés de um provedor de informações.

Os avanços mais recentes no campo dos ILE têm proposto a utilização de sistemas multiagentes (SMA). Um SMA é um sistema composto por múltiplas entidades dotadas de um certo grau de autonomia, onde autonomia significa a capacidade de atuar sem interferência humana direta. Tais entidades são denominadas agentes. Os princípios de SMA mostram-se bastante adequados ao desenvolvimento de ambientes computacionais de ensino, tendo em vista que o problema de ensino/aprendizagem é melhor solucionado de forma cooperativa [4]

A Educação a Distância (EAD) envolve a obtenção de conhecimento fora das formas tradicionais de aprendizagem, através da participação em instituições formais de ensino. Spodick [9] apresenta uma definição simples e abrangente para a EAD: “Educação a Distância é um sistema que deve prover toda e qualquer oportunidade educacional que seja necessária para qualquer um, em qualquer lugar, e a qualquer tempo”.

Desde meados de 1980, as atenções em EAD têm se voltado para a utilização de Redes de Computadores. O atual estágio de desenvolvimento no campo de Redes de Computadores tem contribuído com o progresso de ambientes educacionais baseados em arquiteturas cliente-servidor, os quais são chamados de Ambientes Distribuídos de Aprendizagem (DLE – *Distributed Learning Environments*). Esses ambientes permitem o controle centralizado do processo de ensino e, simultaneamente, a ampla distribuição do material instrucional. [6].

Conforme Silveira [6], um sistema computacional de ensino que procure solucionar o problema de ensino/aprendizagem em ambientes de EAD deve contemplar aspectos tanto de IA quanto de Redes de Computadores. Baseados nesse princípio, começaram a surgir os chamados Ambientes Inteligentes Distribuídos de Aprendizagem (DILE – *Distributed Intelligent Learning Environment*). O ambiente JADE, *Java Agent Framework for Distance Learning Environment* apresentado neste trabalho, propõe um conjunto de recursos para a construção de DILE. Ele utiliza a *WWW* e está baseado numa arquitetura multiagente. Portanto, a maior interatividade e adaptação ao estudante são buscadas através da interação e da cooperação entre os seus agentes.

A arquitetura proposta pelo JADE baseia-se em comunicação entre agentes baseado na linguagem KQML, implementada de forma proprietária, restringindo a evolução do sistema e sua interoperabilidade. Por isso, este trabalho relata a evolução deste projeto baseado na adoção do padrão FIPA, conforme relatado em trabalhos anteriores.

A organização FIPA (*Foundation for Intelligent Physical Agents*) surgiu como um esforço na tentativa de padronizar as tecnologias de comunicação entre agentes, que até o seu surgimento eram completamente heterogêneas. Para tal, ela desenvolveu um conjunto de documentos, entre os quais está a descrição de uma linguagem de comunicação totalmente voltada para a comunicação entre agentes: a FIPA-ACL. Os padrões FIPA vêm crescendo em uso e qualidade, o que pode ser verificado através da observação da grande variedade disponível de plataformas de desenvolvimento de agentes compatíveis com eles.

Este artigo apresenta um modelo de sociedade agentes baseado na sociedade de agentes do JADE e que suporta os padrões para interoperabilidade desenvolvidos pela organização FIPA.

1. Arquitetura de Curso

Antes de ser apresentado o modelo de sociedade de agentes proposto neste trabalho é conveniente que seja feita a apresentação dos aspectos arquiteturais que um curso deve contemplar para que possa ser desenvolvido e utilizar as capacidades adaptativas providas por esse modelo.

Um curso é tratado aqui como uma seqüência de unidades, as quais são formados por unidades, constituídas por diferentes tipos de experiência de aprendizagem, cada qual contendo um conjunto de recursos. Cada conjunto de recursos, referido aqui como tática de ensino, está associada a um agente pedagógico e a um conjunto de objetos de aprendizagem. O agente pedagógico é o responsável pelo trabalho de auxiliar no processo de aprendizagem e os objetos de ensino formam a base para que essa tarefa seja efetuada. Um objeto de ensino é qualquer recurso que possa ser utilizado pelo agente pedagógico na geração de alguma experiência de aprendizagem para o aluno. Assim, os objetos de ensino podem ser arquivos de parâmetros, páginas HTML, *applets* Java que implementam simulações, arquivos multimídia, etc. Cada agente pedagógico pode requerer um tipo especial de objeto de ensino.

As avaliações são efetuadas em caráter formativo, de forma adequada para cada tipo de tática de ensino. Esse recurso é utilizado, em geral, para a avaliação de uma unidade e para que o aluno possa se orientar no andamento do curso e na verificação dos seus objetivos.

O processo de criação de um curso de EAD para o sistema proposto envolve as seguintes etapas:

- Estabelecimento dos objetivos pedagógicos do curso e das atividades a serem desenvolvidas para alcançar esses objetivos;
- Divisão do curso em módulos e unidades;
- Definição das táticas de ensino a serem utilizadas em cada unidade. Essa tarefa é composta pela escolha do agente pedagógico e pela construção dos objetos de ensino;
- Definição das estratégias de avaliação para cada uma das táticas e das possíveis combinações de seqüência de táticas;
- Alimentação da base de conhecimento do sistema.

1.1. Base de Conhecimento

A base de conhecimento é responsável por armazenar o conhecimento do sistema sobre os cursos e os alunos. Ela foi mapeada em um banco de dados relacional e desenvolvida através da construção de um modelo Entidade-Relacionamento (ER). O modelo ER obtido suporta vários cursos, isto é, vários DILE podem ser implementados através dele. Esse modelo ER será apresentado em duas partes para melhor compreensão: os modelos ER1-Aluno e ER2-Curso.

O modelo ER1-Aluno (Figura 1) é responsável pelo armazenamento dos dados de identificação e de histórico do aluno. A entidade “Aluno” representa um aluno, que pode estar matriculado em um curso. A matrícula é representada pela entidade “Matricula”. A entidade “HistNav” representa o histórico de navegação do aluno no sistema, nela são armazenadas todas as informações sobre as ações do aluno e as ações do sistema em relação ao aluno. A entidade “HistTat” representa o histórico das táticas com as quais o aluno já trabalhou. A entidade “Tática” e a entidade “Curso”, são modeladas no diagrama ER2-Curso e, por isso, somente as suas chaves primárias são mostradas na Figura 1.

O modelo ER2-Curso (Figura 2) é responsável pelo armazenamento dos dados que formam um curso e dos professores responsáveis pelos cursos. A entidade “Curso” representa um curso criado por um professor responsável, o qual é modelado pela entidade “Professor”. A entidade “Autoria” modela a relação entre um professor e um curso. Um curso é formado por módulos (entidade “Módulo”), os quais são formados por unidades (entidade “Unidade”). As unidades são compostas por táticas (entidade “Tática”). Uma tática possui um agente responsável (“entidade Agente”) e um conjunto de objetos de ensino (entidade “ObjEns”). Os *links*, utilizados para a inclusão de material de apoio a cada unidade, são modelados pela entidade “Link”.

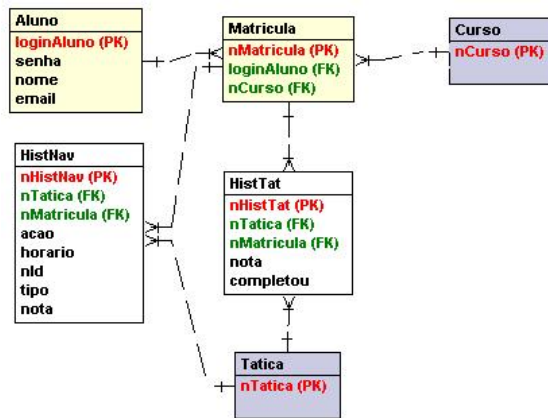


Figura 1. Modelo ER1-Aluno.

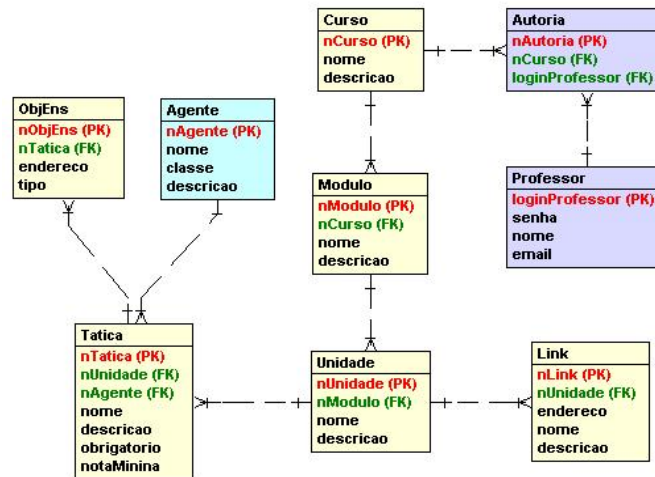


Figura 2. Modelo ER2-Curso.

2. Sociedade de Agentes

A sociedade proposta neste trabalho, como mostra a Figura 3, é composta por dois tipos de agentes: um Agente do Modelo de Aluno e um conjunto de agentes pedagógicos. Os agentes pedagógicos são os responsáveis pelas tarefas de ensino. O Agente do Modelo de Aluno é responsável pela modelagem cognitiva e comportamental do estudante. O ambiente dos agentes corresponde a um ambiente FIPA, o qual provê todos os mecanismos necessários para a comunicação via troca de mensagens.

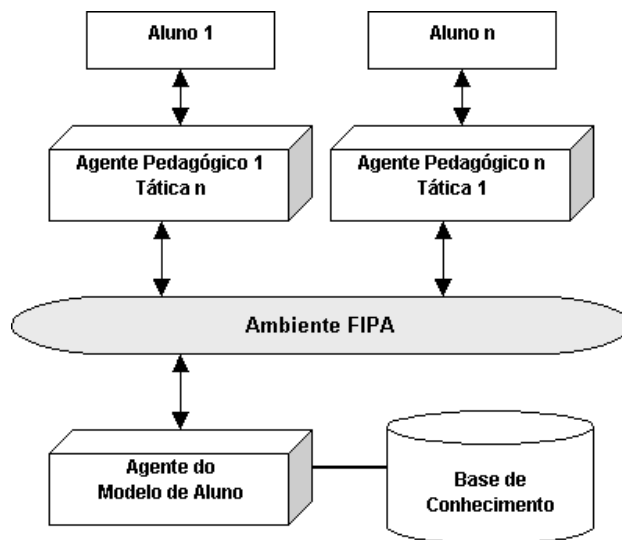


Figura 3. Sociedade de agentes proposta.

2.1. Agente do Modelo de Aluno

O agente do modelo de aluno tem como principal atividade a construção e atualização de uma base de conhecimento acerca do estado cognitivo e comportamental do estudante.

Todas as ações de acesso aos dados do aluno são efetuadas pelo agente do modelo de aluno. Isso significa que sempre que existir a necessidade de um agente pedagógico atualizar os dados de algum estudante, ele deve comunicar esse fato ao agente do modelo de aluno para que este proceda a atualização.

Quando uma sessão tutorial inicia, o agente do modelo de aluno requisita os dados de identificação do aluno para recuperar o Modelo de Aluno correspondente. Numa segunda fase é identificada a parte do curso em que o aluno parou e as táticas de ensino disponíveis para essa parte. Logo após, as táticas selecionadas são apresentadas e, com base numa análise do Modelo de Aluno, é sugerida uma delas. O aluno, então, escolhe uma das táticas, considerando ou não a sugestão do Agente do Modelo de Aluno.

Quando uma nova tática é escolhida pelo aluno ou imposta por algum agente da plataforma, uma instância do agente pedagógico correspondente à tática em questão é criada pelo Agente do Modelo de Aluno e passa a ser a responsável pelo aluno.

Toda a vez que o aluno terminar uma fase do curso, o processo de escolha de tática é feito novamente.

As regras de comportamento que regem a conduta do Agente do Modelo de Aluno são denominadas Regras Macro de Comportamento, pois dizem respeito ao funcionamento do sistema de modo geral. Por exemplo, se um aluno quiser avançar até uma unidade para a qual não tem os pré-requisitos necessários, é tarefa do Agente do Modelo de Aluno, através de suas regras de comportamento, não autorizar esse processo.

2.2. Agentes Pedagógicos

Os agentes pedagógicos são os responsáveis pelas tarefas de ensino. Cada agente pedagógico implementa alguma técnica educacional. Desse modo, existem agentes especializados em geração de exercícios, outros em geração de exemplos e etc.

Dois ou mais agentes podem implementar a mesma técnica educacional, por exemplo: a geração de exercícios. Dessa forma, pode-se ter um agente pedagógico gerando exercícios de múltipla escolha e outro gerando exercícios descritivos.

Um agente pedagógico define as suas próprias regras de comportamento, que são chamadas de Regras Micro de Comportamento, pois dizem respeito somente ao comportamento do agente em relação às ações do aluno durante o seu processo de tutoria.

Como cada agente define as suas próprias regras de comportamento, é possível que dois agentes trabalhem exatamente com a mesma técnica educacional, mas que possuam comportamentos diferentes. Como exemplo, dois agentes podem gerar exercícios de múltipla escolha e terem comportamentos diferentes quanto à avaliação do aluno. Isso evidencia a robustez do modelo de agentes definido neste trabalho.

Uma característica importante de um agente pedagógico é a capacidade de poder avaliar um aluno durante o seu processo de tutoria. Essa propriedade pode ser usada por agentes pedagógicos especializados em avaliação de alunos na tentativa de suprir a falta de um processo explícito de avaliação no sistema.

2.3. Desenho Interno dos Agentes

Os agentes definidos neste trabalho possuem dois componentes para interação externa: o componente de interface e o componente de comunicação. O componente de interface é responsável pela interação entre o aluno e o sistema através da apresentação de uma interface. Já o componente de comunicação é responsável pela interação com os outros agentes e com o ambiente FIPA através de troca de mensagens FIPA-ACL. O desenho e o processamento interno dos agentes é mostrado na Figura 4.

A incorporação das funções de interface pelos agentes assegura maior autonomia de interação com o aluno. A interface pode ser definida de qualquer maneira desde que possa ser visualizada em um browser.

O componente de interface captura e mapeia todas as ações do aluno através da sua interface. As ações mapeadas passam pelas regras de comportamento, que definem quais atitudes o agente deve tomar.

Se não for necessária nenhuma comunicação, o fluxo é passado novamente para o componente de interface. Caso seja necessária a comunicação com os outros agentes, o componente de comunicação é acionado para efetuar essa tarefa. Os resultados obtidos como respostas às funções de comunicação passam pelas regras de comportamento. As atitudes tomadas chegam até o componente de interface que as apresenta ao aluno.

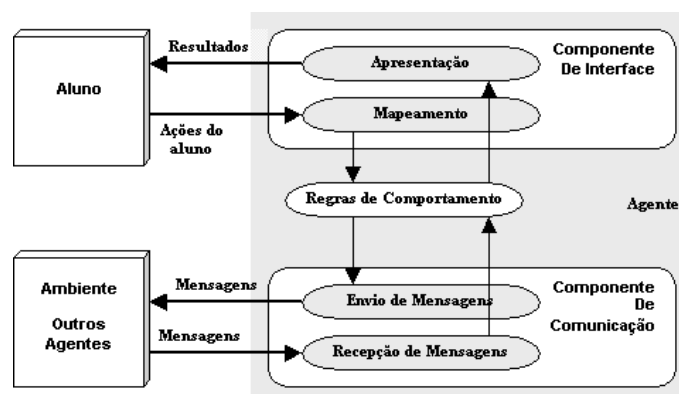


Figura 4. Desenho e processamento interno dos agentes.

2.4. Cooperação

Para que uma sociedade de agentes possa interagir a fim de atingir os seus objetivos, faz-se necessário o estabelecimento de mecanismo de cooperação e comunicação.

O modelo de cooperação deste trabalho adota o mecanismo de reconhecimento do processo executado pelos demais agentes, por meio do recebimento de mensagens que caracterizam determinados eventos e do compartilhamento de informações oriundas do banco de dados, no qual são armazenadas as informações referentes ao Modelo do Aluno.

2.5. Comunicação

Todo o processo de comunicação entre os agentes se dá por troca de mensagens FIPA-ACL através do ambiente FIPA, utilizando, se necessário, os agentes providos por esse ambiente.

A comunicação entre os agentes propostos envolve três fases (Figura 5). Na primeira fase o agente pedagógico solicita o histórico do aluno para decidir por onde começar a sessão de ensino. Na segunda

fase, as ações do aluno e decisões dos agentes geram mensagens de solicitação ou mensagens de informação. As mensagens de solicitação indicam a vontade, do aluno ou de um agente, de que alguma tarefa seja executada, por exemplo: o aluno quer avançar uma unidade, isso implica num pedido do agente pedagógico para o agente do modelo de aluno. As mensagens de informação carregam dados necessários para o funcionamento do sistema e podem ser geradas por ações do aluno ou por decisões do agente, por exemplo: o aluno chegou ao fim de uma fase do estudo, essa informação não implica num pedido do agente pedagógico para o agente do modelo de aluno e sim numa mensagem informativa que, nesse caso, dá início a uma série de ações. A terceira fase compreende a etapa de finalização de um agente pedagógico. O agente do modelo de aluno, por algum motivo, resolve finalizar um agente e exige que esse conclua imediatamente as suas tarefas. Quando um agente é finalizado ele deve atualizar o histórico do aluno com os dados que conseguiu levantar sobre este.

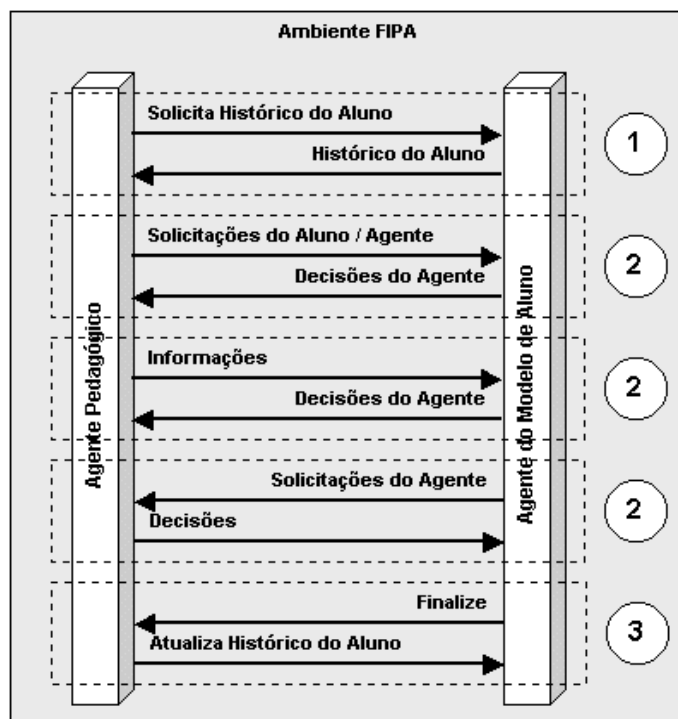


Figura 5. Comunicação

3. Protótipo

Está em fase de construção um protótipo que objetiva validar o modelo de agentes proposto neste trabalho. Inicialmente, o alvo de implementação é a criação de uma sociedade base que possa ser facilmente estendida e aperfeiçoada. Portanto, nessa fase, não foram levados em consideração a construção de agentes avançados do ponto de vista pedagógico. Da mesma forma, nenhum projeto pedagógico de curso está sendo seguido e foram construídos apenas objetos de ensino necessários para validar o modelo de comunicação dos agentes.

3.1. Tecnologias Utilizadas

A fim de facilitar a construção de agentes compatíveis com os padrões FIPA, optou-se por utilizar uma plataforma de desenvolvimento de agentes FIPA, sendo escolhida a plataforma FIPA-OS [2].

A plataforma FIPA-OS disponibiliza, além de um conjunto de classes, um ambiente de execução com todos os recursos necessários para a comunicação entre os agentes.

A plataforma FIPA-OS foi construída na linguagem Java. Portanto, todas as outras tecnologias utilizadas foram escolhidas, também, em função da integração com essa linguagem.

No componente de interface foi utilizada a combinação das tecnologias *Java Servlet* e *Java Server Pages* (JSP). Um *Servlet* é uma classe Java, assim, é possível a utilização de todos os recursos dessa linguagem

para implementar o processamento de uma requisição HTTP. A tecnologia JSP pode ser então utilizada no que ela faz de melhor: controle de *layout* da página com inclusão de conteúdo dinâmico. A combinação das tecnologias *Java Servlet* e *Java Server Pages* (JSP) forma uma ferramenta poderosa para o desenvolvimento de aplicações bem estruturadas e de fácil manutenção [1]. O *servlet* foi mantido como controlador da aplicação e o JSP como o responsável pela montagem dinâmica das páginas. Todas as requisições HTTP são enviadas para o *servlet*. Esse processa a requisição e chama uma página JSP apropriada para mostrar os resultados.

O componente de comunicação, como é o responsável pelo envio e recebimento das mensagens FIPA-ACL, foi encapsulado por um agente FIPA construído através do FIPA-OS. A construção de um agente FIPA através da plataforma FIPA-OS é feita pela extensão da classe FIPAOSAgent, que é a base para qualquer novo agente implementado no ambiente. Esta classe provê os métodos que realizam a efetiva ligação do agente implementado ao ambiente propriamente dito.

O servidor *Web* escolhido foi o Tomcat. O sistema gerenciador de banco de dados Mysql foi utilizado na implementação da base de conhecimento. A integração da base de conhecimento com os agentes foi feita através do uso de JDBC.

3.2. Agentes Implementados

Foram implementados dois agentes no protótipo. Um Agente do Modelo de Aluno e um Agente Pedagógico Apresentador de Conteúdo.

O Agente do Modelo de Aluno implementado, nessa fase inicial de implementação, é responsável pela manutenção da base de conhecimento do sistema e não faz nenhuma espécie de modelagem do aluno. O seu componente de interface compreende a tela inicial do sistema, onde são requisitados os dados de identificação dos alunos, a tela de escolha de tática, na qual o aluno escolhe como quer aprender, a tela de navegação no curso, onde o aluno pode escolher qual a parte do curso que quer estudar, e a tela de material de apoio, onde é apresentado um conjunto de links.

As funcionalidades providas pelo Agente do Modelo de Aluno são as seguintes:

- Autenticação do estudante;
- Consulta, atualização e recuperação dos dados da base de conhecimento;
- Interação com os outros agentes através do envio e recebimento de mensagens;
- Interação com o estudante através da apresentação das telas de navegação no curso (escolha de módulos e unidades), de escolha de táticas e de apresentação do material de apoio.

O agente pedagógico desenvolvido é especializado em apresentação de conteúdo através de páginas HTML. Portanto, qualquer página HTML, contendo gráficos ou outros recursos, pode ser utilizada por ele como objeto de ensino.

O agente pedagógico implementado é o mais simples que pode ser construído. Mesmo assim, ele pode servir de modelo para a implementação de agentes mais complexos e ricos do ponto de vista pedagógico. Isso porque ele apresenta todas as funcionalidades básicas de comunicação e de interface que um agente pode ter no sistema proposto.

O componente de interface do agente pedagógico compreende um menu, no qual o aluno pode interagir com o sistema, e uma parte destinada para a inclusão dinâmica das páginas que constituem os objetos de ensino. O menu de navegação possui as seguintes opções:

- Avançar: O Agente Pedagógico solicita ao Agente do Modelo de Aluno a recuperação do próximo objeto de ensino previsto para a tática em questão;
- Voltar: O Agente Pedagógico solicita ao Agente do Modelo de Aluno a recuperação do objeto de ensino anterior previsto para a tática em questão;
- Mudar Tática: O Agente Pedagógico solicita ao Agente do Modelo de Aluno que este mostre a tela de escolha de táticas para que o aluno possa escolher uma nova tática para a unidade em questão;

- Escolher Módulo/Unidade: O Agente Pedagógico solicita ao Agente do Modelo de Aluno que este mostre a tela de navegação no curso para que o aluno possa escolher uma nova unidade ou módulo;
- Material de Apoio: O Agente Pedagógico solicita ao Agente do Modelo de Aluno que este mostre a tela que contém os links para os materiais de apoio definidos pelo especialista para a unidade em questão;
- Sair: O Agente Pedagógico solicita ao Agente do Modelo de Aluno que este encerre o processo de tutoria do aluno em questão;

Não foi implementada nenhuma regra de comportamento restritiva, ou seja, todas as solicitações dos alunos são atendidas. No Agente Pedagógico implementado não foi prevista nenhuma forma de avaliação.

3.3. Comunicação

A comunicação é feita através da troca de mensagens em FIPA-ACL. É utilizada a metodologia ponto-a-ponto. Como discutido anteriormente na seção 3.5, no ambiente proposto existem mensagens de solicitação e mensagens de informação. Para as mensagens de informação foi usado o ato comunicativo "Inform". Para as mensagens de solicitação foi utilizado o protocolo de interação FIPA-Request (SC00026 - FIPA Request Interaction Protocol Specification) [3]. Os protocolos de interação definem uma seqüência lógica de mensagens, especificando os atos comunicativos que devem ser usados para cada uma delas. Eles são úteis quando alguma ação comunicativa precisa de mais poder semântico e interativo do que o simples envio de uma mensagem.

Um exemplo de mensagem utilizada no sistema pode ser visto na Figura 6. Essa mensagem é usada pelo agente pedagógico quando este quer obter o histórico de navegação do aluno dentro do curso. Com esse histórico, o agente pode identificar o último objeto de ensino com o qual o aluno trabalhou. No campo de conteúdo da mensagem são enviados os objetos Aluno e Tática. Um objeto Aluno representa o aluno com o qual o agente pedagógico está trabalhando e um objeto Tática representa a tática atual.

```
(Request
  :sender agentePedagogico
  :receiver agenteDoModeloDeAluno
  :ontology HistoricoNavegacao
  :content ( Vetor[0] -> Aluno, Vetor [1] -> Tatica )
)
```

Figura 6. Mensagem FIPA-ACL

3.4. Funcionamento do Protótipo

O processamento do sistema começa quando o aluno acessa a página inicial do sistema, que é disponibilizada pelo Agente do Modelo de Aluno. Nela são solicitadas as informações de identificação do aluno. Todas as solicitações HTTP provenientes da página inicial são desviadas para um objeto da classe AgenteMAServlet (componente de interface do Agente do Modelo de Aluno), que deve ser instanciado pelo servidor web para atender à essas solicitações. No seu procedimento de inicialização, o AgenteMAServlet instancia um objeto da classe AgenteMA (componente de comunicação do Agente do Modelo de Aluno) e um objeto da classe Dados (responsável pelos acessos à base de conhecimento).

Após a identificação ter sido realizada com sucesso, é criada, pelo Agente do Modelo de Aluno, uma sessão HTTP. Nessa sessão, é colocado um objeto da classe Aluno. Esse objeto irá funcionar como indicador de que o aluno foi identificado com sucesso e de que ele está em uma sessão válida.

A fase seguinte compreende a apresentação da tela de navegação no curso. É realizada uma consulta na base de conhecimento, onde é recuperada a estrutura do curso. Essa estrutura é apresentada ao aluno, que escolhe por onde quer começar. Após serem escolhidos o módulo e a unidade de ensino, passamos à fase de escolha de táticas. Nela, são apresentadas as táticas de ensino disponíveis para a unidade em questão.

Escolhida uma das táticas, é criado, então, um objeto da classe Tática que a represente. Esse objeto é colocado no contexto da sessão HTTP.

A próxima fase consiste na instanciação do agente pedagógico definido para a tática escolhida pelo aluno, no nosso caso é o Agente Pedagógico Apresentador de Conteúdo. Isso é feito através do desvio da requisição HTTP para a página inicial desse agente. Essa página inicial é responsável pela inicialização do *servlet* que compõem o componente de interface do Agente Pedagógico (objeto da classe AgentePServlet). Esse *servlet*, no seu procedimento de inicialização, instancia um objeto da classe AgenteP, que é o componente de comunicação do Agente Pedagógico. O componente de interface (*servlet*) tem acesso aos objetos de sessão Aluno e Tática, criados pelo Agente do Modelo de Aluno, e passa essa referência para o componente de comunicação.

Quando o componente de comunicação do Agente Pedagógico é instanciado, ele envia uma mensagem de solicitação para o Agente do Modelo de Aluno (ver Figura 5 e 6). Essa mensagem visa recuperar o histórico de navegação do aluno para que o Agente Pedagógico possa decidir com qual objeto de ensino ele deve começar a trabalhar.

A partir dessa fase, as ações do aluno através da interface passam a definir as atitudes do agente pedagógico. Essas atitudes podem gerar a necessidade de tarefas de comunicação com outros agentes, o que é feito através do componente de comunicação.

Quando o aluno sai do sistema, a sessão HTTP é invalidada, os objetos de sessão são perdidos e o sistema volta ao seu estado original.

4. Conclusões

Este artigo apresenta um modelo de sociedade de agentes para a utilização em ambientes computacionais de EAD. Esse modelo foi obtido através da remodelagem da sociedade de agentes do JADE na tentativa de adaptá-la aos padrões de interoperabilidade em sistemas multiagentes desenvolvidos pela organização FIPA.

Os principais avanços obtidos com esse modelo, em relação ao JADE, está justamente na utilização do ambiente FIPA e da linguagem FIPA-ACL. Essa utilização promoveu uma maior compatibilidade com outros sistemas multiagentes, o que favorece a integração com agentes FIPA já existentes. Além disso, existe um grande número de plataformas que visam facilitar a construção de agentes compatíveis com os padrões FIPA e que podem ser utilizadas.

A modelagem da base de conhecimento do sistema permite a criação de vários DILE sem que se tenha a necessidade de construção de novos agentes para cada um deles.

A incorporação das tarefas de interface pelos agentes permitiu um ganho em autonomia, pois a definição dessa interface é livre desde que ela possa ser visualizada em um *browser*.

A descrição e uso de objetos de ensino como geradores de material de aprendizagem contribuiu com o avanço tecnológico dos agentes pedagógicos. Agora eles podem utilizar qualquer formato de objetos de ensino. Assim, um agente pedagógico pode ter como objeto de ensino um arquivo de parâmetros, enquanto outro pode ter uma página HTML.

Cada agente é livre para implementar as suas próprias regras de comportamento. Isso permite que dois agentes pedagógicos trabalhem com a mesma técnica educacional e tenham comportamentos diferentes, o que significa maior autonomia e capacidade de adaptação ao aluno.

A partir da implementação do protótipo podemos ressaltar a facilidade de implementação de agentes FIPA provida pelo *framework* FIPA-OS. As dificuldades enfrentadas se relacionam principalmente com a ligação entre o componente de interface e o componente de comunicação de um agente. Essa construção se tornou muito complicada.

O protótipo implementado preocupou-se em disponibilizar uma estrutura base que possa servir de apoio a estudos mais aprofundados. Da maneira como foi construído, o protótipo ainda precisa de muitos melhoramentos, entre eles: Ferramentas que facilitem a construção de cursos; Geração de relatórios para acompanhamento, pelo professor responsável, do aprendizado dos alunos; e etc.

Atualmente, no campo de conteúdo das mensagens FIPA-ACL são utilizados objetos de implementação proprietária, o que não condiz com os padrões FIPA. A utilização de XML na representação desses objetos deve ser estudada a fim de sanar essa limitação.

O modelo de agentes e a sociedade também precisam de estudos mais aprofundados, principalmente no que diz respeito a: A modelagem do aluno; Construção de agentes pedagógicos avançados; Representação de conteúdo na base de conhecimento; Especificação do fluxo de mensagens entre os agentes bem como de ontologias para a comunicação entre agentes em ambientes de ensino.

Além disso, a utilização do ambiente em sala de aula se faz necessária como forma de validá-lo, porém, ela só será possível quando forem satisfeitos alguns dos itens descritos acima.

Referências

- [1] BERGSTEN, Hans. Java Server Pages. Sebastopol: O'Reilly & Associates, 2001.
- [2] FIPA-OS – FIPA Open Source. Disponível por www em <http://fipa-os.sourceforge.net> (10, dezembro 2002)
- [3] FIPA Specifications Repository. Disponível por www em <http://www.fipa.org/specs/pesspecs.tar.gz> (25, fevereiro 2003)
- [4] OLIVEIRA, F. M.; VICCARI, R. M. Are learning systems distributed or social systems. In: EUROPEAN CONFERENCE ON A.I. IN EDUCATION. Proceedings... Lisbon: [s.n], 1996.
- [5] SELF, John A Computacional Mathematics: the missing link intelligent tutoring systems research? In: COSTA, E.(ED.). New directions in intelligent tutoring systems research. Berlin: Springer-Verlag, 1992.
- [6] SILVEIRA, Ricardo A. Modelagem Orientada a Agentes Aplicada a Ambientes Inteligentes Distribuídos de Ensino: JADE - Java Agent Framework for Distance Learning Environments. Porto Alegre: CPGCC da UFRGS, 2001. Tese de Doutorado.
- [7] SILVEIRA, Ricardo Azambuja; VICARI, Rosa Maria. Developing Distributed Intelligent Learning Environment with JADE - Java Agents for distance Education Framework. Lecture Notes in Computer Science, França, n. 2363, 2002.
- [8] SILVEIRA, Ricardo Azambuja; VICARI, Rosa. Improving interactivity in e-learning systems with Multi-Agent architecture. Lecture Notes in Computer Science, 2002.
- [9] SPODICK, Edward F. The evolution of distance learning. Hong Kong: University of Science & Technology Library. (1995) Disponível por WWW em <http://sqzm14.ust.hk/distance> (05, março, 2003)