

Implementando o Agente de Base de Domínio do Sistema Tutor Inteligente PAT2Math

Gabriel Mello, Talvany Carlotto, Geiseane Rubi,
Henrique M. Seffrin, Patrícia A. Jaques

Programa Interdisciplinar de Pós-Graduação em Computação Aplicada (PIPICA) –
Universidade do Vale do Rio dos Sinos (UNISINOS)
CEP 93.022-000 – São Leopoldo – RS – Brazil

gabrielchites@yahoo.com.br, {talvanynet, geiserubi}@gmail.com,
hseffrin@terra.com.br, pjaques@unisinis.br

Resumo. *A matemática é considerada uma área complexa por muitos alunos, que em geral consideram seus conteúdos muito abstratos, não compreendendo sua aplicação ou não conseguindo relacioná-los com seu dia-a-dia. Mesmo assim, poucos são os ambientes de aprendizagem capaz de auxiliá-los. Faz-se necessário um sistema que leve em consideração as emoções do educando, visto que grande parte das dificuldades enfrentadas por estes se deve a bloqueios afetivos, e que também seja apto a identificar seus equívocos e que possa ser acessado de qualquer lugar. Dentro deste enfoque, este artigo trata do projeto e implementação de um sistema tutor inteligente com uma arquitetura multiagente voltado ao ensino de equações algébricas de 1º e 2º grau. Mais especificamente, o presente trabalho foca no Agente de Base de Domínio, responsável pelo armazenamento e manipulação do conteúdo de matemática do tutor. Isto resulta em um modelo de representação do conhecimento, capaz de oferecer um serviço transparente aos módulos que o requerem, inserido em uma arquitetura composta por módulos independentes e colaborativos.*

Palavras-chaves: *sistema tutor inteligente; sistema multiagente, agentes pedagógicos, aprendizagem de matemática.*

Implementing the Domain Database Agent for the Intelligent Tutoring System PAT2Math

Abstract. *Math is considered by many students a complex subject, since they consider its contents to be too abstract and do not comprehend their use or can not relate these subjects to their daily routine. However, there are only few learning environments that are able to help them. A system that considers the student's emotions is necessary, since a lot of the difficulties faced by them are caused by an affective blockage. This system has to be able to identify the students' misunderstandings and to be accessible from anywhere. Within this approach, this paper describes the project and implementation of an Intelligent Tutoring System with a multiagent architecture for teaching algebraic equations. More specifically, the present work focuses on the Domain Base Agent, responsible for storing and handling the math content of the tutor. This results in a representation model of the knowledge that offers a transparent service to the modules that require it, and which is inserted in an architecture composed by independent and collaborative modules.*

Keywords: *intelligent tutoring system; multiagent systems, pedagogical agents, math learning.*

1. Introdução

O surgimento dos Sistemas Tutores Inteligentes (STI) como alternativa aos *Computed Aided Instruction* (CAI), meros viradores de páginas eletrônicos dos anos 50, vem trazendo uma nova área para diversas experimentações por parte da Inteligência Artificial (IA) no campo da Informática na Educação.

A aplicação de Agentes Pedagógicos Animados em ambientes de STI ocasiona uma nova perspectiva à maneira como os conteúdos são apresentados aos alunos. Por trás dessa apresentação, um planejamento das táticas afetivas e pedagógicas se faz necessário para a adaptação aos diferentes tipos de aprendizado dos usuários.

Este dinamismo pedagógico implica também em conteúdos adaptados especialmente ao formato da apresentação adotado e, através da identificação das necessidades do aluno (seu perfil, conhecimento e fatores emocionais), selecionar em tempo real quais os tópicos que serão ensinados ao aluno naquele momento. Desta maneira, o conhecimento deixa de ser algo estático e torna-se uma entidade dinâmica, necessitando de uma organização diferenciada e de uma manipulação por um agente dedicado à tarefa de obter os conteúdos mais adequados às necessidades do aluno.

Dentro deste enfoque, este trabalho propõe o projeto de um agente responsável pelo armazenamento, organização e manipulação do conteúdo de matemática utilizado pelo tutor. Este agente deve suprir a necessidade do tutor de aquisição de conteúdos personalizados, compostos por fragmentos de documentos guardados na base do sistema. Através de uma arquitetura multiagente, propõe-se uma modularização do STI, em que cada componente é capaz de atuar com independência e possuem capacidade de colaborar de forma ativa.

Esse artigo está organizado como segue. Na seção 2, defini-se o que são STI, discute-se sua arquitetura tradicional e como sistemas multiagentes podem beneficiar esta arquitetura. Na seção 3 encontra-se a motivação para a escolha da matemática como o domínio do tutor, no qual é discutido em detalhes na seção 4, apresentando sua arquitetura, e o porquê do uso de sistemas multiagentes. A seção seguinte trata do objeto principal do projeto apresentado neste artigo: O Agente de Base de Domínio. Na seção 6 mostram-se os resultados de entrevistas com especialistas em matemática sobre o sistema. Por fim, na seção 7 há uma comparação da proposta com trabalhos existentes e na seção 8 apresentam-se as conclusões e os trabalhos a serem realizados no âmbito do projeto PAT2Math.

2. Sistemas Tutores Inteligentes

O objetivo da pesquisa em IA é definido como “construir programas computacionais que apresentam comportamentos inteligentes” (Feigenbaum e Feldman, 1963). Atualmente, há evidências (Viccari e Giraffa, 2003) de que softwares educacionais que não se utilizem de técnicas de IA para se adaptar ao aluno não atingem completamente seu objetivo.

STI são programas que usam técnicas de IA para auxiliar o aluno na sua aprendizagem. Segundo Viccari (1990), os STI são sistemas capazes de, através da interação com o usuário, atualizar sua própria base de conhecimento sobre o aluno (modelo de aluno). Dessa forma, são capazes de reconhecer o estado atual do aluno, diagnosticar lacunas em seu conhecimento e, com o desenrolar do diálogo, aprender, adaptar e aplicar diferentes tipos de estratégias de aprendizagem.

2.1. Arquiteturas de Sistemas Tutores

A arquitetura tradicional de STI possui basicamente os seguintes componentes: modelo do domínio, tutor, modelo do aluno e interface (Psootka e Mutter, 1988). O modelo de

domínio contém o conhecimento sobre o conteúdo pedagógico que será usado pelo sistema na aprendizagem do aluno. A representação pode ser declarativa (para domínio de natureza descritiva e teórica) ou procedural (domínio orientado a uma tarefa). Vários modelos de representação de conhecimento podem ser usados, e a escolha do mais apropriado depende do escopo do projeto.

O Tutor é responsável por organizar um plano de aula para o aluno, buscando formas de apresentar o material instrucional. Esse módulo é responsável por escolher o conteúdo a ser apresentado, considerando a estratégia adotada, o conhecimento sobre o perfil e competência do aluno presente no modelo do aluno e os conteúdos pedagógicos que constam na base do domínio. Além disso, o Tutor é incumbido de acompanhar o aprendizado do aluno, fornecendo *feedback* enquanto o mesmo interage com o ambiente e providenciando medidas corretivas em situações que percebe que o aluno está equivocando em relação ao conhecimento do domínio do sistema

O modelo do aluno contém as informações que o sistema possui acerca do aluno, incluindo os conhecimentos que este já possui e as habilidades adquiridas num dado momento. A partir desse modelo e do conteúdo representado na base do domínio, o módulo tutor será capaz de inferir a melhor estratégia de ação a ser utilizada para cada aluno.

A Interface é a camada responsável pela interação do sistema com o aluno. Esta deve fornecer uma apresentação clara do conhecimento, no formato especificado pelo Módulo Tutor, e deve facilitar o processo de execução de atividades, podendo fazer uso também de Agentes Pedagógicos Animados (Johnson *et al.*, 2003).

Atualmente, a arquitetura de sistemas tutores vem sendo concebida como uma sociedade de agentes (sistemas multiagentes). De acordo com Shoam (1997), um agente de software é uma entidade que funciona autônoma (sem a constante intervenção de um humano) e continuamente em um ambiente particular sempre habitado por outros agentes e processos.

De acordo com Giraffa e Vicari (1998), as arquiteturas baseadas nessa abordagem são variações da tradicional e funcional arquitetura de um STI (base de conhecimento, modelo de aluno, estratégias afetivas), em que um ou mais agentes implementam uma função do sistema. O controle é distribuído entre os agentes, entretanto o usuário vê o sistema como um único, enquanto, internamente, ele é composto por uma sociedade de agentes. Neste contexto, cada agente possui uma função específica e eles se comunicam entre si a fim de atingir o objetivo global que é ensinar ao aluno.

3. Ensino de Matemática

O ensino da matemática é considerado complexo por muitos alunos em diversas áreas, pois apresentam um grau de dificuldade maior, ou, em muitos casos, são considerados abstratos demais para os alunos que não compreendem sua aplicação ou não conseguem relacioná-la com seu dia-a-dia. Dentro destes conteúdos incluímos o ensino/aprendizagem das equações algébricas. Pensando nisto, pesquisadores e professores buscam alternativas para aproximar mais determinados conteúdos da realidade e criar novas alternativas didáticas para o ensino/aprendizagem dos mesmos.

Inicialmente, trazemos a abordagem intuitiva com a metáfora da Balança, Princípio de Euclides, aplicável somente para Números Naturais, pois não há como equilibrar pesos negativos. O objetivo é que os alunos entendam o real significado do sinal de igual ($=$), cujo o equilíbrio é essencial, pois cada prato da balança representa um dos membros da equação. Com isto, compreendem a equivalência de membros de uma equação e a manipulação de incógnitas decorrente para este balanceamento. Desta

forma, o entendimento dos alunos referente ao conceito de equação é importante, fazendo-se necessário uma abordagem concreta para que “vejam” a incógnita como um valor numérico que trará equilíbrio para ambos os membros da equação.

Posterior a introdução com o uso da balança, pode-se incluir os números Inteiros, pois a dificuldade encontrada se deve ao fato de compreenderem o que deverá ser feito com a equação. Ao se depararem com uma equação, os alunos desejam, geralmente, descobrir rapidamente o “x” (incógnita normalmente usada). Porém, quando chegam ao resultado final, ou seja, quando têm acesso ao valor numérico do “x”, não sabem o que fazer com este. Isso acontece, em especial, quando se tem situações-problema (por exemplo, “Uma quantidade somada com a sua quarta parte é igual a 15. Qual é essa quantidade?”), na qual a incógnita representa uma aplicação do conteúdo na realidade. Neste caso, simplesmente calcular o valor da incógnita não é o suficiente, precisam responder adequadamente ao problema.

Para ajudar o aluno a superar as dificuldades encontradas na aprendizagem de resolução de equações algébricas, é importante que os ambientes de aprendizagem firmem bem essas propriedades da aritmética, pré-requisitos para a aprendizagem, assim como devem também, inicialmente, trabalhar com o concreto, com situações do dia-a-dia dos alunos. Para a introdução do conteúdo é interessante a metáfora da Balança. Após, o sistema deve ensinar equações algébricas buscando mostrar sua aplicação, fazendo com que compreendam as generalizações do conteúdo. Na resolução de equações, o sistema deve ser capaz de também resolver uma equação a fim de identificar se os passos realizados pelos alunos estão corretos, se são os mais apropriados e quais as suas concepções incorretas sobre resolução de equações. Para prever todas essas funcionalidades, o ambiente de aprendizagem deve ser mais do que um simples virador de páginas Web. Ele deve possuir uma arquitetura modular e ser dotado de técnicas de IA a fim de oferecer um suporte inteligente e individualizado, o que se caracteriza um sistema tutor inteligente.

4. PAT2Math

O presente trabalho trata do projeto e implementação da arquitetura da Base de Domínio do STI do projeto PAT2Math.

Este STI, chamado de PAT2Math, tem como objetivo o ensino de álgebra elementar a alunos do ensino fundamental em um ambiente capaz de considerar as emoções do aluno. O ensino de álgebra foi escolhido, pois se trata de um domínio no qual é temido por grande parte dos alunos e por facilitar a elaboração de um ambiente com situações de aprendizagem controladas, facilitando a inferência das emoções do aluno.

Além disso, este STI faz uso de um Agente Pedagógico Animado que aplicará táticas pedagógicas para motivar o aluno e engajá-lo em sua interação com o ambiente. Através da inferência das emoções do aluno, o Agente será capaz de assumir personalidades, visando um aprendizado mais personalizado e próximo de uma interação com um tutor real.

PAT2Math está sendo desenvolvido para ser um ambiente de aprendizagem com interface Web, permitindo que o sistema seja utilizado a partir de qualquer computador conectado à internet, disponibilizando o acesso aos alunos, onde quer que estejam.

O enfoque deste artigo é no componente de base de domínio do STI, mais especificamente, no componente declarativo desta base. Portanto nas próximas seções serão introduzidos assuntos necessários para a justificação da arquitetura do agente de domínio.

A seção **Erro! A origem da referência não foi encontrada.** apresenta as razões

que levaram à escolha de uma abordagem multiagente para o tutor proposto, e sua implementação no PAT2Math na seção **Erro! A origem da referência não foi encontrada.** A partir da seção **Erro! A origem da referência não foi encontrada.** discutiremos o agente do domínio e as decisões de soluções tecnológicas para este módulo. Através dessa seção passaremos também por uma apresentação do formato Open Mathematical Document (OMDoc), figura central para a representação do conhecimento do domínio de matemática, sendo este o cerne do desenvolvimento do agente proposto.

4.1. Por que uma Arquitetura Multiagente?

Um sistema baseado na Web deve ser projetado levando em consideração a separação das camadas entre cliente e servidor inerente da arquitetura (Deitel, 2003). Indo além dessa premissa, é objetivo deste trabalho desenvolver os componentes dos módulos do STI de forma que possam ser utilizados de maneira independentes, tanto em sua implementação quanto à sua localização na rede.

Para a construção deste sistema distribuído e autônomo, foi escolhida a arquitetura multiagente. Esta arquitetura permite a construção de entidades distribuídas capazes de cooperarem entre si em um ambiente heterogêneo e coordenarem suas ações em busca de um objetivo.

Excluído: ra

Explorando a natureza distribuída dos Sistemas Multiagentes, pode-se implementar um Sistema Tutor cujos os componentes dos módulos interajam através da rede. Incorporando agentes para a comunicação entre os módulos, estaremos abstraindo complexidades e tecnologias subjacentes de cada componente e tornando-os capazes de colaborar para a realização de um objetivo maior. Utilizando a arquitetura modular do STI aliada a um Sistema Multiagente, podemos reduzir o acoplamento entre os módulos, executá-los em computadores separados, abstraindo complexidades de rede inerentes de sistemas distribuídos. Complementando esta independência, a utilização de agentes permite que os módulos existam sem o conhecimento uns dos outros. Através da utilização de serviços como páginas amarelas, agentes prestadores de serviços inscrevem-se com um Agente Facilitador (Directory Facilitator), e agentes que requisitam serviços utilizam estas páginas amarelas para encontrar outros agentes que possam cumprir suas solicitações (Caire, 2007).

4.2. A arquitetura Multiagente do PAT2Math

A fim de atingir os objetivos citados acima, o STI proposto foi concebido segundo uma abordagem multiagente, no qual cada módulo da arquitetura tradicional do STI (interface, tutor, modelo de aluno e base de domínio) é implementado como um agente neste sistema, dando origem a: Agente de Interface (AGI), Agente Tutor (AGT), Agente Modelo de Aluno (AGMA) e Agente Base do Domínio (AGBD). Além disso, o sistema possui também um agente pedagógico animado, chamado Pat, que será responsável por considerar os aspectos de afetividade na interação entre tutor e aluno. Para isso, outros trabalhos de pesquisa desenvolvidos pelo grupo no sentido de inferir e expressar emoções (Jaques, 2008; Jaques *et al.*, 2008; Oliveira and Jaques, 2008) serão integrados ao STI proposto como funcionalidades do agente pedagógico animado Pat. Na figura 1 pode ser visualizada a arquitetura multiagente proposta para o STI PAT2Math.

A dinâmica da interação entre os agentes se dá da seguinte maneira: O AGI é responsável por acompanhar as ações do aluno na interface e mostrar exercícios e conteúdos pedagógicos definidos pelo AGT. A cada nova ação do aluno no sistema, o AGI envia a informação correspondente ao AGT que a repassa ao AGMA para atualização do modelo de aluno. Se a ação realizada consiste em um passo da resolução de uma equação algébrica, o AGMA verificará se essa resolução está correta ou quais as

concepções errôneas que estão envolvidas na decisão do aluno e enviar essa informação ao AGT. O AGT vai verificar a necessidade de adaptar o plano de aula do aluno baseada nessas novas informações e, caso necessário, um novo plano será gerado dinamicamente. Para a geração dos planos, duas principais informações são consideradas pelo AGT: conhecimento e perfil do aluno no Modelo de Aluno e base de estratégias pedagógicas, além do conteúdo disponível na Base de Domínio. Se a próxima tática a ser aplicada pelo AGT consiste na apresentação de um material pedagógico ou exercício, o mesmo o solicita ao AGBD e envia esse material ao AGI para apresentação.

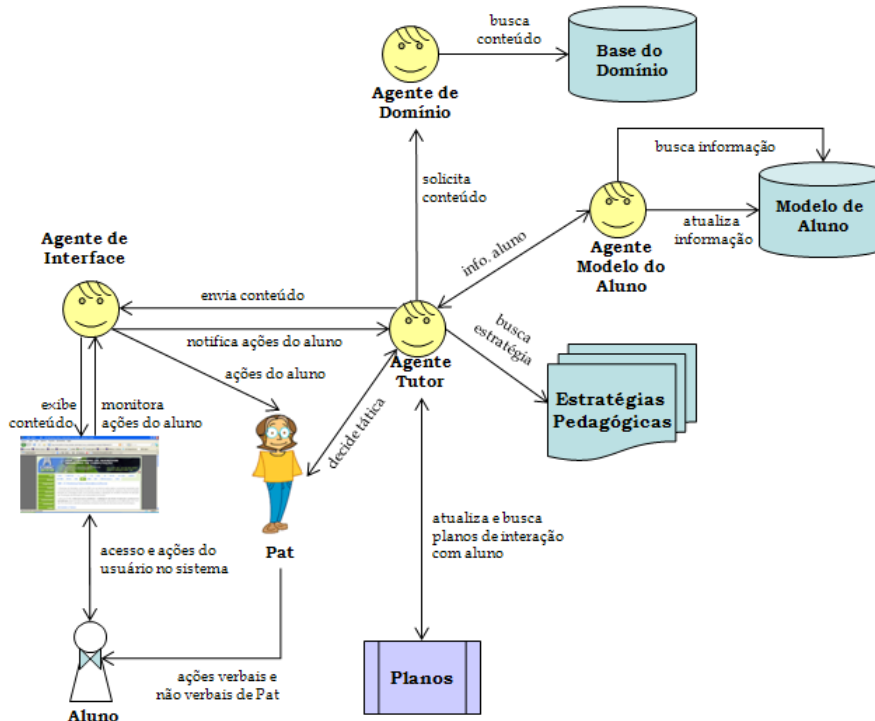


Figura 1 - Visão geral da arquitetura dos agentes do STI PAT2Math

Concorrentemente, as mesmas informações sobre ações do aluno repassadas pelo AGI para o AGT, também são enviadas ao agente pedagógico animado emocional Pat. Baseada nessa informação, Pat decide a melhor tática pedagógica afetiva a aplicar e interage com o AGT para verificar se essa tática entra em conflito com a tática de competência decidida pelo último. Em caso positivo, os agentes negociam uma nova estratégia. Caso contrário, Pat apresenta também sua tática pedagógica afetiva. Esse é o ciclo de vida dinâmico de funcionamento do tutor.

Embora as interações pareçam muito centradas no AGT, essa arquitetura foi escolhida, pois ela obedece também ao padrão Model-View-Controller (MVC), que se configura em separar o que é interface (no sistema proposto representada pelo AGI), da lógica de negócio do sistema (representada pelo AGT), assim como da modelagem de informações (AGBD, AGMA). É importante observar que essa modelagem não acarretará sobrecarga de fluxo de rede no AGT, visto que a comunicação entre os agentes do sistema se dá no servidor, ficando a maior latência de tempo entre a comunicação AGI no servidor e *browser* na máquina local do usuário, já que PAT2Math será um STI Web.

Além disso, outra vantagem da modelagem do STI como um sistema multiagente é a facilidade de integração de novos módulos a ele. Por exemplo, já é prevista a

inserção de um novo agente responsável por verificar se a equação proposta pelo aluno está correta em exercícios que envolvem enunciados de problemas. Para tanto, é necessário apenas que o novo agente conheça a linguagem, protocolos e ontologia dos agentes do sistema.

5. Agente Base de Domínio

No PAT2Math, o armazenamento e a organização do conhecimento declarativo é de responsabilidade do Agente Base de Domínio (AGBD). O papel deste componente é fornecer o conhecimento do domínio matemático àqueles que o requisitarem seguindo um protocolo de comunicação específico, informando critérios de busca pelo conteúdo.

O conhecimento do componente declarativo da Base de Domínio é formado por conteúdos de álgebra e por meta-informações sobre este conteúdo utilizadas para o controle deste. O domínio de álgebra é formado por fragmentos de conhecimento matemático, aqui chamados de tópicos. O AGBD é incumbido de manter o conteúdo organizado de acordo com os atributos dos conteúdos do domínio, tais como: Assunto, dificuldade, tipo de conteúdo, etc. Estes tópicos são armazenados em uma linguagem de marcação para matemática chamada OMDoc.

5.1. Armazenamento do Conhecimento em XML

OMDoc é um formato para representar o conteúdo de documentos matemáticos. Diferentemente de formatos como OpenMath e MathML que representam apenas fórmulas matemáticas, OMDoc é um formato capaz de representar toda gama de conteúdo matemático escrito, tais como explicações, exemplos, teorias e exercícios. Além disso, podemos inserir a semântica do conteúdo matemático e meta-informações sobre ele (Kohlhase, 2006).

O conhecimento matemático declarativo contido no PAT2Math é formado por tópicos, uma pequena unidade de conhecimento a ser apresentado ao aluno. Cada um destes é formado por diversos elementos OMDoc, nos quais podem ser manipulados e combinados pelo agente para a criação de diferentes tópicos. O conjunto de tópicos sobre determinado assunto constituem a aula, na qual é planejada pelo agente tutor.

Documentos OMDoc são aplicações do formato XML (Extensible Markup Language). A partir da necessidade da Base de Domínio armazenar um grande número de fragmentos OMDoc, por consequência XML, foi adotado o uso de um banco de dados XML nativo. A escolha de um banco de dados XML nativo foi motivada pelo desempenho alcançado ao armazenar os documentos em seus formato nativo e pela flexibilidade oferecida para organização, consulta e manipulação de fragmentos dos documentos através da linguagem de consulta do formato XML XQuery, permitindo a composição de novos documentos dinamicamente.

5.2. Organização do Conhecimento: Meta-Informação

Para o completo controle do conhecimento, entretanto é necessário estabelecer uma lógica entre os diversos fragmentos de OMDoc, organizá-los como tópicos, definir atributos que vão além do conteúdo matemático e que dizem respeito apenas ao contexto do ensino de álgebra, tal como atribuir um relacionamento entre os conteúdos, informando quais são os pré-requisitos de um determinado tópico.

Para a estruturação destas informações sobre os tópicos matemáticos sob uma visão de mais alto nível para o STI, utilizamos uma base de dados entidade-relacionamento. A Figura 2 exhibe algumas das informações utilizadas pelo AGBD para a organização de cada tópico: XML_ID significa a identificação do conteúdo na base XML, “pré-requisitos” faz referência para uma lista de tópicos que são bases para o tópico atual, e “nível de dificuldade” é um valor subjetivo, indicando o quão árduo é o

entendimento / resolução do tópico em relação aos outros.

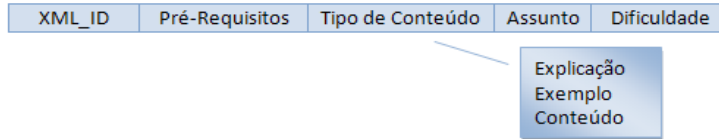


Figura 2 - Informações para Estrutura de Controle do AGBD

5.3. Comunicação do Agente Base de Domínio

Neste contexto, o papel do AGBD é dar suporte para consultas em alto nível para busca de tópicos sobre determinados assuntos, procurando dentre os vários tópicos, qual o conteúdo mais adequado, considerando os outros parâmetros informados. O agente busca o conteúdo escolhido nos fragmentos de documentos OMDoc armazenados no banco de dados XML nativo. A ideia é que, para um mesmo conteúdo, este esteja presente na base de dados em formatos diferentes (explicações, animações, vídeos), com diversos enfoques e perspectivas (por exemplo, há diversas explicações diferentes do mesmo assunto), a fim de que o tutor possa ter um comportamento diferenciado para cada aluno, em cada seção deste.

Por meio da arquitetura multiagente, é possível fazer com que qualquer agente que conheça o protocolo do AGBD tenha acesso ao conhecimento contido neste componente. Classes para controle e manipulação de conteúdo, base de dados Entidade-Relacionamento (ER) para armazenamento de informações de controle dos documentos OMDoc guardados em banco XML e outros pormenores são indiferentes aos solicitantes do serviço do AGBD, pois este entrega aos seus clientes apenas texto, respondendo as solicitações de outros agentes seguindo o protocolo especificado.

Além de abstrair todas as tecnologias e complexidades envolvidas na busca pelo conteúdo mais adequado, o agente deve prover um serviço de colaboração, utilizando o serviço de Páginas Amarelas, onde qualquer agente que conheça o protocolo do AGBD poderá requisitar tópicos de álgebra através da linguagem de comunicação de agentes (Agent Communication Language) FIPA-ACL (FIPA, 2009). Por exemplo:

topic:subject=*Equações do Primeiro Grau*; type=*Explanation* ; difficulty=2;

Através destes três parâmetros na requisição de um tópico (utilizando performativa *request - requisição*), o AGBD é capaz de passar a chamada às classes de controle que por sua vez são responsáveis pela seleção e composição do documento OMDoc que melhor representa o tópico pedido. De posse do documento, o agente de domínio o retorna em formato de texto (*strings*) para os agentes clientes, seguindo o protocolo da solicitação em linguagem ACL.

6. Resultados da Entrevista com Especialistas

Entrevistas qualitativas foram realizadas com professores que trabalham com o ensino de álgebra elementar, objetivando encontrar funcionalidades que o tutor deveria possuir, ou aperfeiçoar se já existente, dado que os professores, conhecendo as dificuldades encontradas na aprendizagem em sala de aula, poderiam contribuir com sugestões didáticas e pedagógicas para o sistema. As entrevistas foram feitas com cinco professores que trabalham em escolas de ensino fundamental e médio de Porto Alegre e contou com o apoio de uma pedagoga em matemática. Os professores entrevistados tiveram a oportunidade de interagir com um protótipo simples do sistema e contribuir com sugestões de melhoria. O protótipo consistia em uma aplicação *stand-alone* que mostrava passo-a-passo a resolução de equações, um dos tópicos mais importantes da álgebra, com uma explicação da operação realizada. Esse protótipo simples visava auxiliar os professores a visualizar as futuras funcionalidades prevista para o sistema.

Quatro dos cinco professores consideram errado usar a linguagem simplificada que muitas vezes é usada na apresentação do conteúdo de álgebra, como "passar para o outro lado da equação". Eles se mostraram favoráveis ao uso dos termos matematicamente corretos, para que o aluno não aprenda o conteúdo de forma errada. Os professores também salientaram a importância da comunicação constante do tutor com o aluno, para, por exemplo, induzir o aluno a encontrar erros cometidos. Ainda, segundo os professores, para que o aluno se sinta a vontade usando o sistema, ele deve ter liberdade, tanto para "testar" o tutor, explorando equações de sua autoria, quanto para fazer o sistema se adaptar a suas preferências na resolução dos exercícios. Além destas, outras sugestões em relação a apresentação das equações e mecanismos de interação do tutor foram também observados.

7. Trabalhos Relacionados

O trabalho proposto difere de outros STI Web, tal como Algebrain (Alpert, 1999) por buscar uma melhoria na usabilidade da interface, enriquecendo a experiência do usuário através de páginas dinâmicas e execução de código em lado cliente. Além disso, PAT2Math será o primeiro STI capaz de considerar as emoções do usuário através de dispositivos próprios, capacitando o tutor a agir ativamente ao estado do aluno, incentivando-o através de fala, gestos, expressões faciais e corporais, facilitando a comunicação tutor-aluno. Além disso, embora o Algebrain também utilize um sistema especialista para auxiliar alunos na resolução de equações, ele não é capaz de detectar as concepções errôneas dos mesmos.

O Live Math (<http://www.livemath.com>) foi implementado para auxiliar o aluno na construção e resolução de cálculos matemáticos em geral. Ele trabalha com uma concepção pedagógica behaviorista, retornando uma recomendação direta ao último passo do usuário, ou seja, não considera o histórico de ações do aluno, diferentemente do STI proposto que segue uma abordagem construtivista.

Em relação a outros tutores que seguem os princípios gerais de arquiteturas de STI (Anderson *et al.*, 1992), existem vantagens que vão além da questão de um sistema Web capaz de considerar emoções: Inovações na área de STI tais como arquitetura multiagente, possibilitando independência entre módulos e base de conhecimento dinâmica, proporcionando uma experiência única para cada aluno, são maneiras de trazer avanços em relação aos sistemas tutores existentes.

8. Conclusão e Trabalhos Futuros

Ao utilizar a arquitetura proposta, tem-se como resultado um serviço transparente de consulta a dados utilizando comunicação entre agentes, provendo facilidades que vão do encapsulamento de diversas tecnologias, incluindo a manipulação de fragmentos de vários documentos OMDoc, até a colaboração entre agentes, através de publicações e requisições de serviços providos pelo *framework* de desenvolvimento de agentes JADE.

Através da implementação da base de domínio do sistema como um agente, obtém-se um módulo autônomo e abstraído do resto do sistema, facilitando, por exemplo, a inserção de novos componentes ao STI. Entretanto o maior ganho na utilização de tal arquitetura, é a possibilidade de colaboração entre os agentes, tornando cada componente um módulo ativo e parte de um grupo com um objetivo em comum: Ensinar comportando-se de maneira especial para cada aluno, resultando em uma experiência única e pessoal para o usuário.

Agradecimentos

O grupo agradece a Unisinos, CNPq e FAPERGS pelo apoio através das bolsas de Iniciação Científica e a Capes pelo apoio dado através do projeto Capes/Cofecub.



Referências Bibliográficas

- ALPERT, S.; SINGLEY, M.; FAIRWEATHER P. Deploying Intelligent Tutors on the Web: An Architecture and an Example. In IJAIED, 10(2), pp. 183-197, 1999.
- CAIRE, G. JADE Programming Tutorial for Beginners. TILAB, CSELT, 2007.
- DEITEL, H. M. Java, como programar. Ed. Bookman, 2003.
- NORMAN, D. Invisible Computer: Why Good Products Can Fail, the Personal Computer Is So Complex and Information Appliances Are the Solution. MIT Press, 1999.
- FEIGENBAUM, A.; FELDMAN, J. Computers and Thought. NY: McGraw-Hill, 1963.
- FIPA Foundation for Intelligent Physical Agents, FIPA ACL Message Structure Spec., 2009. Disponível em: <<http://www.fipa.org/repository/standardspecs.html>> Acesso em 27 jul. 2009.
- GIRAFFA, L. M. ; VICCARI, R. M. . Intelligent Tutoring Systems Modelled Through Agents Techniques. Revista de Ciência, Educação e Cultura, 1998
- JAQUES, P.; JAQUES, F.; LEHMANN, M.; Avaliando a efetividade de um agente pedagógico animado emocional. In: SBIE, 2008, Fortaleza.
- JAQUES, P. Avaliando um Modelo Afetivo de Aluno baseado em uma Abordagem Cognitiva. In: SBIE, 2008, Fortaleza.
- OLIVEIRA, E.; JAQUES, P. Inferindo as emoções do usuário pela face através de um sistema psicológico de codificação facial. In: Simpósio Brasileiro Sobre Fatores Humanos Em Sistemas Computacionais, 2008, Porto Alegre.
- JOHNSON, W.; SHAW, E.; MARSHALL, A.; LABORE, C. Evolution of user interaction: the case of agent adele. In *Proceedings of IUI*. ACM, New York, 2003.
- KOHLHASE, M. An Open Markup Format for Mathematical Documents (Version 1.2). In: (Ed.). Lecture Notes in Artificial Intelligence. Heidelberg: Springer-Verlag, 2006.
- PSOTKA, J.; MASSEY, L.; MUTTER, A. Intelligent Tutoring Systems: Lessons Learned, 1988.
- SHOAM, Y. An Overview of Agent-Oriented Programming. In: J. Bradshaw (Ed.). Software Agents. Menlo Park: AAAI Press/The IT Press, 1997. p.271-290
- VICCARI, R. M. Um Tutor Inteligente para a programação em Lógica – Idealização, Projeto e Desenvolvimento. Universidade de Coimbra (Tese de Doutorado), 1990.
- VICCARI, R. M.; GIRAFFA, L. M. Fundamentos de Sistemas Tutores Inteligentes. In: D. Barone (Ed.). Sociedades Artificiais. São Paulo: Bookman, 2003. p.155-208.