

## Agentificando um Ambiente de Aprendizagem de Matemática

João Carlos de C. e S. Ribeiro<sup>2</sup>, Lucia M. M. Giraffa<sup>1</sup>, Úrsula A. L. F. Ribeiro<sup>2</sup>

<sup>1</sup>Faculdade de Informática (FACIN) – Mestrado em Educação em Ciência e Matemática  
PUCRS – Campus Central  
Av. Ipiranga, 6681, prédio 30 bloco IV – CEP 90619-900 – Porto Alegre – RS – Brasil.

<sup>2</sup>Faculdade de Administração Contabilidade e Informática (FACI)  
PUCRS – Campus Uruguaiana  
BR 472 Km 7 Cx. Postal 249 – CEP 97500-970 – Uruguaiana – RS – Brasil.

giraffa@inf.pucrs.br, {jrubeiro, ursula}@pucrs.br

**Abstract.** *This paper describes the research developed to model and to implement the educational software named “Matematiquinha”. The software was originally developed using a tutorial approach. Due its restrictions a new version was built using multiagent systems architecture. This second version reuse the original database exercises, and include new modules and agent in order to enlarge its pedagogical application.*

**Palavras-chave:** Educação de Matemática, Sistemas multiagentes, Software Educacional, Agentificação.

**Resumo** *Este artigo descreve a experiência desenvolvida com o objetivo de identificar o conjunto de premissas necessárias para agentificação de um software educacional, partindo-se da reutilização do código fonte de um programa educacional pré-existente. Foram reestruturados os requisitos, as características necessárias para modelagem do programa num contexto de sistemas multiagentes, bem como os pré-requisitos referentes à equipe interdisciplinar necessária para conduzir um projeto deste porte. O software educacional utilizado no projeto foi concebido na modalidade tutorial. Ou seja, já existia um domínio e um conjunto de regras de comportamento do tutor modeladas de acordo com objetivos pedagógicos do professor.*

**Key words:** Mathematic Education, Multi-agent Systems, Educational Software, Wrapping.

## 1. Introdução

Existem muitos programas educacionais que necessitam uma remodelação no seu código a fim de poder suportar mudanças tecnológicas e pedagógicas inerentes da sua utilização no contexto de uma escola, onde obteve sucesso. Isto é, existem sistemas que estão incorporados ao cotidiano pedagógico do trabalho de professores e alunos e, que em virtude da evolução tecnológica e a maturidade dos seus usuários necessitam ser redimensionados, sem, no entanto, perder sua característica inicial. Fazer esta adaptação no código, tanto em nível pedagógico como tecnológico, é uma tarefa complexa. A agentificação dos sistemas apresenta uma resposta para resolver esta questão. Ela consiste no reuso e adaptação de sistemas já existentes e uma ampliação do seu potencial pedagógico. No que concerne aos sistemas educacionais isto traz vantagens significativas, pois permite a personalização das atividades, adaptação em função das necessidades do usuário (aluno) e sua utilização fora do escopo de uma aplicação local. Os conceitos utilizados neste projeto basearam-se nos trabalhos de Carvel, Serrano e Oca [1], Chiang [2], Landauer e Bellman [5], Landauer Bellman e Gilliam [4] e Vicari e Giraffa [8]. Estes autores exploraram a questão da agentificação aplicada a ambientes educacionais. Sob o ponto de vista prático, a agentificação cria um conjunto de interfaces personalizadas que permitem utilizar recursos em um sistema heterogêneo. Entretanto, agentificar não é simplesmente agregar uma interface para um recurso. Agentificação permite a criação de uma interface para a “utilização” de um determinado recurso. Logo, o enfoque está centrado na forma pela qual o recurso é utilizado. Ou seja, o estudo das maneiras que os recursos podem ser utilizados. Existirão agentificações específicas e diferenciadas em virtude do que se quer encapsular. Combinações de recursos que frequentemente trabalham juntos podem ter uma única agentificação. Entretanto, podem ser necessárias várias agentificações separadas para se utilizar um único recurso. Quando, por exemplo, houver necessidade de se representar diferentes caminhos.

Este projeto utilizou como elemento base o programa educacional (software legado) denominado *Matematiquinha*, desenvolvido por Ribeiro e Ribeiro [6]. O *Matematiquinha* tem a finalidade de auxiliar o professor no ensino do sistema decimal e das operações de adição e subtração para crianças das séries iniciais. O sistema foi projetado tendo como base o jogo educacional conhecido como *Material Dourado*. O *Material Dourado* é composto por um grande número de cubos e barras confeccionados na maioria das vezes em madeira, representando a unidade, a dezena e a centena. É utilizado no Ensino Fundamental como recurso ao ensino de Matemática fundamental. Possibilita ao aluno construir, de forma prática, o sistema de numeração decimal, através da manipulação de materiais concretos. Os exercícios disponíveis no ambiente do *Matematiquinha* são organizados em níveis de complexidade diferenciados. É necessário realizá-los de forma seqüencial e ordenada. O *Matematiquinha* foi testado com alunos em situação de sala de aula real, nas turmas da especialista que auxiliou a modelagem do sistema. Esta experimentação mostrou uma série de limitações e potencialidades do sistema.

O projeto de agentificação do ambiente *Matematiquinha* surgiu para atender as limitações encontradas, quando da utilização do software em situação de sala de aula real. Isto é, uma aula regular e não apenas um experimento com um grupo de alunos. O

professor trabalha, em média, com uma turma de trinta e cinco alunos. Não sendo possível atender cada um dos alunos individualmente, durante todo o tempo de duração da aula. A opção para resolver este problema foi criar um Assistente Virtual (AV) que pudesse auxiliar o aluno a resolver seus exercícios e dúvidas mais frequentes sem o auxílio do professor. Para desempenhar esta função, o AV precisa registrar as informações geradas durante as interações do aluno com o sistema. Para a construção deste AV foi necessário reprojeter o sistema original. A decisão de se construir um AV como um agente se baseia no fato de que já se identificou a necessidade de agregar outros agentes para desempenhar outras tarefas no ambiente, o que caracteriza o novo ambiente numa especificação de Sistemas Multiagentes (SMA).

Este artigo está dividido em cinco seções. A seção 2 apresenta o processo de agentificação do Matematinha e a construção do Matematinha Multiagente. A seção 3 apresenta como foi realizada modelagem cognitiva do usuário. A seção 4 apresenta as considerações finais. No final do texto, na seção 5, estão elencadas as referências bibliográficas citadas.

## **2. Agentificação do Matematinha**

O Matematinha apresenta um conjunto de restrições relacionadas à personalização das atividades do aluno e sua respectiva avaliação. As limitações do Matematinha estão ligadas ao fato deste ser um sistema tutorial sem contemplar atendimento personalizado. Portanto, o Matematinha foi concebido sem a preocupação de modelar o estado cognitivo do aluno ou, mesmo, procurar variar as estratégias de ensino de forma a adequá-las a um determinado perfil de aluno. Todos os usuários são tratados de forma padronizada no que tange ao processo de avaliação. Ou seja, submetidos a uma mesma seqüência de exercícios, com os mesmos critérios de avaliação. Não existe uma seleção de possíveis exercícios, levando em consideração as necessidades de cada aluno. Logo, gerou-se uma demanda a ser atendida.

Com o intuito de atender a essa demanda, criou-se novos componentes com capacidade de identificar erros cometidos durante a interação com o sistema, modelar o perfil cognitivo de cada jogador, armazenar estas informações em um banco de dados e, com base nestas informações, prestar um atendimento exclusivo (personalizado) a cada usuário do sistema, selecionando possíveis exercícios, levando em consideração as necessidades de cada aluno. O Matematinha passou a interagir com os novos componentes, formando assim, o novo sistema denominado Matematinha Multiagente (MM), que é baseado na arquitetura de um Sistema Tutor Inteligente (STI). Existe uma base de domínio expressa por um conjunto de exercícios, organizada em função das atividades e objetivos do professor que atua em sala de aula presencial. Utilizou-se o trabalho de Schuck e Giraffa [7] para identificar as funcionalidades necessárias para a construção deste assistente no que diz respeito à modelagem do tutor no que tange ao processo de seleção de estratégias de ensino e adoção de táticas associadas.

O projeto do MM levou em conta, as funcionalidades do Matematinha já existentes, que foram preservadas e estendidas para trabalhar em sintonia com os novos componentes desenvolvidos para o ambiente. O esforço para se efetuar o processo de agentificação foi considerável. Foi necessário adicionar ao programa original uma nova camada de software para garantir a comunicação entre os diversos módulos e a

compatibilidade entre as tecnologias envolvidas no projeto original e no novo. Esta nova camada foi adicionada de maneira conceitual, portanto, ela não pode ser visualizada fisicamente, uma vez que esta camada é apenas uma abstração conceitual para representar a agentificação do sistema. No MM existe um assistente virtual composto de dois agentes: o tutor e o assistente. O tutor engloba as funções do tradicional módulo de estratégias de ensino previsto em arquiteturas de STI segundo Giraffa e Vicari [3]. Ele é o responsável pela elaboração do plano de trabalho que cada aluno irá executar durante sua sessão de trabalho. Para tal, ele utiliza as informações armazenadas no banco de informações do aluno (modelo do aluno). Esta base contém as informações relativas ao perfil de cada aluno, considerando os seguintes aspectos: quantidade de vezes que uma tarefa foi refeita até encontrar a resposta correta, uso do sistema de ajuda, tipo de ajuda selecionada e complexidade dos exercícios selecionados. Estes itens foram os escolhidos pela especialista de Educação (professora das séries iniciais) que trabalha com a equipe do projeto.

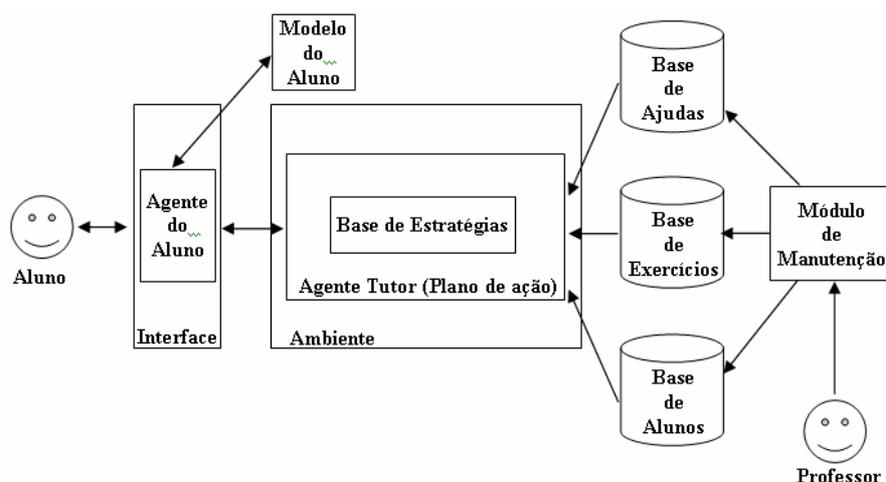
O agente assistente é o responsável pela operacionalização do plano do tutor para um determinado aluno. O Agente Assistente (representado na interface por uma mascote) apresenta os exercícios, controla as informações disponibilizadas na interface, faz as mudanças necessárias associadas a cada execução dos exercícios e as envia para o tutor. A partir de um banco de personagens, o aluno poderá configurar a aparência da mascote ao iniciar o sistema. Sob o ponto de vista do aluno, a mascote é um companheiro que pode auxiliá-lo quando ele precisa. Sob o ponto de vista do sistema, a mascote (visível ou não) trabalha em parceria com o agente tutor.

Uma vez conectado ao ambiente, o aluno passa a interagir com a interface que contém no seu interior o agente assistente. De posse do plano de trabalho elaborado pelo tutor, o agente assistente pode passar ao aluno as informações necessárias ao início das suas atividades. O agente assistente possui a função de monitorar as informações do usuário associadas ao modelo do aluno. O tutor tem o conjunto de informações referentes ao usuário disponível na base de perfis de alunos. Esta base é atualizada pelo agente assistente, ao final de cada sessão de trabalho. Considera-se uma sessão de trabalho aquela onde o aluno executa o plano criado pelo tutor que normalmente consiste de uma seqüência de três exercícios. Ao final de cada sessão, o aluno é informado quanto ao seu desempenho e questionado quanto ao seu desejo de continuar ou não a interagir com o sistema (jogar).

Caso o usuário decida continuar jogando, o sistema disponibiliza a próxima seção de exercícios. Caso contrário, as informações são armazenadas de maneira a permitir que a próxima sessão seja proposta ao usuário quando ele voltar a acessar o sistema.

A coreografia que se estabelece durante as interações de cada aluno vai gerar um conjunto de informações a serem enviadas para o tutor. O tutor possui internamente uma base de estratégias, conforme ilustra a Figura 1. O professor faz a manutenção da base de exercícios. Também é responsabilidade do professor organizar a base de ajudas, a elaboração de elementos de ajuda correspondentes com o padrão de exercícios selecionados. O ambiente possui um módulo de aquisição de conhecimento para que se modifique a base de estratégias. O comportamento do tutor é modelado através da aplicação de táticas que serão utilizadas conforme as categorias de erro transmitidas pelo agente assistente, e que foram estabelecidas pela especialista. Na versão atual esta

atividade fica dependente do programador, uma vez que não se possui um módulo de aquisição de conhecimento para esta finalidade. Este é um ponto crítico do ambiente. O professor especifica as regras e estas deverão ser codificadas por um programador para serem compiladas e o sistema atualizado. Devido a esta dependência, tem-se como trabalho futuro o desenvolvimento de um módulo de aquisição de conhecimento que permita, em alto nível, a modificação destas informações por parte do professor sem a dependência da equipe de desenvolvimento. Este problema não é exclusivo deste projeto e torna-se uma questão em aberto para ser tratada em trabalho futuro.



**Figura 1. Arquitetura Geral do Sistema**

O professor é o único usuário com acesso ao módulo de manutenção. Podendo inserir, retirar e atualizar componentes, na base de alunos e de exercícios. Na base de exercícios encontra-se o tipo de ajuda necessária para que o agente assistente auxilie o aluno. O aluno, ao acessar o ambiente, tem seu nome de usuário e senha verificados. Ocorre, então, a ativação do agente assistente que detém agora, o plano de trabalho criado pelo tutor. Este agente trabalha armazenando as informações do aluno e repassando-as para o tutor que as atualiza a cada sessão.

Quando o aluno encerra sua sessão de trabalho, o agente assistente faz a transferência dos arquivos de dados referentes às informações de cada aluno do computador que o aluno está utilizando para o servidor da escola. O professor possui autorização para fazer alterações nas bases de exercícios e ajudas. Ele pode, também, obter um relatório da base de alunos. O projeto desta versão MM levou em conta, as funcionalidades do Matematicinha já existentes, que foram preservadas e estendidas para trabalhar em sintonia com os novos componentes desenvolvidos para o ambiente. O esforço para se efetuar o processo de agentificação foi considerável. Foi necessário adicionar, ao programa original, novas camadas de software para garantir comunicação entre os diversos módulos e a compatibilidade entre as tecnologias envolvidas no projeto original e no novo.

As novas interfaces do sistema utilizam-se do servidor de banco de dados. As informações provenientes da interação do usuário com estas interfaces devem ser armazenadas pelo Sistema de Gerenciamento de Banco de Dados (SGBD) com a finalidade de obter-se o seu perfil. O SGBD recebe inicialmente o conjunto de

informações provenientes da tela inicial do sistema (tela do cadastro), de maneira a formar o perfil inicial do aluno baseado nas informações fornecidas pelo professor, quando do cadastro do aluno.

A interface do MM mantém as devidas proporções em relação ao tamanho de cada peça virtual representando centena, dezena e unidade, ou seja, a representação gráfica de uma dezena se dará por uma barra contendo dez representações gráficas de unidades (cubos) agrupadas de maneira a representar uma barra (dezena). A representação gráfica de uma centena se dará por um quadro resultante do agrupamento de dez barras (dezenas) e conseqüentemente por cem cubos (unidades). Esta proporcionalidade na representação gráfica das grandezas tem a intenção de permitir que se implementem as funções de troca através da utilização de um lugar específico para este tipo de procedimento denominado “campo de troca”. Justamente as operações de troca são à base do material dourado. Daí a sua importância dentro do contexto do sistema. O campo de troca tem sua visualização indisponível de maneira automática quando o aluno conclui o procedimento de troca ou quando ele opta pelo cancelamento da troca.

Nas novas interfaces do MM o usuário possui opções de personalização das telas do sistema. Ele pode configurar a sua própria área de trabalho escolhendo um ou nenhum entre vários personagens para acompanhá-lo durante o jogo. Também é facultado ao usuário reorganizar qualquer dos três campos principais (campo das centenas, dezenas e unidades). A função “reorganizar” ajusta as posições de todos os elementos para as posições superiores vazias do campo. Desta maneira, as lacunas existentes no campo são eliminadas. Esta função pode ser utilizada depois de colocar modelos na lixeira e depois de procedimentos de troca.

### **3. A Modelagem das Informações do Usuário**

A modelagem do perfil cognitivo do usuário utiliza como base os erros cometidos pelo aluno quando da resolução dos exercícios propostos pelo sistema MM. Foram implementadas rotinas capazes de executar, passo a passo, a correta resolução de cada um dos exercícios. Estas rotinas são executadas logo que o exercício é apresentado na tela e de maneira imperceptível ao usuário. Ou seja, são executadas para que o agente assistente tenha subsídios para a sua tomada de decisão, quando o aluno executa um exercício proposto. Os resultados obtidos através da execução destas rotinas são, então, comparados aos resultados obtidos pelo aluno. Através destas comparações o sistema pode detectar possíveis erros cometidos pelo aluno. Estas rotinas são capazes de fazer comparações em resultados parciais. Ao final de cada um dos exercícios propostos, os erros cometidos pelo usuário são reportados ao agente assistente, que os classifica em categorias. Como o mesmo erro pode ser cometido em exercícios diferentes e erros diferentes podem relacionar-se com uma mesma parcela do conteúdo, optou-se pela abordagem de categorização de erros. Esta categorização, definida pela especialista, leva em conta o tipo de tratamento que cada erro terá por parte do sistema. Cada categoria busca identificar a falta de conhecimento do usuário em relação a uma parcela específica do conteúdo apresentado. Após a classificação e categorização dos erros pelo agente assistente, ele envia ao agente tutor estas informações.

O tutor do MM adota estratégia de ação não invasiva. Ou seja, ele atua como um companheiro que assiste o aluno e fornece orientações para que ele resolva seu

exercício. O tutor se comporta como um assistente/facilitador e não como um guia propriamente dito, aquele que resolve passo-a-passo o exercício para o aluno. Para colocar em prática esta estratégia, o tutor utiliza táticas baseadas em dicas e exemplos diversos.

O conjunto dos erros cometidos pelo aluno até um determinado momento nos permite inferir o estado cognitivo corrente do usuário com relação ao conceito especificado. Fez-se necessária uma ampla discussão com a especialista no sentido de se identificar qual o procedimento mais adequado para auxiliar o usuário a compreender o conceito em questão, prevendo-se que o aluno possa necessitar de vários exercícios e/ou dicas e exemplos até que ele possa construir o conhecimento necessário à compreensão de um determinado conceito. Cabe salientar que o ambiente foi projetado para ser utilizado como recurso da aula prática. Pressupondo-se que a parte conceitual foi trabalhada previamente em sala de aula. E, que o aluno irá utilizar o sistema por mais de uma vez.

No caso dos ambientes baseados em STI, estes exploram a questão da fixação de conteúdos, uma vez que trabalham com uma base de domínio composta por exercícios/problemas organizados, intencionalmente, por níveis de dificuldade e com tratamento da sua resolução acompanhado pelo módulo tutor. Módulo este que na abordagem de STI multiagentes pode ser desempenhado por um ou mais agentes. No caso do MM, esta atividade de tutoria é compartilhada pelo agente tutor.

#### **4. Considerações Finais**

Após a realização deste trabalho reforçando a idéia da necessidade de planejar e documentar o projeto a fim de permitir o trabalho cooperativo e sincronizado da equipe multidisciplinar. O esforço de agentificar o sistema educacional permitiu ao grupo dimensionar aspectos envolvendo pré-requisitos e restrições a serem consideradas para quem deseja reproduzir estes resultados. Resumindo, têm-se como premissas:

- Necessidade de se compor uma equipe interdisciplinar, tendo o especialista tempo disponível para reuniões e organização do material, bem como, interesse e domínio do conteúdo para guiar a equipe nos problemas que surgem em função da tecnologia.
- Os programadores devem dominar as escolhas tecnológicas e devem estar habituados a documentar projetos e programas de forma cooperativa, organizando e padronizando o código fonte;
- O ambiente a ser agentificado deve ter sido avaliado e testado em situação de sala de aula e, possuir resultados positivos e promissores que justifiquem o futuro investimento;
- A base de domínio deve ser organizada na forma de exercícios categorizados por grau de complexidade. E, a eles associadas às regras de comportamento do tutor/assistente. Estas regras e exercícios são definidos pela especialista.

Observaram-se como implicações:

- Aumento do grau de complexidade do projeto no que tange a modelagem e implementação e, por consequência,

- A necessidade de aumentar a quantidade de membros da equipe, pois o volume de trabalho e o tempo gasto no projeto são consideravelmente maiores.
- Aumento do potencial pedagógico do sistema uma vez que o professor passa a ter um ambiente que registra as escolhas e interações do aluno ao longo das sessões.

Este registro permite ao professor ampliar a avaliação de cada aluno, permitindo refazer suas estratégias e táticas na sala de aula. Ele pode ter uma idéia do desempenho e dificuldades dos alunos. Permite um *feedback* do estado cognitivo dos alunos no que tange aos exercícios propostos, associados desta forma, com o seu domínio do conteúdo trabalhado em sala de aula. O processo de agentificação permitiu ao autor e aos demais participantes solidificar conceitos e aprendizagens no que concerne à importância e complexidade inerentes a um projeto deste porte e natureza. Uma visão equivocada de quem observa um software educacional é se atender a aplicação em si, no conteúdo. O que se buscou enfatizar é que o projeto de um software educacional utiliza princípios, métodos e conhecimentos oriundos da Ciência da Computação como qualquer outra aplicação (software). Desta forma, este trabalho foi realizado dentro do contexto de um programa de pós-graduação em Ciência da Computação, pois as habilidades e expertises para sua concepção foram altamente dependentes do perfil de um profissional desta área.

Este artigo não apresentou detalhes da modelagem realizada para construção do sistema, no que tange à Engenharia de Software, tendo em vista o público-alvo a qual se destina (professores que participam de equipes interdisciplinares de desenvolvimento de software educacionais). No entanto, seguiu-se o conjunto de melhores práticas e diretivas da Engenharia de Software no que concerne à modelagem do sistema e organização de toda informação do projeto. Uma das restrições que encontramos, na comunidade de Informática na Educação, é justamente a documentação dos sistemas. Muito informais e não obedecendo aos padrões que permitam o reuso de código, a fim de que os resultados possam ser replicados por outro grupo, ou até mesmo, se conseguir complementar o projeto desenvolvido com outra equipe de pessoas, especialmente com outros desenvolvedores (programadores). A próxima fase do projeto envolverá a complementação das funcionalidades projetadas, e o aumento da “inteligência” dos agentes para desempenho de seus papéis dentro do contexto de ajuda e supervisão das atividades dos alunos.

## 5. Referências

- [1] Carver, Doris; Serrano, Miguel; Oca, Carlos. Reengineering legacy systems for distributed environments. Proceedings of the Journal of Systems and Software, 2002. V. 64 p. 37-55.
- [2] Chiang, Chia-Chu. Wrapping legacy systems for use in heterogeneous computing environments. Proceedings of the Information and Software Technology, 2001. V. 43 p. 497-507.
- [3] Giraffa, L. M. M. (1999) Uma Arquitetura de Tutor Utilizando Estados Mentais. Porto Alegre: CPGCC/UFRGS. (Tese de Doutorado).

- [4] Landauer, C., Bellman K. L. and Gilliam, A. (1993) Software Infrastructure for System Engineering Support. In Proceedings. AAAI '93 Workshop on Artificial Intelligence for Software Rengineering, 12 July, Washington, D. C.
- [5] Landauer, Christopher; Bellman Kirstie. Wrappings for One-of-a-Kind Software Development. Proceedings of the 35th Annual Hawaii International Conference on System Sciences (Hawaii, HI, Jan. 7–10, 2002), pp. 3987–3995.
- [6] Ribeiro, J. C. C. S. and Ribeiro, U.A. L. F. (2002) *Matematiquinha — Uma Ferramenta de Auxilio Ao Ensino de Matemática Para Séries Iniciais*. III Simpósio de Informática do Planalto Médio (SIPM'2002), 21 – 24 de maio, Passo Fundo – RS.
- [7] Schuck, P. W. and Giraffa, L. M. M. (2001) *Construindo um Sistema Tutor Inteligente para suporte ao ensino de Matemática Financeira: da modelagem à validação*. XII Simpósio Brasileiro de Informática na Educação (SBIE), 21 - 23 de novembro, Vitória - ES, Brasil. Sociedade Brasileira de Computação. pp. 418-426.
- [8] VICCARI, Rosa; GIRAFFA, Lucia. The Use of Multi-Agent Systems to Build Intelligent Tutoring Systems. In *Computing Anticipatory Systems: CASYS 2001 - Fifth International Conference*. AIP Conference Proceedings. Liege, Belgium. 13-18 August, 2001. Volume 627. p. 340-348.