
Um Sistema de Autoria para Construção de Ambientes Interativos de Aprendizagem baseado em Agentes

Ig Bittencourt, Camila Nunes, Evandro Costa, Marcos Tadeu, Camila Bezerra

¹Instituto de Computação – Universidade Federal de Alagoas (UFAL)
Campus A. C. Simões, BR 104 - Norte, Km 97, C. Universitária, Maceió, AL – Brasil

{ibert, cpbn, evandro, mts, cbs}@tci.ufal.br

Abstract. *Building Interactive Learning Environment is, normally, a hard task. Aspects such as architecture design, components implementation, development of knowledge representation strategies makes it complex and ineffective to develop, administrate and deploy them effectively in terms of time and costs. To approach these aspects, this paper proposes an authoring system for building agent-based interactive learning environments. This approach provides collaboration tools, an ontology-based infra-structure for educational systems and an agents society that help domain modeling and problems solving process, through Artificial Intelligence technics.*

Resumo. *O esforço envolvido na construção de Ambientes Interativos de Aprendizagem é, normalmente, uma tarefa árdua. Aspectos como definição da arquitetura, implementação de componentes, estratégias de representação do conhecimento tornam seu desenvolvimento, administração e distribuição complexos e ineficazes em termos de tempo e custo. Para abordar tais aspectos, propõe-se neste artigo um Sistema de Autoria para Construção de Ambientes Interativos de Aprendizagem baseado em Agentes. Essa abordagem possui ferramentas de colaboração, uma infra-estrutura baseada em ontologia para sistemas educacionais e uma sociedade de agentes que auxiliem na modelagem do domínio e resolução de problemas, através técnicas de Inteligência Artificial.*

1. Introdução

A construção de Ambientes Interativos de Aprendizagem é, normalmente, uma tarefa árdua, requerendo esforços dos engenheiros de *software* para desenhar toda a arquitetura do sistema, além de implementar todos os componentes, desenvolver as estratégias de representação do conhecimento e mecanismos de raciocínio, falta de interoperabilidade, dificuldade de compartilhamento de materiais, alto custo de manutenção, entre outros [Rodrigues et al. 2005].

A partir dessa constatação, passou-se a investir em ferramentas para automatizar alguns aspectos envolvidos na construção dos ambientes em questão. Neste contexto, surgiu a categoria de ferramentas denominada sistemas de autoria, dando suporte ao desenvolvimento, administração e distribuição de *softwares* instrucionais de uma forma efetiva em termos de tempo e custo. A idéia básica destas ferramentas é permitir ao usuário que não possui experiência em programação poder construir de forma ágil *softwares* instrucionais.

Ambientes interativos de aprendizagem têm recebido grande importância na área de educação, principalmente no que diz respeito ao suporte à aprendizagem colaborativa. Ambientes colaborativos focam no trabalho em grupo, com vistas a promover a aprendizagem. Como exemplo destes ambientes, pode-se citar o de comunidades virtuais de aprendizagem.

Atualmente, observa-se a chegada de uma geração mais nova de ambientes computacionais de suporte à aprendizagem cooperativa dotada de recursos ainda mais sofisticados e, conseqüentemente, trazendo mais complexidade no seu desenvolvimento. Estes ambientes, em geral, adotam o paradigma de agentes inteligentes [Plekhanova 2002]. A construção de sistemas baseados em agentes ainda se constitui numa tarefa complexa, devido principalmente à falta de recursos oferecendo facilidades aos seus desenvolvedores, como encapsulamento de técnicas de Inteligência Artificial em componentes.

A proposta deste artigo é a construção de um Sistema de Autoria para Construção de Ambientes Interativos de Aprendizagem baseado em Agentes. Tal abordagem possui ferramentas de colaboração, uma infra-estrutura baseada em ontologia para sistemas educacionais e sociedades de agentes que auxiliem na modelagem do domínio e no processo de resolução de problemas, através de técnicas de Inteligência Artificial.

O artigo está organizado da seguinte forma. Os trabalhos relacionados são apresentados na Seção 2. Na Seção 3, descreve-se a arquitetura do sistema. Um cenário ilustrativo abordando o uso do sistema é apresentado na Seção 4. Finalmente, trabalhos futuros e conclusões são apresentados na Seção 5.

2. Trabalhos Relacionados

Um importante relato do estágio relativamente atual da pesquisa pode ser visto em [Murray et al. 2003]. Mais recentemente novos ambientes surgiram, dentre os quais se destacam:

[Aroyo et al. 2004] equivale a um sistema de autoria evolutivo, possuindo características de raciocínio sobre o próprio comportamento e fazendo uso de regras para manutenção nas diferentes partes do processo de autoria. Apesar do ambiente comportar os diversos aspectos de aprendizagem (domínio, estudante, pedagógico e plano instrucional), não possui características de extensibilidade, bem como suporte para utilização de diversos mecanismos de inferência no processo de resolução de problemas.

[Aleven et al. 2006] possui duas (Tutor Cognitivo e Tutoria baseada em Exemplos) características de autoria, dando suporte para engenheiros de *softwares* e autores. Porém, o processo de criação dos agentes de interfaces ainda ocorre de forma lenta. Além disso, não possui um formalismo de representação do conhecimento baseado em padrões de compartilhamento.

Uma arquitetura de aprendizagem baseada em agentes, componentes e *framework* é proposta em [do Socorro Silva et al. 2003]. A utilização de componentes e *framework* nesse ambiente têm como finalidade a (re)utilização de ferramentas disponíveis na arquitetura, além da construção de espaços de aprendizagem seguindo diferentes abordagens e ferramentas pedagógicas. Apesar de possuir características de extensibilidade, a evolução do conhecimento é um aspecto limitado no ambiente.

Além disso, ambientes educacionais de autoria com objetivos específicos foram

desenvolvidos, como ferramentas de autoria destinados a apoiar o ensino de Xadrez e o sistema brKChess (ferramenta de autoria para ensino de Xadrez), que foram adotados em [Hartmann et al. 2005].

A fim de explorar as limitações encontradas nos trabalhos citados, é proposto neste artigo um sistema para construir ambientes interativos de aprendizagem baseado em agentes, dando facilidades tanto para o programador quanto para o autor (não programador). O sistema leva em consideração aspectos de Engenharia de Software, como *framework* e *shell*, além de possuir um formalismo de conhecimento baseado em ontologias OWL (objetivando a evolução e compartilhamento do conhecimento). Outro aspecto importante é que no sistema proposto o autor pode configurar seu mecanismo de inferência através de interfaces gráficas bem definidas, além de poder integrar diversas técnicas de Inteligência Artificial, como Raciocínio Baseado em Casos e Regras.

Como foi dito anteriormente, as características citadas possuem relevância com relação aos outros ambientes. Além disso, o mesmo possui um aspecto inovador referente a integração de diversas técnicas de Inteligência Artificial através de interfaces gráficas adaptativas.

3. Arquitetura do sistema

Na Figura 1 é apresentada a arquitetura do sistema baseado em agentes para ambientes de aprendizagem interativa. A arquitetura está dividida em camadas, e destacam-se os recursos presentes no núcleo do sistema. Tais recursos são: conjunto de ferramentas interativas, uma sociedade de agentes tutores autônomos (SATA), uma sociedade de agentes de decisão e aprendizagem (SADA) e um agente de coordenação. Os agentes e sociedades de agentes foram desenvolvidos utilizando o *framework* Jadex [Braubach et al. 2002]. Jadex foi escolhido por ter sido estendido do ambiente Jade [Tilab 2005]¹, pois os agentes são baseados em *BDI* (*Belief, Desires and Intentions*) e possuem ferramentas para manipular ontologias criadas no Protégé [Stanford 2000]. O uso de ontologias é motivado pelos diversos recursos providos, possibilitando solucionar os problemas atualmente enfrentados por pesquisadores da área em questão [Mizoguchi and Bourdeau 2000].

Na camada de segurança, o componente de segurança é responsável pela autenticação e autorização dos usuários no sistema. Utilizou-se a tecnologia *Java Authentication and Authorization Service* (JAAS) [MicroSystems 2006], que fornece um conjunto de interfaces para autenticação e autorização bem definidas

A camada de infra-estrutura contém o componente que implementa o serviço para atender o requisito não-funcional do sistema, que é o serviço de persistência de objetos. Este componente é responsável por manter as conexões com o mecanismo de persistência, como um banco de dados relacional.

3.1. Agente Coordenador

Este agente possui duas funcionalidades: *i*) garantir a comunicação entre os agentes da SATA e SADA, permitindo aos agentes de uma sociedade solicitarem ao agente coordenador qual agente tem habilidade para determinada tarefa; *ii*) facilitar a utilização da

¹Esta extensão faz com que os padrões de interoperabilidade para comunicação entre agentes (FIPA) sejam herdados.

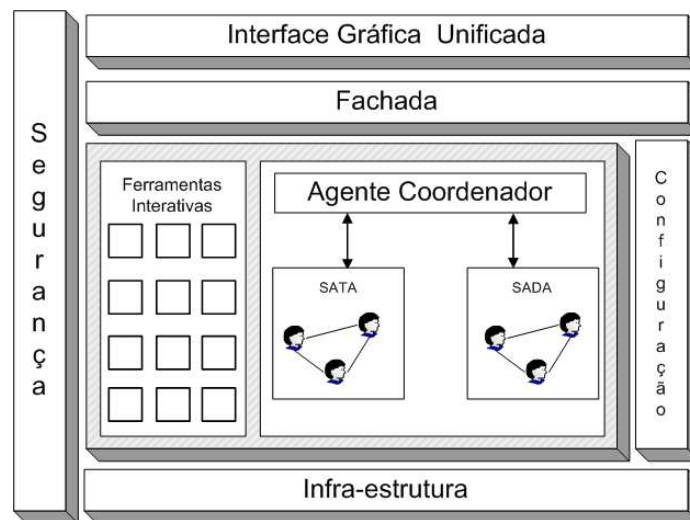


Figura 1. Arquitetura do sistema de autoria.

abordagem de autoria, pois a partir do momento em que diversos recursos são disponibilizados para o autor, a complexidade e usabilidade tornam-se inversamente proporcionais, ou seja, a complexidade aumenta e a capacidade de gerenciamento dos recursos diminui. Com isso, o agente coordenador tem total controle de todas as atividades que devem ser executadas pelos agentes e a quantos passos de distância o autor está da configuração do ambiente educacional.

A configuração deste agente se dá através da utilização de uma ontologia, possuindo informações do tipo:

- **Metodologia:** é utilizada pelos autor/engenheiro de *software* para construir o sistema educacional. Entende-se por metodologia, dois viés diferenciados, sendo eles: processos de Engenharia de Software e metodologias educacionais. Processo de Engenharia de Software é voltado para equipes de desenvolvimento, objetivando a implementação de novas funcionalidades. Como exemplo pode ser definida uma metodologia RUP [Kruchten 2003] ou XP [Beck 2001]. Metodologias educacionais são voltadas para o autor, que define qual a seqüência de construção do ambiente educacional e do conhecimento;
- **Tipo de Ambiente Educacional:** pode ser definida a construção de diversos tipos de ambientes educacionais, como Sistemas Tutores Inteligentes, Sistemas Colaborativos, entre outros. A escolha do tipo de ambiente tem influência direta na metodologia educacional escolhida;
- **Informações dos Agentes:** características dos agentes presentes nas sociedades de agentes são fundamentais para que o agente coordenador saiba que tipo de agente pode ser solicitado para uma tarefa.

3.2. Sociedade de Agentes Tutores Autônomos (SATA)

A Sociedade de Agentes Tutores Autônomos é responsável pelo conhecimento sobre os aspectos cognitivos envolvidos no processo de ensino-aprendizagem. Dentre as preocupações referentes ao módulo do domínio, destacam-se: *i) Resolução de Problemas:* característica importante que pode estar presente na construção de sistemas educacionais.

Para isso, diversas técnicas de Inteligência Artificial podem ser utilizadas. Isto justifica a necessidade de integração com SADA; ii) *Informações do domínio*: equivale a características sobre o que será ensinado para o estudante, ou seja, o conteúdo curricular que será repassado.

SATA foi modelada baseada no modelo MATHEMA [Costa 1997], porém a construção da ontologia, ilustrada na Figura 2 se deu através da avaliação dos trabalhos [Dillenbourg and Self 1992, Chen and Mizoguchi 2004, de Barros Costa et al. 1998]. Além disso, houve a integração da ontologia feita em SATA com SCORM [Learning 2004], para utilização de objetos de aprendizagem.

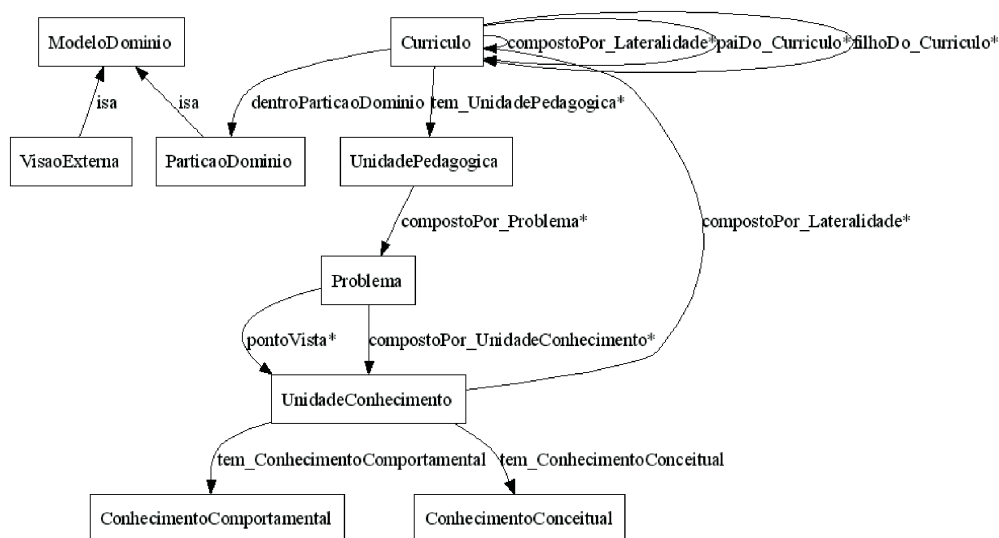


Figura 2. Ontologia do Modelo do Domínio

3.3. Sociedade de Agentes de Decisão e Aprendizagem (SADA)

SADA permite a integração dinâmica de diversas técnicas de Inteligência Artificial, objetivando a resolução de problemas, como agrupamento de usuário, extração de conteúdo de *chat*, avaliação, entre outros. Dentre os algoritmos presentes atualmente em SADA, citam-se:

- **Raciocínio Baseado em Regras (RBR)**: trata-se de uma abordagem para solucionar problemas. Para isto, o sistema precisa ter uma base de regras consistente armazenada, por exemplo, em um banco de dados, e também de fatos dados pelo usuário para se chegar à conclusão do problema. Um sistema baseado em regras é composto por base de conhecimento, ferramenta de inferência e o subsistema de explicação.
- **Raciocínio Baseado em Casos (RBC)**: trata-se de uma abordagem que se utiliza de um tipo de raciocínio analógico, contrastando com o convencional paradigma de resolução de problemas baseado em regras. Portanto, a abordagem para a solução de problemas no Raciocínio Baseado em Casos é baseada na resolução de problemas valendo-se de experiência passada [Ig Ibert Bittencourt 2006].
- **Busca**: aborda a resolução de problemas através de algoritmos de busca. As implementações de busca vigentes em SADA equivale a busca cega e informada,

com algoritmos como, largura, profundidade, profundidade limitada, A*, t mpera simulada, subida de encosta, entre outros.

Outras t cnicas podem ser adicionadas no ambiente atrav s da configura o da ontologia e adi o do agente espec fico da t cnica.

A integra o das t cnicas ocorre de forma din mica atrav s da utiliza o da ontologia ilustrada na Figura 3. Para isso, quatro tipos de informa es t m que ser levados em considera o para cada algoritmo:

1. Entrada/Sa da: equivale aos dados de entrada e sa da do algoritmo e seus tipos.
2. Processamento: refere-se   forma no qual o algoritmo   processado internamente. Cada fase do Ciclo 4R [von Wangenheim and von Wangenheim 2003] presente em um algoritmo de RBC seria um exemplo.
3. *Feedback*: equivale a um retorno dado para o algoritmo ap s determinado processamento. Este *feedback* pode se dar de forma s ncrona ou ass ncrona, conseq entemente, tanto o usu rio quanto outro agente pode dar esta informa o para o agente.
4. Estat stica: equivale a dados pr -estabelecidos para avaliar o desempenho do algoritmo.

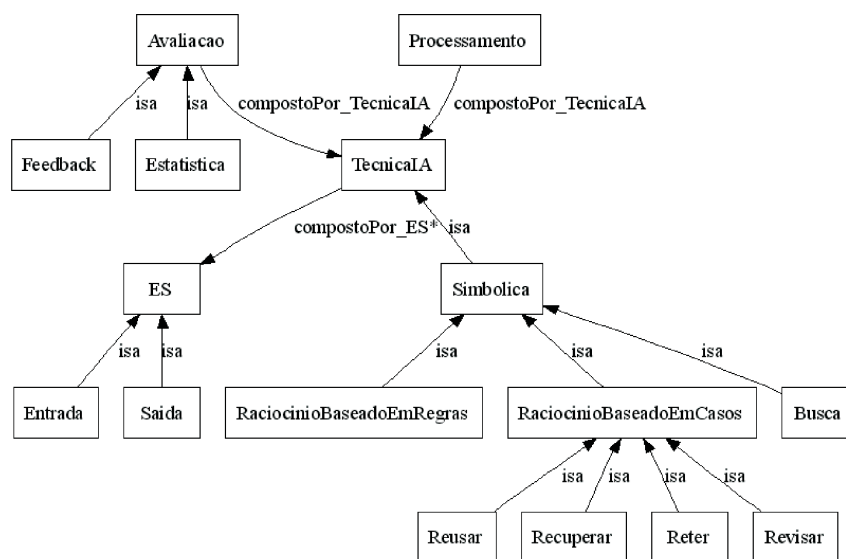


Figura 3. Ontologia Das T cnicas de Decis o e Aprendizagem

3.4. Ferramentas Interativas

Ambientes de aprendizagem colaborativa, como as comunidades virtuais, necessitam de ferramentas que possibilitem a intera o dos usu rios com o sistema.

As ferramentas dispon veis no sistema s o apresentadas:

- *f rum*: permite aos usu rios postarem t picos com discuss es relativas aos interesses de determinada comunidade;
- *scrap, blog, enquete e e-mail*: para intera o entre usu rios;
- *biblioteca digital*: onde podem ser armazenadas informa es em formato de arquivo digital que poder o ser compartilhadas entre os usu rios de uma comunidade virtual. Esses arquivos devem estar relacionados com o contexto da comunidade e os usu rios podem iniciar discuss es sobre eles.

4. Cenário Ilustrativo

Nesta seção apresenta-se um cenário ilustrativo a fim de demonstrar a utilização do sistema proposto. Neste cenário ilustram-se os diversos recursos disponibilizados pelo sistema, consistindo de um sistema de aprendizagem colaborativo para o domínio jurídico, mais especificamente no Código Penal Brasileiro [Bittencourt et al. 2006]. No ambiente, o professor cadastra a comunidade (Subseção 4.1) referente a um código da legislação penal. Após criação da comunidade, o mesmo configura o modelo do domínio de aprendizagem (Subseção 4.2) e mecanismos de inferência (Subseção 4.3).

4.1. Criação de uma comunidade

Neste cenário, o usuário propõe a criação de uma comunidade virtual. Os passos para a criação da comunidade de *Homicídio* são descritos abaixo.

- **Passo 1:** o professor digita o nome da comunidade e a sua descrição (Figura 4);
- **Passo 2:** o professor cria os papéis e atribui permissões para o mesmo. Papéis: Estudante e Monitor; Permissão: Estudante: Acesso ao sistema educacional; Monitor: Acesso as avaliações e desempenho educacional do estudante;
- **Passo 3:** o professor da comunidade define a categoria da sua comunidade. Por exemplo, Categoria: Direito Penal.

Nome da comunidade:	Homicídio
Descrição:	Comunidade destinada ao estudo do artigo 121 do Código Penal Brasileiro.

Figura 4. Ilustração do passo de criação de comunidade.

4.2. Modelagem do Domínio

A modelagem do domínio ocorre através da utilização de SATA. Neste cenário o professor define informações acerca do conteúdo de Homicídio que será ensinado ao estudante. Na Tabela 1 é definido um modelo de domínio simplificado da comunidade proposta².

Associado a cada unidade pedagógica, um conjunto de problemas é definido pelo professor para avaliação do conhecimento do estudante. Para viabilizar o processo de resolução de problemas são utilizados dois mecanismos de inferência (Raciocínio Baseado em Casos e Raciocínio Baseado em Regras). Os passos para a configuração dos mecanismos são descritos na Subseção 4.3.

4.3. Integração com IA

Após as fases especificadas anteriormente, o professor define as regras e os casos referentes aos problemas das unidades pedagógicas.

A metodologia de especificação do algoritmo de inferência é baseada em uma estrutura de *Shell*. Assim, o professor tem mais facilidade para manipular regras e casos.

Abaixo segue a forma de configuração dos algoritmos.

²Detalhes com relação a modelagem descrita na Tabela 1, vide Subseção 3.2

Tabela 1. Dados do Domínio da Comunidade de Homicídio

Classe	Conteúdo
Domínio	Direito Penal
Partição do Domínio	Homicídio
Currículo ₁	Simple
Unidade Pedagógica ₁	Problema ₁
Unidade Pedagógica ₂	Problema ₂
...	...
Currículo ₂	Culposo
Unidade Pedagógica ₁	Problema ₁
Unidade Pedagógica ₂	Problema ₂
...	...
Currículo ₃	Qualificado
Unidade Pedagógica ₁	Problema ₁
Unidade Pedagógica ₂	Problema ₂
...	...

4.3.1. Agente de Raciocínio Baseado em Regras

Os passos para a utilização do componente de RBR são:

- **Passo 1:** o usuário cadastra as variáveis e seus respectivos valores;
- **Passo 2:** o usuário cria as regras específicas do seu domínio;
- **Passo 3:** neste passo, o usuário escolhe o método de inferência, que são: encadeamento para a frente ou o encadeamento para trás;
- **Passo 4:** é configurado o módulo de explanação através do mapeamento de regras em linguagem natural, facilitando o entendimento dos estudantes com relação a explanação das regras.

Segue abaixo um exemplo (Homicídio Simple) de como as regras são estruturadas no sistema.

Regra *Se HouveAssassinato = 'Sim' e IdadeVitima > 18 Entao Homicidio = 'Simple'*

4.3.2. Agente de Raciocínio Baseado em Casos

Para a utilização do componente de RBC são especificados sete passos indicados na Figura 5. São eles:

- **Passo 1:** o usuário seleciona a base de casos;
- **Passo 2:** é definido os atributos;
- **Passo 3:** define-se o peso de cada atributo. Os valores padrões dos pesos equivalem a 1, pois o usuário pode não querer diferenciar os atributos;
- **Passo 4:** logo após são definidas as funções de similaridade local, ou seja, o cálculo de similaridade entre os atributos. Além disso, é definida nesta fase a função de similaridade global, sendo o cálculo de similaridade entre os casos;
- **Passo 5:** define a forma de recuperação dos casos, podendo ser seqüencial, dois níveis, entre outros;
- **Passo 6:** local onde se define um arquivo jar para a fase de reuso e revisão dos casos similares. Estas fases não são disponibilizadas no ambiente na forma de configuração padrão, devido a grande especificidade das mesmas;
- **Passo 7:** define o limiar de retenção do caso.

Agente de Raciocínio Baseado em Casos

The image shows a graphical user interface for configuring a Case-Based Reasoning (RBC) component. It consists of seven distinct steps:

- Passo 1:** Base de Casos. A list box containing 'Livros', 'Jurisprudências', and 'Comunidades Virtuais'. 'Jurisprudências' is selected.
- Passo 2:** Seleção dos atributos. Three list boxes: 'Modalidade do Crime', 'Qualificação do Crime', and 'Tipo de Ação'.
- Passo 3:** Coloque o peso dos atributos. Four input fields for 'Modalidade do Crime', 'Qualificação do Crime', 'Tentativa', and 'Tipo de Ação', each with a '1' entered.
- Passo 4:** Atributos. A 'Casos' dropdown set to 'Vizinho mais próximo'. Four dropdowns for 'Modalidade do Crime' (simbólica), 'Qualificação do Crime' (simbólica), 'Tentativa' (sim), and 'Tipo de Ação' (Taxonômica).
- Passo 5:** Recuperação. A 'Teste' dropdown set to 'Em dois níveis' and an 'Arquivo...' button.
- Passo 6:** Reúso and Revisão. Two 'Arquivo...' buttons.
- Passo 7:** Limiar de Retenção. An input field containing '0.75'.

Figura 5. Passos para utilização do componente de RBC

5. Conclusões e Trabalhos Futuros

Este artigo descreveu um sistema de autoria para construção de ambientes interativos de aprendizagem baseado em agentes. O sistema foi desenvolvido utilizando as tecnologias *Java*, *Jadex* e *Protégé*. A utilização das sociedades de agentes para modelagem do domínio educacional e utilização de algoritmos inteligentes são aspectos fundamentais para a construção de ambientes interativos de aprendizagem. Com o sistema proposto, espera-se ter contribuído para a área de informática na educação, em ambientes interativos de Aprendizagem e ambientes de autoria de atividades de aprendizagem.

Como trabalhos futuros, objetiva-se construir agentes de decisão e aprendizagem contendo as técnicas de IA, como Q-Learning. Além disso, objetiva-se finalizar as telas de assistência de SATA, dando suporte à configuração dos objetos de aprendizagem SCORM através do sistema. Pretende-se também fazer parcerias com outras universidades com o objetivo de divulgar e testar ainda mais o sistema.

Referências

- Aleven, V., McLaren, B. M., Sewall, J., and Koedinger, K. R. (2006). The cognitive tutor authoring tools (ctat): Preliminary evaluation of efficiency gains. In *Intelligent Tutoring Systems*, pages 61–70.
- Aroyo, L., Inaba, A., Soldatova, L., and Mizoguchi, R. (2004). Ease: Evolutional authoring support environment. In *Intelligent Tutoring Systems*, pages 140–149.
- Beck, K. (2001). *Programação eXtrema (XP) eXplicada*. Bookman.
- Bittencourt, I. I., Tadeu, M., Costa, E., Nunes, R., and Silva, A. (2006). Combining ai techniques into a legal agent-based intelligent tutoring system. In *Eighteenth International Conference on Software Engineering and Knowledge Engineering - SEKE, 2006, San Francisco*, volume 18, pages 35–40.
- Braubach, L., Pokahr, A., and Walczak, A. (2002). *Jadex*. <http://vsis-www.informatik.uni-hamburg.de/projects/jadex/>. último acesso em Agosto de 2006.

-
- Chen, W. and Mizoguchi, R. (2004). Learner model ontology and learner model agent. *Cognitive Support for Learning - Imagining the Unknown*, pages 189–200.
- Costa, E. B. (1997). *Um Modelo de Ambiente Interativo de Aprendizagem Baseado numa Arquitetura Multi-Agentes*. Tese de doutorado, Universidade Federal da Paraíba, Campina Grande.
- de Barros Costa, E., Perkusich, A., and Ferneda, E. (1998). From a tridimensional view of domain knowledge to multi-agents tutoring systems. *Lecture Notes in Artificial Intelligence, 1515, Springer, Berlin, Germany*, pages 61–72.
- Dillenbourg, P. and Self, J. (1992). A framework for learner modelling. *Interactive Learning Environments*, 2:111–137.
- do Socorro Silva, A., de Brito, S. R., Favero, E. L., Hernández-Domínguez, A., de Lira Tavares, O., and Francês, C. R. L. (2003). Uma Arquitetura para desenvolvimento de Ambientes Interativos de Aprendizagem baseado em Agentes, Componentes e Framework. In *XIV Simpósio Brasileiro de Informática na Educação*, pages 203–212, Rio de Janeiro, RJ.
- Hartmann, C., Direne, A., Bona, L. C. D., Silva, F., and dos Santos and Marcos Castilho, G. (2005). Linguagem e ferramenta de autoria para promover o desenvolvimento de perícias em xadrez. In *Anais do XVI Simpósio Brasileiro de Informática na Educação, Juiz de Fora*, volume 1, pages 656–665.
- Ig Ibert Bittencourt, Evandro Costa, E. F. e. R. A. (2006). Sistemas de informação e apoio a decisão baseado em conhecimento. In *Escola Regional de Informática de Minas Gerais, aceito para apresentação em Minicurso*.
- Kruchten, P. (2003). *The Rational Unified Process: An Introduction*. Addison Wesley.
- Learning, A. D. (2004). Scorm. <http://www.adlnet.gov/downloads/272.cfm>.
- MicroSystems, S. (2006). Java Authentication and Authorization Service. <http://java.sun.com/products/jaas/>. último acesso em Agosto de 2006.
- Mizoguchi, R. and Bourdeau, J. (2000). Using ontological engineering to overcome common ai-ed problems. *International Journal of Artificial Intelligence in Education*, 11(2):107–121.
- Murray, T., Ainsworth, S., and Blessing, S. (2003). *Authoring Tools for Adv. Tech. Learning Env.*, pages 493–546. Kluwer Academic Publishers, Netherlands.
- Plekhanova, V. (2002). *Intelligent Agent Software Engineering*. Idea Group Publishing, Hershey, PA, USA.
- Rodrigues, M., Novais, P., and Santos, M. F. (2005). Future challenges in intelligent tutoring systems - a framework. In *m-ICTE2005 3rd International Conference on Multimedia and Information and Communication Technologies in Education*. FORMATEX.
- Stanford (2000). Protégé Ontology Editor and Knowledge Acquisition System. <http://protege.stanford.edu>.
- Tilab (2005). Java Agent Development Framework. <http://jade.tilab.com/>.
- von Wangenheim, C. G. and von Wangenheim, A. (2003). *Raciocínio Baseado em Casos*. Manole.