Uma Abordagem para Provimento de Recursos em um Ambiente Interativo de Aprendizagem

Priscylla Silva[1,2], Evandro Costa[1], Baldoino Fonseca[1]

[1]Instituto de Computação – Universidade Federal de Alagoas (UFAL)
Campus A. C. Simões – Av. Lorival Melo Mota, s/n, Maceió – AL – Brazil

[2]Instituto Federal de Alagoas (IFAL) – Murici, AL – Brazil

{pmss,evandro,baldoino}@ic.ufal.br

Abstract. *This paper describes an approach to provide adaptive resources in the context of problem solving, involving students in an Interactive Learning Environment on mathematical domain. Specifically, this work proposes the use of multiple support resources considering the ways that influencied the difficulties presented by the students in problem solving situations. These resources aim to help a student to solve problems in a success way. Hence, it has been used information from the current problem, as well as, from both the student model and the difficulty type of the student, and then the intelligent support system select and offer the most appropriate resource for helping the student. We have considered resources, such as: textual hints, videos and analogous solutions by using case-based reasoning, taking into account the use of a database of solved problems in order to help the student in his current problem.*

Resumo. *Este artigo descreve uma abordagem para provimento de recursos oportunos e adaptativos circunstanciados ao processo de resolução de problemas de estudantes posicionados em um ambiente interativo de aprendizagem no domínio da matemática. Especificamente, este trabalho propõe o uso de suporte com múltiplos recursos, visando auxiliar o estudante com dificuldade na resolução de um determinado problema, tendo em consideração os motivos que influenciaram as dificuldades por ele apresentadas. Esses recursos, portanto, prestam-se a ajudar um estudante a resolver problemas de forma bem sucedida. Assim, foi desenvolvido um sistema de suporte inteligente que utiliza como fontes o modelo do estudante, o estado atual da solução do problema e os motivos da dificuldade do estudante. A partir dessas fontes são selecionadas as formas de ajuda mais adequadas, sendo compostas por dicas textuais, videoaulas e soluções análogas. Este módulo se encontra integrado a um sistema de raciocínio baseado em casos que utiliza como base problemas já solucionados para auxiliar o estudante em seu problema atual.*

1. Introdução

Ambientes Interativos de Aprendizagem são sistemas de software com propósitos educacionais, tendo como característica central oferecer interações personalizadas, considerando principalmente os estados cognitivos dos estudantes com os quais interagem. O presente trabalho se situa no contexto de ambiente multiagente voltado para aprendizagem de expressões aritméticas, privilegiando uma abordagem de aprendizagem baseada

DOI: 10.5753/CBIE.SBIE.2013.386

em problemas. Nesta abordagem o estudante soluciona problemas e, a partir de suas soluções, o sistema pode inferir informações sobre o seu conhecimento e alimentar um modelo que será utilizado para personalizar a interação com o estudante.

No processo de resolução de problemas, surgem algumas questões que podem ser identificadas em cenários nos quais o estudante pode passar, tais como: o estudante não compreender o enunciado do problema ou não possuir conhecimento suficiente para solucioná-lo, até mesmo situações em que ele, apesar de dominar o assunto, "*esquece*"a maneira como a solução deve ser elaborada. Todos esses fatores podem atrapalhar o estudante em seu aprendizado e até mesmo desmotivá-lo. Partindo dessas constatações, surge o desafio de como auxiliar o estudante a superar as suas dificuldades durante os exercícios nos quais ele deve colocar em prática o seu conhecimento. Algumas pesquisas na literatura têm abordado este tema e realizado propostas para auxiliar o estudante no processo de resolução de problemas. Neste sentido, foram propostos alguns mecanismos de suporte para o estudante durante esse processo, entre eles, o uso de dicas [Seffrin et al. 2012, Aleven and Koedinger 2000, Zhou et al. 1999].

Este trabalho propõe o uso de múltiplos recursos de suporte alinhado com os motivos que levam o estudante a ter dificuldade na resolução de problemas. Estes recursos visam ajudar a estudante a solucionar os problemas com sucesso, para isso, são utilizadas as informações do problema corrente, o modelo do estudante e o tipo da dificuldade do aluno. De posse dessas informações o sistema oferece o recurso de ajuda mais apropriado.

Este artigo está estruturado do seguinte modo. Na Seção 2 faz-se uma contextualização do presente trabalho, posicionando-o no contexto do Ambiente Interativo de Aprendizagem, e apresentando o ambiente com sua arquitetura e funcionalidades. Na Seção 3 é apresentado o sistema de suporte inteligente proposto neste trabalho. Na Seção 4 são discutidos alguns trabalhos relacionados comparando-os com o aqui proposto. Por fim, na seção 5 são apresentadas as considerações finais e proposição de trabalhos futuros.

2. Ambiente Interativo de Aprendizagem

Nesta seção apresenta-se o Ambiente Interativo de Aprendizagem (AIA) projetado e desenvolvido neste trabalho. Em primeiro lugar, fornece-se uma visão geral do ambiente e suas funcionalidades. Posteriormente descreve-se a modelagem do domínio proposto e, por fim, apresenta-se a arquitetura do sistema.

2.1. Visão Geral

O AIA desenvolvido neste trabalho possui uma abordagem multiagente. Ele tem por objetivo auxiliar o processo de ensino/aprendizagem fornecendo recursos tanto para o professor, para ajudá-lo no acompanhamento da evolução do conhecimento dos estudantes, quanto aos estudantes, através de recursos e suporte para tornar a sua participação neste processo mais proveitosa e eficaz. O estudante é o principal ator do sistema, podendo usufruir das seguintes funcionalidades:

- O estudante pode apresentar problemas e receber como resposta do sistema a solução para o mesmo com uma explicação passo-a-passo da resolução.
- O estudante pode por em prática os seus conhecimentos através de exercícios e receber *feedbacks* sobre o seu desempenho.

- O estudante pode solicitar ajuda durante o processo de resolução de exercícios e receber dicas e materiais de suporte contextualizados. Alternativamente o sistema pode detectar a necessidade de ajuda do estudante sem que seja preciso que o mesmo requisite ajuda de forma explícita.

Na figura 1 encontra-se uma das interfaces do AIA vista pelo estudante durante a resolução de problemas.

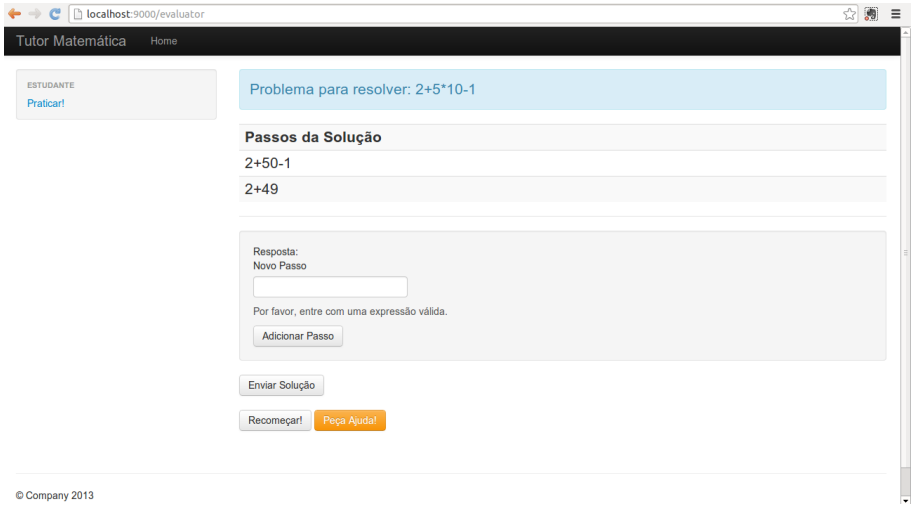

Figura 1. Interface utilizada pelo estudante para resolver problemas no Ambiente.

2.2. Arquitetura

O AIA possui uma arquitetura modular e contém seis módulos principais como pode ser visto na figura 2, onde é possível observar a arquitetura conceitual do sistema e os atores que interagem com o ele por meio de sua interface. O professor alimenta o AIA com novos problemas e configura o modelo pedagógico e o modelo de domínio de acordo com as suas preferências. O especialista (*expert*) é responsável pelas configurações gerais e manutenção do sistema.

Cada módulo da arquitetura do ambiente é descrito a seguir:

- **Interface** - Responsável por intermediar o diálogo entre o ambiente e os seus usuários.
- **SATA** - Este módulo é conhecido como Sociedade de Agentes Tutores, sendo formado por um sistema multiagente. A construção de SATA é baseada em uma estrutura de *curriculum*, onde cada tópico presente na estrutura é mapeado em um agente tutor.
- **Tools** - Este módulo contém todas as ferramentas disponíveis no sistema para os usuários. Elas podem ter seus serviços requisitados pelos usuários ou podem ser acionadas pelos agentes tutores contidos em SATA.
- **Kernel** - Contém serviços que são fornecidos para as ferramentas e para os agentes tutores, incluindo o acesso a informações contidas nas bases de conhecimento. Também é responsável por intermediar a comunicação entre os agentes e as ferramentas. Essa comunicação ocorre pois ambos podem acionar o outro quando

necessário, por exemplo, se o agente tutor julgar necessário que o estudante adquira um maior conhecimento sobre um assunto, ele pode acionar uma ferramenta de recomendação de videoaulas.

- **Infra** - Gerencia o acesso às bases de conhecimento.
- **Knowledge Base** - Contém os modelos de domínio, pedagógico e do estudante.

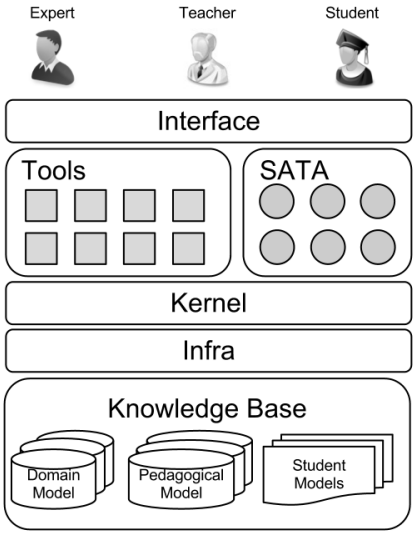

Figura 2. Arquitetura Conceitual do AIA.

2.3. Modelagem do Domínio

A modelagem do domínio de expressões aritméticas foi realizada utilizando uma estrutura de *curriculum*, sendo esta dotada de três tipos de entidades inter-relacionadas, a saber: tópico, problema e recurso, estando cada uma definida por uma estrutura de árvore. No caso de tópico, há uma estrutura de dados em árvore, onde cada nó contém um tópico do domínio que o estudante necessita aprender. Por sua vez, cada tópico possui uma ligação a um conjunto de problemas/exercícios e a um conjunto de recursos de suporte de conhecimento associados a ele, sendo que tanto problema quanto recurso, estão também representados por árvores. Os tópicos estão dispostos conforme uma relação de ordem e, assim, possuem um sequenciamento determinado pelo professor. Este sequenciamento será utilizado pelo ambiente, via módulo pedagógico, para guiar o estudante através da árvore de *curriculum* desde o momento em que ele inicia sua primeira interação até o momento em que tenha aprendido todos os assuntos do domínio e encerre o uso do ambiente.

2.4. Agente Tutor

Cada tópico da estrutura de *curriculum* é mapeado em um agente tutor correspondente. Um agente tutor baseia-se na arquitetura clássica de um STI, sendo ele formado por três módulos: módulo de domínio ou especialista, módulo pedagógico e o módulo estudante.

O módulo especialista contém o conhecimento do agente em um tópico específico do domínio, sendo composto por três sub-módulos: resolvedor, avaliador e diagnosticador. Cada um desses sub-módulos é composto por um sistema especialista tradicional, que contém uma base de conhecimento, um motor de inferência e um módulo de explicação.

Esses sub-módulos foram criados utilizando o framework *Inabit* [Rocha et al. 2012]. Este software faz a integração de um shell de sistema especialista baseado em regras e um framework para desenvolvimento de aplicações inteligentes. O sub-módulo resolvedor é responsável por solucionar problemas, neste caso, expressões aritméticas, referentes ao domínio. Por exemplo, um agente de multiplicação recebe problemas apenas do seu domínio e retorna a solução. O avaliador recebe uma solução e retorna um *feedback* sobre a mesma. Já o diagnosticador recebe uma solução incorreta e retorna os possíveis motivos de seus erros.

O módulo pedagógico é responsável pelo gerenciamento da interação entre o aluno e o ambiente. Baseado no sequenciamento dos tópicos do *curriculum* e nas informações do modelo do estudante, o agente tutor decide quais problemas o estudante deve responder e quando é o momento certo do estudante avançar para o próximo tópico (que é gerenciado por outro agente tutor).

O módulo estudante tem a função de alimentar e manter o modelo do estudante com informações do desempenho dos alunos na atividade de resolução de problemas. Cada estudante possui um modelo individual, este modelo é construído baseado na estrutura de *curriculum*, onde para cada tópico o aluno possui um conceito associado. As informações contidas no modelo são utilizadas pelo módulo pedagógico e por algumas ferramentas presentes na camada *Tools* (ver seção 2.2). Mais informações sobre o modelo do estudante podem ser vistas na seção 3.1.2.

3. Sistema de Suporte Inteligente

Tal como já mencionado, propõe-se neste artigo uma abordagem para provimento adapatativo de recursos, operacionalizada por meio de um sistema de suporte inteligente, visando ajudar um dado estudante em situação de dificuldade, diante do processo de resolução de problema. Tais dificuldades podem ser ocasionadas por diversos motivos, desde uma situação na qual o estudante apresenta dificuldade na interpretação do enunciado do problema até mesmo por ele temporariamente "esquecer"como o problema deve ser solucionado, ou ainda dúvida ou equívoco em alguma etapa da solução do problema. Essa constatação, sugere a necessidade da existência de mecanismos de suporte para auxiliar o aluno na atividade de resolução. Nesse sentido, alguns trabalhos na literatura abordam soluções inteligentes para essa necessidade através do fornecimento de dicas, tal como será discutido na seção 4.

O presente trabalho, em particular, busca utilizar múltiplos recursos de suporte para auxiliar o estudante na resolução de problemas. Os recursos atualmente disponíveis em nosso sistema de suporte são: um conjunto de dicas organizadas através de níveis, sendo tal conjunto também utilizado em outros trabalhos [Aleven and Koedinger 2000, Vanlehn 2006, Seffrin et al. 2012]), videoaulas recuperadas do *Youtube* e problemas com soluções prontas que possam ser utilizados para ajudar o estudante a compreender e solucionar seu problema atual.

O sistema de suporte pode ser acionado tanto pelo estudante, que pode pedir ajuda a qualquer momento durante a resolução, quanto pelo sistema, através de um agente em SATA. Na figura 3 são apresentadas algumas dos recursos apresentados na interface do sistema, em *(a)* é apresentada uma dica e em *(b)* é apresentada uma videoaula.

Dois termos são muito comuns no contexto de sistemas de fornecimento de

ajuda para o estudante na resolução de problemas, são eles: *help refuse* e *help abuse* [Aleven and Koedinger 2000, Roll et al. 2011]. O *help refuse* consiste na recusa do estudante em requisitar ajuda. Para combater esse empecilho o sistema possui a caracteristica de proatividade, ou seja, fornecerá suporte ao estudante sem que ele requisite. Neste caso, o sistema tutor fornecerá ajuda ao estudante em duas situações: quando o estudante cometer dois ou mais erros consecutivos em um passo específico da solução e quando o estudante obtiver uma quantidade total grande de erros acumulados na solução. Já o *help abuse* consiste no uso abusivo dos recursos de ajuda por parte do aluno. Para contorná-lo o sistema de suporte controla a quantidade de ajudas que o estudante pode pedir. Quando o estudante requisitar ajuda em excesso ou pedir uma ajuda em um passo da solução sem que tenha tentado solucioná-lo anteriormente, o sistema apresentará a seguinte mensagem: "*Tente um pouco mais antes de pedir uma ajuda*".

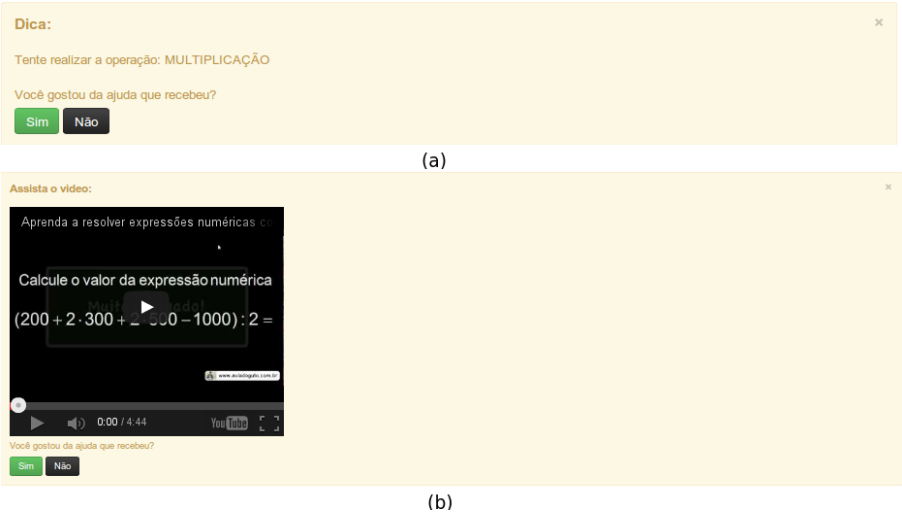

Figura 3. Recursos de suporte: (a) Dica textual e (b) Videoaula.

3.1. Fontes de Informação

O sistema utiliza três fontes de informação para determinar qual o recurso mais adequado para oferecer ao estudante em uma dada situação de dificuldade, são elas: a) informações do problema corrente e o estado da solução que o estudante está elaborando, b) o modelo do estudante e c) o tipo da dificuldade do aluno.

3.1.1. Informações do Problema e do Estado da Solução

Um problema é composto pelos seguintes atributos: enunciado, expressão aritmética, nível de dificuldade e o tópico do domínio que ele está associado. Um problema pode ter um conjunto de soluções associadas a ele, que são compostas por uma lista de passos formados por expressões aritméticas.

3.1.2. Modelo do Estudante

O modelo do estudante é construído a partir da estrutura de *curriculum*, tendo para cada tópico do domínio um conceito associado que reflete a opinião do sistema sobre o conheci-

Tabela 1. Dificuldades apresentadas pelos estudantes no questionário.

Tipo	Dificuldade	Alunos	Percentagem
1	*Ás vezes esqueço o modo como o exeercício deve ser resolvido*	28	56%
2	*Tenho dificuldades para interpretar o enunciado do exercício*	25	50%
3	*Costumo esquecer o assunto depois de algum tempo sem estudar*	16	32%
4	*Não estudei o suficiente para resolver os exercícios*	10	20%
5	*O exercício possui uma dificuldade maior do que posso resolver*	9	18%
6	*Estou desatento ou distraído na hora de resolver os exercícios*	5	10%

mento do estudante naquele tópico. No processo de resolução de problemas o sistema seleciona e apresenta um problema para que o estudante solucione, após receber o problema o estudante elabora a sua solução e a submete para o sistema. Em nosso trabalho o modelo de cada estudante é criado utilizando redes bayesianas que possuem suas probabilidades atualizadas com as avaliações das soluções submetidas pelo estudante [Costa et al. 2012].

3.1.3. Tipos da Dificuldade

Estudantes diferentes podem ter dificuldades distintas ao tentar solucionar o mesmo problema. Para tentar identificar as dificuldades dos estudantes na resolução de problemas matemáticos foi realizado um questionário com 50 alunos de ensino médio, no qual eles responderam às seguintes questões:

- *Você possui facilidade para resolver exercícios de matemática?*
- *Indique os motivos que levam você a ter alguma dificuldade na resolução de exercícios de matemática ou que levam você a desistir de solucionar o exercício.*

Dos alunos que participaram da pesquisa 39% declararam ter dificuldades ou não resolver exercícios de matemática com facilidade e 29% declararam ter facilidade para resolver alguns exercícios e dificuldades em outros. Na tabela 1 é apresentado o resultado da questão que pedia para que eles indicassem os motivos de suas dificuldades, ou seja, quais os motivos que levam o estudante a solucionar um exercício da forma errada ou até mesmo a desistir de resolvê-lo. As respostas obtidas dos estudantes foram agrupadas por suas semenlhanças gerando as seis respostas apresetadas na tabela 1, elas foram inseridas no sistema como tipos de dificuldade. Para tentar identificar o tipo de dificuldade que o estudante está tendo durante a resolução o sistema utiliza um conjunto de regras baseadas nos erros cometidos pelo estudante durante a elaboração da solução corrente e no modelo do estudante.

3.2. Seleção do Recurso de Suporte

Este mecanismo de seleção é utilizado tanto quando o aluno requisita ajuda do sistema, quanto quando o sistema detecta automaticamente a necessidade de ajuda. A seleção dos recursos de suporte é realizada utilizando três fatores: o estado corrente da solução do estudante, o modelo do estudante e o tipo da dificuldade do estudante na resolução. O modelo do estudante fornece um valor em porcentagem sobre o quanto o estudante domina um determinado tópico do domínio. Quando o estudante requisita ajuda o sistema de suporte verifica o quanto o estudante domina o tópico do problema corrente, caso o

estudante domine menos de 50% e já tenha cometidos mais de dois erros na solução o tutor irá recomendar uma video-aula para que o estudante melhore seu conhecimento do assunto antes de voltar a tentar solucionar o problema, isso ocorrerá porque o sistema irá considerar que o estudante está com dificuldades porque não possui conhecimento o suficiente sobre o assunto. As videoaulas também serão indicadas ao estudante quando o mesmo acumular excessivos erros consecutivos na solução o que é considerado pelo sistema como o estudante está praticando o método de "tentativa e erro".

Os recursos de dicas estão organizados em níveis tal como tratado em [Seffrin et al. 2012, Aleven and Koedinger 2000, Zhou et al. 1999]. Cada tópico do assunto possui um conjunto de dicas relacionadas que são instanciadas de acordo com as especificações do problema em questão. Cada nível possui um grau de especificidade que vão desde uma mensagem mais abstrata como "*Qual operação possui maior prioridade, adição ou multiplicação?*"até uma mais especifica, como, por exemplo, "*Tente efetuar primeiro a operação 3x15*".

O recurso de problemas análogos será utilizado quando o sistema detectar que o estudante está tendo dificuldades para solucionar um problema de um assunto que o estudante domina com sucesso e demonstrou ter habilidade para solucionar problemas semelhantes em outras ocasiões. Nesta situação o estudante pode estar com dificuldades por não ter conseguido interpretar o problema ou por, apesar de ter o domínio sobre o assunto, não "lembrar"o modo de resolução desse tipo de problema. Nestes casos, o sistema de suporte seleciona por meio de RBC um problema análogo ao corrente, a partir de uma base de casos contendo problemas já solucionados. O sistema apresenta o problema selecionado e a sua solução informando ao estudante que a solução do problema atual é semelhante a que está sendo apresentada. Essa abordagem foi utilizada por Santos [Santos et al. 2008] no contexto de um ambiente para o ensino de programação.

4. Trabalhos Relacionados

Mecanismos com o objetivo de prover ajuda ao estudante durante seu processo de aprendizado tem sido considerados por alguns trabalhos na literatura, alguns dos quais foram selecionados para aqui serem discutidos.

O trabalho prospoto por Seffrin [Seffrin et al. 2012] fornece um módulo de ajuda para auxiliar o estudante na resolução de equações no sistema tutor inteligente *PAT2Math*. Neste trabalho o recurso disponível para os estudantes são dicas textuais que estão organizadas através de níveis de abstração. Cada tópico de conhecimento do sistema tutor possui um conjunto finito de ajudas que possuem um grau de granularidade de informação diferente em cada nível. O sistema seleciona a dica do nível mais apropriado para ser apresentada dependendo do conhecimento do estudante na operação matemática em questão.

O trabalho apresentado por Aleven e Koedinger [Aleven and Koedinger 2000] mostra um STI com o objetivo de auxiliar no ensino geometria chamado de *PACT Geometry Tutor*. O sistema proposto neste trabalho oferece dois tipos de ajuda classificadas como ingeligentes e não inteligentes. As ajudas inteligentes consistem de dicas para solucionar o problema e dicas para correção de erros cometidos. As ajudas não inteligentes consistem em recomendar a leitura de um glossário de termo do domínio de geometria.

Analogus é um ambiente para o auxílio do ensino de programação para iniciantes [Santos et al. 2008, Santos et al. 2009] que possui um sistema de ajuda baseado em

Tabela 2. Comparação entre o trabalho proposto e os relacionados.

Abordagens	Domínio	Recursos Disponíveis	Critério de Seleção
PACT	Geometria	Dicas e Glossário	Nível da Ajuda e informações do uso do sistema
Analogus	Programção	Problemas Análogos	Seleção através de raciocínio baseado em casos
PAT2Math	Álgebra	Dicas	Conhecimento do aluno e nível da ajuda
Melo et al.	Variado	Artigos e *posts* de Blogs	Interações dos alunos com o ambiente
Nossa Proposta	Expressões Aritméticas	Dicas, videoaulas e problemas análogos	Conhecimento do estudante e tipo da dificuldade

raciocínio por analogia. Seu sistema de ajuda é constituido por um *chartterbot* que guia o estudante durante a elaboração de uma solução algorítmica na linguagem de programação *Python*. O sistema procura por um problema que o estudante já tenha solucionado que seja semelhante ao que ele está tentando resolver nesse instante e guia o estudante fazendo perguntas sobre a semelhança entre eles, a medida que o diálogo acontece o estudante tenta fazer ligações semânticas entre os problemas para que consiga solucionar o problema corrente.

Melo em [Melo et al. 2012] propõe um sistema integrado a um ambiente virtual de aprendizagem (AVA) que recomenda artigos e *post* de blogs para estudantes referentes ao assunto que o professor está abordando no ambiente. O sistema funciona integrado a um fórum que seleciona um dos dois recursos disponíveis baseado nas interações dos estudantes e na frequencia de acesso dos mesmos.

Uma das limitações das abordagens descritas consistem na ausência de múltiplos recursos de suporte para auxiliar o estudante no processo de resolução de problemas. O uso de múltiplos recursos é aqui assumido como um mecanismo importante para melhorar o processo de aprendizagem, principalmente porque é possível a utilização de recursos diferentes para suprir necessidades e dificuldades distintas de cada estudante. O trabalho proposto por Melo [Melo et al. 2012], apesar de utilizar dois tipos de recursos de suporte, não utiliza informações individuais do estudante para essa seleção, tendo os mesmos critérios para todos. A tabela 2 apresenta uma comparação entre o trabalho proposto e os relacionados.

5. Considerações Finais e Trabalhos Futuros

Apresentou-se neste artigo um estudo sobre mecanismos de assistência inteligente a estudantes diante de situações de dificuldades no processo de resolução de problemas no contexto de uso de um ambiente interativo de aprendizagem. Nesse sentido, foi proposto um sistema inteligente de suporte de recursos para apoiar estudantes envolvidos com resolução de problemas no domínio de expressões aritméticas. Este sistema se mostrou efetivo em suas escolhas a partir de testes realizados para verificar a qualidade das recomendações e da seleção dos recursos, representando um avanço sobre as propostas relacionadas, pois trouxe como principal diferencial e contribuição em sua abordagem, a utilização de múltiplos recursos de ajuda ao estudante contextualizados com seu estado

cognitivo e com suas dificuldades.

Como trabalho futuro imediato, necessita-se avaliá-lo mais com experimentos para aferir melhor e efetuar melhorias nos critérios de seleção do recursos. Particularmente, com tais experimentos melhorias significativas podem ser incorporadas ao módulo de raciocínio baseado em casos.

Referências

Aleven, V. and Koedinger, K. R. (2000). Limitations of student control: Do students know when they need help? In *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, ITS '00, pages 292–303, London, UK. Springer-Verlag.

Costa, E. d. B., Silva, P., Magalhães, J., and Silva, M. (2012). An open and inspectable learner modeling with a negotiation mechanism to solve cognitive conflicts in an intelligent tutoring system. In Herder, E., Yacef, K., Chen, L., and Weibelzahl, S., editors, *UMAP Workshops*, volume 872 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Melo, J., Ferreira, R., Costa, E., Brito, P., Pontes, J. P., and Freitas, F. (2012). Mining text from student-system interactions to recommend blogs and papers. *IEEE Technology and Engineering Education (ITEE)*, 7(3):1–12.

Rocha, R., Costa, E., Brito, P., Silva, M., Silva, P., and de Barros Paes, R. (2012). Improving construction and maintenance of agent-based applications through an integration of shell and software framework approaches. In *Encontro Nacional de Inteligência Artificial*.

Roll, I., Aleven, V., McLaren, B. M., and Koedinger, K. R. (2011). Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and Instruction*, 21(2):267–280.

Santos, G. P. d., Costa, E. d. B., and Fechine, J. M. (2008). Raciocínio baseado em casos para auxílio a alunos na resolução de problemas por analogia â uma abordagem para representação e recuperação de casos. In *XIX Simpósio Brasileiro de Informática na Educação (SBIE 08)*, Fortaleza, CE.

Santos, G. P. d., Fechine, J. M., and Costa, E. d. B. (2009). Analogus: Um ambiente para auxílio ao ensino de programação orientado pelo raciocínio por analogia. In *XXIX Congresso da Sociedade Brasileira de Computação (CSBC 2009)*, Bento GonÃ§alves, RS.

Seffrin, H., Rubi, G., Ghilardi, C., Morais, F., Jaques, P., Isotani, S., and Bittencourt, I. I. (2012). Dicas inteligentes no sistema tutor inteligente pat2math. In *XXIII Simpósio Brasileiro de Informática na Educação (SBIE 12)*, Rio da Janeiro, RJ.

Vanlehn, K. (2006). The behavior of tutoring systems. *Int. J. Artif. Intell. Ed.*, 16(3):227–265.

Zhou, Y., Freedman, R., Glass, M., Michael, J. A., Rovick, A. A., and Evens, M. W. (1999). Delivering hints in a dialogue-based intelligent tutoring system. In *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, AAAI '99/IAAI '99, pages 128–134, Menlo Park, CA, USA. American Association for Artificial Intelligence.