

Um Agente Pedagógico Animado de Apoio à Aprendizagem Baseada em Problema Utilizando o Moodle

Laysa Mabel de Oliveira Fontes¹, Francisco Milton Mendes Neto^{1,2}, Fábio A. Diniz¹, Danilo Gomes Carlos², Luiz Jácome Júnior¹, Luiz Cláudio N. da Silva²

¹Programa de Pós-Graduação em Ciência da Computação (PPgCC)
Universidade do Estado do Rio Grande do Norte (UERN)
Universidade Federal Rural do Semi-Árido (UFERSA)

²Laboratório de Engenharia de Software (LES) - UFERSA
BR 110 - Km 47 Bairro Pres. Costa e Silva - Mossoró, RN - Brasil

{laysa, miltonmendes}@ufersa.edu.br, {fabio.abrantes.diniz, danilogcarlos, luizjunior05}@gmail.com, luizclaudio@ufersa.edu.br

Resumo. *A aprendizagem baseada em problema (Problem-Based Learning - PBL) é uma teoria de aprendizagem que enfatiza a colaboração e o trabalho em grupo para resolução de um problema. No entanto, a implantação de um método de ensino com base na PBL não é uma tarefa trivial. Em ambientes virtuais de aprendizagem (AVAs), a complexidade de implantação deste método é ainda maior, pois o facilitador nem sempre pode detectar possíveis problemas na colaboração. Dessa forma, este trabalho apresenta uma arquitetura baseada em agentes de apoio à PBL, com o objetivo de detectar e corrigir problemas inerentes à implantação desta teoria de aprendizagem.*

Abstract. *The Problem-Based Learning (PBL) is a learning theory that emphasizes collaboration and teamwork to solve a problem. However, the implementation of a teaching method based on the PBL is not a trivial task. In virtual learning environments (VLEs), the complexity of implementation of this method is even greater, as the facilitator cannot always detect possible problems in the collaboration. Thus, this paper presents an agents-based architecture to support PBL, with the objective of detecting and correcting problems inherent in the implementation of this learning theory.*

1. Introdução

Segundo Hmelo-Silver (2004), a aprendizagem baseada em problema (*Problem-Based Learning - PBL*) é um método no qual os estudantes aprendem através da resolução de um problema que, em geral, não possui uma solução trivial e uma única solução correta. A aprendizagem é centrada no estudante e o conhecimento é adquirido de forma auto dirigida. Os estudantes trabalham em pequenos grupos colaborativos para identificar o que eles necessitam aprender para resolução do problema. O professor atua como facilitador do processo de aprendizagem ao invés de apenas transmitir conhecimentos.

Na literatura, existem muitas pesquisas que apresentam novas alternativas de ensino. Esses estudos buscam, além de um alcance maior para as pessoas que por algum

motivo não podem frequentar um ensino presencial, tornar esses ambientes mais eficazes [Jaques et al. 2002].

O conceito de agentes pedagógicos tem se mantido como um importante tema de pesquisa no âmbito educacional. Os agentes pedagógicos oferecem instrução personalizada, aumentam a motivação dos estudantes e agem pedagogicamente, por conta própria ou com o auxílio do facilitador. Por outro lado, ambientes virtuais de aprendizagem (AVAs) agregam valor ao processo educativo, gerando novas possibilidades de educação. Sendo assim, a combinação de agentes pedagógicos e AVAs consiste em uma abordagem promissora para o aprendizado eficaz auxiliado por computador [Soliman e Guetl 2010].

A PBL destaca o trabalho em equipe como um dos principais requisitos para o sucesso do processo de aprendizagem, ou seja, a colaboração é essencial [Savery 2006]. Contudo, a implantação de um método de ensino com base na PBL não é uma tarefa trivial. Em AVAs, a complexidade de implantação deste método é ainda maior, pois o facilitador nem sempre pode detectar possíveis problemas na colaboração, nem possui todas as informações necessárias para aplicar as técnicas de aprendizagem deste método.

Desta forma, este artigo apresenta uma arquitetura baseada em um agente pedagógico animado e outros quatro agentes de software para auxiliar na aplicação correta da teoria de aprendizagem PBL. Este trabalho apresenta uma abordagem para aperfeiçoar a aplicação da PBL nos seguintes aspectos: detecção de estudantes passivos, detecção de conversações fora do contexto, formação de grupos e recomendação de objetos de aprendizagem (OAs) sensível ao contexto do estudante. Esses agentes irão trabalhar colaborativamente com o objetivo de detectar e corrigir situações indesejadas, melhorando o processo de aprendizagem.

Este trabalho está organizado da seguinte forma: na Seção 2, são descritos os principais conceitos sobre a PBL; a Seção 3 apresenta trabalhos que utilizam agentes pedagógicos em AVAs; a Seção 4 descreve a arquitetura baseada em agentes apresentada neste trabalho; e, por fim, na Seção 5, são apresentadas as considerações finais e os trabalhos futuros.

2. Aprendizagem Baseada em Problema

A aprendizagem baseada em problema é uma teoria de aprendizagem na qual os estudantes aprendem através da resolução de um problema [Hmelo-Silver 2004]. Na PBL, o facilitador tem o papel de guiar os estudantes neste processo, identificando possíveis deficiências de conhecimento e habilidades necessárias à solução do problema proposto. Assim, neste método, ao invés de termos o professor simplesmente repassando os conhecimentos e depois testando-os através de avaliações, ele faz com que os estudantes apliquem o seu conhecimento em situações novas. Os estudantes se deparam com problemas muitas vezes mal estruturados e tentam descobrir, através da investigação e pesquisa, soluções úteis.

Para o sucesso da aplicação da PBL como estratégia pedagógica, os seguintes estágios devem ser cumpridos [Hmelo-Silver 2004]: i) o facilitador propõe um problema mal estruturado para o grupo de estudantes; ii) os estudantes tentam gerar fatos e identificar hipóteses sobre o problema, através de um *brainstorming* inicial. Os

estudantes também identificam, baseados no problema proposto, outros assuntos que servirão de base para a realização de pesquisas e, conseqüentemente, aquisição dos conhecimentos necessários para a resolução do problema; iii) os estudantes formulam e analisam o problema, objetivando gerar ideias para sua solução; iv) os estudantes, auxiliados pelo facilitador, identificam deficiências de conhecimento para solução do problema; v) os estudantes procuram por novos conhecimentos relacionados ao domínio e tentam gerar fatos sobre este novo conhecimento; vi) ao final de cada problema, os estudantes refletem sobre os conhecimentos adquiridos. A Figura 1 ilustra o ciclo de desenvolvimento da PBL.

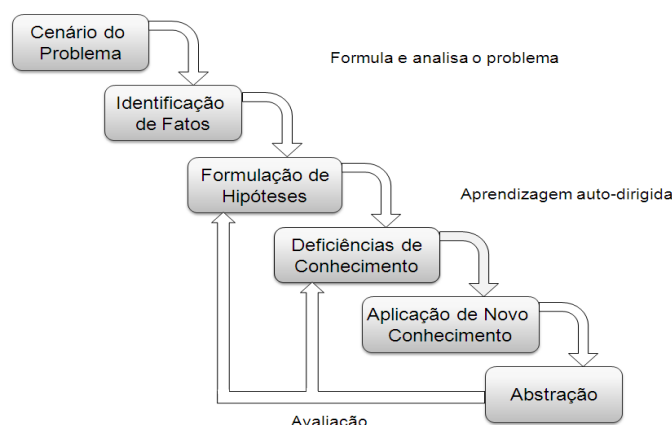


Figura 1. Ciclo da PBL

3. Uso de Agentes Pedagógicos Animados em AVAs

Alguns esforços têm sido realizados para criar agentes pedagógicos animados, mas a avaliação de seu impacto ainda é preliminar [Gresser et al. 2007]. Em [Silva e Bernardi 2009], os autores apresentam um agente pedagógico animado, chamado Cal, que foi desenvolvido com o objetivo de interagir afetivamente com o estudante, de modo a facilitar a relação ensino-aprendizagem, além de auxiliar o estudante na utilização do OA no qual o agente está inserido.

Em [Frozza et al. 2009], é apresentada a proposta de modelagem e desenvolvimento de um agente pedagógico, chamado Dóris. Esse agente consiste em um personagem *cartoon* 3D que expressa emoções na interação dos estudantes com um AVA. Os autores também apresentam os resultados iniciais sobre as primeiras expressões animadas do agente proposto.

Em [Arroyo, Woolf e Cooper 2011], são apresentados os resultados de uma avaliação, realizada com estudantes de ensino médio, do impacto da utilização de agentes pedagógicos integrados a um sistema de tutoria inteligente de matemática. Os resultados apresentados pelos autores indicaram que agentes pedagógicos melhoraram os aspectos afetivos dos estudantes em geral, mas tendo um impacto maior com os estudantes do sexo feminino.

Como diferencial do nosso trabalho, podemos destacar que, diferentemente dos outros trabalhos discutidos nessa seção, o presente trabalho é voltado especificamente para aplicação da PBL, ou seja, apresenta um agente pedagógico animado e outros quatro agentes de software para auxiliar na aplicação correta da PBL, uma teoria de aprendizagem comprovadamente eficaz [Strobel e Van Barneveld 2009].

4. Arquitetura Baseada em Agentes de Apoio à PBL

A arquitetura baseada em agentes de apoio à PBL apresentada neste artigo está esquematizada na Figura 2.

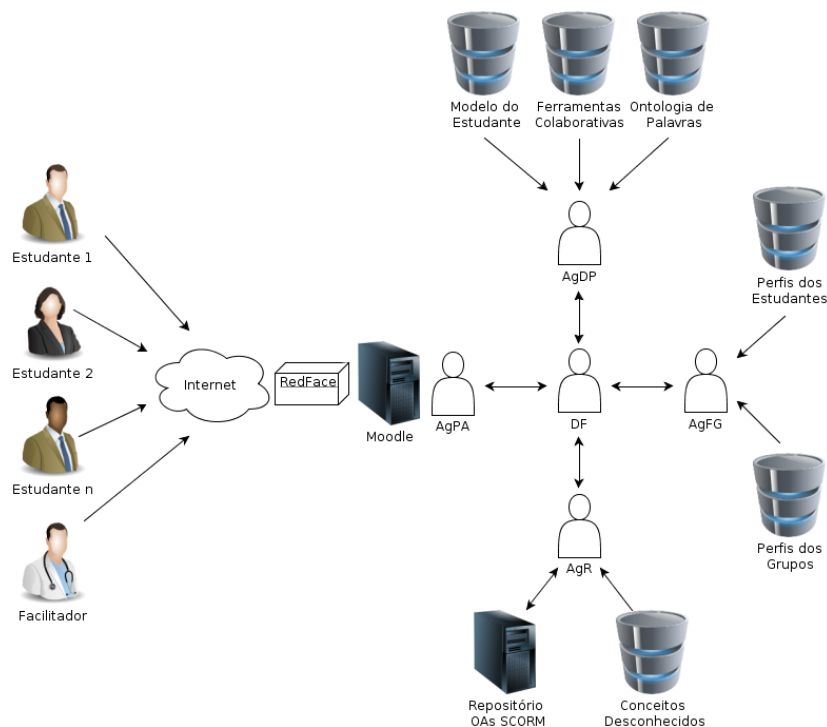


Figura 2. Arquitetura de apoio à PBL

Conforme ilustrado na Figura 2, a arquitetura apresentada nesse trabalho é composta de cinco tipos de agentes: Agente Pedagógico Animado (AgPA), Agente Detector de Problemas (AgDP), Agente Formador de Grupos (AgFG), Agente Recomendador (AgR) e Agente DF (*Directory Facilitator*).

Os agentes presentes na arquitetura foram implementados utilizando o JADE (*Java Agent Development Framework*) [JADE 2012]. Esses agentes serão descritos em mais detalhes nas próximas subseções.

4.1. Agente Pedagógico Animado - AgPA

O AgPA foi implementado com o intuito de apoiar os estudantes na resolução de problemas, através da teoria de aprendizagem PBL. O AgPA consiste em um modelo humanoide tridimensional animado responsável por acompanhar os estudantes durante o processo de aplicação da PBL, além de manter os estudantes sempre motivados. Para obter sucesso nesse último caso, o AgPA expressará emoções similares às dos seres humanos, conforme ilustrado na Figura 3 (a).

Até a escrita deste trabalho, foram modeladas e implementadas quatro animações para expressar as emoções do AgPA, conforme ilustrado na Figura 3 (a): felicidade, que remete, por exemplo, momentos em que o estudante esteja interagindo com o ambiente; tristeza, quando algum problema de colaboração for detectado, como, por exemplo, a detecção de estudantes passivos; expectativa, durante os questionamentos do AgPA para o estudante; e dúvida, quando o estudante permanecer muito tempo sem interagir com o ambiente.

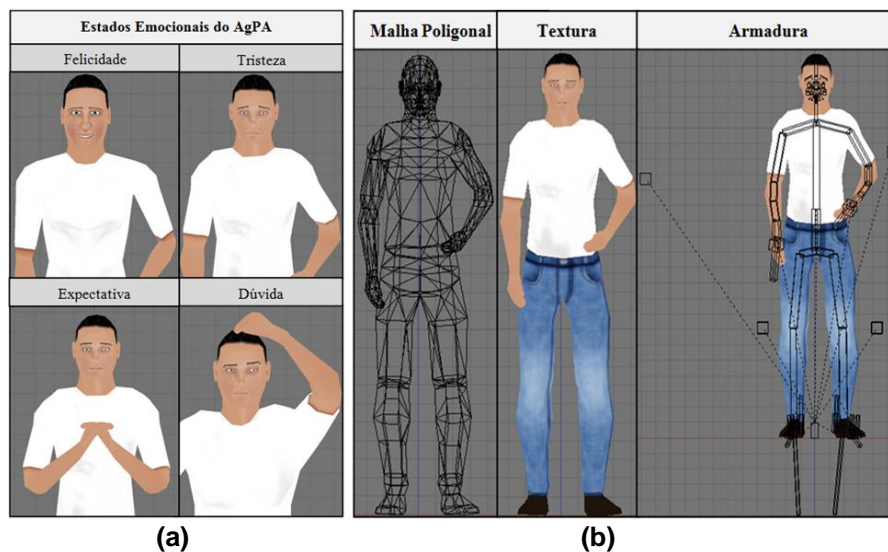


Figura 3. (a) Animações referentes aos estados emocionais do AgPA. (b) Malha poligonal, textura e armadura do AgPA

O AgPA se mantém ativo durante toda a permanência do estudante no ambiente. Ele se comunica com os outros agentes, de forma colaborativa, e atua de acordo com o que for constatado no ambiente. O AgPA possui um comportamento do tipo *Cyclic Behaviour* para ficar monitorando e tratando as requisições dos agentes AgDP e AgR. A seguir apresentamos dois cenários de atuação do AgPA:

- Uma vez que o AgDP tenha detectado um estudante passivo ou uma conversação fora do contexto, o AgPA será acionado para tentar motivar este estudante. Para obter sucesso neste caso, o AgPA expressará emoções que remetem tristeza e exibirá mensagens textuais como forma de instigar este estudante a interagir mais com o ambiente;
- Durante a fase de identificação dos fatos, o AgR tentará encontrar OAs sensíveis ao contexto do estudante, como forma de sanar possíveis dúvidas durante esta fase. Uma vez que o AgR tenha detectado OAs adequados ao contexto do estudante, este irá acionar o AgPA. O AgPA, por sua vez, irá se comunicar com o estudante, através de mensagens textuais, avisando-o sobre a existência de tais OAs. Neste caso, o AgPA expressará emoções que remetem felicidade.

Para a modelagem e animação do AgPA foi utilizada a ferramenta Blender [Brito 2010]. Na modelagem da forma do agente, foi utilizada a técnica de malha poligonal [Funk e Aymone 2010], conforme ilustrada na Figura 3 (b). Essa malha poligonal foi produzida utilizando uma *Blueprint* [Funk e Aymone 2010]. Durante a modelagem da malha, foi empregada a técnica de modelagem denominada *Low Poly* [Totten 2012].

A confecção da textura foi feita utilizando a ferramenta Gimp [Kylander e Kylander 1999] (ilustrada na Figura 3 (b)), tendo como referência a representação 2D do modelo gerado a partir da técnica de mapeamento UV [Brito 2010]. O controle das poses foi feito utilizando o modificador *Armature* (armadura) [Brito 2010] (ilustrado na Figura 3 (b)). As animações foram realizadas utilizando as técnicas *Key Frames* [Brito 2010]. Para renderizar o AgPA, foi utilizado o motor de jogos *jMonkey* [Donglai, Gouxi e Qiuxiang 2010]. Para isso, foi necessário adaptar e exportar o modelo do Blender para o formato do *Ogre3D* [Junker 2006].

4.2. Agente Detector de Problemas - AgDP

O AgDP possui dois comportamentos: detecção de estudantes passivos e detecção de conversações fora do contexto do problema. Esses dois comportamentos são do tipo *Ticker Behaviour*, ou seja, são executados a cada determinado período de tempo. Neste caso, o AgDP executa estes comportamentos uma vez por dia, durante toda a realização de um curso. Esse agente foi criado com o intuito de auxiliar o facilitador na avaliação do comportamento dos estudantes durante o processo de aplicação da PBL. Desta forma, uma vez sendo detectado um comportamento indesejado, o AgDP irá notificar o facilitador, e esse, por sua vez, poderá tomar uma providência cabível. As subseções a seguir descrevem esses comportamentos em mais detalhes.

4.2.1. Detecção de Estudantes Passivos

O AgDP tem a função de detectar os estudantes passivos. Para que o AgDP consiga cumprir com esta meta é necessário que ele atualize os perfis dos estudantes conforme o uso das ferramentas colaborativas disponíveis no ambiente. Nesta abordagem, para cada ação executada pelo estudante no ambiente, este é pontuado com base em uma tabela cujos valores são previamente definidos.

Esta informação será usada posteriormente pelo AgDP para realizar a detecção dos estudantes passivos propriamente dita. Os passos do processo de detecção de estudantes passivos executados pelo AgDP são: (i) inicialmente é calculada a média de participação dos estudantes (conforme o uso das ferramentas colaborativas), eliminando os valores discrepantes (*outliers*); (ii) a detecção de valores discrepantes é realizada calculando-se a mediana dos valores da série (conjunto de pontos de um grupo de estudantes). A discrepância é eliminada de acordo com um limiar pré-estabelecido em relação à mediana dos valores; (iii) em seguida, é calculada a média aritmética dos valores restantes, refletindo melhor a tendência da série; e (iv) com base na média dos valores restantes, é possível detectar um estudante passivo que diste do limiar pré-estabelecido pelo facilitador.

Após detectar um estudante passivo, o AgDP notifica o facilitador, via e-mail, informando todas as informações inerentes ao referente estudante. Em seguida, todas as informações relativas ao estudante passivo são enviadas para o AgPA.

4.2.2. Detecção de Conversações Fora do Contexto

O AgDP também é responsável por detectar as conversações fora de contexto baseado no uso das ferramentas colaborativas disponíveis no ambiente e em uma ontologia do problema que está sendo resolvido.

O AgDP monitora constantemente as ferramentas usadas pelos estudantes para cooperação e comunicação durante a resolução do problema. Então, o AgDP compara as palavras usadas pelos estudantes nas suas interações com um conjunto de palavras previamente instanciadas na ontologia.

Em seguida, o AgDP calcula a porcentagem de palavras fora do contexto do problema usadas pelo estudante nas ferramentas disponíveis no ambiente. Isso é útil para identificar o nível de fuga do contexto do estudante em relação aos assuntos relacionados ao problema em discussão. Este nível pode ser obtido pela expressão $NF =$

$PNR/(PR-\Delta)$, onde NF = nível de fuga; PNR = quantidade de palavras não relacionadas; PR = quantidade de palavras relacionadas e Δ é um fator que o facilitador pode gerenciar para aumentar ou diminuir o impacto de palavras não relacionadas.

Uma vez detectada uma conversação fora do contexto, caso o NF possua um valor superior ao definido pelo facilitador, o AgDP envia uma mensagem automaticamente para o facilitador, via e-mail, notificando todas as informações inerentes aos estudantes dispersos. Em seguida, todas as informações inerentes aos estudantes dispersos são enviadas para o AgPA.

4.3. Agente Formador de Grupos - AgFG

O processo de formação de grupos é efetuado da seguinte forma: os estudantes, através de uma interface Web, preenchem seu perfil, que alimenta uma base de perfis que será usada no processo de formação de grupos. O perfil dos estudantes é composto por habilidades, conhecimentos e deficiências, onde cada um possui um nível, que pode ser baixo, médio ou alto, podendo um estudante ter uma ou mais habilidades, deficiências e conhecimentos.

Por outro lado, o facilitador preenche os perfis dos grupos desejados para cada problema a ser solucionado através de uma interface Web, da mesma forma que o estudante. Um perfil de grupo é composto por habilidades, conhecimentos e deficiências, cada um possuindo um nível, que pode ser baixo, médio ou alto, bem como um valor *fuzzy*, que varia de 0.1 a 1.0 e está vinculado aos valores baixo, médio e alto. Após o facilitador construir os perfis dos grupos, haverá uma base de perfis de grupos que poderá ser consultada pelo AgFG no processo de formação de grupos. É importante salientar que o perfil desejado construído pelo facilitador é o que melhor se adéqua à resolução do problema; assim, um estudante que tenha um perfil aproximado ao desejado terá as competências necessárias à resolução do problema proposto.

O AgFG é o responsável pela formação automática dos grupos e possui dois comportamentos, um implementado em Java [Veronese et al. 2002] e outro em Prolog [Gomes et al. 2002]. O comportamento do AgFG implementado em Java é responsável pela geração de candidatos que estão aptos a participar de determinado grupo. Esse processo é feito analisando os perfis dos estudantes e os perfis dos grupos. Após essa análise, ele gera um arquivo, que será o arquivo de entrada para o comportamento do AgFG implementado em Prolog.

O comportamento do AgFG em Prolog é responsável pela alocação dos estudantes aos grupos propriamente dita. No final desse processo é gerado um arquivo que contém os *rankings* para formação de grupos. O facilitador analisará este resultado, que é exibido através de uma página Web, e decidirá se acata ou não a sugestão do AgFG. Os dois comportamentos do AgFG são do tipo *One Shot Behaviour*, ou seja, de execução única. O AgFG é acionado a cada novo perfil do grupo criado.

4.4. Agente Recomendador - AgR

O AgR tem o intuito de detectar OAs adequados ao contexto do estudante. O AgR encontra, inicialmente, o OA que seria mais adequado de acordo com a recomendação baseada em conteúdo.

O mecanismo de recomendação baseada em conteúdo considera as informações de horário preferido de estudo e área de interesse do estudante, as quais estão contidas na ontologia de contexto estático. Essas informações são ponderadas de acordo com o nível de influência que cada uma exerce no modelo de aprendizagem do estudante. A estratégia para identificar os OAs adequados ao contexto do estudante é realizada de acordo com a Equação 1.

$$FR = ((AI * 5) + (CD * 3) + (HP * 2)) / 10 - FA \quad (1)$$

O Fator de Recomendação (*FR*), o qual é determinado pelo AgR, é influenciado, principalmente, pela Área de Interesse (*AI*) do estudante, tendo, portanto, peso 5. Os Conceitos Desconhecidos (*CD*) por possuir grande contribuição para o cálculo do fator de recomendação, recebe peso 3, uma vez que identifica pontos de dificuldade do estudante. Por fim, o Horário Preferido (*HP*) de estudo também é de interesse pelo fato de poder exercer influência no nível de concentração e, conseqüentemente, influenciar positiva ou negativamente a recomendação de um OA. Já o Fator de Ajuste (*FA*) diz respeito a um fator que pode ser estabelecido pelo facilitador a fim de aumentar (quando o *FA* for menor) ou diminuir (*FA* maior) o impacto que o contexto do estudante exerce para a recomendação de OAs. Os valores numéricos de *AI* e *HP* são obtidos com base nos valores capturados dinamicamente e naqueles previamente definidos na ontologia de contexto estático dos estudantes. O valor de *CD* é definido considerando informações fornecidas pelo estudante e as informações contidas nos OAs.

Para definir, de forma dinâmica, o valor que representa o quão adequado determinado OA é em relação à área de interesse de um estudante, são consideradas três características do OA: descrição, título e palavras-chave. O AgR, por sua vez, verifica a incidência de palavras de interesse do estudante, contidas na ontologia de contexto estático, nessas três características do OA. Em seguida, esses valores são ponderados pelo AgR conforme a Equação 2.

$$AI = ((PC * 3) + (D * 2) + (T * 1)) / 6 \quad (2)$$

A Equação 2 mostra que o maior peso é dado às Palavras-Chave (*PC*), visto que representam melhor os assuntos tratados no OA. A Descrição (*D*) do OA nos fornece uma visão geral de como os seus diversos assuntos estão integrados. Por fim, o Título (*T*) representa uma influência menor, dentre as três características, por não conter uma gama de palavras relacionadas ao OA tão abrangente quanto às *PCs*.

O valor de *CD* é definido a partir da incidência dos conceitos desconhecidos na descrição, no título e nas palavras-chave do OA. A ponderação ocorre conforme definido na Equação 2, sendo cada variável a representação da incidência dos conceitos desconhecidos. Esses conceitos desconhecidos são fornecidos pelo próprio estudante, via interface do Moodle, durante a aplicação da PBL. Tal característica irá auxiliar os estudantes durante a fase de identificação dos fatos da PBL, conforme apresentado na Seção 2. Por último, a captura dinâmica do horário atual é feita também pelo AgR no momento de autenticação do estudante no sistema. Essa informação servirá para que o AgR defina o valor numérico atribuído ao fator *HP*. Após identificar OAs adequados, o AgR informa ao AgPA quais são os possíveis OAs a serem sugeridos. Este comportamento é do tipo *One Shot Behaviour*. O AgR é acionado durante a fase de identificação dos fatos.

4.5. Directory Facilitator (DF)

O papel de mediador da comunicação entre os agentes é realizado pelo agente DF, o qual é provido pela própria plataforma JADE, conforme exigência da especificação FIPA (*Foundation for Intelligent Physical Agents*) [FIPA 2012]. Na arquitetura multiagente proposta, foi necessário apenas codificar a forma como os agentes criados se comunicam com o DF.

4.6. Avaliação dos Estudantes

Este trabalho também apresenta um módulo de avaliação dos estudantes. Este módulo é composto por duas interfaces: uma para o facilitador e outra para o estudante.

A interface para o facilitador consiste em um formulário com vários critérios de avaliação, como, por exemplo, pontualidade, participação, etc. Nesse mesmo formulário também existem alguns critérios de avaliação direcionados especificamente para o coordenador e o relator da sessão, como, por exemplo, condução e distribuição de tarefas para o coordenador, organização do relatório da sessão e pontualidade na entrega para o relator. Essa avaliação é realizada a cada nova sessão concluída. Vale ressaltar que, ao final de cada sessão, os estudantes realizam uma autoavaliação e uma avaliação dos outros membros do grupo, chamada também de avaliação de pares. Essas avaliações são enviadas para o facilitador de forma que o auxiliará na avaliação dos membros do grupo.

Na interface do estudante, será exibido o mesmo formulário com os critérios de avaliação, porém com algumas peculiaridades. Cada critério de avaliação será exibido em uma cor que condiz com a avaliação dada pelo facilitador. A cor verde indica que o estudante recebeu uma boa pontuação para este critério. A cor amarela indica uma pontuação regular e a cor vermelha indica que o estudante recebeu uma baixa pontuação no referido critério. Desta forma, ao visualizar suas pontuações, os estudantes irão se conscientizar em quais pontos eles precisam melhorar para atingir boas pontuações.

4.7. Sistema de Reconhecimento Facial

Os AVAs, em geral, adotam como prática de segurança a utilização de mecanismos de autenticação por *login* e senha. O uso deste tipo simples de autenticação aumenta a vulnerabilidade a fraudes, tanto no acesso ao sistema quanto durante a participação do estudante nas atividades do curso. Assim, a não presença dos estudantes torna a fraude fácil e tentadora, pois outra pessoa pode substituir facilmente o estudante que deveria ser avaliado.

Por outro lado, estudos têm sido conduzidos sobre possíveis aplicações da biometria para autenticação em ambientes Webs [Rolim e Bezerra 2008]. Baseado nisso, este trabalho propõe um módulo de reconhecimento facial, denominado de RedFace, que adiciona a funcionalidade de autenticação biométrica tanto no acesso ao sistema quanto de forma contínua, durante a realização do curso.

Diferentes técnicas foram utilizadas para a construção do módulo RedFace. Utilizou as técnicas de Análise de Componentes Principais (PCA) [Stan e Anil 2012] e *Eigenfaces* [Stan e Anil 2012] no processo de extração de características da face. Na classificação das imagens, foi usado o classificador *K-Nearest Neighbors* (K-NN)

[Jiangsheng 2002]. O sistema de reconhecimento facial proposto consiste basicamente das seguintes etapas [Stan e Anil 2012]:

- Aquisição das imagens: através da *webcam*, o sistema captura a imagem da face do estudante, que será utilizada como entrada para o módulo RedFace;
- Pré-processamento: nesta etapa, as imagens são normalizadas e corrigidas para melhorar o reconhecimento da face;
- Extração de características: devido à alta dimensionalidade dos vetores, foi utilizada a técnica de PCA, juntamente com a técnica *Eigenfaces*;
- Classificação e verificação: nesta etapa, foi utilizado o algoritmo de reconhecimento de padrão K-NN. A validação do algoritmo foi realizada em uma base de dados contendo 1.280 imagens de 64 classes distintas. Ficou notório que o desempenho do classificador, testado para sistemas de reconhecimentos de face baseado em PCA, foi muito satisfatório, atingindo uma taxa de reconhecimento acima de 90% em condições ideais, com baixos tempos de execução e com pequena quantidade de informação trafegada entre cliente e servidor, com imagens em torno de 30KB.

O RedFace foi proposto com o intuito de permitir que os administradores, os coordenadores e, principalmente, os facilitadores tenham uma certificação da identidade dos estudantes durante a realização de suas atividades e avaliações no processo da PBL.

5. Considerações Finais e Trabalhos Futuros

Neste artigo, foi descrita uma arquitetura baseada em um agente pedagógico animado e outros quatro agentes de software para auxiliar na aplicação correta da teoria de aprendizagem PBL.

Este trabalho apresentou uma abordagem para aperfeiçoar a aplicação da PBL nos seguintes aspectos: detecção de estudantes passivos, detecção de conversações fora do contexto, formação de grupos e recomendação de OAs sensível ao contexto do estudante. Este trabalho também propôs um módulo de reconhecimento facial, o RedFace, com o intuito de permitir uma certificação da identidade dos estudantes durante a realização de suas atividades e avaliações no processo da PBL.

Como trabalhos futuros, pretende-se abordar outras metas relacionadas ao auxílio no cumprimento da aplicação da PBL, conforme apresentado em [Pontes 2010]. Por fim, objetiva-se realizar um estudo de caso como forma de validar a eficácia da solução apresentada neste trabalho. Esse estudo de caso está integrado ao projeto intitulado “Um Agente Pedagógico 3D de Apoio a Estudantes de Medicina na Resolução de Problemas na Área da Oncologia Utilizando a Aprendizagem Baseada em Problema” aprovado pelo COHM (Centro de Oncologia e Hematologia de Mossoró). Pretende-se realizar o estudo de caso com estudantes do curso de medicina, sendo essa validação apoiada pelo COHM. Nessa ocasião, os estudantes utilizarão a ferramenta, com a solução integrada, para resolverem problemas relacionados à área de Oncologia. Dessa forma, será possível avaliar a solução proposta e verificar se a ferramenta provê um auxílio eficaz aos estudantes do curso de medicina na resolução de problemas da área de oncologia.

Referências

- Arroyo, I., Woolf, B. P. e Cooper, D. G. (2011) “The Impact of Animated Pedagogical Agents on Girls’ and Boys’ Emotions, Attitudes, Behaviors and Learning”, In: IEEE International Conference on Advanced Learning Technologies, Athens, p. 506-510.
- Brito, A. (2010), Blender 3D: Guia do Usuário, Novatec, 4ª edição.
- Donglai, F., Gouxi, C. e Qiuxiang, Y. (2010) “A Robust Software Watermarking for jMonkey Engine Programs”, In: International Forum on Information Technology and Applications (IFITA), p. 421-424.
- Frozza, R., Silva, A. K., Lux, B., Cruz, M. E. J. K. e Borin, M. (2009) “Dóris 3D: Agente Pedagógico baseado em Emoções”, In: Anais do Simpósio Brasileiro de Informática na Educação (SBIE).
- Funk, S. e Aymone, J. L. F. (2010). Proposta de Diretrizes para o Processo Criativo do Design Virtual de Embalagens. In *Design & Tecnologia*, v. 1, n. 2, p. 55-68.
- Gomes, A. K., Bernardini, F. C., Monard, M. C. e Batista, G. E. A. P. A. (2002) “Uma Sintaxe Padrão Prolog para Classificadores Simbólicos”, In: Relatórios Técnicos do ICMC.
- Gresser, A., Chipman, P., King, B., McDaniel, B. e D'Mello, S. (2007) “Emotions and Learning with AutoTutor”, In: Proceedings of Artificial Intelligence in Education, p. 569-571.
- Hmelo-Silver, C. E. (2004). Problem-Based Learning: What and How Do Students Learn?. In *Educational Psychology Review*, v. 16, n. 3, p. 235–266.
- Jaques, P. A., Andrade, A., Jung, J., Bordini, R. e Vicari, R. (2002) “Using pedagogical agents to support collaborative distance learning”, In: Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community, p. 546–547.
- Java development framework: an open-source platform for peer-to-peer agent based applications, JADE (2012). Disponível em: <<http://jade.tilab.com/>>. Acesso em: 24 Jul. 2012.
- Jiangsheng, Y. (2002) “Method of k-Nearest Neighbors”, In: Institute of Computational Linguistics, Peking University, China.
- Junker, G. (2006), Pro OGRE 3D Programming, Apress.
- Kumar, S., Gankotiya, A. K. e Dutta, K. (2011) “A Comparative Study of Moodle with other e-Learning Systems”, In: International Conference on Electronics Computer Technology - ICECT, 3. Kanyakumari: IEEE.
- Kylander, K. e Kylander, O. S. (1999), GIMP User's Manual: The Complete Guide to Gimp, Coriolis Group.
- Pontes, A. A. A. (2010) “Uma Arquitetura de Agentes para Suporte à Colaboração na Aprendizagem Baseada em Problemas em Ambientes Virtuais de Aprendizagem”, Dissertação de Mestrado, Programa de Pós-Graduação em Ciência da Computação, Universidade Federal Rural do Semiárido (UFERSA), Mossoró, Brasil.

- Rolim, A. L. e Bezerra, E. P. (2008) “Um sistema de identificação automática de faces para um ambiente virtual de ensino e aprendizagem”, In: Simpósio Brasileiro de Sistemas Multimídia e Web.
- Savery, J. R. (2006). Overview of Problem-based Learning: Definitions and Distinctions. In *The Interdisciplinary Journal of Problem-based Learning*, v. 1, n. 1, p. 9-20.
- Silva, T. G. e Bernardi, G. (2009) “Cal: um Agente Pedagógico Animado para Apoio em um Objeto de Aprendizagem para o Ensino de Matemática”, In: Anais do Simpósio Brasileiro de Informática na Educação (SBIE).
- Soliman, M. e Guetl, C. (2010) “Intelligent Pedagogical Agents in Immersive Virtual Learning Environments: A Review”, In: Proceedings of the International Convention (MIPRO), Opatija, Croatia, p. 827-832.
- Stan, Z. L. e Anil, K. J. (2012), Handbook of Face Recognition, Springer, 2ª edição.
- Strobel, J. e Van Barneveld, A. (2009). When is PBL More Effective? A Meta-synthesis of Meta-analyses Comparing PBL to Conventional Classrooms. In *Interdisciplinary Journal of Problem-based Learning*, v. 3, n. 1.
- Totten, C. (2012), Game Character Creation with Blender and Unity, Sybex.
- Veronese, G., Correa, A., Werner, C. e Jezinni, F. (2002) “ARES: Uma Ferramenta de Engenharia Reversa Java-UML”, In: Simpósio Brasileiro de Engenharia de Software (SBES), Sessão de Ferramentas, p. 347–352.
- Welcome to the Foundation for Intelligent Physical Agents, FIPA. (2012) “Site Oficial do Padrão FIPA”. Disponível em: <<http://www.fipa.org/>>. Acesso em: 24 Jul. 2012.