

UM AGENTE ANIMADO PARA AMBIENTES DE APRENDIZAGEM COLABORATIVOS

¹ **Eliseo Berni Reategui**
ebreateg@ucs.br

¹ **Rafael Fernando Ceron**
rfceron@ucs.br

^{1,2} **Elisa Boff**
eboff@ucs.br

² **Rosa Maria Vicari**
rosa@inf.ufrgs.br

¹ Departamento de Informática
Universidade de Caxias do Sul (UCS)
Rua Francisco Getúlio Vargas, 1130
95070-560 - Caxias do Sul - RS - Brasil

² Instituto de Informática - UFRGS
Bloco IV - Av. Bento Gonçalves, 9500
Porto Alegre - RS

Resumo: Este artigo detalha a modelagem e implementação de um agente de interface sócio-afetivo capaz de interagir com alunos de um ambiente de aprendizagem colaborativa de algoritmos. O agente monitora a interação com o usuário e utiliza as informações coletadas para fazer recomendações personalizadas de conteúdo. Além disso, o agente é capaz de identificar outros estudantes conectados no sistema que podem servir como tutores em determinados tópicos, e sugere a interação entre estes e alunos com dificuldades. Através destas recomendações, o agente motiva a formação de grupos e promove a aprendizagem colaborativa.

Palavras-Chave: Agentes Animados de Interface, Ambientes de Aprendizagem, Sistemas de recomendação, Afetividade, Interação Social e Formação de Grupos

Abstract: This paper describes a social-affective animated agent that is able to interact with students in a collaborative learning environment for algorithms. The agent monitors the interaction with the user and employs the information collected to make personalized recommendations of contents. In addition, the agent is capable of identifying other students that may act as tutors in certain topics, and suggests the interaction between them and the students that are having difficulties in their studies. Through these recommendations, the agent motivates group formation and promotes collaborative learning.

Keywords: Interface Agents, Learning Environments, Recommender Systems, Affective, Social Interaction and Workgroups.

1. INTRODUÇÃO

Ao longo dos anos, vários pesquisadores, tanto da Psicologia quanto da Educação, têm levantado evidências acerca da importância da socialização na construção do conhecimento. Neste grupo podemos incluir Vygotsky [23] e Maturana [14]. Piaget [15] também não desconheceu a importância das trocas sociais para o desenvolvimento cognitivo, apesar de tê-la abstraído na grande maioria dos seus estudos [4]. Esse autor

aponta o conhecimento humano como sendo essencialmente coletivo e diz que a vida social constitui um dos fatores essenciais da formação e do crescimento dos conhecimentos pré-científicos e científicos [4].

As trocas sociais são enfatizadas no processo de construção do conhecimento, mesmo sabendo-se que a aprendizagem ocorre individualmente. Estas atividades coletivas de aprendizagem normalmente são divididas em aprendizagem cooperativa ou colaborativa, existindo

diferentes conceitos para estes termos. Na literatura, podem-se encontrar autores que consideram ambos com o mesmo significado, outros que diferenciam tais termos. A aprendizagem colaborativa ocorre como resultado do compartilhamento do conhecimento entre múltiplos participantes. Para tanto, aparece a tecnologia computacional que apóia e organiza as situações, facilitando este processo. Um processo de aprendizagem nesses moldes, baseado na coletividade, difere-se dos tradicionais pelo fato de apresentar ausência de hierarquia formal, possuir um objetivo em comum, respeitar as diferenças individuais e possibilitar a liberdade de expressão na exposição de idéias e questionamentos.

Pelas idéias expostas acima, podemos afirmar que se a socialização é importante no contexto da aprendizagem, também o é em ambientes informatizados que visem a facilitação deste processo, tais como os Sistemas Tutores Inteligentes (STI) e os Ambientes Inteligentes de Aprendizagem (AIA). Observa-se nos principais meios de divulgação de avanços científicos uma crescente utilização de técnicas de Inteligência Artificial para adequar sistemas computacionais às necessidades individuais dos usuários, ou seja, na individualização do processo de aprendizagem.

Baseado nos conceitos já apresentados e ressaltando a idéia de que a interação social é fator importante para o desenvolvimento do conhecimento, este trabalho propõe um agente animado sócio-afetivo, que ancora seu processo de decisão em características individuais do aluno, como humor e desempenho (*performance*), e em aspectos sociais, como a aceitação do aluno pelo grupo e sua habilidade social. Informações individuais e do grupo são armazenadas e empregadas por um sistema de recomendação para sugerir conteúdos de algoritmos e recomendar a interação entre os alunos, a fim de que os próprios alunos assumam o papel de tutores. O objetivo é tentar amenizar as dificuldades pedagógicas que possam acontecer com algum estudante no decorrer de uma sessão de ensino, através da recomendação de alunos potencialmente capacitados e amigáveis e, a partir daí, proceder a formação de grupo entre o aluno com dificuldades e o “colega tutor”. À medida que esta dinâmica vai se estabelecendo, novos grupos vão se formando. Com o passar do tempo, as recomendações do agente animado se tornam mais adequadas, visto que o perfil do aluno consultado para gerá-las vai sendo atualizado a cada interação com o sistema.

Este artigo está organizado da seguinte forma: a seção 2 aborda os trabalhos relacionados que influenciaram a modelagem do agente animado proposto, a seção 3 apresenta o modelo para o agente animado e o Ambiente para Aprendizagem de Algoritmos 3A, nosso estudo de caso. A seção 4 detalha dois experimentos em que o ambiente foi utilizado com alunos. Na seção 5 são apresentadas as considerações finais e possibilidades para trabalhos futuros.

2. TRABALHOS RELACIONADOS

A área de agentes de interface é voltada para o desenvolvimento de aplicações que visam aperfeiçoar o processo de interação entre humanos e computadores através da utilização de personagens que podem ser animados ou não.

Cosmo [13] e Adele [22] são alguns exemplos clássicos desses agentes. Cosmo é um agente voltado para área da informática, mais especificamente para área de redes no roteamento de pacotes no domínio da Internet. Em tempo real, ele procura demonstrar e aconselhar a melhor forma de enviar pacotes para um determinado destino. Adele, por sua vez, é um agente com características humanas projetado para trabalhar com os estudantes na área da medicina. Suas funções são auxiliar na resolução de problemas e disponibilizar material para estudo. Além disso, Adele é capaz de destacar assuntos importantes, monitorar o aluno e realizar testes de avaliação com o objetivo de verificar o grau de compreensão do aluno sobre o assunto que está sendo estudado.

Outros pesquisadores têm investigado o impacto destes agentes em ambientes de aprendizagem interativo. André [1], por exemplo, demonstraram que estudantes podem considerar o assunto em estudo significativamente menos difícil e a apresentação mais lúdica na presença de um assistente virtual. Neste mesmo experimento a maior parte dos estudantes afirmou que o assistente os auxiliou a prestar atenção a informações relevantes. Já Craig [5] mostraram efeitos distintos no aprendizado de alunos submetidos a interações com agentes estáticos e animados.

Outros pesquisadores focalizam a importância de aspectos sociais e afetivos nas experiências de aprendizagem dos estudantes em ambientes computacionais interativos. Elliot [7] investigaram a importância da dimensão afetiva em personagens virtuais, desenvolvendo estudos sobre agentes pedagógicos que são sensíveis aos estados emocionais dos estudantes e são capazes de utilizar dados aspectos afetivos na resolução de problemas. De Angeli [6] também apresentaram argumentos que dão suporte à utilização de personagens virtuais, enfatizando que, através da introdução de estímulos sociais, os agentes de interface podem melhorar a comunicação entre o usuário e o computador.

O I-Help [10] é um ambiente multiagente formado por assistentes pessoais virtuais em que agentes probabilísticos, representando usuários, buscam e ofertam ajuda a outros usuários. Neste sistema, os usuários buscam ou oferecem ajuda a outros usuários através de seus agentes pessoais. Porém, os assistentes pessoais do I-Help não foram projetados como agentes animados de interface. No I-Help, o agente que solicita ajuda negocia, com outro agente, em termos de tempo e dinheiro. O protocolo de negociação normativamente descreve a ordem das ações. O mecanismo de tomada de decisão descreve o conjunto de estratégias do agente, que correspondem às ações que

um agente pode realizar numa determinada situação: aceitar, rejeitar ou sugerir uma contraproposta. Os agentes, em determinados momentos, podem solicitar ajuda e em outros, oferecê-la. Na fase de negociação, o processo de tomada de decisão é baseado na avaliação de um diagrama de influências, onde se define a função utilidade dos entendimentos. A negociação termina quando o agente aceita ou rejeita uma oferta. Se o agente faz uma contraproposta, esta nova informação é armazenada, e o processo retorna à fase de negociação. Para se tomar uma decisão, o agente consulta as interações do passado buscando compreender suas ações, considerando e analisando as propostas rejeitadas, aceitas e negociadas.

Conati [3] desenvolveu um modelo probabilístico para monitorar as emoções dos usuários e seu entrosamento na interação com um jogo educacional que objetiva a aprendizagem da fatoração de números. O modelo avalia a afetividade através da evidência das possíveis causas da manifestação de emoção do usuário e de seus efeitos (expressões corporais influenciadas por reações emocionais). O modelo probabilístico desenvolvido é usado na decisão do agente pedagógico para gerar intervenções visando alcançar o melhor entre a aprendizagem do aluno e seu engajamento durante sua interação com o jogo. O agente pedagógico do jogo “Prime Climb” foi desenvolvido com os recursos do Microsoft Agent, assim como o agente proposto neste trabalho, e busca estimular a interação entre o aluno e o jogo. No caso do agente proposto neste artigo (seção 3), o modelo de emoções é utilizado com o propósito de motivar a colaboração entre os alunos, de forma que o agente leva em consideração o estado afetivo de um aluno como parte da estratégia de formação dos grupos.

A Educação é uma área onde os agentes de interface podem ser aplicados como um recurso interessante para aliviar a carga cognitiva do aluno, tanto em relação à navegação na interface, como no acesso do conteúdo desejado. Neste caso, o agente atua como um tutor virtual que guia o aluno pelos ambientes interativos, como em [22], onde instrutores virtuais guiam os estudantes em histórias interativas online.

Na pesquisa realizada por Ball [2] são elencadas as características de interfaces sociáveis, a saber:

- Suporte à interatividade através de perguntas e respostas: Os agentes não somente podem responder quando lhes é perguntado alguma coisa, mas também realizam perguntas para melhor compreender sua tarefa, descrever seus planos e antecipar problemas;
- Reconhecimento dos custos de interação e atraso: o agente deve modelar a significância de suas decisões e o potencial custo de um possível erro. Deste modo, ele só buscará uma interação como o usuário quando realmente necessário, não o desconcentrando, nem atrasando uma tarefa à espera de uma resposta caso o usuário esteja ausente;

- Gerenciamento de interrupções: o agente deve ser capaz de perceber se o usuário está ativo ou inativo no ambiente para escolher o momento apropriado de realizar a interrupção;
- Reconhecimento dos aspectos sociais e emocionais da interação: um agente de interface deve saber qual o “comportamento adequado” para executar determinada tarefa. Também deve saber o momento de dar *bom dia* ou *boa tarde*, e conhecer o tipo de humor a apresentar para melhorar sua interação com o usuário. São estas características que proporcionam boa parte da vivacidade de uma interface.

Normalmente ambientes de aprendizagem são sistemas de múltiplos usuários e, portanto, os agentes devem adaptar-se à forma de estudo de cada um. A próxima seção apresenta nosso experimento com o agente de interface *Kurrupako* em um ambiente multi-usuário para aprendizagem de algoritmos.

3. O AGENTE ANIMADO SÓCIO-AFETIVO

O 3A é um ambiente virtual de aprendizagem utilizado simultaneamente por vários usuários, onde o agente proposto é capaz de aprender e se adaptar a cada estudante. Este agente foi batizado de *Kurrupako* e possui como uma de suas principais funcionalidades a capacidade de recomendar conteúdos e tutores aos usuários de forma animada.

3.1. AMBIENTE DE APRENDIZAGEM DE ALGORITMOS - 3A

O ambiente interativo de aprendizagem de algoritmos 3A vem sendo desenvolvido como forma de apoio à disciplina de algoritmos do Departamento de Informática da Universidade de Caxias do Sul, uma disciplina basicamente presencial. O principal objetivo do 3A é tornar as aulas mais dinâmicas, aumentando o interesse e participação dos alunos e provendo um ambiente onde eles possam resolver dúvidas na ausência do professor. O sistema apresenta aos alunos conteúdos regulares da disciplina, propõe exercícios, disponibiliza um mecanismo de interação síncrona (*chat*), um mecanismo de interação assíncrona (*fórum*), além de uma ferramenta para teste e execução de algoritmos. Através das ferramentas de interação, os alunos podem colaborar com os colegas e desenvolver atividades colaborativas, apoiados também pelo simulador de algoritmos.

O 3A é um sistema desenvolvido para Web utilizando-se da linguagem Java (J2SE), que além de ser uma linguagem robusta, provê meios de implementação para a Internet através de recursos da plataforma J2EE (Java Enterprise Edition). Além disso, utiliza o Tomcat como servidor para que esse ambiente possa ser iniciado e acessado a partir de um navegador. Como banco de dados foi escolhido o

MySQL, por ser um banco leve e de simples utilização. Basicamente, pode-se dizer que o 3A é constituído por quatro componentes, descritos nas subseções a seguir.

3.1.1. MÓDULO DE CONTEÚDO

O Módulo de Conteúdo é responsável pela organização e armazenamento dos conteúdos e atividades do sistema. É constituído basicamente de uma apostila de algoritmos e de exercícios cadastrados para cada tópico por nível de dificuldade.

3.1.2. MÓDULO PARA RECOMENDAÇÕES DE CONTEÚDO

Este módulo emprega um sistema de recomendação para identificar e sugerir tópicos de interesse dos alunos. Em tempo real, ele identifica padrões de navegação dos alunos e os interpreta como caminhos alternativos na leitura dos textos e execução das atividades. Ao encontrar e sugerir tais caminhos a outros alunos que mostram padrões similares de navegação, o sistema dá apoio a alunos com diferentes estilos de aprendizagem, sem forçá-los a seguir tais caminhos nem alterar a estrutura de navegação pré-definida pelos professores.

Uma das técnicas mais antigas utilizadas pelos sistemas de recomendação é conhecida como filtragem colaborativa [12]. Esta técnica prevê as preferências de um usuário com base nas preferências de outros com comportamento semelhante. A técnica de filtragem colaborativa tem sido utilizada com sucesso em diversos projetos de pesquisa. O projeto Tapestry [8], uma das primeiras implementações de filtragem colaborativa em sistemas de recomendação, utilizava opiniões de pessoas de uma comunidade pequena (e.g. escritório ou grupo de trabalho) para encontrar recomendações. Já o projeto GroupLens da Universidade de Minnesota [20] auxiliava o usuário a encontrar filmes de seu interesse. Hoje em dia, no entanto, os sistemas de recomendação já migraram dos laboratórios de pesquisa para o mercado da internet, e podem ser encontrados em grandes *websites* como o da livraria Amazon.com ou da loja de discos CDNow.com [21].

No ambiente 3A o sistema de recomendação é utilizado não para identificar preferências, mas para recomendar conteúdo apropriado, a partir da análise de seqüências de navegação executadas por outros estudantes. O módulo de recomendação de conteúdo opera através da sugestão de links, exercícios ou outros materiais relacionados ao assunto consultado pelo aluno [18]. Para viabilizar este processo, um sistema de monitoramento armazena todas as interações dos alunos, indicando o dia e a hora em que cada seção do sistema é acessada por cada um. Um sistema de mineração de dados vasculha este repositório em busca de seqüências de acesso frequentes. Quando encontradas, estas seqüências são armazenadas em descritores de itens [17]. A abordagem de descritores

de itens usa estruturas semelhantes a registros para representar conhecimento sobre como fazer as recomendações.

Um descritor de item representa o conhecimento sobre quando recomendar um item em particular, listando as características que os usuários provavelmente interessados nos item deveriam ter. Eles são os principais componentes empregados na representação do perfil dos alunos. Estas características podem ser classificadas como *demográficas* (dados que descrevem um indivíduo, como idade, sexo, ocupação, endereço), *comportamentais* (dados que descrevem os acessos aos textos, atividades realizadas e preferências de um indivíduo) ou *sócio-afetivas* (aceitação, habilidade interpessoal, estado afetivo). Enquanto as características demográficas e comportamentais são empregadas na recomendação de conteúdos, as informações sócio-afetivas sobre o usuário são utilizadas na identificação e recomendação de tutores, tópico apresentado na seção 3.1.3. A Tabela 1 apresenta um exemplo de um descritor de item onde características comportamentais e demográficas são consideradas na recomendação de um certo item (representado pelo descritor “Programação de Jogos”).

Tabela 1: Descritor de item para o texto “Programação de Jogos”.

Programação de Jogos (Texto)		
Tópicos de Estudo		
Item Relacionado	Classe	Confiança
Introdução ao Desenvolvimento de Jogos	Texto	0.70
Princípios de IA e Jogos	Texto	0.66
C++ ou Java?	Texto	0.52
Características Demográficas		
Item Relacionado	Classe	Confiança
Sexo = “masculino”	Demográfico	0.67
Faixa etária = [18,25]	Demográfico	0.42

O descritor expressa a idéia de que o usuário que se interessa pelos Tópicos de Estudo “Introdução ao Desenvolvimento de Jogos”, “Princípios de IA e Jogos” e “C++ ou Java?” também poderá se interessar pelo texto sobre “Programação de Jogos”. Além disso, ele indica que a maioria dos usuários que leram o texto “Programação de Jogos” eram alunos do sexo masculino na faixa etária de [18-25] (Característica Demográfica). O fator de confiança de um termo é apresentado também no descritor, e representa a força com que um termo está relacionado a outro, neste caso ao texto “Programação de Jogos”. Este fator é obtido através do cálculo da probabilidade condicional $P(d_i/e)$, ou seja, a probabilidade de um usuário

que tenha se interessado pelo item e também se interessar pelo item representado por d_i .

Considerando-se que cada item possui seu próprio descritor, o processo de recomendação é realizado da seguinte forma: dada uma lista de usuários $U=\{u_1, u_2, \dots, u_m\}$ e uma lista de descritores $D=\{d_1, d_2, \dots, d_n\}$, é iniciada a coleta de informações sobre um usuário u_i para quem se deseja fazer alguma recomendação. Em seguida, um mecanismo competitivo inicia, no qual o sistema calcula a similaridade para cada descritor d_j quando comparado aos dados coletados sobre as interações com o usuário, encontrando a lista de termos relacionados $T=\{t_1, t_2, \dots, t_k\}$. O sistema calcula então um *escore* para o descritor d_i , que varia entre diferente (0) até muito similar (1), considerando k os termos semelhantes, de acordo com a fórmula:

$$Escore(d_i) = 1 - \prod_{j=1}^k (1 - w_j * P(d_i | t_j))$$

onde $Escore(d_i)$ é o valor final computado para o descritor d_i ; w_j é o peso associado ao termo t_j ; e $P(d_i | t_j)$ é a probabilidade condicional de d_i dado o termo t_j . O uso de pesos diferentes para os termos t_j possibilita o ajuste manual do sistema a fim de considerar alguns itens ou características mais importantes que outros no cálculo dos escores dos descritores. Por exemplo, considerando um usuário do sexo masculino com 28 anos que tenha acessado os seguintes tópicos de estudo: “C++ ou Java?” (Texto); “Introdução ao Desenvolvimento de Jogos” (Texto); “2ª Lista de Exercícios de Programação” (Atividade Prática). Quando comparamos as informações sobre o usuário com o descritor “Programação de Jogos”, apresentado na Tabela 1, três termos comuns são encontrados:

- Introd. ao Desenvolvimento de Jogos - Confiança 0.7
- C++ ou Java? – Confiança 0.52
- Sexo = “masculino” – Confiança 0.67

Considerando diferentes pesos para as informações Comportamentais e Demográficas (1.0 e 0.8 respectivamente), o $Escore(d_i)$ representando o quão apropriado para a recomendação é o texto “Programação de Jogos”, pode ser calculado como:

$$\begin{aligned} Escore(d_i) &= 1 - \{[1 - (1 * 0.7)] * [1 - (1 * 0.52)] * [1 - (0.8 * 0.67)]\} \\ &= 1 - \{0.3 * 0.48 * 0.464\} \\ &= 0.933 \end{aligned}$$

Esta abordagem é baseada na hipótese de que qualquer termo encontrado na lista de informações do usuário deve aumentar a confiança na recomendação representada no descritor. Em um exemplo real, é possível assumir com um certo grau de confiança que um consumidor que compra uma lixa de unhas possivelmente vai querer comprar esmalte para unhas. Ao sabermos que trata-se de um consumidor do sexo feminino, o grau de confiança na recomendação do item esmalte deve aumentar.

Calculado o escore para cada descritor de item, aqueles que obtiverem o maior valor serão recomendados ao usuário.

3.1.3. MÓDULO RECOMENDAÇÃO DE TUTORES

Este módulo tem como objetivo encontrar alunos que possam auxiliar outros na resolução das tarefas propostas, estimulando a cooperação e a aprendizagem colaborativa. A função de recomendação de tutores tem como parâmetros de entrada o identificador do aluno e o tópico no qual se necessita de ajuda. Os alunos conectados no sistema são identificados e cria-se um escore individualizado para cada um. Este escore é baseado nos seguintes aspectos:

- **Estado de humor:** O humor de uma pessoa pode variar diversas vezes em um dia, dependendo do conjunto de ações que esta realiza. Mesmo tendo sucesso na maior parte de suas atividades, o humor da pessoa pode ser afetado negativamente, deixando-a com uma espécie de resíduo emocional desagradável. Aqui, utilizamos o estado de humor como indício da capacidade do estudante em ser tutor em determinado momento, já que um aluno num mau estado de humor pode ter mais dificuldade em auxiliar seus colegas. Atualmente, o estado de humor não é inferido pelo sistema, mas é informado espontaneamente pelo estudante no momento do *login* através de uma interface gráfica. A variável pode assumir valores no intervalo [0,1], onde 0 representa mau humor e 1 representa ótimo humor.
- **Performance:** Representa o desempenho do aluno nas diversas atividades desenvolvidas. Um maior número de exercícios concluídos e um maior número de acertos denotam uma performance melhor. Utilizamos dois graus de performance relativos para estimar o desempenho do aluno. O primeiro mede o quanto o aluno apresentou uma boa performance em seu percurso durante o semestre/ano. O outro estabelece o quanto o aluno conseguiu aprender sobre o tópico específico no qual está sendo solicitada a ajuda. Ambos os graus são calculados pela mesma fórmula, uma delas aplicada a todas as leituras e atividades propostas no semestre/ano, e a outra aplicada às leituras e atividades propostas para o tópico em questão.

$$Pf = \frac{P1 * Lc}{Lp} + \frac{P2 * Ac}{Ap}$$

onde Pf é a performance (desempenho) do aluno, Lc são as leituras concluídas, Lp são as leituras propostas, Ac são as atividades concluídas com êxito e Ap são as atividades propostas. $P1$ e $P2$ são pesos que permitem ponderar a importância de cada fator. Deste modo, é possível, por exemplo, atribuir um peso maior para o fator Ac (atividades concluídas com êxito), já que o que se busca

é verificar o desempenho do aluno. O grau de performance final é calculado através da média entre os dois graus relativos.

- **Aceitação:** No final de cada sessão em que um aluno atuou como tutor, uma nota lhe é atribuída pelo colega que solicitou o auxílio, indicando o quão proveitosa foi a interação. O grau de aceitação é calculado pela média destas notas, indicando o quanto o aluno é aceito como tutor por seus colegas.
- **Média Interações:** Refere-se à média total de comunicações iniciadas e respondidas por um aluno, e pode ser calculada pela expressão:

$$Mi = \frac{P1 * Ci}{MGi} + \frac{P2 * Cr}{MGr}$$

onde Mi é a média de interações do aluno, Ci representa o número de comunicações iniciadas pelo aluno, MGi a média geral de comunicações iniciadas pelos alunos, Cr é o número de comunicações respondidas pelo aluno e MGr é a média geral de comunicações respondidas pelos alunos, enquanto $P1$ e $P2$ são pesos atribuídos para ponderar a importância de cada fator.

O escore final do aluno é obtido através de uma fórmula simples que calcula o somatório de todos os dados considerados para a recomendação de tutores.

$$EscoreAluno = P1 * Pf + P2 * Eh + P3 * Ac + P4 * Mi$$

onde Pf é a performance calculada para aluno, Eh corresponde ao seu estado de humor, Ac seu grau de aceitação e Mi sua média de interações. No caso específico da recomendação de tutores, apenas para efeito de teste e sem a pretensão de levantar questionamentos sobre a importância de cada um destes fatores no processo de recomendação, todos os fatores são multiplicados pelo mesmo peso com valor 1.

3.1.4. O AGENTE DE INTERFACE

O Agente de Interface tem o principal objetivo de apresentar aos usuários as recomendações de conteúdo e as sugestões de alunos tutores geradas pelo sistema. *Kurrapako* foi desenvolvido de forma a proporcionar aos estudantes um meio mais amigável de interação, uma vez que o agente é capaz de tomar iniciativas e executar determinadas tarefas sem que isso tenha sido solicitado.

Pelo fato de reproduzir animações, gestos e representações conhecidas, salientando bem suas características sociais, o agente ganha maior aceitação e maior grau de credibilidade por parte do usuário, gerando nele maior confiabilidade. *Kurrapako* realiza quatro tarefas bem definidas, ilustradas na Figura 1.

A recomendação de conteúdo e de alunos tutores é feita através do acesso aos módulos apropriados, descritos nas seções anteriores (3.1.2 e 3.1.3).

Já o mecanismo de guia do usuário é capaz de conduzi-lo pelo ambiente, apresentando-lhe a organização do conteúdo, as ferramentas disponíveis, bem como a possibilidade e a importância de interação com os colegas na realização das atividades. Além disso, o mecanismo de guia do usuário utiliza o estado afetivo do aluno (ex. feliz, triste) para tentar atingi-lo psicologicamente com alguma frase que o faça refletir sobre determinada situação. Estas frases estão armazenadas numa tabela no banco de dados e são escolhidas de acordo com o estado afetivo do aluno. Desta forma, pretende-se motivar o aluno e auxiliá-lo a ultrapassar barreiras relacionadas a dificuldades encontradas em determinado assunto, ou mesmo relacionadas ao seu estado afetivo.

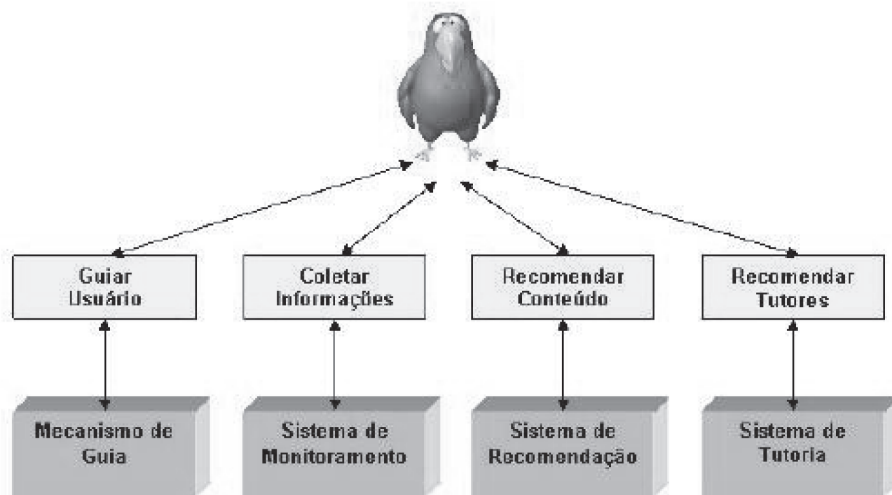


Figura 1: Agente Kurrapako.

O módulo do agente responsável pela coleta de informações é por sua vez responsável pelo monitoramento de todas as ações do usuário, fornecendo os subsídios necessários para que o agente possa tratar cada aluno de maneira personalizada.

Kurrapako foi implementado a partir dos modelos de Agentes da Microsoft. Esses agentes são componentes prontos que contêm uma série de recursos como: animações, expressão através da fala, expressões faciais e gestos.

Tabela 2: Tabela de Animações

	acoesAgente	estadoAfetivo	tipoAcao	fala	movimento
1		A	F	Olá, que bom vê-lo novamente!!!	GetAttention
▶ 2		T	F	O que eu posso fazer para lhe deixar feliz?	Thinking
3		NA	T	Por que você não conversa com	LookLeftReturn
4		NA	T	Um colega seu poderia lhe ajudar:	Uncertain
5		NA	R	Gostaria de lhe recomendar um exercício:	LookRightReturn

O agente *Kurrapako* teve suas ações programadas através de scripts em linguagem VBScript e inseridas em páginas HTML. Para criar as ações do agente foi desenvolvido um mecanismo de geração e interpretação dinâmica de scripts. Como o A3 foi construído através de páginas JSP, que por sua vez geram como saída código HTML, a idéia principal do projeto de implementação do agente foi utilizar a potencialidade de programação em JSP e Java para gerar scripts dinâmicos.

O agente possui um Módulo de Animação que seleciona uma animação apropriada para a tarefa que será executada. Para este módulo foi criada uma tabela no banco de dados representada na Tabela 2. A coluna da tabela representando **estadoAfetivo** é utilizada para estabelecer o tipo de fala e de animação que mais se enquadra ao estado

afetivo do aluno: A (alegre) ou T (triste). A coluna **tipoAcao** indica o tipo de tarefa que o agente pode realizar: F (Guiar), R (Recomendar Conteúdo), T (Recomendar Tutor), N (Fazer Nada – útil para não interromper o usuário a todo o momento). A coluna **fala** contém a frase a ser utilizada pelo agente quando executa determinada tarefa. Por fim, a coluna **movimento** armazena o identificador da animação a ser executada para a tarefa.

Os movimentos apresentados na última coluna da Tabela 2, são recursos pré-estabelecidos do Microsoft Agent. Para o *Kurrapako* foram escolhidos alguns movimentos, em função do tipo de ação e do estado afetivo do aluno. Além disso, foram associadas diferentes animações para a mesma ação, tornando a atuação do agente menos previsível. Os movimentos utilizados pelo agente são detalhados na Tabela 3.

Tabela 3: Conjunto de animações utilizadas pelo agente.

Animação	Exemplo de Uso	Exemplo de animação
<i>Blink</i>	Quando o personagem termina uma fala ou estado de espera (<i>idle</i>)	Personagem pisca os olhos naturalmente
<i>Confused</i> ¹	Quando o personagem não entende o que deve fazer	Personagem coça a cabeça
<i>GetAttention</i> ¹	Quando o personagem precisa notificar o usuário sobre algo importante	Personagem acena com as mãos ou pula para cima e para baixo
<i>LookLeft</i>	Quando o personagem precisa olhar para a esquerda	Personagem olha para a esquerda
<i>LookRight</i>	Quando o personagem precisa olhar para a direita	Personagem olha para a direita
<i>LookUp</i>	Quando o personagem precisa olhar para cima	Personagem olha para cima
<i>MoveDown</i>	Quando o personagem se prepara para se mover para baixo	Personagem se move caminhando ou voando para uma posição abaixo (definida pelo programador)
<i>MoveLeft</i>	Quando o personagem se prepara para se mover para esquerda	Personagem se move caminhando ou voando para uma posição a esquerda (definida pelo programador)
<i>MoveRight</i>	Quando o personagem se prepara para se mover para direita	Personagem se move caminhando ou voando para uma posição a direita (definida pelo programador)
<i>MoveUp</i>	Quando o personagem se prepara para se mover para cima	Personagem se move caminhando ou voando para uma posição acima (definida pelo programador)
<i>Speak</i> ¹	Quando o personagem fala de forma neutra	Personagem permanece relaxado com postura atenta
<i>Think</i> ¹	Quando o personagem está pensando sobre algo	Personagem olha para cima e coloca as mãos na cabeça
<i>Uncertain</i> ¹	Quando o personagem precisa a confirmação do usuário a respeito de uma solicitação	Personagem olha zombeteiro, no sentido de "Tem certeza?"
<i>Writing</i>	Quando o personagem escreve uma informação para o usuário	Personagem escreve em um pedaço de papel (esta animação segue em <i>loop</i> , se repete)

¹ Animação requer balões de fala e movimentos de lábios

Ao acessar o 3A o agente *KurrupaKo* é iniciado. Ele inicia saudando o usuário e logo verifica se é a primeira vez que o usuário acessa o ambiente. Caso seja sua primeira vez, a tarefa inicial do agente após sua apresentação é introduzir e guiar o usuário no ambiente, mostrando como o usuário pode navegar e começar a estudar. Esta interação está representada nos detalhes 1 e 2 da Figura 2.

Algumas animações são representações que combinam várias ações diferentes, como no exemplo da Figura 2 (detalhe 1) em que a ação *LookDownBlink* é uma combinação das animações *LookDown* e *Blink*. A partir dela, o agente executa a seqüência de ações de olhar para baixo e finaliza a animação em estado de espera, piscando os olhos.

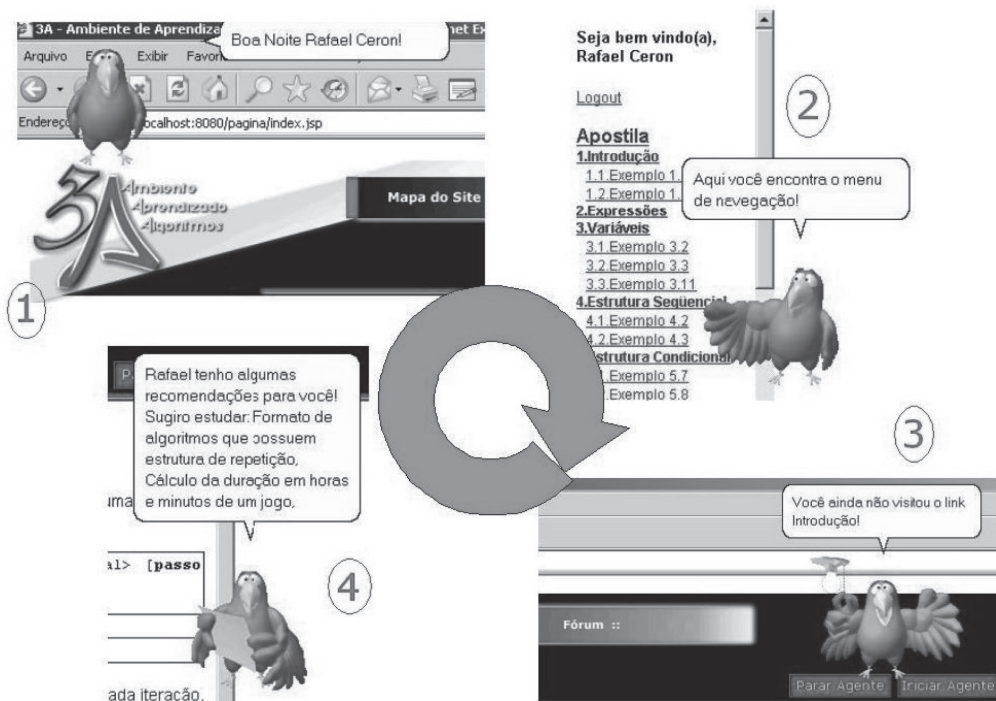


Figura 2: Seqüência de ações do Agente.

O detalhe 3 da Figura 2 mostra o assistente trazendo uma sugestão de link, e no detalhe 4 a recomendação de dois tópicos de estudo para o aluno Rafael: “Formato de algoritmos que possuem estrutura

de repetição” e “Cálculo da duração em horas e minutos de um jogo”. A imagem completa da tela do sistema com o agente realizando estas recomendações é apresentada na figura 3.

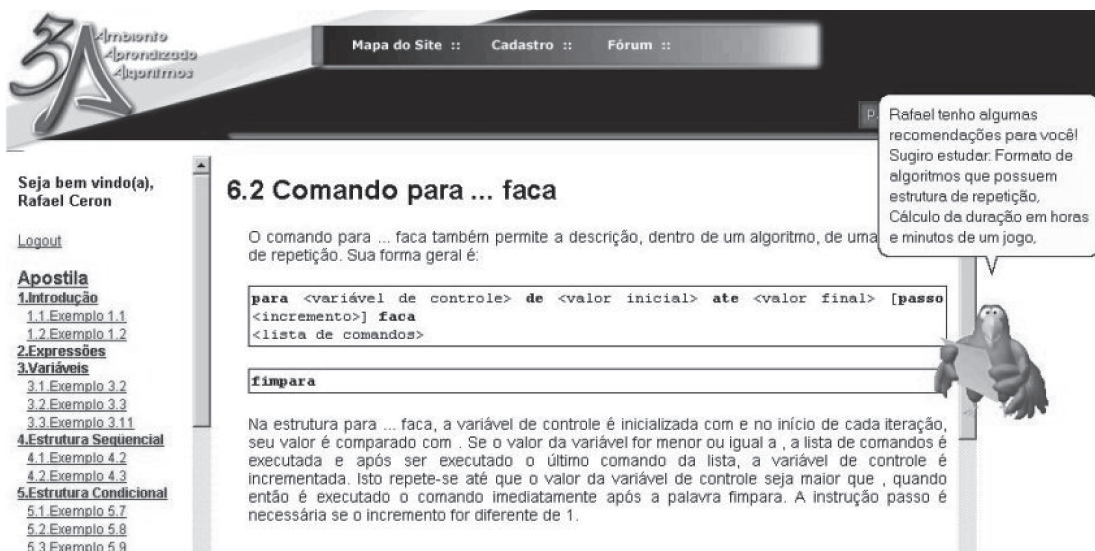


Figura 3: Recomendação de Conteúdo.

Para realizar essas recomendações, todos os módulos da arquitetura proposta foram acionados: (1) o usuário foi identificado; (2) seu histórico de leituras e atividades propostas foi recuperado; (3) uma das ações do agente foi selecionada, no caso *recomendar conteúdo*; (4) após a seleção da ação de recomendação de conteúdo, os itens relacionados ao assunto consultado pelo aluno foram encontrados pelo sistema de recomendação; (5) a fala do agente *Kurrupako* foi então construída pelo gerador

dinâmico de scripts com base nos dados da tabela de controle das ações do agente e nas recomendações advindas do sistema de recomendação; (6) as animações e falas do agente *Kurrupako* foram então acionadas para gerar a visualização apropriada.

A Figura 4 mostra outro exemplo de interação com o agente *Kurrupako* em que ele recomenda um aluno tutor para auxiliar o usuário na execução de uma tarefa.

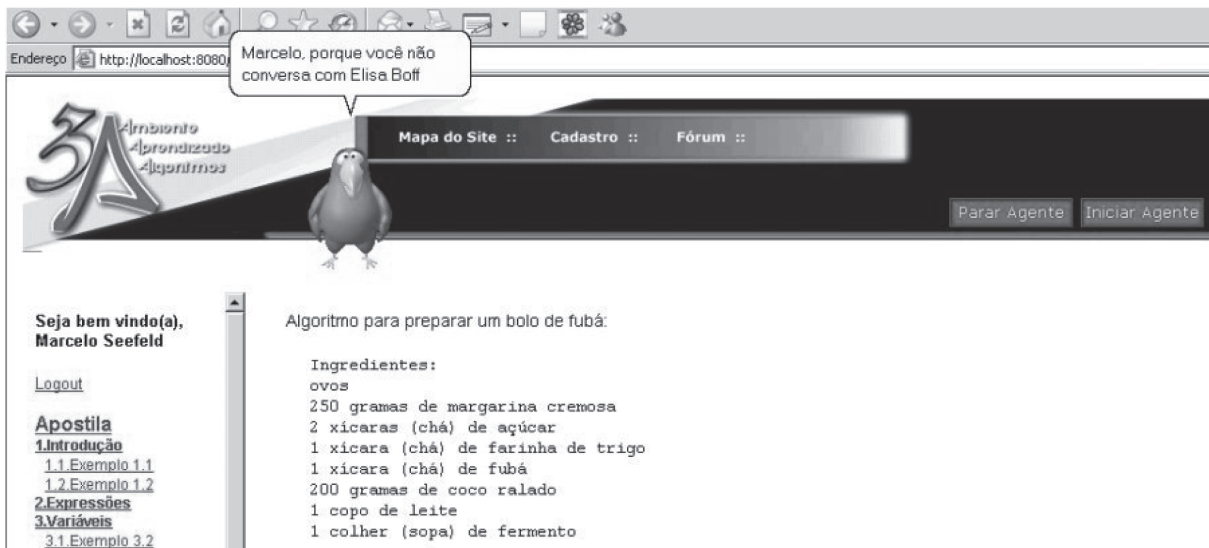


Figura 4: Recomendação de Tutor.

Caso o usuário já tenha utilizado o ambiente, o agente é capaz de identificar e sugerir um novo assunto a partir dos que já foram estudados. Isso se deve ao fato de o agente monitorar o usuário através dos registros de acesso aos materiais e conteúdos do ambiente. Assim, o usuário ao entrar no ambiente pode aceitar este conselho e continuar estudando a partir de onde havia parado, ou simplesmente ignorar e continuar navegando pelo ambiente. Essa tarefa representa a capacidade do agente em realizar o acompanhamento dos estudos do usuário.

4. EXPERIMENTOS REALIZADOS

Organizamos um primeiro experimento para avaliar certos aspectos do ambiente 3A relacionados ao assistente virtual e seu mecanismo de recomendação. Os resultados do experimento foram apresentados em [18], e serão relatados novamente aqui. Cinco grupos de estudantes da Universidade de Caxias do Sul utilizaram o ambiente, todos alunos de cursos de graduação relacionados à Informática (Bacharelado em Ciência da Computação, Bacharelado em Sistemas de Informação e Bacharelado em Tecnologias Digitais). Um total de 53 estudantes participou do experimento, todos matriculados na disciplina de Algoritmos.

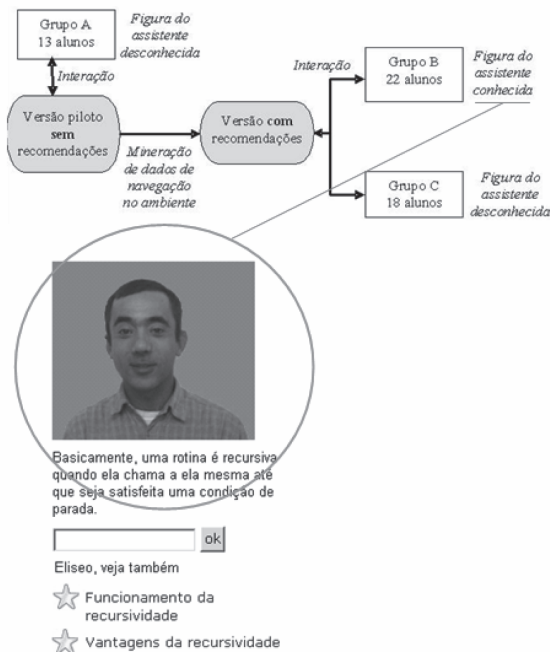


Figura 5: Organização do Experimento.

Uma versão piloto do sistema foi preparada, onde o assistente virtual estava presente, mas não havia

recomendações de contúdo e de tutores. Solicitou-se ao primeiro grupo de estudantes, em número de 13, que estudassem o tópico de recursão utilizando o ambiente, e que resolvessem um dos dez exercícios propostos. Depois disso, os alunos responderam um questionário para avaliação do sistema. Os dados coletados da navegação destes alunos, totalizando 482 transações, foram minerados e empregados na construção de descritores de itens para o servidor de recomendações. Uma segunda versão do sistema foi então gerada, esta capaz de fazer recomendações a partir dos descritores gerados através do processo de mineração do experimento inicial.

Escolhemos como assistente virtual a imagem não animada de um professor de Algoritmos (apresentado no detalhe da Figura 5) que estava ministrando a disciplina a 22 dos 40 alunos restantes, excluindo-se os 13 alunos que participaram do experimento inicial. Para os outros 18 alunos, este professor era uma figura desconhecida. O grupo que conhecia o assistente foi chamado de **grupo B**, o outro grupo foi denominado **grupo C**, como mostra a Figura 5.

Os alunos dos grupos B e C foram solicitados a realizar a mesma tarefa do grupo inicial (A), ou seja, estudar recursão utilizando o ambiente de aprendizagem e responder um dos exercícios propostos. Os dados de navegação e as respostas dos questionários dadas pelos estudantes que utilizaram a versão do ambiente com recomendações, nos permitiram chegar a algumas conclusões.

Em relação à utilização do assistente virtual, pudemos verificar o impacto positivo no emprego de um assistente com figura conhecida. Para o grupo de 18 alunos que não conheciam o assistente, apenas 28% indicaram que o assistente auxiliou-os nas tarefas. 38% destes alunos apontaram o assistente como inútil. Já no grupo de 22 alunos que conheciam o assistente, 90% indicaram que o assistente auxiliou-os na tarefa, e nenhum deles apontou o assistente como inútil. Do grupo de 18 alunos que não conheciam o assistente, 33% indicaram que com certeza voltariam a utilizar o ambiente, 55% assinalaram que talvez voltassem a utilizá-lo, e 12% indicaram que não o utilizariam novamente. No grupo de 22 alunos que conheciam a figura do assistente, 86% apontaram que com certeza voltariam a utilizar o ambiente de aprendizagem, 14% indicaram que talvez voltassem a utilizá-lo, e nenhum aluno indicou que não voltaria a utilizar o ambiente.

Em relação à capacidade do assistente em recomendar conteúdo adequadamente, não verificamos diferenças significativas nas respostas dos alunos dos grupos B e C. Logo, os dados de ambos os grupos foram analisados em conjunto. Dos 40 alunos que participaram do experimento, 8 não perceberam as recomendações. Dos 32 estudantes restantes, 22% apontaram as recomendações como muito apropriadas, 78% as apontaram como apropriadas, e nenhum aluno as considerou inapropriadas. Dos 32 alunos, 97% assinalaram ser capaz de lembrar pelo menos de algumas recomendações, e 90% indicaram seguir as recomendações. Estes resultados mostram eficiência do

mecanismo de recomendação utilizado. Se compararmos as respostas do grupo A (alunos que utilizaram o assistente sem capacidade de recomendação de conteúdo) com os grupos B e C (alunos que utilizaram o assistente com capacidade de recomendação de conteúdo), podemos verificar o impacto do sistema de recomendação na percepção dos alunos no que diz respeito à capacidade do assistente em auxiliá-los: 92% dos alunos do grupo A informaram que o assistente não foi capaz de ajudá-los na realização das tarefas propostas.

Nos grupos B e C (assistente com capacidade de recomendação), apenas 38% dos alunos indicaram que o assistente não foi capaz de auxiliou-os na resolução das tarefas. Estes dados destacam a importância de um sistema de recomendação de conteúdos em um ambiente de aprendizagem.

Atestada a validade do sistema de recomendação, partimos para um novo experimento, a fim de validar a utilização do agente animado *Kurrupako*. Neste experimento, foi proposta uma atividade para um grupo de 30 alunos da disciplina de Algoritmos. A atividade consistiu na exploração do ambiente e na resolução de exercícios, seguida do preenchimento de um formulário de avaliação do ambiente.

Neste grupo de alunos, 63% acharam o ambiente de aprendizagem de *fácil* utilização e navegação e 37% apontaram o fato da utilização do ambiente ser *muito fácil*.

Os dados deste experimento indicaram que:

- **Em relação à presença do assistente:** 73% dos alunos achou sua presença útil, 20% acharam muito útil e somente 7% acharam sua presença inútil. Os alunos que apontaram a presença do assistente como inútil justificaram sua posição informando que o ambiente já é de fácil navegação sem o assistente.
- **Em relação ao uso do assistente:** 73% dos alunos indicaram que gostariam da ajuda de um assistente para apoiar sua aprendizagem através de um ambiente virtual, enquanto 27% dos alunos disseram que *talvez* utilizassem a ajuda de um assistente em um ambiente de aprendizagem. Nenhum dos alunos indicou que *não* gostaria de utilizar um assistente.
- **Em relação à forma de atuação do assistente:** a questão “*a presença de um assistente virtual poderia se tornar inconveniente prejudicando seu desempenho*” foi apresentada para avaliar a forma de atuação do assistente, ao lado da questão anterior. Os dados coletados mostraram que 67% dos alunos não acharam inconveniente a presença do assistente, 30% dos alunos indicaram que o assistente *talvez* pudesse se tornar inconveniente e somente 3% indicaram inconveniente sua presença. Alguns alunos comentaram que acharam importante a opção que o assistente possui de poder

ocultá-lo, se julgassem desnecessária sua atuação no momento, e acioná-lo conforme a necessidade.

Os comentários mais representativos foram relacionados ao fato do assistente sanar dúvidas de conteúdo e de navegação do ambiente. Além disso, observamos o aspecto afetivo em relação ao assistente, pois os alunos se sentiram motivados a interagir com o ele, considerando-o “meigo”, “engraçado” e “divertido”.

Alguns comentários dos alunos ilustram os diagnósticos apresentados acima:

“...ele é meigo e nos lembra alguns avisos...”

“...facilita o acesso à apostila, dúvidas e exercícios...”

“...incentiva o uso do site...”

“...ajuda nas horas que é realmente necessário”

“...você pode ter alguma dúvida, meio idiota, e tem vergonha de perguntar, aí ele (assistente) esclarece...”

“...economiza tempo”

“Acho desnecessário (o assistente), o ambiente está bem claro”

“... gostei pois encontramos a matéria com facilidade, facilitando nosso trabalho”.

O primeiro experimento que fizemos com assistentes não animados indicaram que a maioria dos alunos gostou das recomendações do assistente, mas nem sempre perceberam as recomendações feitas por ele. Este foi o principal diferencial entre o assistente virtual estático e o animado, visto que no animado os alunos perceberam todas suas intervenções.

No futuro, pretendemos desenvolver assistentes animados com figuras conhecidas dos nossos alunos, pois observamos a relação afetiva entre estes e o personagem virtual.

5. CONSIDERAÇÕES FINAIS

Este artigo apresentou um modelo de agente de interface sócio-afetivo capaz de interagir com usuários com o objetivo de motivar a formação de grupo e promover a aprendizagem colaborativa. A recomendação de tutores é utilizada como estratégia para estimular o início da colaboração entre os alunos. Em [9] os autores também propuseram um mecanismo de recomendação de tutores. No entanto, utilizaram uma abordagem diferente já que não enfatizaram a dimensão sócio-afetiva na construção do modelo do usuário. Além de buscar identificar alunos que melhor possam cumprir a tarefa de tutores para determinada tarefa, o agente proposto também está apto a encontrar e recomendar conteúdos adequados para cada estudante.

Atualmente, evidências científicas começam a apontar que a presença de um personagem em uma interface pode aumentar a confiança do usuário [19] e melhorar a comunicação

entre homem e máquina através da introdução de estímulos sociais [6]. No entanto, para que tais efeitos de interação possam ocorrer é necessário que os personagens tenham papéis e comportamentos adaptados à comunicação com cada usuário [16]. Em nosso sistema, o agente *Kurrupako* não apenas memoriza toda a interação do usuário com o ambiente para lhe fazer recomendações personalizadas de conteúdo, como também identifica outros alunos que melhor se adaptam à categoria de tutor para determinada tarefa. Ao considerar características sócio-afetivas do usuário para modelar e selecionar suas ações, o agente *Kurrupako* adquire propriedades julgadas fundamentais na produção de comportamento verossímil [11].

Os experimentos realizados até então não são conclusivos, mas indicam a capacidade do assistente em despertar interesse e capturar a atenção dos usuários, servindo como um forte componente motivador para os alunos. Um de nossos próximos desafios está relacionado à inferência de estados afetivos do usuário a partir de dados coletados durante a interação, com base no trabalho de Conati [3]. Porém, enquanto o modelo probabilístico de inferência de emoções proposto pela autora busca melhorar a interação do aluno com o sistema, em nossa pesquisa o estado afetivo é utilizado para motivar as interações entre os estudantes, estimulando a colaboração entre usuários e a formação de grupo.

Além disso, estamos interessados na utilização de outros modelos de assistentes com características gráficas mais ricas na busca de uma melhor comunicação com o usuário e na obtenção de resultados mais expressivos no que diz respeito à percepção do usuário das capacidades e utilidade dos agentes.

REFERÊNCIAS

- [1] E. André, T. Rist and J. Muller. Employing AI methods to control the behavior of animated interface agents. *Applied Artificial Intelligence*, Volume 13, Numbers 4-5, May 1999, 415-448.
- [2] G. Ball, D. Ling, D. Kurlander, J. Miller, D. Pugh, T. Skelly, A. Stankosky, D. Thiel, M. V. Dantzich and T. Wax. *Lifelike computer characters: the persona project at microsoft*. pages 191–222, 1997. Disponível em: <http://research.microsoft.com/research/pubs/view.aspx?pubid=439>.
- [3] C. Conati. Probabilistic assessment of user’s emotions in educational games. *Journal of Applied Artificial Intelligence*. 16 (7-8):555–575. 2002.
- [4] A. C. da R. Costa. A Teoria Piagetiana das Trocas Sociais e sua Aplicação aos Ambientes de Ensino-aprendizagem. In: *Informática na educação: teoria & prática*, vol.6, num.1, 2003.
- [5] S. D. Craig, B. Gholson and D. M. Driscoll. Animated Pedagogical Agents in Multimedia Educational Environments. In: *Journal of Educational Psychology*, 2002, Vol. 94, No. 2, 428-434.
- [6] A. De Angeli, P. Lynch and G. Johnson. Personifying the e-market: A framework for social agents. In: *Interact*

- 2001 - Eighth IFIP TC.13 Conference on Human-Computer Interaction, Tokyo, Japan, July 9-13, 2001.
- [7] C. D. Elliott, J. Rickel and J. Lester. Lifelike Pedagogical Agents and Affective Computing: An Exploratory Synthesis. In: M. Wooldridge and M. Veloso, editors, *Artificial Intelligence Today*, number 1600 in Lecture Notes in Computer Science, pages 195—212. Springer-Verlag, 1999.
- [8] D. Goldberg, D. Nichols, B. M. Oki and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, Vol 35, Num 12. Dezembro 1992.
- [9] J. E. Greer, G. McCalla, J. E. Cooke, J. Collins, V. S. Kumar, A. Bishop and J. Vassileva. “The Intelligent Helpdesk: Supporting Peer-Help in a University Course”, in *Proc. of the International Intelligent Tutoring Systems Conference (ITS’98)*, San Antonio, TX, 1998, pp. 494-505.
- [10] J. E. Greer, G. McCalla, J. Vassileva, R. Deters S. Bull and L. Kettel. Lessons Learned in Deploying a Multi-Agent Learning Support System: The I-Help Experience, *Proceedings of AI in Education AIED’2001*, San Antonio, IOS Press: Amsterdam, 410-421. 2001.
- [11] B. Hayes-Roth, V. Johnson, R. V. Gent and K. Wescourt. Staffing the web with interactive characters. *Commun. ACM*, 42 (3):103–105. 1999.
- [12] J. Herlocker, J. Konstan and J. Riedl. Explaining Collaborative Filtering Recommendations. In *Proceedings of ACM 2000 Conference on Computer Supported Cooperative Work*, December 2-6, 2000.
- [13] J. C. Lester, J. L. Voerman, S. G. Towns and C. B. Callaway. Cosmo: A life-like animated pedagogical agent with deictic believability. In *Proc. of the IJCAI97 Workshop on Animated Interface Agents: Making them Intelligent*. Nagoya, Japan, August 23-29. 1997.
- [14] H. Maturana and F. Varela. *Árvore do conhecimento*. Ed. Psy, Campinas, 1995.
- [15] J. Piaget. *Estudos Sociológicos*. Rio de Janeiro: Forense. 243p. 1973.
- [16] R. Picard. *Affective Computing*. Cambridge: MIT Press, 1997. 262 p.
- [17] E. Reategui, R. Torres and J. A. Campbell. Using Item Descriptors in Recommender Systems. In: Nineteenth National Conference on Artificial Intelligence, 2004, San José, CA, Estados Unidos. *Proceedings of the Workshop on Semantic Web Personalization - AAAI04*, 2004.
- [18] E. Reategui, E. Boff and R. M. Vicari. Proposta e Avaliação Preliminar de um Assistente Virtual para Recomendação de Conteúdos. In: *Simpósio Brasileiro de Informática na Educação*, 2005, Juiz de Fora. *Proceedings XVI Simpósio Brasileiro de Informática na Educação*, 2005.
- [19] R. Rickenberg and B. Reeves. The Effects of Animated Characters on Anxiety, Task Performance, and Evaluations of User Interfaces. In *Proceedings of CHI 2000 - Human Factors in Computing Systems*. The Hague, Amsterdam, The Netherlands. 1-6 APRIL 2000.
- [20] B. Sarwar, J. Konstan, A. Borchers, J. Herlocker, B. Miller and J. Riedl. Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System. *Proceedings of the 1998 Conference on Computer Supported Cooperative Work*. New York, 1998.
- [21] J. Schafer, J. Konstan and J. Riedl. Recommender systems in e-commerce. In *Proceedings of ACM E-Commerce*, 1999.
- [22] E. Shaw, W. L. Johnson and R. Ganeshan. Pedagogical agents on the web. In: *AGENTS ’99: Proceedings of the third annual conference on Autonomous Agents*, pages 283–290, New York, NY, USA. ACM Press.
- [23] L. S. Vygotsky. *A Formação Social da Mente: o Desenvolvimento dos Processos Psicológicos Superiores*. Editora Martins Fontes, São Paulo. 1998.