

M-SYNCHROTALK: UMA FERRAMENTA PARA ENSINO A DISTÂNCIA BASEADA EM COMPUTAÇÃO MÓVEL E EXCLUSÃO MÚTUA

Berto de Tácio P. GOMES; Omar Andrés C. CORTES; Rafael Fernandes LOPES

Centro Federal de Educação Tecnológica do Maranhão (CEFET-MA),
Avenida Getúlio Vargas N° 04, Monte Castelo, São Luís-MA-Brasil, CEP 65.030-000
fone/fax: (98) 3218-9000

bertodetacio@gmail.com

{[@cefet-ma.br](mailto:omar_rafael)}

RESUMO

O objetivo deste artigo é descrever a implementação de algoritmos de exclusão mútua distribuída para coordenar a troca de mensagens de texto entre usuários de um Chat para dispositivos móveis denominado M-SynchroTalk. Esse Chat possibilita que tutores e aprendizes estejam em contato mesmo em situações de mobilidade.

A coordenação de mensagens de texto ajuda o tutor a organizar o ambiente virtual, visto que a comunicação é muito prejudicada quando vários usuários “falam” ao mesmo tempo. Os algoritmos mencionados sincronizam o acesso ao broadcast de mensagens de texto, de tal forma que apenas um aprendiz por vez tem acesso a esse recurso tido com região crítica no bate-papo.

Desenvolvido com linguagem de programação Java entre outras tecnologias livres e de código aberto, o M-SynchroTalk é ainda um software livre, favorecendo a redução custos de aquisição do sistema, o que pode ser um fator considerável para que seja adotado em cursos à distância. O sistema poderá ser utilizado em uma ampla variedade de dispositivos móveis e de sistemas operacionais, visto que as tecnologias utilizadas garantem a independência de plataforma.

Os resultados preliminares mostram que as soluções adotadas dão ao tutor de ensino-aprendizagem um maior controle sobre o ambiente virtual, de forma que este é capaz de torná-lo mais organizado, no entanto existe a possibilidade do controle ser feito sem a intervenção do tutor, através de processos computacionais, o que suscita discussões acerca de qual metodologia deverá ser a mais adequada no contexto da educação a distância.

Palavras-chave: Ensino a Distância, Computação Móvel, Chat, M-SynchroTalk.

1. INTRODUÇÃO

O desenvolvimento das redes de computadores e das telecomunicações aumentou a quantidade e a qualidade de serviços voltados para a Internet. Surgiram softwares em variadas plataformas de desenvolvimento que possibilitam a comunicação entre usuários de formas assíncronas (E-Mail e Fóruns) e síncronas (Chat, Videoconferência e Teleconferência).

O paradigma da computação móvel e o aumento das capacidades de memória e processamento dos dispositivos móveis contribuíram para que esses serviços também fossem utilizados num contexto cada vez mais comum neste século: a mobilidade durante as atividades, ou seja, a capacidade das pessoas executarem tarefas importantes do cotidiano ainda que estejam em pleno movimento.

O emprego do Chat em Educação a Distância (EAD), na sua forma mais convencional, pressupõe a utilização de computadores pessoais conectados a Internet. No entanto, essa ferramenta muito utilizada do dia-a-dia de estudantes e profissionais pode apresentar problemas sérios quando muitos usuários nas salas de bate-papo enviam mensagens muitas simultaneamente, dificultando a leitura e o entendimento acerca de: quem está falando? Com quem? e; Sobre o que?. Nesse contexto, percebe-se que o Chat requer estratégias para coordenar a troca de mensagens de texto como forma de tornar reuniões virtuais mais organizadas e produtivas.

Quando imaginado no contexto da computação móvel, as possibilidades de utilização do Chat aumentam significativamente, pois a presença desse sistema em celulares e PDAs tornam-no mais flexível, já que tutores e aprendizes poderão comunicar-se mesmo em situações de mobilidade.

Entretanto, a adaptação dessa ferramenta para que seja usada em dispositivos móveis, herda além das dificuldades mencionadas de coordenação, outros problemas adicionais como as limitações de conectividade e de tela desses dispositivos.

Um dos problemas, em uma situação real de bate-papo com celulares ou PDA's, consiste no fato de que muitas mensagens enviadas podem preencher rapidamente a tela do dispositivo, forçando uma rolagem de tela mais rápido do que a capacidade de leitura dos usuários. Provavelmente, nesses casos, muitas mensagens deixariam de ser lidas, o que poderia mudar completamente o contexto das idéias que estão sendo debatidas e atrapalhar o processo de ensino-aprendizagem.

Outra problemática que deve ser considerada é a perda de transmissão de dados, ou seja, a troca de mensagens pode ser interrompida em momentos de conexão intermitente, situação muito comum em redes *wireless*. Nesses casos, um Chat móvel necessita ainda de um serviço eficiente de entrega de mensagens off-line.

Diante do contexto apresentado, o objetivo deste artigo é mostrar contribuições para alguns dos problemas mencionados, que estão sendo tratados no desenvolvimento do Chat para dispositivos móveis denominado M-SynchroTalk. Para auxiliar na resolução dos problemas, lança-se mão de algoritmos de exclusão mútua distribuída para coordenar a troca de mensagens e reduzir o tráfego de dados na rede.

Para alcançar tal objetivo, este artigo encontra-se dividido da seguinte maneira: A seção 2 mostra pesquisas correlatas. A seção 3 trata de descrever a arquitetura e as funcionalidades do sistema M-SynchroTalk. A seção 4 descreve em detalhes a implementação dos algoritmos de exclusão mútua e explica ainda o funcionamento dos mesmos durante as sessões de bate-papo. A seção 5 mostra quais as tecnologias utilizadas e como elas favoreceram a implementação dos algoritmos implementados. A seção 6 descreve os testes do M-SynchroTalk para avaliação do sistema e discute a aplicação dos mesmos na educação a distância. Por fim têm-se as considerações acerca da pesquisa e os novos rumos do projeto.

2. TRABALHOS CORRELATOS

Muitas foram as tentativas de utilização de Chat's na educação a distância ao longo dos anos. No entanto, poucas pesquisas no sentido de solucionar os problemas relacionados a ela foram encontradas, especialmente quanto se trata da organização nas trocas de mensagens entre tutores e aprendizes.

Os trabalhos correlatos mostram sistemas desenvolvidos apenas para computadores pessoais e não abrangem a área da computação móvel, ou seja, não foram encontrados sistemas de Chat que pudessem ser utilizados a partir de celulares e pda's, ou que permitissem a integração entre estes e os computadores pessoais.

Utilizando-se de “tipos de entonação” para o ato de “pedir palavra” em sessões de bate-papo, Vahl Junior (2003) criou a ferramenta ChatEd, onde tinha mais chances “falar”, o usuário que escolhesse a entonação “responder para”, em contrapartida os usuários que escolhiam a entonação “falar para” ficavam mais distantes do topo da fila de prioridades.

O grande problema nesta primeira versão do ChatEd diz respeito a frustração que muitos usuários sentiam quando repentinamente eram recolocados para trás na fila ao invés de avançarem. Isso acontecia sempre que um usuário escolhia um tipo de entonação maior do que a de um outro usuário da fila.

O mecanismo utilizado pelo ChatEd até então não parece muito flexível, visto que o tutor de ensino-aprendizagem não possui métodos eficazes para dar a oportunidade de enviar mensagens a um aprendiz que tivesse “falado menos” ou que ainda não tivesse tido a oportunidade. Outra desvantagem é que o mecanismo poderia ser facilmente burlado, pois qualquer aprendiz poderia escolher a opção “responder para”, mesmo que de fato ela não estivesse respondendo a nenhuma mensagem para outro usuário.

Em uma segunda versão do ChatEd (Oeiras et al, 2004), adicionaram a essa ferramenta o controle de envio de mensagens baseado em cotas de tempo, ou seja, cada usuário teria um período determinado antecipadamente ao início da sessão de bate-papo, isso porque o tempo era definido no código-fonte da aplicação e não havia como modificar os valores após o início do sistema. Percebe-se que esse mecanismo, apesar de tentar ser mais justo, pois todos os usuários teriam a mesma quantidade de tempo para enviar mensagens, essa metodologia é inflexível, pois não dá ao tutor uma segunda opção ou chances de modificar o tempo de acordo com a conveniência do processo de ensino-aprendizagem.

Gomes e Cortes (2006) adotaram, pela primeira vez no Chat SynchroTalk, mecanismos de coordenação baseados em algoritmos de exclusão mútua distribuída que tiveram boa aceitação dos usuários durante os testes. A utilização desses algoritmos impedia que mais de um usuário por vez enviasse mensagens para o bate-papo e dava ao tutor de ensino-aprendizagem um papel de destaque na reunião virtual, pois este tinha privilégios para coordená-la.

No entanto, pelo fato de ainda ser uma ferramenta em desenvolvimento, a mesma ainda não possui formas de transição eficiente entre os mecanismos de coordenação implementados. Mais ainda, por se tratar de uma ferramenta que utiliza tecnologias voltadas para a utilização em computadores pessoais e servidores, o SynchroTalk não pode executar a partir de dispositivos móveis, devido as limitações de memória e processamento desses dispositivos, o que motivou o desenvolvimento do **M (Móvil) - SynchroTalk**, versão do SynchroTalk para celulares e pda's.

O M-SynchroTalk também implementa algoritmos de exclusão mútua distribuída. Porém, com melhorias em relação ao SynchroTalk. Além disso, o algoritmo que foram implementados e que apresentaram baixo desempenho na versão para computadores pessoais não estão na versão móvel.

3. O CHAT M-SYNCHROTALK

As tecnologias de computação móvel encontram-se em franca evolução e parecem destinadas a se transformar no novo paradigma dominante da computação atual e, provavelmente, das gerações futuras (Keegan, 2006).

Um dos principais responsáveis pela disseminação e uso de aparelhos celulares é o baixo custo dos aparelhos e dos serviços de telefonia móvel em comparação aos valores dos computadores e serviços de acesso à internet. Nos últimos anos, os celulares tiveram uma redução significativa de preço, ao passo que os preços dos computadores não acompanharam esta redução, mesmo considerando os valores atuais dos computadores de baixo custo no Brasil.

Marçal et al (2005) apontam alguns objetivos de utilização de dispositivos móveis na educação: (1) expandir o corpo de professores e as estratégias de aprendizado disponíveis; (2) desenvolver métodos inovadores de ensino utilizando novos recursos de computação e de mobilidade; e, (3) na promoção de inclusão digital.

Nesse contexto e diante do leque de possibilidades de sua utilização, foi desenvolvido o Chat Móvel M-SynchroTalk, cujo objetivo é dar ao tutor de ensino-aprendizagem formas de coordenar a reunião virtual, tornando as salas virtuais de bate-papo mais organizadas, e, portanto, tornando o processo de ensino-aprendizagem mais produtivo, comprovando que tecnologia da informação pode ser aplicada com sucesso no âmbito da educação a distância.

O grande diferencial do M-SynchroTalk em relação a outros Chat's, é que no M-SynchroTalk existe um protocolo social que define o tutor de ensino-aprendizagem como coordenador da reunião. Esse tutor poderá (caso necessário) utilizar estratégias de coordenação que partem de teorias de sistemas distribuídos, onde apenas um processo por vez deve ter acesso exclusivo a um dado recurso compartilhado na rede e tido como a região crítica (Tenenbaum, 2006), e que para garantir isso recorre à **Exclusão Mútua Distribuída**.

O M-SynchroTalk implementa dois dos três algoritmos mais comuns na literatura para o problema de exclusão mútua: o centralizado e o token-ring. Existe ainda na literatura um terceiro algoritmo denominado distribuído que fora implementado na versão do SynchroTalk para computadores pessoais obtendo baixo desempenho, por esse motivo não foi implementado na versão móvel. O M-SynchroTalk considera o recurso de broadcast de mensagens como sendo a região crítica e cada dispositivo móvel como um processo computacional.

A literatura não menciona a intervenção humana na execução dos algoritmos, mas neste trabalho foi considerado que o tutor não pode ser excluído do processo de gerenciamento, pois os fatores humanos na educação devem ser considerados. Uma menor quantidade de mensagens em fluxo ajuda ainda a amenizar o problema ocasionado pela limitação de tela, já que menos mensagens diminuem o tempo de rolagem dos *displays* e ainda facilita a leitura.

A arquitetura do M-SynchroTalk é composta de clientes móveis que se conectam a um servidor para ter acesso às funcionalidades: autenticação e autorização (Figura 1), envio (Figura 2) e recebimento de mensagens (Figura 3), mecanismos de coordenação (Figura 4), visualização da fila de prioridades para envio de mensagens (Figura 5), pedidos de autorização para envio de mensagens (Figura 6), entre outras. O M-SynchroTalk é dividido em perfis. Tutores e aprendizes são autorizados a diferentes serviços. Por exemplo, apenas os tutores podem criar salas de bate-papo, tem acesso aos mecanismos de coordenação e ainda podem mudar de mecanismo durante uma reunião.

O M-SynchroTalk é um software multiplataforma livre, que foi desenvolvido com tecnologias livres e de código aberto que irão permitir: (1) execução em um conjunto amplo de dispositivos móveis (celulares, pda's, etc.); (2) execução independentemente do sistema operacional; (3) disponibilidade do código fonte para pesquisas e melhorias do sistema.



Fig. 1. Autenticação



Fig 2. Recebimento de Mensagens

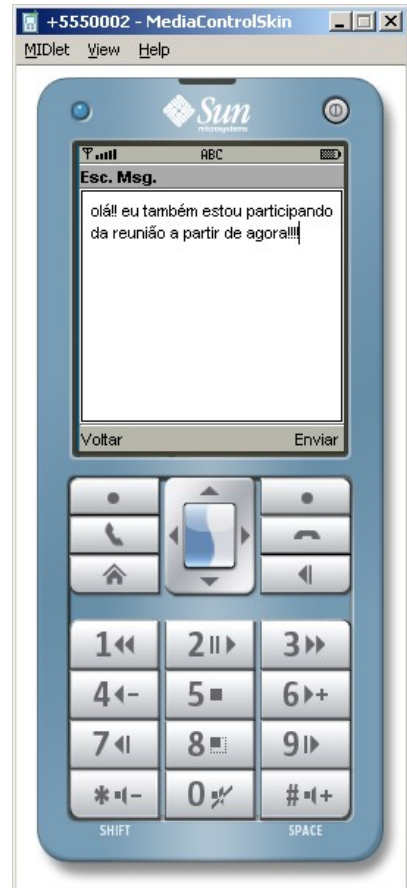


Fig. 3. Envio de Mensagens



Fig. 4. Modos de Coordenação



Fig. 5. Fila de Prioridades



Fig. 6. Menu: "Pedindo Palavra"

4. IMPLEMENTAÇÃO DA EXCLUSÃO MÚTUA

4.1. Algoritmo Centralizado

Em teoria, um algoritmo centralizado deve possuir os seguintes elementos: um coordenador central para controle de acesso a região crítica e uma fila de processos que aguardam um sinal do coordenador para ganhar o acesso exclusivo ao recurso desejado. Cada processo deve enviar uma solicitação para o coordenador, que atende aos pedidos por ordem de chegada. Apenas um processo por vez ganha acesso à região crítica, que só acontece após uma notificação do último processo que a utilizou, informando ao coordenador que este já pode atender o próximo da fila.

Baseado nesse princípio, o M-SynchroTalk implementa dois tipos de algoritmo centralizado: no primeiro um processo computacional exerce a função de coordenador. No segundo, esse papel é exercido pelo próprio tutor de aprendizagem.

Na primeira abordagem, chamada de **Centralizada Computacional**, o M-SynchroTalk utiliza uma *thread* para cada sala de bate-papo criada por um tutor autenticado e autorizado. Essa *thread* mantém um objeto fila (*Arraylist*) para armazenar as solicitações de acesso ao broadcast (região crítica do bate-papo) e, portanto, exerce a função de coordenador.

Todos os objetos mencionados encontram-se numa aplicação servidora, na qual os clientes (processos) deverão se conectar utilizando sockets/TCP-IP (por meio de redes wireless GPRS ou 3G). Em funcionamento, os clientes enviam solicitações a *thread* coordenadora que verifica se o objeto *usuarioAtual* (que representa o usuário com o acesso exclusivo) do tipo Cliente é *null*, o que significa para o algoritmo que nenhum cliente está usando o broadcast. Nesse caso, o próximo cliente é atendido e o objeto que o representa é removido do *Arraylist* de pedidos.

Cada cliente móvel mantém duas *threads* no servidor para que haja duas sessões: uma para recepção de dados no sentido cliente-servidor e outra para envio de dados no sentido servidor-cliente. As opções de “Pedir Palavra” (para alunos) e “Seleção do Modo de Coordenação” (para tutores) devem ser escolhidas no menu da tela de conversas do M-SynchroTalk (clientes).

Nos clientes existem mensagens de *feedback*, que permitem que os usuários tenham conhecimento do contexto de ações do Chat (ganhar palavra, perder palavra, pedido enviado, pedido recebido, etc.), sendo que cada usuário conhece sua posição na fila.

Por fim, o tutor pode definir o tempo de postagem de mensagens que variam entre um e quinze minutos. Por padrão o tempo é ilimitado. Assim que o tempo de postagem para cada cliente termina, a *thread* de coordenação atende o próximo cliente da fila automaticamente. Caso um aluno queira entregar a palavra antecipadamente, basta que este selecione a opção “Devolver Palavra”.

Na outra abordagem, chamada de **Centralizada Tutor**, o controle da *thread* de coordenação fica por conta do próprio tutor. O tutor visualiza no dispositivo móvel a fila de clientes que desejam acessar a região crítica. Para tornar essa implementação mais flexível o tutor pode (caso julgue conveniente) atender um determinado cliente independente de sua posição na fila. Essa decisão contraria de certa forma os princípios teóricos do algoritmo centralizado, mas pode ser benéfica para o contexto da aprendizagem, visto que o entendimento do tutor acerca do que pode ser melhor para o bate-papo está sendo considerado.

4.2. Algoritmo Token-Ring

Segundo a literatura, um algoritmo token-ring organiza todos os processos computacionais em forma de anel sem que haja um coordenador central. A autorização para acesso a região crítica é representada por um *token* (bastão) que circula pelo anel, de tal forma que apenas o processo que estiver com o bastão poderá acessá-la, garantindo assim a exclusão mútua. Quando um processo recebe o *token*, ele o retém caso queira acessar a região crítica. O processo que detém o *token* deverá

repassá-lo ao próximo participante do anel caso não queira acessar a região crítica ou caso já tenha terminado a utilização do recurso.

Baseado neste princípio foi implementado o modo de Coordenação Token Ring. Neste caso, todos os objetos que representam clientes móveis são organizados em forma lógica de anel (*Ring*). Em funcionamento, os alunos que desejam usar o broadcast de mensagens devem selecionar a opção “Pedir Palavra”. O envio de uma solicitação implica na entrada do cliente móvel que a enviou para o anel de usuários que pleiteiam o broadcast.

O anel de clientes na visão dos usuários é uma fila, na qual se desloca para o fim o último aprendiz que usou o broadcast, mas em termos de aplicação é uma lista encadeada circular.

5. TECNOLOGIAS

Para o desenvolvimento do M-SynchroTalk utilizou-se a linguagem Java (Sun, 2008) em duas de suas principais plataformas: **JavaSE e JavaME**. A linguagem Java é livre e de código aberto, sendo que em todas as suas plataformas permite o desenvolvimento de softwares segundo o modelo da orientação a objetos (OO), adotado por diversas empresas de software. A linguagem Java é uma das mais utilizadas no mundo, tornando-se referência para o desenvolvimento de sistemas.

A plataforma JavaSE é destinada ao desenvolvimento de sistemas desktop e por isso foi aplicada no desenvolvimento do servidor do M-SynchroTalk. Uma das vantagens da utilização da plataforma JavaME é independência de plataforma, uma vez que devido a existência da **Máquina Virtual Java** ou simplesmente JVM, sistemas desenvolvidos nessa plataforma podem ser executados em qualquer sistema operacional desde que possua uma JVM. Outro benefício da JVM é a maior segurança que esta proporciona, uma vez que esta impede a execução de código malicioso, ou seja, de aplicações que comprometam a segurança.

Para a organização dos usuários em na forma lógica de anel, foi utilizado o framework ProActive (Objectweb, 2008), desenvolvido em JavaSE e que permite acessar objetos a esquerda ou a direita do anel, o que permitiu a implementação do algoritmo token-ring.

A plataforma JavaME é destinada ao desenvolvimento de sistemas voltados para computação móvel e foi utilizada no desenvolvimento dos clientes M-SynchroTalk. Uma das vantagens da utilização da plataforma JavaME é que pode ser aplicada a uma grade variedade de dispositivos móveis, logicamente, desde que os dispositivos atendam a perfis e configurações de cada máquina virtual, conhecidas como KVM's, que impedem ainda o acesso a dados pessoais, como por exemplo, a agenda.

Para o M-SynchroTalk utilizou-se a configuração **CLDC 1.0** e o perfil **MIP 2.0**, que definem dispositivos com no mínimo 128 kilobytes de memória para executar aplicações Java, no mínimo 32 kilobytes de memória para a alocação de memória em tempo de execução, conectividade de rede sem fio com largura de banda baixo e acesso intermitente.

A conectividade entre cliente e servidor é realizada utilizando a API de sockets de ambas as plataformas, que permitem entre outras, o estabelecimento de conexões mais persistentes como TCP/IP, adotada pela maioria das aplicações de troca de mensagens em tempo real. Apesar do bom desempenho de *sockets* sem a utilização de **Chamadas de Procedimento Remoto (RPC)**, o tempo de desenvolvimento foi maior do que o gasto no desenvolvimento da versão desktop, pois o perfil CLDC não dá suporte à reflexão computacional, presente na maioria das tecnologias RPC.

Para a persistência das mensagens off-line foi utilizado o banco de dados relacional FireBird (FirebirdSQL, 2008) e o framework Hibernate (Hibernate, 2008), que permite realizar a persistência dos dados de forma orientada a objetos (ainda que o banco de dados seja relacional). Como ferramenta de desenvolvimento foi utilizada a IDE Netbeans (Netbeans, 2008), que fornece um conjunto de emuladores de dispositivos móveis para a realização de testes. Vale destacar que todas as tecnologias escolhidas no desenvolvimento deste trabalho são livres.

6. TESTES E DISCUSSÕES DOS RESULTADOS

Para analisar o comportamento da ferramenta M-SynchroTalk foram realizadas simulações de reuniões virtuais educativas com alunos e professores voluntários do ensino médio, nas quais os mesmos deveriam fazer sugestões de alterações no sistema e ainda compará-lo com a ferramenta ChatEd. Cada simulação contava com a participação de um professor e dez alunos.

Para realizar as simulações foram utilizados emuladores de dispositivos móveis fornecidos pela ferramenta Netbeans em um laboratório com microcomputadores em rede cabeada. Recorreu-se a utilização de emuladores não pela carência de dispositivos móveis, mas sim pela falta de uma infraestrutura de redes wireless do tipo GPRS ou 3G.

Em termos de usabilidade, o consenso dos usuários foi que a ferramenta M-SynchroTalk expõe as funcionalidades de forma bastante simples, uma vez que todos já estavam acostumados à utilização de celulares. Além disso, os usuários não encontraram dificuldades para realizar as tarefas de autenticação e trocar mensagens entre si.

Naturalmente, os alunos chegaram à conclusão de que a conversa não flui com a mesma dinamicidade entre dispositivos móveis quando comparado à troca de mensagens utilizando computadores pessoais. Porém, os usuários acreditaram que a mobilidade é muito importante visto que é cada vez maior a necessidade das pessoas de estarem em movimento.

Em comparação com a ferramenta ChatEd (mesmo esta não sendo voltada para dispositivos móveis), o consenso foi que os mecanismos de coordenação implementados pelo M-SynchroTalk são mais eficientes pelos seguintes motivos:

- I. O tutor de ensino-aprendizagem não fica descaracterizado dentro da reunião virtual. Com o M-SynchroTalk ele ganha um papel de destaque podendo intervir no curso da reunião, sendo um agente na organização e não um dependente de mecanismos de coordenação que trabalhem sem o seu consentimento.
- II. No ChatEd a troca de posições na fila de alunos que desejam a palavra é muito intensa, o que confunde os usuários e gera muita frustração. Já no M-SynchroTalk a fila de prioridades é respeitada pela ordem de chegada dos pedidos. Apesar da flexibilidade que o tutor tem para escolher um aluno com menor prioridade na fila, pode-se dizer que ainda assim a permuta de usuário é menor, pois depende apenas do tutor e não dos usuários como acontece no ChatEd quando as posições se alteram dependendo das entonações. Por esse motivo, os mecanismos implementados pelo M-SynchroTalk foram considerado também mais justos.
- III. A possibilidade de determinar o tempo na ferramenta M-SynchroTalk é mais flexível, visto que o tutor de ensino-aprendizagem pode a qualquer hora, de acordo com a conveniência, alterar as cotas de tempo para envio de mensagem. Na ferramenta ChatEd o tempo não pode ser alterado depois do início da sessão de bate-papo.

Apesar dessas vantagens, a avaliação mais importante nessa etapa da pesquisa é a avaliação dos algoritmos de exclusão mútua distribuída que foram implementados. Todos os algoritmos (centralizado e token-ring) mostraram-se capazes de auxiliar o tutor na tarefa de organizar melhor o ambiente.

Alunos e professores concordaram que o algoritmo centralizado com intervenção do tutor é o mais eficiente, pois supõe uma participação mais ativa do tutor, que por unanimidade é uma figura que deve ser destacada.

Quanto aos demais algoritmos, apesar de contribuírem também para uma melhor organização, alguns voluntários afirmaram que em certos casos o uso dos mesmos pode dar a impressão de que o tutor não estaria “prestando atenção”, ainda que a utilização destes não impeça o tutor de participar

ativamente, visto que todos os mecanismos se aplicam apenas aos aprendizes. Por outro lado, também é possível alterar o algoritmo utilizado de acordo com as necessidades.

Porém todos os voluntários foram unânimes em dizer que as demais formas de organização (token-ring e centralizado computacional) poderão ser importantes em períodos de falha do tutor. Sugeriram então que os mesmos poderiam ser utilizados como alternativa caso o tutor falhe, o que ainda não acontece.

Tal como é encontrado na literatura, os testes mostraram ainda que os algoritmos implementados são vulneráveis quando expostos a certas condições de falhas de hardware ou software, e, portanto, deve-se tomar cuidado para evitar as situações de falhas ou ainda utiliza-se de mecanismos de tolerância a falhas como forma de minimizar os efeitos produzidos por elas.

No algoritmo centralizado, estuda-se no momento uma forma de se poder adotar um novo coordenador caso o tutor falhe, ou ainda, migrar automaticamente para o algoritmo centralizado controlado por processo computacional ou mesmo o token-ring, de forma que logo após a recuperação do tutor este venha a retomar a coordenação do Chat.

7. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Do ponto de vista pedagógico o M-SynchroTalk representa uma ferramenta que em um futuro próximo poderá servir de apoio a tutores e alunos de cursos de ensino a distância, colaborando-se assim para a expansão dessa modalidade de ensino e colaborando na inclusão digital.

O M-SynchroTalk poderá significar um elo entre tutores e aprendizes em momentos que teoricamente os mesmos poderiam estar ociosos, passando a ocupar o tempo livre com a discussão de temas relevantes de seus cursos ou realizando atividades em grupo.

Do ponto de vista tecnológico, o M-SynchroTalk permitiu o contato com diversas tecnologias disponíveis no mercado, todas elas livres e de código aberto. O impacto dessa utilização será a disponibilidade do M-SynchroTalk de forma gratuita, fortalecendo a política de software livre.

O desenvolvimento de um Chat móvel não é uma tarefa trivial devido às inúmeras limitações dos dispositivos móveis. A plataforma Java ME mostrou-se adequada para o desenvolvimento, pois é possível reutilizar código e é independente de plataforma, o que permitirá integrar clientes móveis, desktop e web com maior facilidade.

A aplicação de algoritmos de exclusão mútua para organizar o bate-papo virtual é uma estratégia inovadora, visto que existem poucos trabalhos relacionados no campo da computação móvel. Os resultados preliminares mostram que os mesmos foram relativamente bem aceitos, embora uma avaliação mais formal ainda dependa de uma infra-estrutura de rede wireless para os testes.

O uso de algoritmos de exclusão mútua mostra ainda que certas teorias dos sistemas distribuídos, principalmente no tocante a sincronizações e formas de organização, podem ser aplicados no desenvolvimento de software. No caso do M-SynchroTalk, a aplicação sofreu adaptações em função dos fatores humanos existentes no processo de ensino-aprendizagem, mostrando que levar em consideração a importância dos agentes educacionais de aprendizagem pode fazer grande diferença no desenvolvimento de softwares educacionais, principalmente quando envolvem interação social.

Apesar do M-SynchroTalk e do SynchroTalk (Desktop) serem projetos paralelos, ainda não existe integração para a troca de mensagens entre as duas plataformas, o que é algo desejável no futuro a fim de aumentar a flexibilidade de utilização de uso do sistema em reuniões virtuais.

Outro ponto a ser tratado é a implementação de mecanismos de tolerância a falhas, que trará uma maior confiabilidade ao sistema diante de problemas de hardware ou de software.

8. AGRADECIMENTOS

Os autores agradecem o apoio financeiro do CNPq sem o qual o desenvolvimento deste trabalho não seria possível.

9. REFERÊNCIAS

VAHL JÚNIOR, J. C. **Uso de agentes de interface para adequação de bate-papos ao contexto de educação à distância**. Campinas: Unicamp, 2003. 146 p. Dissertação (Mestrado em Ciência da Computação).

OEIRAS, J. Y. Y.; ROCHA, H. V. **Uma Ferramenta de bate-papo com mecanismos de coordenação para apoio a discussões on-line**. In: XV Simpósio Brasileiro de Informática na Educação, 2004, Manaus-AM. Anais: Instituto de Computação da Universidade de Campinas.

GOMES, Berto de T. P; CORTÊS, Omar Andrés. C. **Sincronização em uma Aplicação Distribuída para Educação a Distância usando Recursos do Middleware ProActive**. VII Workshop em Sistemas Computacionais de Alto Desempenho, 2006, Anais. Ouro Preto-MG.

KEEGAN, D. **The future of learning: From eLearning to mLearning**, 2006. Disponível em: http://learning.ericsson.net/mlearning2/project_one/book.html.

MARÇAL et. al. **Aprendizagem utilizando Dispositivos Móveis com Sistemas de Realidade Virtual**. In RENOTE: Revista Novas Tecnologias na Educação: V.3 Nº 1, Maio, Porto Alegre: UFRGS, Centro Interdisciplinar Tecnologias na Educação, 2005.

TANENBAUM, Andrew S, **Sistemas Operacionais Modernos**, São Paulo: Prentice Hall, 2006.

SUN: **Web site das tecnologias Java**: <http://www.sun.com>, ultimo acesso 01 de agosto de 2008.

OBJECTWEB: **Web site do Grid Middleware Proactive**: <http://www.objectweb.org/ProActive>, último acesso 25 de junho de 2008.

FIREBIRDSQL: **Web site do Banco de Dados Firebird**: <http://www.firebirdsql.org/>, último acesso 01 de agosto de 2008.

HIBERNATE: **Web site do framework Hibernate**: <http://www.hibernate.org/>, último acesso 01 de agosto de 2008.

NETBEANS: **Web site da IDE Netbeans**: <http://www.netbeans.org/>, último acesso 01 de agosto de 2008.