



MÉTODOS DE SINCRONIZAÇÃO EM UMA FERRAMENTA COMPUTACIONAL AUXILIAR PARA O ENSINO A DISTÂNCIA.

Berto de Tácio Pereira Gomes

Curso de Licenciatura em Informática.

Centro Federal de Educação Tecnológica do Maranhão – CEFET-MA
Av. Getúlio Vargas s/n – Monte Castelo CEP 65.000-000 - São Luís-MA
E-mail: bertodetacio@gmail.com

Omar Andrés Carmona Cortês

Departamento Acadêmico de Informática.

Centro Federal de Educação Tecnológica do Maranhão – CEFET-MA
Av. Getúlio Vargas s/n – Monte Castelo CEP 65.000-000 - São Luís-MA
E-mail: omar.carmona@gmail.com

RESUMO

O objetivo deste trabalho é apresentar algumas formas de sincronização em reuniões virtuais. No contexto de educação à distância, a sincronização é utilizada para coordenar a comunicação entre membros de uma reunião virtual, sendo que a falta de sincronização pode vir a atrapalhar o processo de ensino-aprendizagem se vários núcleos de conversas são gerados em paralelo. Os algoritmos de sincronização implementados são baseados na sincronização utilizada em sistemas distribuídos. Nesse contexto, três algoritmos foram utilizados: centralizado, distribuído e *token-ring*. Os aspectos técnicos e pedagógicos dos algoritmos envolvidos também serão discutidos.

PALAVRAS-CHAVE: Sincronização, Sistemas Distribuídos, Bate-papo, Educação à Distância.

1. INTRODUÇÃO

A educação a distância começou a ser amplamente utilizada na década de 50, quando o principal meio de comunicação era o correio tradicional (Litwin, 2001). Com o advento dos computadores e especialmente da internet, as possibilidades para efetivar e incentivar o ensino à distância aumentaram significativamente, pois dependendo da forma como o curso está estruturado, muitas vezes os gastos com equipamento nas cidades afastadas é zero, pois o curso pode ser acompanhado pelo estudante a partir do seu próprio computador, na sua casa ou escritório. Além disso, o estudante pode acompanhar o curso nos horários que mais lhe convier.

Apesar da flexibilidade de horários e de local proveniente dos cursos à distância, encontros periódicos através da Internet podem também ser solicitados. Nesse caso, uma das ferramentas de baixo custo mais utilizadas é o bate-papo, que permite realizar reuniões virtuais entre alunos e professores. Entretanto, essa ferramenta pode causar sérias dificuldades quando, por exemplo, alunos passam a discutir outros assuntos em paralelo ou quando todos “falam” ao mesmo tempo. Quando isso ocorre, o professor tentará controlar a situação, embora isso vá depender exclusivamente da colaboração dos alunos. Nesse contexto, observa-se que apesar da evolução do uso da tecnologia na educação, a sua aplicação não é garantia de qualidade (Stafford, 2005). Sendo assim, nota-se que este tipo de ferramenta carece de mecanismos que coordenem o acesso ao recurso de envio e recebimento de mensagens. Oeiras e Rocha (2000), Pimentel e Sampaio (1999), entre outros, mostram as dificuldades encontradas em ferramentas de bate-papo quando vários usuários enviam mensagens simultaneamente. Dessa forma, percebe-se nitidamente a necessidade de controlar o acesso ao recurso de envio de mensagens.

Os trabalhos de Smith e Burkhalter (2002) e Pimentel e Sampaio (2001), por exemplo, utilizam um mecanismo para sincronizar as conversas onde o usuário deve clicar na mensagem que deseja responder. Essa metodologia demonstrou-se ineficiente, pois os usuários tiveram dificuldades em se adaptar a nova forma de conversar. Já, no trabalho de Vahl Junior (2003), apresentam-se modelos para organizar as conversas nos seguintes moldes: assembléia, seminário, tradicional e personalizado. Os modelos assembléia e seminário exigem que o usuário peça a palavra e sua requisição é colocada em uma fila. A forma tradicional é a mesma dos bate-papos comuns e a personalizada mistura os demais modelos. O problema apresentados neste trabalho é que os usuários ficavam perdidos com relação ao seu próprio posicionamento na fila. No trabalho de Oeiras e Rocha (2004) é introduzida a figura do coordenador para sincronizar a comunicação. Na maioria dos casos pesquisados, observa-se que os participantes têm dificuldades na utilização do bate-papo, pois os mecanismos utilizados não são transparentes.

Nesse contexto, vislumbrou-se que o problema de sincronização em bate-papo é semelhante ao encontrado em sistemas distribuídos para o tratamento de exclusão mútua, ou seja, vários processos (programas em execução) tentam acessar o mesmo recurso (Tenenbaum, 2003). No caso da ferramenta de bate-papo, a região crítica é considerada como sendo o ato de “falar”. Nesse âmbito, este artigo pretende contribuir com uma proposta para aplicação dos algoritmos de sincronização encontrados na teoria dos sistemas distribuídos (Coulouris, 2000) em uma ferramenta de bate-papo direcionada à educação à distância. O software implementado é denominado *SynchroTalk*. Além disso, os mecanismos implementados são transparentes aos usuários, sendo que o máximo esforço de um usuário é pedir a palavra.

Para cumprir seu papel este artigo está dividido da seguinte maneira: a Seção 2 apresenta os mecanismos de sincronização utilizados; a Seção 3 justifica o porquê das tecnologias utilizadas; a Seção 4 descreve as funcionalidades e a arquitetura do *SynchroTalk*; na Seção 5 discutem-se os resultados encontrados até o presente momento; a Seção 6 mostra as conclusões deste trabalho; finalmente, a Seção 7 aborda os trabalhos futuros.

2. MECANISMOS DE SINCRONIZAÇÃO

Para realizar a sincronização na ferramenta *SynchroTalk*, fez-se uma analogia ao problema de exclusão mútua em sistemas distribuídos, onde apenas um processo por vez pode ter acesso a um mesmo recurso, tido como região crítica. O recurso de envio de mensagens é considerado como sendo a região crítica do *SynchroTalk*, para o qual foram adotados algoritmos de sincronização similares aos que podem ser utilizados em sistemas distribuídos. Os algoritmos utilizados foram: centralizado, distribuído e *token ring*. Para cada algoritmo criou-se um mecanismo análogo aos quais são detalhados nas próximas seções.

2.1 Mecanismo Centralizado

No algoritmo centralizado há um coordenador para controlar o acesso à região crítica. Um processo antes entrar na região, envia uma mensagem de solicitação de recurso ao coordenador, que coloca todas as mensagens recebidas em uma fila. O processo solicitante deverá aguardar a resposta do coordenador, que verifica se algum outro processo

ocupa o recurso solicitado. Quando um processo deixa a região crítica, envia uma mensagem ao coordenador, informando que liberou o recurso o que permite que o coordenador atenda o próximo pedido.

Baseado neste algoritmo, o *SynchroTalk* contém um mecanismo centralizado de controle, onde há um coordenador humano e/ou processo computacional para controlar o acesso ao recurso de envio de mensagens. Os participantes agem como se fossem processos. Para qualquer que seja o coordenador, pode-se determinar o tempo que os participantes terão para postar suas mensagens, neste caso, o recurso será liberado automaticamente após o término do tempo, permitindo que o coordenador atenda a outra solicitação. No contexto educativo, o coordenador poderá ser um professor ministrando uma aula à distância. Esse professor poderá definir um aluno, para fazer perguntas ou permitir exposições de opinião, com o apoio deste mecanismo.

2.2 Mecanismo Distribuído

O segundo mecanismo de sincronização aplicado no *SynchroTalk* é semelhante a um algoritmo distribuído. Neste algoritmo, quando um processo deseja entrar na região crítica, ele envia uma mensagem com seu pedido e o tempo atual para todos os outros e para si mesmo. Quando um processo recebe uma mensagem desse tipo, ele reagirá de acordo com o seu estado podendo:

- I. Enviar uma mensagem de OK ao processo emissor caso não esteja na região crítica e não queira entrar nela;
- II. Não responder a mensagem recebida e inserir-la numa fila caso esteja na região crítica;
- III. Caso o processo deseje entrar na região crítica e ainda não o tiver feito, comparar o tempo da mensagem recebida, com o tempo constante da mensagem que ele enviou aos demais processos, enviando uma mensagem de OK como resposta a mensagem recebida se esta tiver um tempo menor que a mensagem enviada, caso contrário, ele não responde a mensagem e insere-a numa fila.

Na implementação do mecanismo distribuído, os participantes enviam mensagens especiais (diferentes das mensagens normais de um *Chat*) para os outros integrantes e para eles mesmos quando querem ter acesso ao recurso. Quando um participante recebe um pedido, seu *software* de *Chat* reage automaticamente de forma semelhante a um processo na mesma situação. Quando um participante envia seu pedido aos demais, ele só poderá ter acesso ao recurso quando receber de todos os outros uma mensagem de OK, devendo enviar uma mensagem de liberação quando não quiser mais enviar mensagens (mensagens normais), ou caso o tempo de envio termine (essa possibilidade foi adotada caso se queira evitar que um participante monopolize o recurso).

2.2 Mecanismo *Token-Ring*

No algoritmo *token ring* cria-se um anel de processos pelo qual circula um *token*. Quando um processo recebe o *token*, ele poderá entrar na região crítica caso queira (só poderá fazê-lo quem estiver com o *token*). O processo deverá “soltar” o *token* ao deixar a região crítica, ou passá-lo imediatamente caso não deseje acessá-la.

No *SynchroTalk*, o modelo análogo a esse algoritmo, organiza todos os participantes em um anel, pelo qual circula uma permissão (um agente representando o *Token*). Quando a permissão chega a um participante, ele a repassa ao próximo participante caso não queira postar mensagens, ou, a retém caso contrário, e deverá solta-lá ao término da utilização do recurso. Semelhante aos demais mecanismos, pode-se determinar o tempo de retenção da permissão, o que permite que a mesma seja repassada periodicamente.

3. TECNOLOGIAS UTILIZADAS

3.1 Linguagem de Programação

Para o desenvolvimento da ferramenta *SynchroTalk*, foi utilizada a linguagem de programação Java (Sun 2006). A adoção dessa linguagem permitiu desenvolver um *Software Desktop* capaz de funcionar independente de plataforma. Isto significa que tanto alunos quanto professores que utilizarem o *SynchroTalk* terão total liberdade na escolha do sistema operacional.

Apesar do uso da linguagem Java, fez-se necessário um ambiente de desenvolvimento que permitisse a utilização dessa linguagem de forma que os recursos oferecidos auxiliassem na qualidade e no tempo de desenvolvimento da ferramenta *SynchroTalk*. O ambiente de desenvolvimento adotado foi o Eclipse, o qual é apresentado na Figura 1.

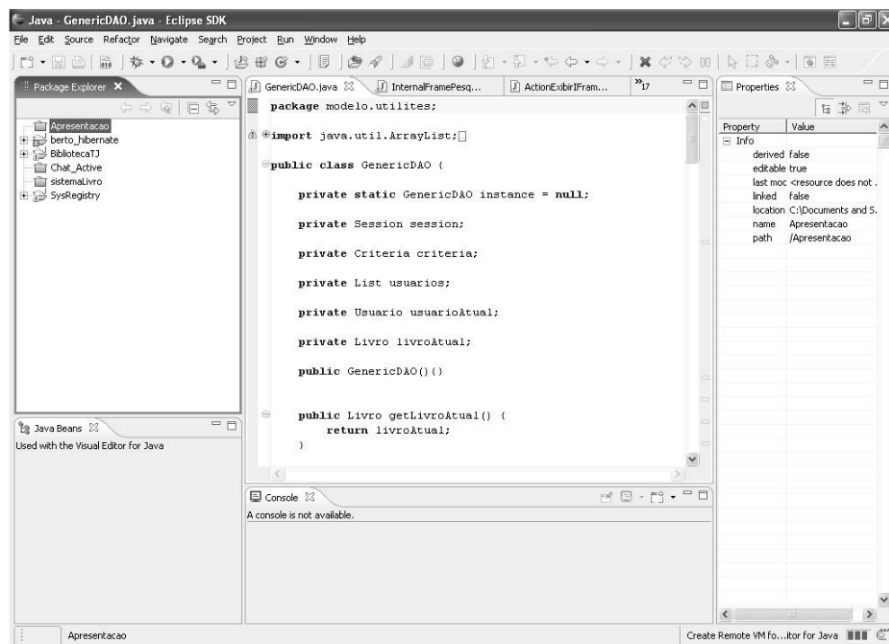


Figura 1 – Eclipse - Ambiente de Desenvolvimento utilizado no projeto SynchroTalk

3.2 Tecnologia de Comunicação a Distância

A linguagem Java permite que um computador execute métodos em outros computadores como se a chamada a esses métodos fosse local. Esse mecanismo é chamado de RMI (Remote Method Invocation – Chamada a Métodos Remotos) e é implementada com classes da biblioteca padrão do Java. Entretanto, essa tecnologia não permite o desenvolvimento de aplicações com um alto grau de abstração (Huet et al, 2004).

Além disso, os mecanismos de comunicação do RMI são ponto-a-ponto. Isso é adequado para a maioria das aplicações cliente-servidor. Entretanto, em um contexto onde o desempenho, a comunicação assíncrona e a comunicação coletiva devem estar acessíveis ao programador, a utilização do RMI não é suficiente. (Baduel et al, 2002)

Nesse contexto, decidiu-se pela utilização da tecnologia ProActive (ObjectWeb, 2006), a qual permite: realizar a comunicação em um alto nível de abstração e provê comunicação coletiva ao programador. A comunicação coletiva à distância é algo fundamental para a finalidade da ferramenta considerada, uma vez que a troca de mensagens entre pessoas geograficamente distantes é a base para qualquer software de conversação à distância.

Outro ponto de destaque é que a tecnologia ProActive é livre, e isto é um ponto chave no desenvolvimento de software, uma vez que tecnologias proprietárias implicam em maiores gastos para a obtenção dessas tecnologias, bem como o pagamento mensal de licenças de uso de software. O ProActive fornece ainda uma ferramenta de monitoramento capaz de verificar o fluxo de envio de mensagens e ainda conhecer a estrutura de distribuição e a situação do sistema, o que permite visualizar possíveis problemas na comunicação através da rede de computadores.

4. ARQUITETURA DO *SYNCHROTALK*

A arquitetura do software *SynchroTalk* é uma arquitetura distribuída, necessária para que pessoas geograficamente distante possam interagir utilizando os mecanismos de comunicação oferecidos pelo software. O sistema possui atualmente quatro componentes básicos: cliente, servidor, coordenador e monitor.

O software cliente, mostrado na Figura 2, é o software utilizado pelos usuários comuns, ou seja, os participantes do bate-papo. Ele contém os recursos que são necessários para o envio de mensagens e para a requisição de permissão para utilizar tal recurso.



Figura 2 – Software SynchroTalk Cliente

O software servidor, apresentado na Figura 3, é o responsável em distribuir as mensagens que serão enviadas a todos os participantes. Ele serve ainda como interceptador das requisições de recurso, que serão mostradas ao coordenador da reunião. Este servidor possui toda infra-estrutura necessária para controlar a reunião, visto que ele contém as permissões necessárias para bloquear ou banir um participante, dado que o mesmo armazena todas as informações sobre os participantes.

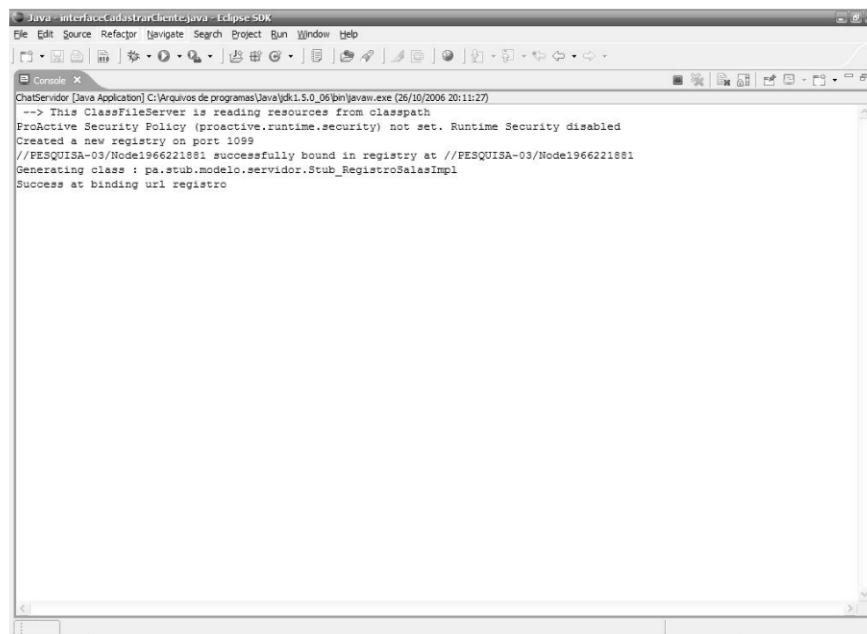


Figura 3 - Software Servidor do SynchroTalk funcionando no Ambiente Eclipse

Pelo fato de prover muitas funcionalidades, o servidor pode tornar-se um ponto de falhas, pois uma vez que este falhe as salas de bate-papo não podem ser mantidas. Por esse motivo, foi implementado também um mecanismo de

tolerância à falhas, que permita a aplicação mudar para um outro servidor disponível; ou permitir que cada software cliente envie suas mensagens diretamente ao destinatário, ou seja, sem o intermédio do servidor.

A utilização de um servidor ajuda a diminuir a carga dos clientes, pois ele se encarrega de replicar uma mensagem recebida para todos os destinatários, do contrário cada cliente teria que utilizar mais recursos computacionais para realizar a mesma tarefa, acarretando em uma diminuição do desempenho e da estabilidade da conversação. O uso de um servidor também aumenta a segurança, pois o mesmo pode controlar a autenticação de usuários, garantindo que apenas pessoas autorizadas participem do bate-papo.

O software administrador é apresentado na Figura 4. Ele é utilizado pelo coordenador de cada sala de bate-papo. Ele permite que um coordenador crie uma sala de bate-papo e seja responsável apenas pela sua própria sala. Ele contém a interface necessária à visualização de todos os participantes. Dessa forma o coordenador sabe quem está pedindo a palavra, podendo avaliar se este merece a concessão do recurso solicitado. Para impedir um aluno de enviar mensagens basta que o coordenador clique no botão bloquear. Para permitir que um usuário volte a enviar mensagens, basta escolher o participante e depois clicar em desbloquear.



Figura 4 - Software Coordenador do SynchroTalk

O software monitor é o IC2D que é mostrado na Figura 5. Com ele é possível conhecer o atual estado do sistema, visualizar o tráfego de informações na rede de computadores e ainda verificar falhas que possam ter ocorrido durante a comunicação e tomar as devidas providências.

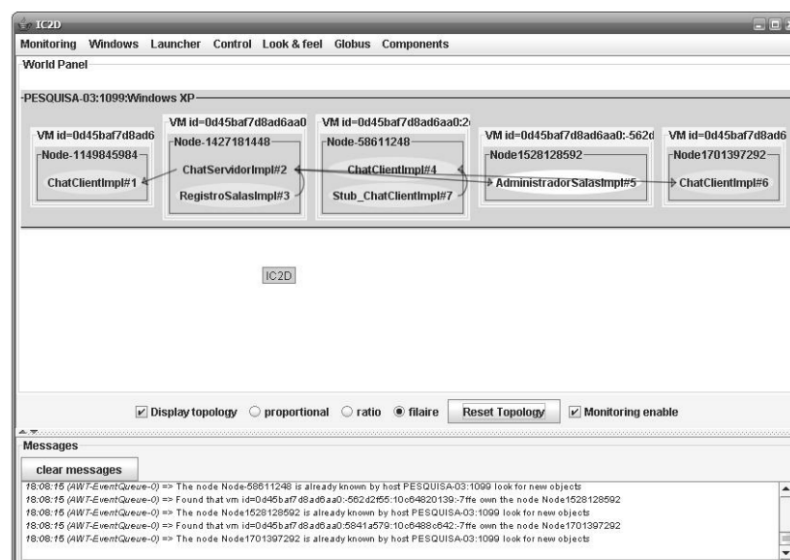


Figura 5 - Software IC2D. Monitorando o Sistema SynchroTalk

A divisão do sistema em módulos permite que os ajustes no sistema sejam feitos de forma descentralizada. Cada vez que um novo recurso tiver que ser inserido no *SynchroTalk*, existe apenas a preocupação em saber qual módulo será modificado, permitindo ainda um trabalho mais especializado em cada componente do sistema.

5. RESULTADOS E DISCUSSÃO

Os algoritmos foram avaliados de acordo com dois critérios: aspectos técnicos e aspectos pedagógicos. Ambos estão atrelados, principalmente porque os aspectos técnicos impactam diretamente nos aspectos pedagógicos.

O algoritmo centralizado possui duas vantagens claras: é relativamente simples de ser utilizado, e o coordenador pode a qualquer momento bloquear um ou mais participantes. Sob o ponto de vista pedagógico, isso representa algo desejado por parte dos educadores: um ambiente didático onde não haja problemas causados por alunos que conversam paralelamente às aulas ou falem todos ao mesmo tempo, provocando a falta entendimento. Por outro lado, um professor pode impedir um aluno de postar mensagens podendo ceder oportunidade a um aluno que participou menos. Baseado neste algoritmo, o *SynchroTalk* abre a possibilidade para que o coordenador seja tanto um humano (professor) quanto um processo computacional. No caso de um humano (professor) ele pode priorizar mensagens de um aluno, por exemplo, o que estiver participando de maneira mais ativa e esteja contribuindo com o debate.

O algoritmo centralizado apresenta também uma desvantagem: o coordenador por ser um ponto de convergência de requisições, pode tornar-se um ponto crítico de falhas dependendo da quantidade de participantes. Isso impacta diretamente no processo de ensino-aprendizagem, pois se o número de participantes for relativamente alto o professor pode não conseguir atender a todas as requisições; e de acordo com o fluxo de requisições, o coordenador poderia não suportar a carga e perder sua conexão com o servidor deixando a sala de bate-papo sem controle.

No mecanismo *token-ring*, o participante que deseja falar envia uma requisição. Se o token identificar que o participante pelo qual está passando enviou uma requisição, seu pedido será automaticamente atendido. Este mecanismo pode ser um alternativa caso o coordenador assuma uma posição mais passiva, como por exemplo, em atividades onde todos os alunos devem discutir pontos de vistas a respeito de um determinado assunto. Outra possibilidade de uso seria caso o coordenador precise ausentar-se temporariamente; ou se a reunião for apenas entre alunos; ou ainda, caso a conexão do coordenador com o servidor seja interrompida.

O mecanismo *token-ring* também possui algumas desvantagens, pois se num dado momento um aluno que estiver com a permissão (*token*) perder conexão com a rede, o mesmo não poderá ser movimentado para outros alunos, impedindo assim que os demais recebam permissões para falar. Uma alternativa para esse caso é criar um agente de emergência que circula na direção inversa a do *token* de certo em certo tempo. A movimentação na direção inversa visa encontrar o *token* mais rapidamente. Se após uma volta o agente token não for encontrado o agente de emergência cria um novo token.

Com relação ao algoritmo distribuído, deve-se mandar a requisição de pedido de palavra para todos os participantes quando se deseja falar. Controlar esse envio não é um processo trivial, pois participantes podem se desconectar a qualquer momento. Em teoria isso é uma desvantagem. Entretanto, esse problema foi solucionado com a utilização de grupos de comunicação e com a utilização de exceções não funcionais. Nesse contexto, quando um participante é desconectado os demais deixam de esperar por sua resposta.

Genericamente falando, o algoritmo distribuído funciona de maneira análoga ao algoritmo centralizado, desde que o coordenador seja um processo computacional. Em outras palavras, o primeiro que pedir a palavra a terá. Entretanto, perde-se a possibilidade de dar preferência a grupos de alunos. Isso pode comprometer pedagogicamente o debate se os alunos que conseguirem a palavra não contribuírem significativamente com o debate. Ressalta-se que a contribuição pode vir tanto na forma de afirmações quanto de questionamentos interessantes sobre o tema em discussão.

6. CONCLUSÃO

Do ponto de vista pedagógico o sistema de bate-papo proposto vem a contribuir com a educação à distância, pois além de permitir recursos normais de comunicação à distância, a ferramenta possui mecanismo de sincronização que permitem uma melhor organização dos encontros virtuais, uma vez que seus recursos de controle impedem que várias pessoas falem ao mesmo tempo. Nesse contexto, debates, aulas, discussões, entre outros tipos de atividades

cognitivas poderão ser desenvolvidas com certo grau de segurança caso falte à colaboração do(s) aluno(s). Sendo assim, o professor terá formas de garantir a organização do espaço virtual.

Não obstante, abre-se espaço para discussões acerca de como a educação à distância mostra caminhos para a expansão do ensino. No entanto, é necessário pensar além de números. A qualidade do ensino é um fator fundamental, pois a educação à distância deve visar muito mais que comunicação entre indivíduos geograficamente distantes, mas também prover maneiras de comunicação organizada que atenda as necessidades de professores e alunos.

Contemplando o ponto de vista tecnológico, a ferramenta oferece completa portabilidade, ou seja, a possibilidade de funcionamento em qualquer sistema operacional. Além disso, possibilita aos envolvidos neste trabalho experiências nas novas tecnologias disponíveis.

7. TRABALHOS FUTUROS

O sistema *SynchroTalk* continua em pleno desenvolvimento, sendo que a cada dia funcionalidades em potencial são pensadas e analisadas. Nesse contexto, observando-se que sistemas Web provêem uma maior facilidade de acesso à internet a usuários comuns, vislumbrou-se a transformação da aplicação considerada de um *Chat Desktop* para um *Web-Chat*, ou seja, possibilitar aos participantes o acesso ao bate-papo através de um site Web.

Sabe-se ainda que sistemas distribuídos estão sujeitos a falhas. Por esse motivo pretende-se implementar mecanismos que permitam que as reuniões virtuais mantenham um determinado nível de estabilidade, mesmo em caso de problemas na rede de computadores, tais como falhas em computadores remotos, perda do servidor ou alguma possível falha dos mecanismo de sincronização.

Pretende-se ainda fazer uma análise da qualidade da participação das pessoas. Isso poderá ser útil em uma aula distância, pois permitirá ao professor analisar o nível de participação dos alunos, baseado na qualidade das mensagens enviadas por cada aluno, facilitando assim o processo de avaliação.

Por fim, após melhorias na interface gráfica, o *SynchroTalk* será efetivamente testado por meio de aulas à distância, permitindo a verificação da eficiência das soluções e coletar informações acerca da opinião dos usuários sobre a facilidade de utilização do sistema.

Agradecimentos

Os autores agradecem o apoio financeiro dado pelo CNPq sem o qual este trabalho não seria possível.

8. REFERÊNCIAS BIBLIOGRÁFICAS

BADUEL, L.; BAUDE, F.; CAROMEL, D., **Efficient, Flexible and Typed Group Communications for Java**, Joint ACM Java Grande - ISCOPE 2002.

COULOURIS, George & DOLLIMORE, Jean & KINDBERG, Tim, *Distributed Systems: Concepts and Design*, Adson Wesley, 3^{ed}, 2000.

HUET, F.; CAROMEL, D.; BAL, H., **A High Performance Java Middleware with a Real Application**, SuperComputing Conference, November 2004.

LITWIN, E. **Educação à Distância: Temas para o Debate de uma Nova Agenda Educativa**, Artmed, Porto Alegre-RS, 2001.

OEIRAS, J. Y. Y.; ROCHA, H. V. **Uma modalidade de comunicação mediada por computador e suas várias interfaces**. In: WORKSHOP SOBRE FATORES HUMANOS EM SISTEMAS COMPUTACIONAIS, 2000, Gramado. Anais... Porto Alegre: Instituto de Informática da UFRGS, p. 151-160.

OEIRAS, J. Y.; LACHI, R. L.; da ROCHA, H. V. **Uma ferramenta de bate-papo com mecanismos de coordenação para apoio a discussões online**. In: Simpósio Brasileiro De Informática Na Educação, Manaus-AM, 2004.

OBJECTWEB: **Web site do Middleware Proactive**: <http://www.objectweb.org/ProActive>, ultimo acesso 25 de maio de 2006.

PIMENTEL, M. G.; SAMPAIO, F. F. **Hiperdiálogo uma ferramenta de bate-papo para diminuir a perda de contexto**. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – SBIE'2001, 2001, Vitória (ES). Anais: Vitória, nov. 21-23, 2001. p. 255-266.

SUN MICROSYSTEMS, <http://www.sun.com>, Visitado em 30/10/2006.

SMITH, M.; CADIZ, J. J.; BURKHALTER, B. **Conversation trees and threaded chats**. **SIGCHI Bulletin, Minneapolis**, v. 31, n. 3, p. 21-23, Jul., 1999. Capturado em 9 dez. 2002. On-line. Disponível em. <http://www.acm.org/sigchi/bulletin.3/morales.pdf>

TANENBAUM, Andrew S, **Sistemas Operacionais Modernos**, São Paulo: Prentice Hall, 2003.

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) - Instituto de Computação, Universidade Estadual de Campinas.