UNIVERSIDADE FEDERAL DO CEARÁ

DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA

CURSO DE GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

AMBIENTE VIRTUAL COLABORATIVO COM SUPORTE À TRANSMISSÃO DE ÁUDIO

Rodolfo Galvão Aurélio

Fortaleza - Ceará
Maio/2011

UNIVERSIDADE FEDERAL DO CEARÁ

DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA

CURSO DE GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

AMBIENTE VIRTUAL COLABORATIVO COM SUPORTE À TRANSMISSÃO DE ÁUDIO

Autor:

Rodolfo Galvão Aurélio

rodolfogalvao@ymail.com

Orientador:

José Marques Soares

marques.deti@gmail.com

Projeto de Final de Curso submetido à Coordenação do Programa de Graduação em Engenharia de Teleinformática da Universidade Federal do Ceará como parte dos requisitos para a obtenção do grau de Engenheiro de Teleinformática.

Fortaleza - Ceará

Maio/2011
Rodolfo Galvão Aurélio

AMBIENTE VIRTUAL COLABORATIVO COM SUPORTE À TRANSMISSÃO DE ÁUDIO

Este Trabalho foi julgado adequado para a obtenção do título de Engenheiro de Teleinformática e aprovado em sua forma final pelo Departamento de Graduação em Engenharia de Teleinformática da Universidade Federal do Ceará.

_____

Rodolfo Galvão Aurélio

Banca examinadora:

_____

Prof. Dr. José Marques Soares

Orientador

_____

Prof.

Universidade Federal do Ceará

_____

Prof.

Universidade Federal do Ceará

Fortaleza, 28 de abril de 2011

RESUMO

Dados os recursos tecnológicos atuais, é cada vez mais comum encontrarmos situações em que pessoas geograficamente distantes desempenham alguma atividade em conjunto, seja ela de caráter comercial ou educacional. Sistemas de áudio, videoconferência, chats e sistemas de navegação Web cooperativos são exemplos de aplicações que tornam possíveis essas atividades. No entanto, as ferramentas de colaboração não fornecem ao usuário uma real noção da presença de todos os demais usuários com os quais ele coopera, bem como de quais ações esses usuários estão realizando em um dado momento.

O advento dos mundos virtuais tridimensionais (3D) constitui um passo importante na evolução dos ambientes colaborativos. Neles, os usuários têm a sensação de estarem realmente se encontrando com os outros a partir da noção de presença proporcionada por seus representantes virtuais denominados avatares. Um ambiente virtual tridimensional aliado a recursos de áudio imersivo reforçam o processo de colaboração e torna muito mais eficaz o trabalho do que um sistema baseado em telas 2D padrão.

O ambiente virtual 3D, escolhido para ser usado como base para o desenvolvimento desse trabalho, além de oferecer a noção de presença citada anteriormente, permite o compartilhamento de aplicações dos usuários remotos ampliando a interação entres os participantes da conferência. Porém, apenas comunicação por texto (chat) está disponível aos usuários desse ambiente virtual o que prejudica o estabelecimento de comunicação eficiente.

Assim, o presente trabalho pretende elaborar uma infra-estrutura para conferência de áudio e incorporá-la ao ambiente virtual 3D.

ABSTRACT

Dados os recursos tecnológicos atuais, é cada vez mais comum encontrarmos situações em que pessoas geograficamente distantes desempenham alguma atividade em conjunto, seja ela de caráter comercial ou educacional. Sistemas de áudio, videoconferência, chats e sistemas de navegação Web cooperativos são exemplos de aplicações que tornam possíveis essas atividades. No entanto, as ferramentas de colaboração não fornecem ao usuário uma real noção da presença de todos os demais usuários com os quais ele coopera, bem como de quais ações esses usuários estão realizando em um dado momento.

O advento dos mundos virtuais tridimensionais (3D) constitui um passo importante na evolução dos ambientes colaborativos. Neles, os usuários têm a sensação de estarem realmente se encontrando com os outros a partir da noção de presença proporcionada por seus representantes virtuais denominados avatares. Um ambiente virtual tridimensional aliado a recursos de áudio imersivo reforçam o processo de colaboração e torna muito mais eficaz o trabalho do que um sistema baseado em telas 2D padrão.

O ambiente virtual 3D, escolhido para ser usado como base para o desenvolvimento desse trabalho, além de oferecer a noção de presença citada anteriormente, permite o compartilhamento de aplicações dos usuários remotos ampliando a interação entres os participantes da conferência. Porém, apenas comunicação por texto (chat) está disponível aos usuários desse ambiente virtual o que prejudica o estabelecimento de comunicação eficiente.

Assim, o presente trabalho pretende elaborar uma infra-estrutura para conferência de áudio e incorporá-la ao ambiente virtual 3D.

DEDICATÓRIA

Aos meus amados pais, Evaldo e Galvão, que sempre participaram de todos os momentos importantes na minha vida além de terem sido modelos para a formação do meu caráter. Ao meu irmão Rafael pelo companheirismo, DEDICO

AGRADECIMENTOS

- Aos meus pais, pelo apoio incondicional;
- Ao meu irmão, pelo exemplo de dedicação;
- A minha namorada, pela compreensão;
- A Família, pela força transmitida apesar da distância;
- Ao meu orientador, pela paciência;
- A todo o time do Laboratório de Engenharia de Sistemas de Computação (LESC) da Universidade Federal do Ceará (UFC) em especial aos Engenheiros: Jardel, Alexandre.
- E principalmente a Deus, pelo dom da vida.

“Para que levar a vida tão a sério
se ela é uma incansável batalha
da qual jamais sairemos vivos ?!"

Bob Marley

SUMÁRIO

LISTA DE FIGURAS

1. Introdução

O número de usuário com acesso à Rede Mundial de Computadores tem crescido bastante nos últimos anos. Segundo IBGE, entre 2005 e 2009, o número de domicílios com acesso à Internet no Brasil cresceu 71% [3]. Por isso, é cada vez mais comum encontrarmos pessoas geograficamente distantes desempenhando alguma atividade em conjunto, seja ela de caráter comercial, educacional ou mesmo empresarial. Aplicações ou ferramentas colaborativas, tais como sistemas de áudio, videoconferência, chats, sistemas de navegação Web cooperativos e ambientes virtuais colaborativos (AVC), são geralmente utilizadas como meio para promover tal interação. Porém, segundo Santos [4] as ferramentas de colaboração não fornecem ao usuário uma real noção da presença de todos os demais usuários com os quais ele coopera bem como quais ações esses usuários estão realizando em um dado momento.

As aplicações virtuais tridimensionais (3D) têm contribuído para a evolução dos ambientes colaborativos. Nelas, os usuários têm a sensação de estarem realmente se encontrando com os demais a partir da noção de presença proporcionada por seus representantes virtuais, denominados avatares [5]. Além disso, um ambiente virtual 3D aliado a recursos de áudio imersivo reforçam o processo de colaboração e torna o trabalho muito mais eficaz do que um sistema baseado em telas 2D padrão [7]. Porém, a diminuição da percepção das ações individuais de usuários é ainda um ponto crítico. Um fator importante que contribui para o aumento dessa dificuldade é a restrição, devido ao afastamento geográfico, de um dos recursos comunicativos mais importantes do homem: a comunicação gestual, também conhecida como comunicação não verbal [6].

Com o intuito de minimizar as limitações inerentes às ferramentas de trabalho colaborativo, a comunicação gestual sintética aliada à imersão de áudio ao ambiente ganha destaque neste trabalho.

1.1 Colaboração à distância e suas limitações

Os recursos de áudio, vídeo ou o simples texto, isolados ou combinados, são empregados comumente na comunicação entre colaboradores distantes que compartilham um mesmo espaço virtual. Uma dificuldade natural na construção de tais ambientes é a concepção de arquiteturas capazes de integrar, de maneira consistente, os diversos canais

de comunicação e os objetos compartilhados. Além disso, é necessário construir interfaces intuitivas e eficazes do ponto de vista perceptivo.

Segundo Schmidt [9], presumiu-se há tempos que o vídeo conjugado ao áudio poderia oferecer um meio de comunicação fluida e não restritiva entre interlocutores distantes, e que um ambiente compartilhado entre usuários remotos, dispondo de recursos multimídia, permitiria uma cooperação comparável àquela existente em um mesmo espaço físico. Todavia, como discutido por Peraya [11], nem os dispositivos técnicos de videoconferência, nem as numerosas aplicações para colaboração à distância permitem de fato a restituição das informações ligadas à percepção visual. Além dos problemas técnicos ligados à necessidade de gerir múltiplos fluxos de vídeo, pode-se ressaltar a dificuldade de integração dos diferentes elementos do espaço de compartilhamento em uma interface coesa, capaz de representar visualmente a inter-relação entre cada um dos elementos a cada instante. Em geral, estes elementos compreendem as janelas contendo o vídeo de cada usuário e os objetos compartilhados. Com o uso exclusivo do vídeo, em especial, nem sempre é possível perceber a origem das ações de colaboradores distantes sobre um objeto compartilhado, menos ainda acompanhar o desenvolvimento dessas ações. A posição da câmera, a qualidade do vídeo e a iluminação do ambiente são outros aspectos que podem também interferir na qualidade das interfaces baseadas em vídeo. Além disso, é necessário um fluxo de vídeo para cada participante conectado, o que pode exigir uma largura de banda nem sempre disponível.

1.2 Ambiente de trabalho colaborativo

Sistemas complexos podem oferecer sensação de imersão em ambientes virtuais, mas são, em geral, custosos e difíceis de serem construídos não sendo, portanto, razoável pensar atualmente em soluções desta natureza para o trabalho de grupo do dia-a-dia, como é o caso das aplicações remotas de cunho educacional.

O progresso constante dos computadores pessoais equipados com placas gráficas cada vez mais acessíveis e de maior capacidade de processamento permite a utilização da realidade virtual em sistemas conhecidos como não imersivos os quais o usuário visualiza o ambiente colaborativo, porém não tem a sensação de estar realmente inserido naquela realidade. Isso simplifica a implementação de ambientes virtuais com os quais o usuário pode interagir através dos dispositivos de entrada convencionais, como o mouse e o teclado. Este tipo de configuração começa a se apresentar como elemento comum em

laboratórios de escolas e centros de inclusão digital, o que representa uma possibilidade real de utilização de ambientes virtuais não imersivos em contextos educacionais. Nestes ambientes virtuais, a comunicação gestual pode ser estabelecida a partir de modelos sintéticos. Em forma de humanóides, tais modelos, freqüentemente articulados, podem ser animados, simulando ou reproduzindo de maneira simplificada os gestos humanos. As animações corporais aliadas às animações faciais sincronizadas com a voz tornam eficiente a interação entre os humanóides.

Com o uso da realidade virtual, imersiva ou não, é possível a construção de ambientes integrados, onde cada participante é representado por um objeto chamado avatar, dentro de um cenário que pode conter ainda as representações virtuais dos objetos compartilhados. Um ambiente virtual colaborativo que permite a percepção de interações com aplicações bidimensionais é proposto por Soares et al [1] [2] e otimizado por Anselmo [8]. Uma experimentação foi conduzida para inserir comunicação auditiva sincronizada aos movimentos corporais do avatar e publicada em [20]. Essa versão do ambiente virtual utilizava *Java Media Framework* (JMF) [14], mas não era estável e, além disso, os módulos relativos à comunicação por áudio não eram adequadamente integrados à infra-estrutura do ambiente, não permitindo uma sincronização eficiente entre características do sinal de áudio à animação dos avatares. Outras limitações podem ser elencadas:

- O JMF é uma *Standard Extension* e, como tal, não acompanha o Kit de Desenvolvimento Java, necessitando uma instalação adicional. Além disso, o desenvolvimento do JMF foi descontinuado pela SUN em 2001, ficando a sua manutenção na responsabilidade de uma comunidade [19];

- A instalação e configuração do JMF não são simples, apresentando freqüentemente erros de difícil manutenção. Além disso, não existe versão da JMF para Macintosh, o que limitaria a compatibilidade original do ambiente virtual;

- Na ocasião, o ambiente virtual apresentado em [20] não continha a animação da face de avatares, não havendo a preocupação com a sincronização da animação com o áudio.

Dessa forma, este trabalho apresenta o uso de uma tecnologia alternativa para tratamento do sinal de áudio com objetivo de mitigar as limitações apresentadas.

1.3 Objetivo

Ampliar o potencial de interação entre os participantes de uma conferência a distância que utilizam o Ambiente Virtual Colaborativo (AVC) tridimensional proposto por Soares et Anselmo [1] [2] [8] [18] através da inserção de comunicação auditiva e representação da origem da interlocução através de movimentos da face do *avatar*. Visa-se reduzir as limitações da dinâmica do trabalho em grupo ocasionadas pela perda de informação devido ao afastamento geográfico dos participantes, principal problema desse tipo de ambiente colaborativo.

Este trabalho tem como ponto de partida o protótipo desenvolvido por Anselmo [8], desenvolvido durante o projeto intitulado Comunicação Gestual a Distância com Humanóides Virtuais: Aplicações na Educação, fomentado com recursos do CT-INFO: CNPq 31/2004 – PDPG-TI de 2004 a 2006. Este projeto é referenciado ao longo deste texto pela sua sigla: COGEST.

Como objetivos específicos, elencam-se:

- Concepção e especificação dos módulos complementares da arquitetura para prover a comunicação por áudio integrada a movimentos da face;
- Implementar a função de comunicação privada entre dois interlocutores do espaço virtual;
- Implementar a função de comunicação em conferência entre todos os participantes do ambiente virtual.

1.4 Organização do texto

Este trabalho está dividido em seis capítulos, sendo este o que contextualiza a proposta, apresentando seus objetivos e também discute o tema Ambientes Virtuais Colaborativos, abordando as dificuldades encontradas no trabalho a distância. O ambiente virtual COGEST é apresentado no Capítulo 2, onde são detalhadas suas características, recursos disponíveis e tecnologias empregadas no seu desenvolvimento. Alem disso, os benefícios que a conferência de áudio sincronizada com animações faciais proporciona a dinâmica do trabalho em grupo são analisadas. No Capítulo 3 será apresentada a biblioteca de funções usada para manipulação de som em Java. O Capítulo 4 apresenta a solução

para distribuição de áudio adotada nesse trabalho além de apresentar como a proposta desse projeto está inserida ao ambiente virtual original do COGEST. É ainda apresentada a técnica para detecção de silêncio. No capítulo 5 é exposta a solução adotada para o estabelecimento de conversa reservada. Finalmente, o Capítulo 6 finaliza esse trabalho apresentando um resumo do trabalho proposto bem como analisa possíveis trabalhos futuros.

2. O ambiente virtual COGEST

Neste capítulo é apresentada a infra-estrutura para atividades colaborativas a distância desenvolvida por Anselmo [8] e utilizada como base para o desenvolvimento deste trabalho. Serão abordados seus aspectos funcionais e as tecnologias usadas na implementação do ambiente gráfico 3D.

O COGEST é um ambiente virtual tridimensional em que diversos meios de comunicação como áudio, vídeo e texto são integrados em um mesmo espaço e associados com a comunicação gestual. O compartilhamento de aplicações é realizado de forma que as ações do usuário são convertidas em gestos, reproduzidos por um modelo humanóide virtual, contribuindo para o aumento da percepção dessas ações. Ainda é possível a inserção, seleção e manipulação de objetos 3D no mundo virtual, possibilitando a presença de elementos interativos de acordo com tema escolhido. Diferentes formas de navegação pelo ambiente permitem que os usuários explorem o espaço virtual de maneira simples, facilitando a interação com todos os objetos disponíveis.

2.1 Visão Geral

Em sua concepção, o projeto considera cinco elementos básicos em sua arquitetura. São eles:

a) Canais clássicos de comunicação;

b) Ambiente virtual tridimensional;

c) Comunicação gestual;

d) Compartilhamento de aplicações;

e) Interface de comunicação e gestão do ambiente.

Os *canais clássicos de comunicação*, tais como áudio, vídeo e texto, juntamente com o *compartilhamento de aplicações*, são reunidos em um mesmo *espaço virtual* onde é agregada a eles a *comunicação gestual* através da representação dos humanóides virtuais. O gestor do ambiente colaborativo mantém a comunicação e o sincronismo entre os participantes da atividade colaborativa.

O protótipo desenvolvido consiste em um aplicativo do tipo cliente/servidor em que o lado cliente é formado por um ambiente virtual tridimensional onde são inseridos vários objetos que representam funcionalidades disponíveis. Os usuários conectam-se ao servidor e este trata de manter o ambiente sempre atualizado e sincronizado, de forma que cada um dos clientes possa visualizar o que os demais estão fazendo. Ao se conectar ao servidor, um determinado usuário tem um humanóide virtual como representante dentro do mundo 3D (seu objeto comunicativo). É a partir desse humanóide que o usuário se comunica gestualmente com os demais participantes (também representados por humanóides virtuais) através de animações corporais e faciais as quais podem ser pré-definidas ou em tempo real.

Dentro do ambiente 3D, o usuário também encontra outros objetos interativos como o quadro virtual, no qual é exibido um aplicativo qualquer que esteja sendo compartilhado, como um editor de texto ou uma ferramenta de desenho.

2.2 Interface Gráfica do Usuário

A interface gráfica do usuário (GUI), composta por painéis, botões, caixas de texto, permite o controle das configurações e acesso às funcionalidades do ambiente. Na Figura 2.1 são mostrados os elementos integrantes dessa interface que são:

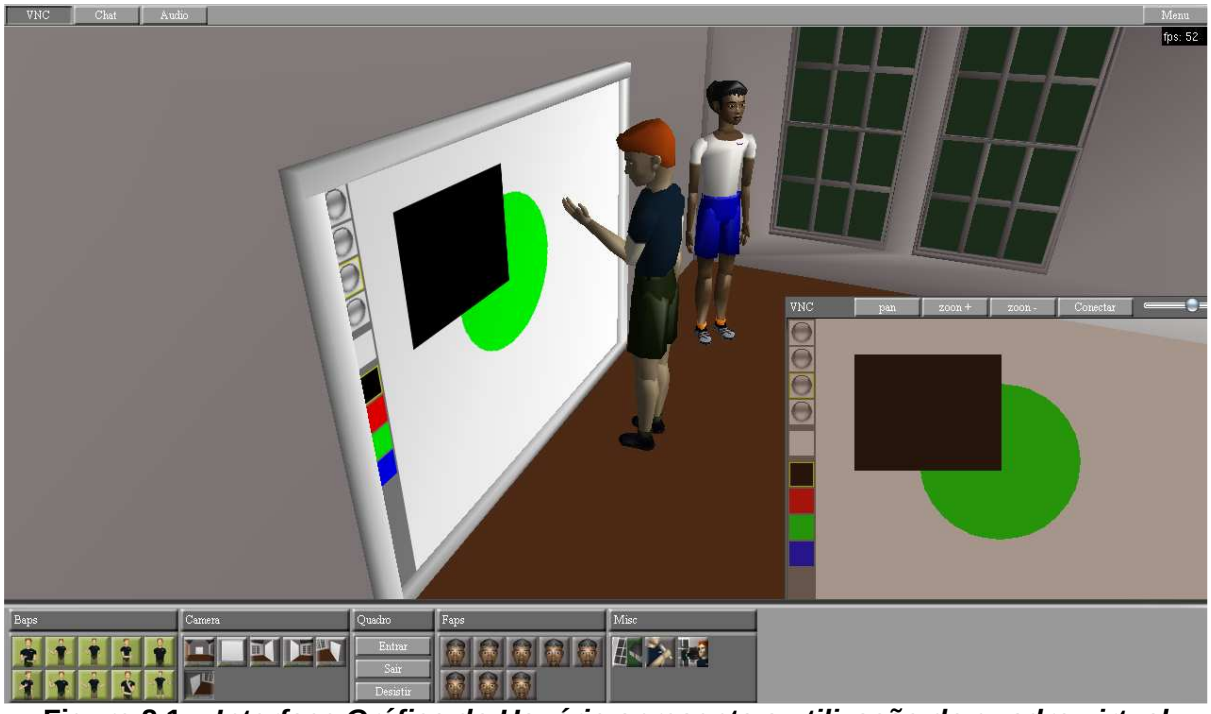


Figura 2.1 – ***Interface Gráfica do Usuário apresenta a utilização do quadro virtual***

a) Mundo virtual 3D ao centro, ocupando a maior parte da janela da aplicação. Nesta área o usuário visualiza o cenário habitado por humanóides virtuais, objetos interativos como o quadro virtual que compartilha aplicações e observa o trabalho do grupo através da representação gestual dos avatares.

b) Área de trabalho, ao centro, sobreposta ao mundo 3D. É composta por painéis flutuantes usados para facilitar a visualização e manipulação de objetos 3D interativos, além de exibir informações sobre eventos ocorridos no cenário. Cada painel possui um ajuste de transparência de forma a não prejudicar a visualização do mundo 3D em segundo plano.

c) Painel de controle, na parte inferior da janela, contendo botões para a execução de animações corporal e facial pré-definidas, os controles de uso do quadro virtual e as propriedades dos objetos selecionados.

d) Na parte superior da janela encontra-se a barra de ferramentas contendo o menu principal, onde o usuário pode definir suas configurações pessoais, iniciar o processo de conexão com o servidor e a opção de mostrar ou esconder o painel 2D da aplicação compartilhada ou a janela de conversação. Após o usuário definir suas configurações pessoais e conectar, o ambiente virtual é atualizado para o estado atual mantido pelo servidor. O humanóide virtual do usuário é carregado na cena junto com os humanóides dos demais usuários, caso o ambiente já esteja em uso.

2.3 Compartilhamento de aplicações

Um cliente VNC (Virtual Network Computing) [17] é integrado ao ambiente para compartilhar uma aplicação qualquer em execução em algum computador com um servidor VNC ativo. Porém, sua utilização pode não ser adequada a todas as aplicações por limitações de velocidade da rede, uma vez que, além dos eventos de mouse e teclado, as imagens da aplicação compartilhada são também transmitidas a cada modificação, assemelhando-se a um fluxo de vídeo. Como alternativa é possível a adição de aplicações em todos os clientes do ambiente virtual de forma que apenas os eventos gerados são transmitidos. É o caso da ferramenta de desenho mostradas nas Figuras 2.1, 2.2 e 2.3.

O aplicativo compartilhado é executado em dois espaços distintos: um painel (2D) flutuante e um quadro virtual imerso no ambiente 3D. A interface 2D da aplicação compartilhada evita que as distorções decorrentes da mudança de perspectiva no ambiente

3D prejudiquem a visibilidade e a manipulação do aplicativo. Já a execução no quadro virtual 3D imerso proporciona uma melhor percepção das interações do usuário sobre o objeto compartilhado, através da representação gestual dos humanóides virtuais. Sendo assim, o compartilhamento de aplicações (ou outros objetos interativos) é constituído por uma composição de interfaces 2D e 3D, como mostrado na Figura 2.1 e Figura 2.2.

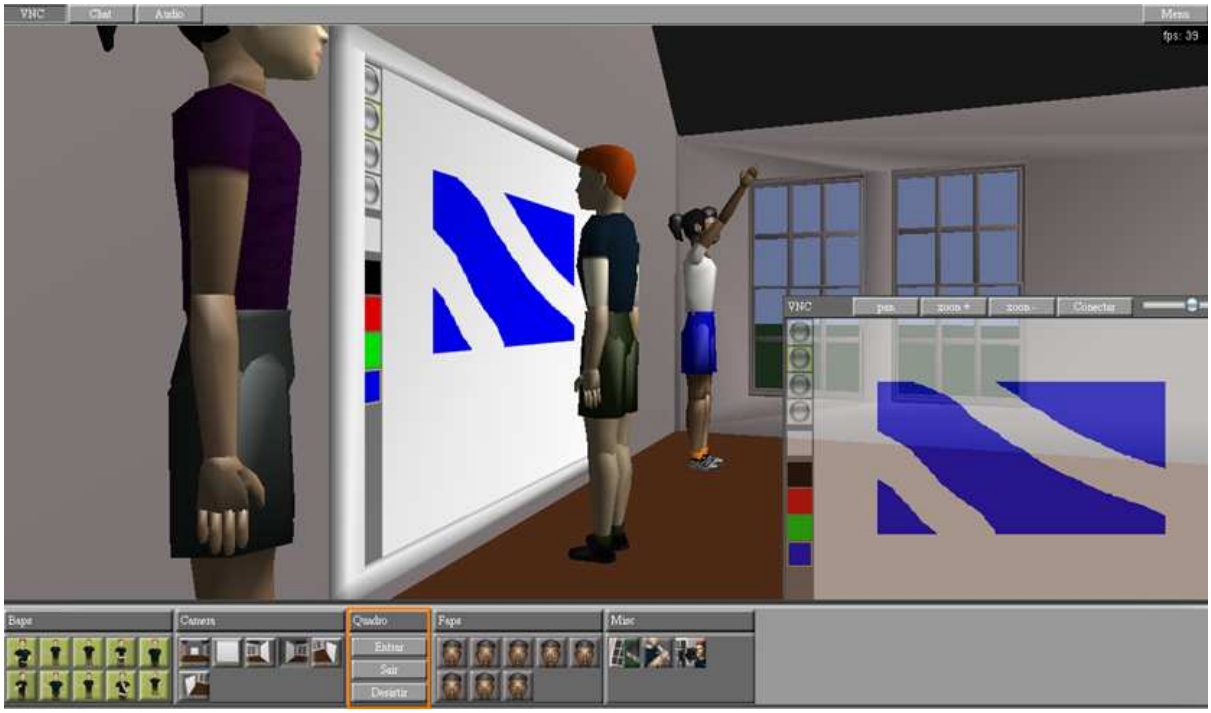


Figura 2.2 – Manifestação do desejo de utilização do quadro virtual

O usuário que deseja manipular o aplicativo compartilhado solicita o uso do quadro virtual através do botão “Entrar” no painel de controle. Se a aplicação estiver livre, o humanóide do cliente é automaticamente posto diante do quadro virtual e o painel 2D da aplicação é liberado para interação. A partir deste instante, o usuário pode utilizar a janela de aplicação como se estivesse em seu computador local. As interações sobre esta janela são convertidas em gestos reproduzidos pelo humanóide virtual no ambiente 3D (Figura 2.2).

Quando outro usuário solicita o uso do quadro virtual com este já em uso, o servidor o põe em uma fila de espera e seu humanóide sinaliza a ação erguendo o braço, de forma que todos saibam de sua intenção. O usuário que aguarda na fila pode abandoná-la a qualquer momento, caso esteja cansado de esperar, por meio do botão “Desistir” no painel de controle. O usuário que possui o controle da aplicação pode liberá-la acionando o botão “Sair” no painel de controle do quadro virtual.

2.4 Animações pré-definidas

No painel de controle existem botões que permitem ao usuário executar animações pré-definidas em seu humanóide virtual. Cada botão apresenta uma captura da animação executada. Há também uma breve descrição de cada botão no *menu* suspenso, como mostrado na Figura 2.3. As animações pré-definidas representam gestos comuns em uma conversa como, por exemplo, dar as boas vindas, aplaudir ou se despedir. Caso o humanóide em questão esteja usando o quadro virtual ou na fila de espera para usá-lo, as animações não serão executadas.

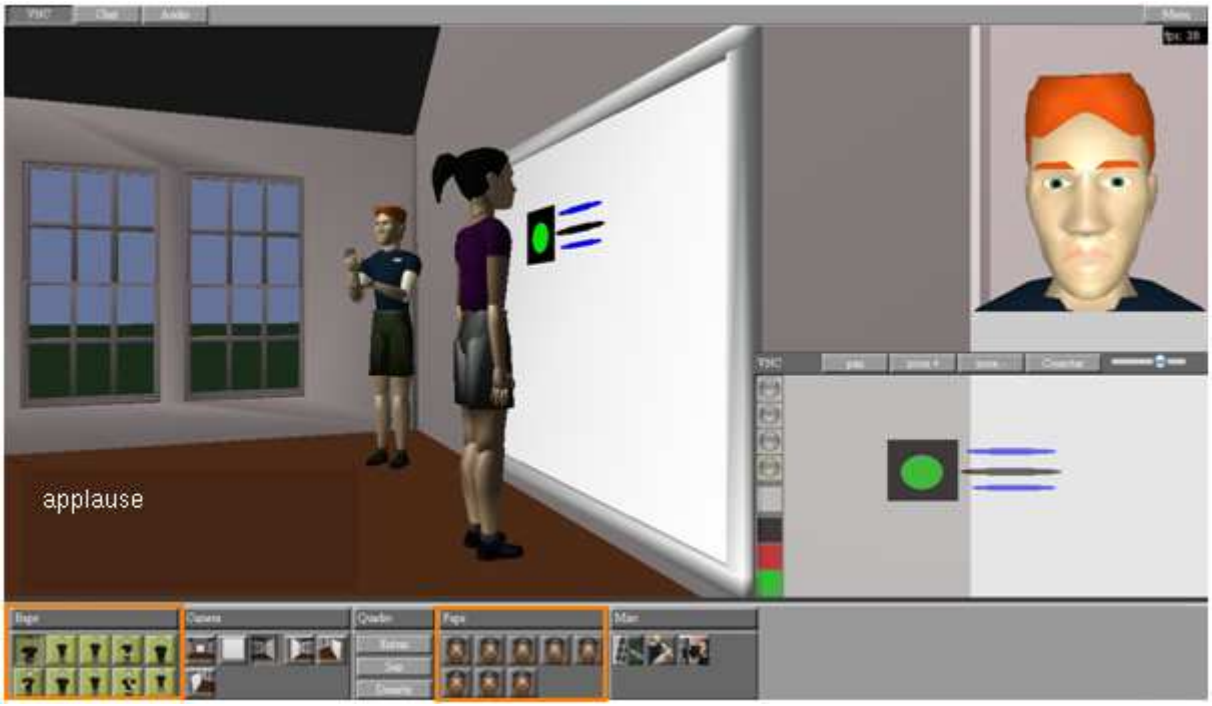


Figura 2.3 – Animações pré-definidas

2.5 O padrão MPEG-4 FBA

MPEG, acrônimo de *Moving Picture Experts Group*, designa uma família de padrões internacionais para codificação e compressão de objetos audiovisuais. Atualmente a especificação do padrão MPEG-4 é composta por mais de 20 partes, sendo que as duas primeiras, Systems (Parte 1) e Visual (Parte 2) tratam da animação de caracteres virtuais, os chamados objetos FBA (*Face and Body Animation*) [10].

Um objeto FBA é composto por dois fluxos de dados (*bitstreams*) independentes, denominados Parâmetros de Definição e Parâmetros de Animação. Os Parâmetros de Definição (*Definition Parameters*), especificados na Parte 1 (*Systems*), trazem informações

intrínsecas dos objetos virtuais tais como geometria, métodos de deformação da malha 3D, formas de calibração dos parâmetros de animação recebidos, entre outros. São definidos uma única vez para cada modelo antes do início de qualquer processo de animação. Neste grupo estão os FDPs (*Face Definition Parameters*) e os BDPs (*Body Definition Parameters*) que representam respectivamente, as definições de um modelo facial e corporal.

Os Parâmetros de Animação (Parte 2 - Visual) trazem informações genéricas, independente da forma e tamanho dos modelos, e são responsáveis pelas animações. Estão neste grupo os FAPs (*Facial Animation Parameters*) e BAPs (*Body Animation Parameters*). Esses parâmetros são transmitidos a cada quadro de animação e devem produzir as mesmas animações em diferentes modelos virtuais.

O padrão MPEG-4 define um conjunto amplo de parâmetros de animação. São 68 FAPs e 296 BAPs que representam o estado de um objeto FBA em um determinado quadro de animação [10]. Dependendo do nível de detalhamento exigido pela aplicação, é possível a seleção de um subconjunto desses parâmetros.

2.5.1 Animação Facial

Como forma de aprimorar a comunicação gestual, os humanóides virtuais ganham suporte para a animação facial, permitindo que os mesmos expressem o estado emocional de seus respectivos usuários ou simplesmente “conversem” uns com os outros. Os movimentos faciais podem ser pré-definidos ou gerados em tempo real. Um exemplo típico da animação em tempo real é a chamada animação facial sincronizada com a fala, em que os humanóides executam movimentos labiais de acordo com parâmetros extraídos de uma fonte sonora. Esse recurso torna a interação virtual mais realística favorecendo a dinâmica em grupo.

Em uma face humana real, ações musculares fazem com que determinadas regiões apresentem perceptíveis modificações em seu formato, gerando deformações que, quando combinadas, resultam em diferentes tipos de expressões. Cada região é representada por um único ponto, denominado Ponto Característico (*Feature Point* - FP). Esses pontos representam os locais de maior movimentação na estrutura do tecido e que são capazes de influenciar os “pontos” vizinhos para promover uma deformação. Esses “pontos” formam a região de influência de um FP. Em um modelo de face sintética, a estrutura facial corresponde a uma malha 3D de pontos interligados entre si formando polígonos. Desta forma, com o objetivo de simular a dinâmica de uma face real, a animação facial

especificada pelo padrão MPEG-4 é baseada na movimentação de pontos característicos da face sintética.

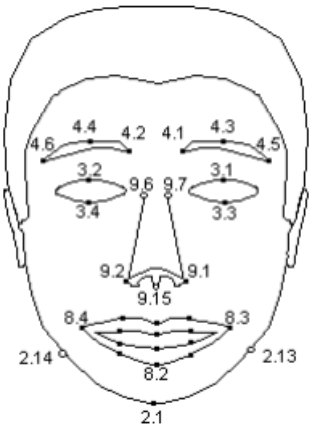


Figura 2.4 – Pontos característicos – Feature Points

Os demais pontos da malha facial não-marcados como característicos são agrupados em torno dos FPs vizinhos compondo suas respectivas regiões de influência. Ao movimentar um FP, todos os pontos de sua região de influência acompanham seu movimento de acordo com um fator de deslocamento (peso) pré-definido provocando a deformação da malha facial. A forma como estes FPs são manipulados gerando as mais variadas expressões é determinada por um conjunto de parâmetros denominados FAPs (Facial Animation Parameters). Cada FAP atua sobre um determinado ponto característico movimentando-o em uma direção específica pelo valor indicado. Por fim, o padrão MPEG-4 define um conjunto de unidades usadas nos deslocamentos dos pontos. As unidades garantem que um mesmo FAP produza efeitos similares quando aplicado em diferentes modelos faciais.

2.6 Comunicação e Gerenciamento

A consistência e o sincronismo das ações dos participantes do ambiente virtual COGEST são gerenciados através do uso de um protocolo que define conjuntos de mensagens relacionadas aos tipos de interação de usuário e à distribuição de informações de estado pelos participantes do ambiente virtual. O protocolo é implementado por mensagens serializadas enviadas por *sockets*. O compartilhamento de aplicações é feito mediante uma conexão independente com um servidor VNC, podendo esta vir antes ou depois da conexão com o servidor responsável pela gestão do ambiente, aqui chamado servidor de eventos.

Para cada novo pedido de conexão, o servidor inicia um novo *thread* responsável pelo tratamento de um determinado cliente. Com isso, tem-se a divisão do servidor de eventos em três partes (Figura 2.5):

- Gestor do ambiente: constitui o módulo base do servidor de eventos, onde são iniciados os demais módulos. O gestor do ambiente atua como meio de interligação entre os gestores de cliente e de conexão, além de manter um banco de dados relativo aos clientes conectados, métodos de autenticação, fila de espera para uso de recursos, dentre outros.

- Gestor de cliente: constitui um sub-processo dedicado a um determinado cliente. Para cada cliente conectado, um novo módulo gestor é iniciado para gerenciar a troca de informações entre o servidor de eventos e o cliente em questão. Após o término da conexão este sub-processo é finalizado.

- Gestor de conexão: encarregado de verificar se novas conexões de clientes estão sendo solicitadas, requisitando, ao gestor do ambiente, a criação de um novo gestor para o cliente recém conectado.

Na Figura 2.5 é apresentada a dinâmica da comunicação entre cliente e servidor com dois clientes (B e C) conectados e um cliente (A) em processo de conexão.

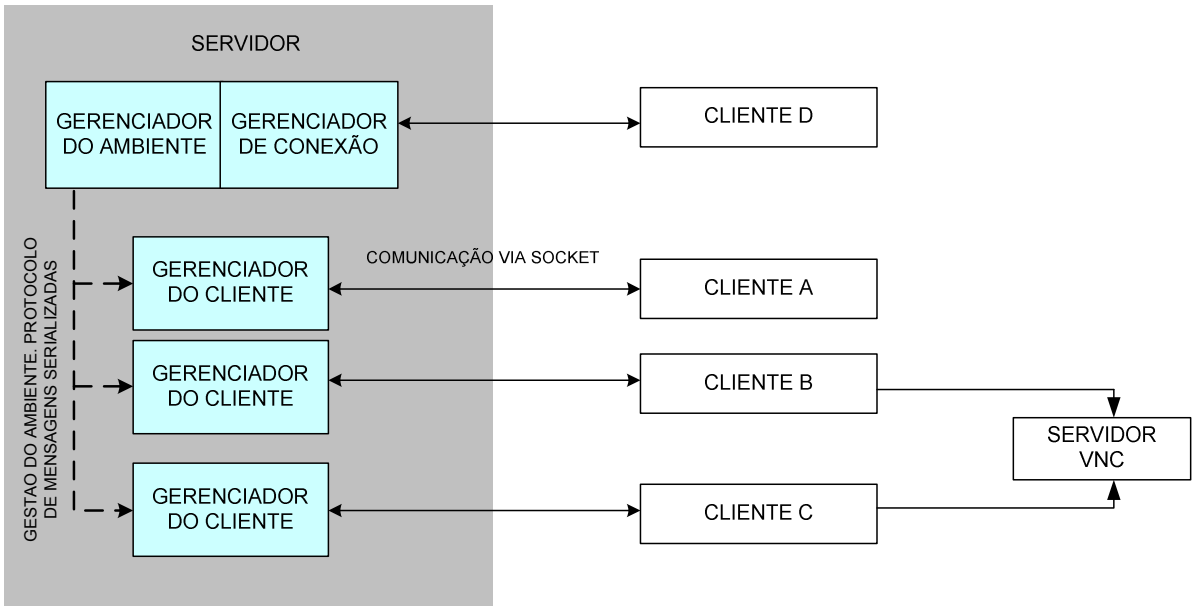


Figura 2.5 – Arquitetura original do COGEST

2.6.1 O protocolo de mensagens serializadas

Representada na Figura 2.5 pelas setas tracejadas, as mensagens serializadas são responsáveis por manter a atualização do ambiente virtual, bem como o controle de acesso aos recursos compartilhados. O protocolo utilizado é baseado em objetos serializados, onde cada objeto determina o tipo de evento a ser difundido e qual ação deve ser tomada, seja ela pelo servidor ou pelos clientes.

O cliente e o servidor ficam em estado de espera até que um objeto seja recebido. Com base na identificação do tipo de objeto recebido, as ações correspondentes são iniciadas pelo cliente ou pelo servidor. Para permitir a identificação do objeto, todas as classes de objetos de mensagem são herdeiras de uma mesma classe abstrata que, por sua vez, implementa a interface *Serializable*. Esse mecanismo permite a fácil extensão do ambiente através da inclusão de novas classes de mensagens, o que pode ser usado para associar novos comportamentos aos avatares que, desta maneira, podem representar tipos específicos de interação ou de evento.

2.7 Benefícios da inserção de canais áudio ao ambiente virtual

Apresentada a arquitetura e os principais recursos do ambiente virtual COGEST, enfatiza-se a inexistência de uma infra-estrutura que permita aos participantes da conferência interagir entre si usando a voz. Os elementos de interação originalmente presentes são: quadro virtual, mensagens de texto (*chat*), expressões faciais e gestos corporais pré-determinados. A proposta deste trabalho é adicionar canais de áudio como mais uma forma de interação além de estabelecer a animação das faces dos avatares em tempo real de acordo com o sinal de voz detectado.

O benefício que a proposta insere ao ambiente virtual se traduz em uma ampliação da dinâmica do trabalho em grupo uma vez que é permitido a cada participante expressar-se usando a voz. Além disso, os movimentos faciais dos *avatares* que estão “falando” sinalizam a origem do interlocutor. Nos capítulos seguintes são detalhadas as tecnologias utilizadas bem como a arquitetura para distribuição coerente dos canais de áudio.

3. Técnicas de áudio usando o Java Sound

A transmissão de áudio através de uma rede de dados remonta ao século XIX com a invenção do telefone. Naquela época eram usados dois fios de cobre para enviar e receber o som sobre os quais só era possível estabelecer apenas uma ligação por vez. Atualmente, a infra-estrutura para o estabelecimento de comunicação de áudio entre usuários geograficamente separados é completamente diferente, a Internet é bom exemplo de mudança de perspectiva. Com alcance global, ela proporciona uma alternativa à rede de telefonia tradicional.

Para a transmissão da voz em uma rede de computadores é necessário que o sinal sonoro passe por um processo de conversão em que é necessária a utilização de princípios de modulação e codificação do sinal analógico, a esse processo é dado o nome de digitalização. Uma vez na forma de sinal digital, os bits referentes ao sinal de voz podem ser empacotados e transmitidos pela rede da mesma forma que bits representativos de texto, por exemplo.

Neste capítulo é apresentada uma API de baixo nível para controlar a entrada e a saída bem como a execução de mídias de som, incluindo dados de áudio amostrado e *Musical Instrument Digital Interface* (MIDI) em uma estrutura que promove a extensibilidade e flexibilidade [13]. Assim, os recursos disponíveis nessa API são utilizados no desenvolvimento aplicações de alto nível, como por exemplo:

- Frameworks de comunicação (conferência e telefonia);

- Programas de aplicação interativa (games e web sites);

- Sistema de entrega de conteúdo para usuário final (tocadores de mídia e música em streaming);

Entende-se por dados de áudio amostrados uma série de valores que representam a amplitude ou a intensidade da pressão da onda de som. MIDI é uma especificação criada para tornar possível a interoperabilidade entre os diversos instrumentos musicais baseados em microprocessador: os instrumentos eletrônicos. A API Java Sound oferece facilidades para trabalhar com essas duas perspectivas, porém discutiremos apenas sobre os dados de áudio amostrados.

A API Java Sound não assume uma configuração de hardware de áudio específica, ela é projetada para permitir que diversos componentes de áudio sejam instalados nos

sistemas e acessados pela API. Existem outras APIs Java que permitem manipulações em dados de som, por exemplo, Java 3D e APIs para telefone e voz. Normalmente, essas APIs utilizam as funcionalidades da Java Sound internamente [13].

Essa API reúne suas funcionalidades em apenas quatro pacotes, são eles:

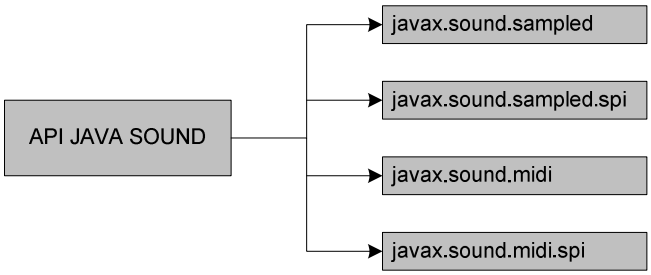


Figura 3.1 – Estrtura da API JavaSound

As suas duas maiores funcionalidades estão disponíveis em pacotes separados:

- javax.sound.sampled: esse pacote especifica interface para captura, mixagem, e reprodução de áudio digital (amostrado). Mais detalhes são apresentados na Seção 3.1.

- javax.sound.midi: esse pacote provê interface para síntese, seqüenciamento e manipulação de eventos sobre dados MIDI.

Os outros dois pacotes permitem a provedores de serviço (em oposição aos desenvolvedores de aplicação) criar componentes de software que extendam as capacidades de implementação da API. São eles:

- javax.sound.sampled.spi

- javax.sound.midi.spi

3.1 O pacote javax.sound.sampled

Esse pacote disponibiliza interfaces e classes para captura, processamento e reprodução de dados de áudio amostrados. Dessa forma, todos os recursos necessários ao desenvolvimento da aplicação proposta nesse projeto estão presentes nesse pacote. A sua tarefa central é transportar os bytes de áudio para dentro e para fora do sistema. Isso

envolve abrir uma entrada de áudio e um dispositivo de saída, além do gerenciamento de *buffer* que são preenchidos em tempo real. Pode envolver também mixagem de múltiplos de fluxos de áudio em um único canal de saída. O transporte do som para dentro e para fora do sistema deve ser corretamente tratado quando o usuário requisitar que o fluxo de áudio comece, pause, retorne ou pare.

Para utilizar esse pacote, basta adicioná-lo ao cabeçalho de importações como ocorre com qualquer outro pacote em Java. Para reproduzir e capturar som usando API Java Sound é necessário pelo menos três elementos: dados de áudio formatado, um mixer e uma linha. A seguir será apresenta uma descrição de cada um deles.

3.1.1 Dados de Áudio Formatados

Formatar uma seqüência de dados significa organizá-la de uma forma padrão para que possa ser interpretada posteriormente. No caso de dados de áudio, essa formatação torna possível, por exemplo, saber se uma determinada seqüência de bytes teve origem na leitura de um arquivo em disco ou foi capturado diretamente pelo microfone. Pode ser necessário saber a quantidade de bits que representam uma amostra (a representação da menor porção de som) ou a taxa com a qual o som foi amostrado.

Na API Java Sound, a formatação dos dados é tratada pelo objeto da classe *AudioFormat*, que inclui os seguintes atributos, além dos explicitados anteriormente:

- Técnica de codificação utilizada: PCM é a mais comum;

- Número de canais: 1 para mono, 2 para stereo, etc;

- Taxa de Frame;

- Tamanho do frame;

- Byte order: big-endian ou little-endian

3.1.2 Mixer

Fisicamente, o *mixer* é um dispositivo que combina diversos fluxos de entrada de áudio em diversos fluxos de saída de áudio (tipicamente dois canais para um som *stereo*).

Para exemplificar, em um concerto musical, os cabos de diversos microfones e instrumentos convergem para esse aparelho que possibilita ao engenheiro de som ajustar ganho e o *pan* (contribuição de cada linha de entrada para a saída), por exemplo, antes de direcionar o fluxo de saída configurado para amplificadores e posteriormente para os alto-falantes. Da mesma forma, na API Java Sound, o propósito de um *mixer* é controlar um ou mais fluxos de entrada e saída de áudio. Um objeto *Mixer* pode representar a capacidade de mixagem de som de um dispositivo físico, como por exemplo, uma placa de som que manipula diversos dispositivos de entrada e saída de som.

Em aplicações que transportam informação via streamming é disponibilizada uma classe chamada *AudioSystem* que atua como ponto de entrada para acessar os recursos do sistema que manipulam áudio amostrado. Ela permite consulta e acesso aos *mixers* que estão instalados no sistema. Além de disponibilizar métodos para obtenção de linhas sem a necessidade de acessar diretamente os mixers, o que representa uma maior facilidade de programação.

3.1.3 Linha

Uma Linha é o caminho para mover o áudio para dentro ou pra fora do sistema. Pode ser também as portas de entrada e saída de áudio além de um caminho de dados chegando até um *mixer* ou partindo dele. As informações contidas no objeto *AudioFormat* são passadas à classe *DataLine.Info* para informar que tipo de informação trafegará pela linha.

Do ponto de vista da API, uma Linha é uma interface que apresenta a seguinte hierarquia:

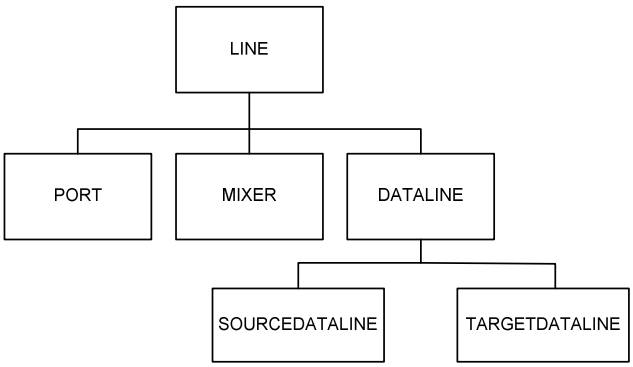


Figura 3.2 – Hierarquia da interface Line da API JavaSound

A interface base, Line, descreve as funcionalidades comuns a todas as linhas [13]:

- Controle: *DataLine*s e *Port*s possuem um conjunto de controles que afetam o sinal de áudio que passa pela Linha. A API especifica as classes de controle que podem ser usadas para manipular aspectos do som, tais como ganho (influencia o volume do sinal), pan (influencia o posicionamento esquerdo-direito do som) e ressonância (que adiciona reverberação ao som para emular diversos tipos de acústica).

- Status de aberto e fechado: a abertura de uma linha com sucesso garante a correta alocação de recurso para a linha.

- Evento: uma linha gera evento quando é aberta ou fechada. Quando um alinha gera um evento, o evento é enviado para todos os objetos que se registraram para "escutar" por eventos naquela linha.

Sobre as subinterfaces, temos:

- *Port*: linha simples pra entrada ou saída de áudio de ou para dispositivos de áudio. Por exemplo: microfone, alto-falante, *headfone*.

- *Mixer*: como dito anteriormente, é definido como tendo um ou mais linhas de entrada e uma ou mais linhas de saída. Essa interface disponibiliza métodos para obter as linhas do mixer, isso inclui as linhas de entrada e as linhas de saída.

- *DataLine*: uma interface Line genérica não oferece uma forma de começar ou parar um toque ou uma gravação de som. Essa é a função das interfaces *DataLine* que ainda oferece recursos para produção de eventos, manipulação de buffer, formato de dados de áudio, status da linha entre outros. Destaca-se, pra propósito desse projeto, duas sub-interfaces:

    - *TargetDataLine*: recebe os dados de áudio do Mixer que os captura de um Port.

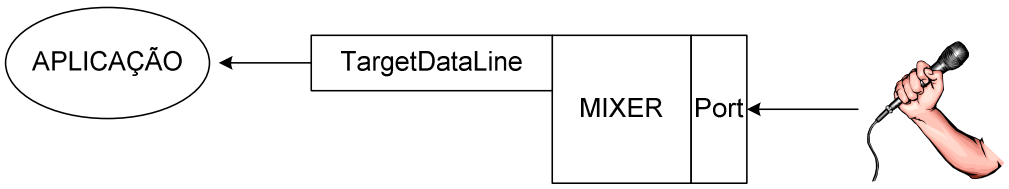


Figura 3.3 – Elementos da API JavaSound utilizadas para captura de áudio

    - *SourceDataLine*: recebe os dados para posterior reprodução.

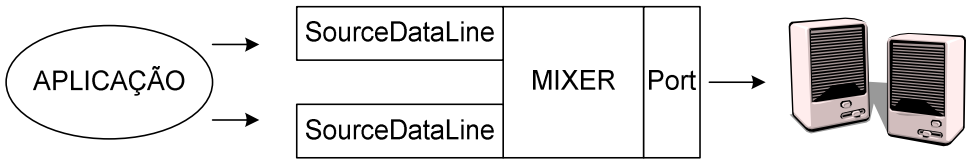


Figura 3.4 – Elementos da API JavaSound utilizadas para reprodução de áudio

Observa-se que a terminologia adotada pra nomenclatura dos *DataLine* tem como referência o *Mixer*. Esse ponto de vista adotado pela Sun pode causar certa confusão pois, do ponto de vista da aplicação, ou seja, do programador, os *TargetDataLine* são a fonte de dados para a aplicação. Da mesma forma, os *SourceDataLine* são o destino dos dados da aplicação e fonte para o *Mixer*.

3.2 Reprodução do Som

Existem dois tipos de linhas que podem ser usadas para reproduzir o áudio: *Clip* e *SourceDataLine*. Usam-se objetos do tipo Clip quando os dados de áudio estão pré-alocados na memória. *SourceDataLine* são usados quando temos um fluxo de dados em tempo real, ou seja, em *streaming*. Dessa forma, como a aplicação proposta pretende trafegar dados de áudio pela rede, devemos utilizar objetos do tipo *SourceDataLine*.

3.3 Captura do Som

A captura de áudio é o processo de obter um sinal de fora do computador. Uma aplicação típica de captura de áudio é a gravação de voz em um arquivo obtida usando um microfone. Cabe ressaltar que uma aplicação que captura áudio não necessita armazenar os dados na memória, podendo mantê-los em *buffer* durante o momento em que são enviados para reprodução em um alto-falante, por exemplo, sendo posteriormente descartados.

3.4 Considerações finais

Pelo que foi apresentado, a utilização da API JavaSound atende aos requisitos esperados para inclusão de um canal de áudio no ambiente COGEST. Todas as funcionalidades dessa API estão inseridas na JDK Java, sem necessidade de instalação de bibliotecas adicionais ao ambiente. Os Capítulos 4 e 5 apresentam a arquitetura do novo ambiente para comunicação por áudio em conferência ou como conversa reservada. O desenvolvimento se apóia na Java Sound que, para a comunicação, oferece vinte e cinco classes e oito interfaces. Entretanto, foram utilizadas apenas três classes (*AudioFormat*, *DataLine*.*Info e AudioSystem*) e duas interfaces (*TargetDataLine e SourceDataLine*) para a inclusão das funcionalidades referidas, o que demonstra a facilidade de utilização do Java Sound em relação á API JMF.

4. Comunicação por áudio em conferência no ambiente COGEST

Neste Capítulo, é apresentada a arquitetura de comunicação em modo de conferência inserido no ambiente COGEST. Neste modo de comunicação, diversos usuários podem se comunicar por áudio simultaneamente, sento todos emissores e receptores. Por se tratar de uma aplicação desenvolvida com solução diferente, a comunicação reservada é apresentada apenas no capítulo seguinte.

Para promover o compartilhamento de voz entre os participantes da sala virtual do COGEST, utilizou-se uma arquitetura de software do tipo cliente-servidor. A comunicação entre os processos cliente e servidor foi estabelecida usando *sockets* TCP.

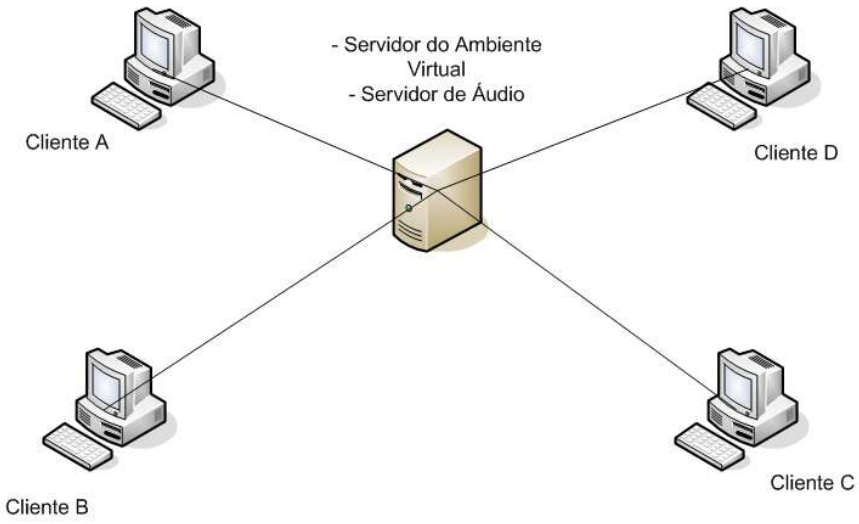


Figura 4.1 – Topologia física da arquitetura proposta

O protótipo desenvolvido sobre um modelo de rede centralizada apresenta conexões ponto a ponto entre os clientes (os usuários da aplicação) e o servidor e usa endereçamento *unicast* com o encaminhamento dos pacotes realizado na entidade central, o servidor. O motivador fundamental para utilização dessa modelo de rede é o fato de aproveitar a infra-estrutura do projeto COGEST que já apresentava um elemento central coordenando a coerência do ambiente virtual. Assim, foi acrescentada mais uma funcionalidade ao servidor original, visando ao suporte da conferência de áudio.

Cabe ressaltar que é necessário estar conectado ao ambiente virtual para que seja possível estabelecer uma chamada de voz entre os participantes. Assim, pode ocorrer de apenas uma parte dos clientes estarem participando da conferência, embora todos eles estejam representados na sala virtual com seus respectivos *avatares.*

O servidor de áudio mantém uma lista dos usuários conectados, ou seja, dos usuários que pretendem usar os recursos da voz para a comunicação em grupo e que se registraram pra isso. Dessa forma, quando o servidor recebe um pacote contendo dados de áudio ele simplesmente o repassa aos demais clientes registrados. A Figura 4.2 apresenta como o Servidor de Áudio encaminha os pacotes considerando que apenas três clientes estão participando da conferência. Nesse caso, o Cliente D participa apenas da sala virtual.

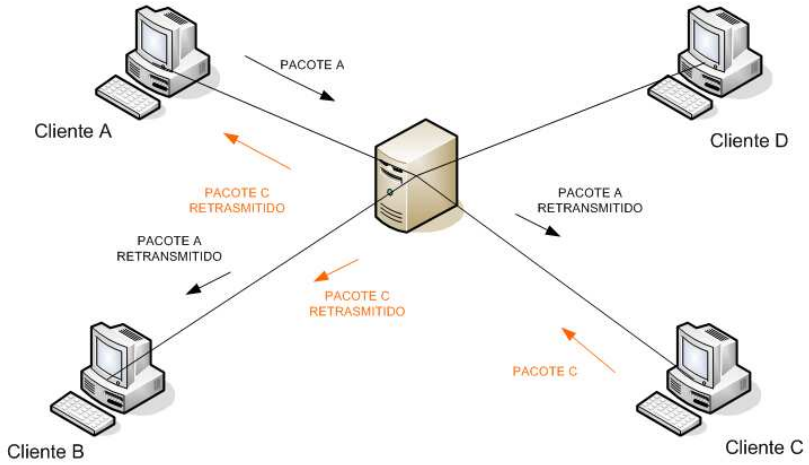


Figura 4.2 – Encaminhamento de pacotes para estabelecimento de conferência

Para tratar cada cliente, o servidor mantém um *thread* para receber os dados de cada conexão. Dessa forma, dois pacotes podem chegar simultaneamente ao servidor sem que haja perda de informação. No caso apresentado acima, o Cliente B recebe os dados de voz provenientes dos Clientes A e C.

Um problema decorrente da arquitetura cliente/servidor é a falta de escalabilidade. Quando o número de clientes cresce, o servidor passa a trabalhar de forma sobrecarregada. Além de manter a *thread* responsável por coletar os dados de voz de cada cliente, o que consome bastante recurso (memória e processamento), o servidor deve encaminhar os diversos pacotes enviados pelos participantes da conferência. Nesse cenário de sobrecarga,

pode ocorrer de algumas requisições de repasse não sejam atendidas. Porém, o COGEST é utilizado é um cenário restrito a poucos participantes da conferência.

4.1 Adaptação ao ambiente COGEST

O diagrama de blocos da Figura 4.3 apresenta como o modelo de comunicação em conferência foi inserido ao ambiente COGEST. Ao comparar o diagrama apresentado na Figura 2.5, que representa o projeto COGEST antes das mudanças propostas nesse trabalho, e o diagrama abaixo, concluí-se que as alterações apenas adicionam novas funcionalidades, sem promover modificações no código base, o que demonstra a modularidade e a característica de fácil extensão do ambiente.

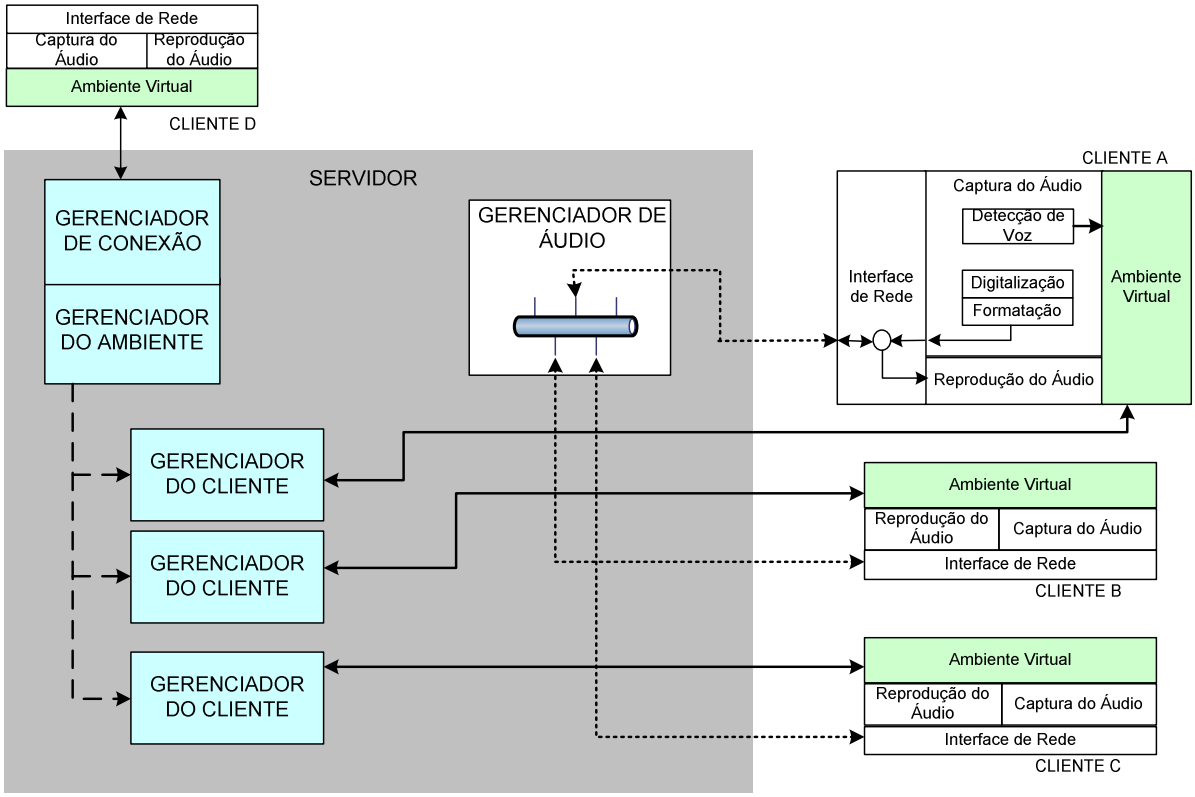


Figura 4.3 – Arquitetura da solução proposta para conferência

Os blocos inseridos na arquitetura do ambiente COGEST para a realização da conferência são apresentados em seguida.

4.2 O Gerenciador de Áudio

Ao observar o Servidor, destacamos o Gerenciador de Áudio, um módulo de software responsável pelo encaminhamento dos dados de voz enviados pelos clientes. Ele representa a única alteração proposta ao servidor original do projeto COGEST.

A retransmissão dos pacotes de voz aos destinos só é possível porque o Gerenciador mantém uma lista com o nome e o endereço IP dos clientes conectados. Dessa forma, depois de identificado o emissor da informação, ocorre o repasse dos dados aos usuários de destino da aplicação. Note que não é coerente para quem envia os dados de voz tê-los reproduzidos em seu ambiente.

A comunicação entre a Interface de Rede no cliente e o Gerenciador de Áudio no Servidor é representada na figura acima pela linha pontilhada bidirecional. Na perspectiva do código fonte, essa linha é formatada por quatro descritores de arquivos responsáveis pela comunicação via *socket*. A figura abaixo apresenta uma perspectiva mais realista da linha bidirecional.

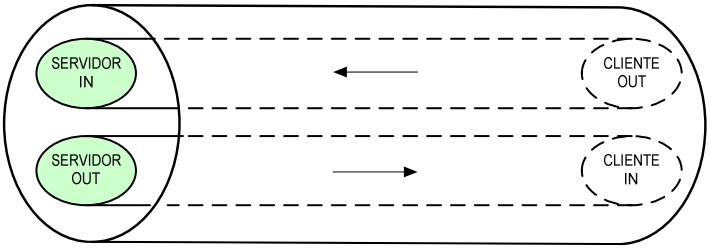


Figura 4.4 – Abstração do canal de comunicação ente o cliente e o servidor de áudio

A quantidade de clientes suportados pelo Gerenciador de Áudio não é limitada por software, ou seja, não existe restrição no código-fonte que impeça o atendimento de um número maior de clientes. O número de clientes apresentados no diagrama reflete apenas uma comodidade gráfica.

4.3 O Cliente

As principais modificações propostas nesse projeto se concentraram no lado cliente da aplicação. Nele os dados de voz são digitalizados, formatados e posteriormente enviados para que o servidor de áudio possa encaminhá-los para os demais clientes. Ele também deve ser capaz de captar os dados enviados pelos demais participantes da conferência e

reproduzi-los. Além disso, o movimento facial do *avatar*, notadamente o movimento da boca, deve ser coerente à voz do usuário.

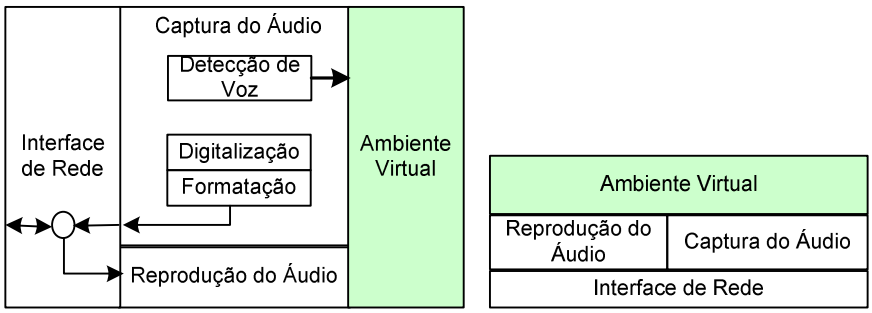


Figura 4.5 – Diagrama de blocos do cliente. As duas representações são equivalentes.

No diagrama apresentado na Figura 4.5, procurou-se enfatizar em blocos as funcionalidades adicionadas ao cliente que são: “Interface de Rede”, “Reprodução do Áudio”, “Captura do Audio”.

4.3.1 Interface de Rede

Responsável por receber os dados de áudio formatados e encaminhá-los ao servidor. Mantém um canal exclusivo para receber todos os dados oriundos do servidor e encaminhá-los para reprodução (Figura 4.4). Tanto o envio como o recebimento de dados é estabelecida com o uso de Sockets TCP.

4.3.2 Captura do Áudio

Todo o processo de captura de áudio é realizado por intermédio da API JavaSound. No Capítulo 3 foi apresentado um estudo detalhada dessa biblioteca de funções.

4.3.2.1 Digitalização e Formatação

Esse processo tem início no microfone que converte as ondas sonoras em sinais elétricos por meio de um transdutor localizado no microfone. Normalmente o microfone é a única entrada disponível e também o dispositivo padrão para a API JavaSound. Após o

processamento preliminar que ocorre na placa de som, os bytes resultantes representam a pressão do sinal sonoro amostrado. A esses dados precisam ser adicionadas informações para que eles possam ser corretamente interpretados por outras aplicações. A Modulação PCM é usada como padrão da API JavaSound. Alguns parâmetros importantes para o processo de digitalização são apresentados a seguir:

a) Taxa de amostragem = 8000 amostrados por segundo

b) Tamanho de cada amostra = 16 bits

c) Número de canais = 1

d) Organização do byte = little endian

Todas essas informações são reunidas em cada pacote contendo a voz digitalizada prontos para serem enviados ao Servidor de Áudio para seu posterior encaminhamento aos demais clientes.

4.3.2.2 Detecção de Voz

Logo após a digitalização da voz, os bytes armazenados em buffer passam por um processo que analisa se esses dados são característicos da voz humana. A detecção da voz tem por objetivo principal informar ao Gestor do Ambiente, presente no Servidor, que determinado usuário está usando a voz para se comunicar e que por conseqüência é necessária a atualização da face do seu respectivo avatar inserido no Ambiente Virtual de cada participante da conferência. A atualização da face do avatar é realizada com a reprodução de animação pré-definida. Dessa forma, não há nenhum mecanismo de reconhecimento de fonemas.

Na figura 4.5, a seta que parte do bloco representativo da detecção de voz representa o procedimento inicial para atualização da face dos avateres de cada cliente. Ao ser acionado, o Ambiente Virtual envia uma mensagem, usando o protocolo de objetos serializados descrito na Seção 2.6.1, ao Gerenciador do Cliente que a repassa ao Gerenciador de Ambiente para que, finalmente, ela seja encaminhada ao Gestor dos demais clientes. Esse, por sua vez, envia a requisição de atualização da face para o seu respectivo cliente que reproduz o comando requisitado. A Figura 4.6 auxilia a compreensão da seqüência apresentada anteriormente.

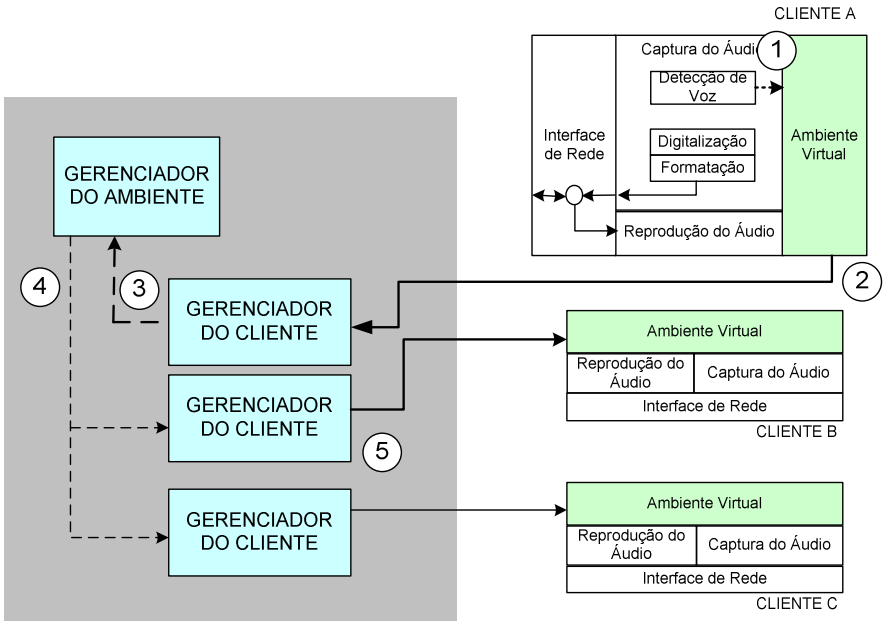


Figura 4.6 – Seqüência temporal de eventos para movimentação da face dos avatares

Após a inspeção sobre cada conjunto de dados, se não for confirmada a presença de voz humana, esses dados não serão transmitidos ao Servidor, sendo simplesmente descartados. Com isso, economiza-se largura de banda e processamento desnecessário no lado cliente e, principalmente, no lado servidor. Para perceber que o desperdício de recursos na máquina servidora é ainda maior, pode-se analisar um cenário típico de conferência em que apenas um entre vários participantes está falando. Em casos como esse, a quantidade de pacotes que chegariam ao servidor, oriundos dos clientes que estavam em silêncio, causaria uma sobrecarga desnecessária. A sobrecarga no lado cliente se reflete ao encaminhamento de dados que não carregam informação útil e na posterior reprodução do ruído.

Para detectar se os dados analisados representam a voz humana, foi utilizado um algoritmo de Transformada de Fourier Rápida [21]. O resultado da transformada é um número complexo que tem a sua magnitude analisada. Caso esse valor supere um valor de limiar estabelecido, temos então que os dados originalmente analisados correspondem à voz humana.

A escolha do valor de limiar mais adequado foi resultado de uma análise do espectro de magnitude da voz. Para isso, os valores inteiros que representam os dados de voz amostrados foram gravados em um arquivo e posteriormente exportados ao MATLAB [22].

O algoritmo da transformada rápida de Fourier foi aplicado a duas amostras de áudio: uma amostra com voz humana e outra amostra com silêncio. O resultado é apresentado a seguir:

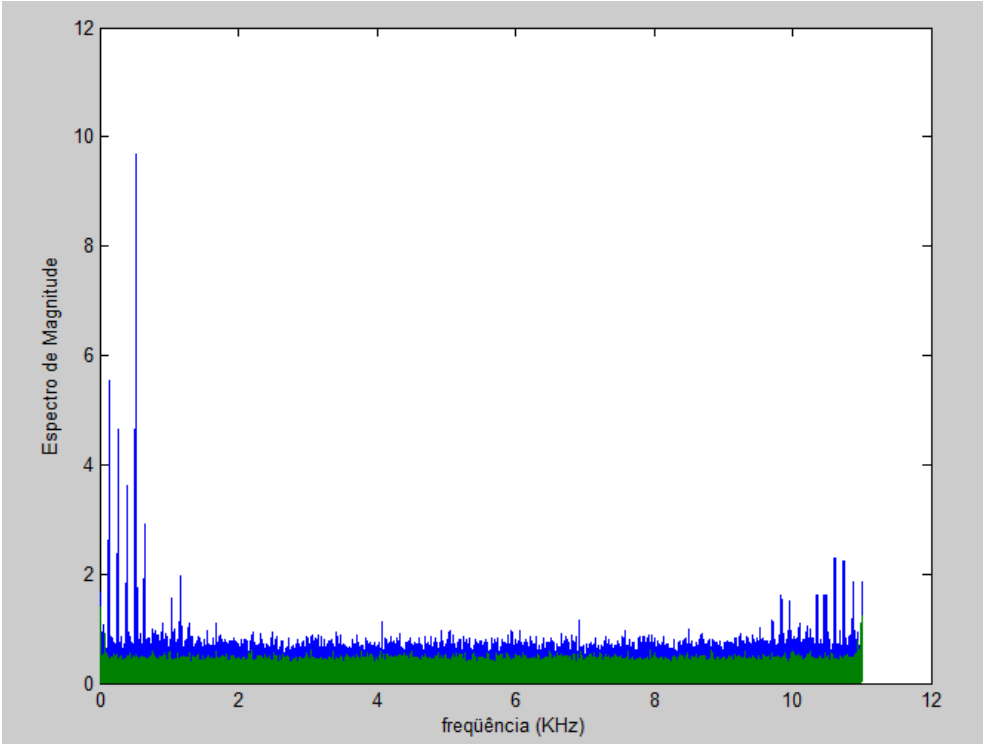
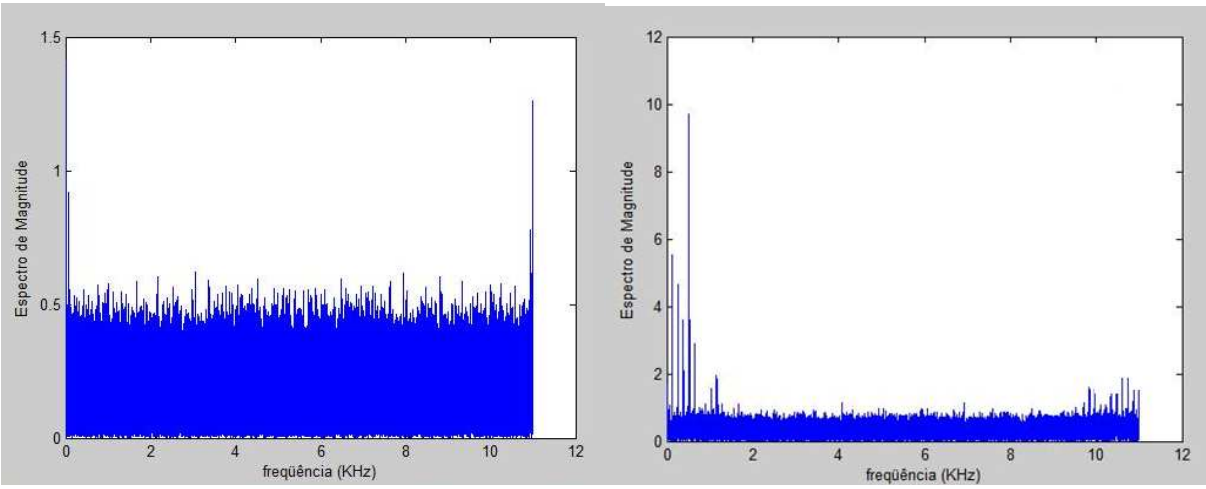


Figura 4.7 – Representação na freqüência de uma amostra de áudio com voz (em azul) e uma amostra de áudio sem voz, ou seja, silêncio (em verde)



a) b)

Figura 4.8 – Representação na freqüência de uma amostra de áudio com voz (b) e uma amostra de áudio sem voz, ou seja, silêncio (a). Note a diferença de escala entre as figuras

Ao analisar o espectro do sinal com ausência de voz, observamos que nenhuma amostra possui magnitude superior a 1,5. Um valor médio para as amostras é de aproximadamente 0,5. Ao analisar agora o espectro do sinal com voz, observamos uma

região com pico de magnitude em uma faixa de baixas freqüências representativos da voz. Com base nessa análise dos gráficos, foi adotado como limiar para decisão da presença de voz o valor de magnitude 3,5. Esse valor é usado pelo aplicativo cliente para decidir se os dados armazenados em buffer representam a voz ou apenas ruído devido ao silêncio.

4.3.3 Reprodução do Áudio

A Interface de rede mantém um canal exclusivo para monitorar o recebimento de dados do Gerenciador de Áudio no Servidor. Efetuada a leitura do pacote de dados, ele é encaminhado para a reprodução. Assim como no processo de captura de áudio, para reproduzi-lo, usou-se a API JavaSound.

De posse dos dados a serem reproduzidos, é necessário formar um canal para enviá-los até o *Mixer*, responsável pelo encaminhamento até a porta de reprodução. Esse canal que leva os dados até o *Mixer* é uma Linha do tipo *SourceDataLine*. Mais uma vez, para a abertura da Linha, é necessário informar os parâmetros do sinal da voz que irão trafegar por ela tais como: taxa de amostragem, tamanho de cada amostra, número de canais, organização do byte, além do processo de modulação por qual passou os dados do som, que, no caso, foi PCM.

4.3.4 Ambiente Virtual

Finalmente, o último bloco que compõe o cliente é o Ambiente Virtual. Esse bloco representa o cliente da aplicação virtual e foi desenvolvido por Anselmo [8] não sofrendo nenhuma alteração, do ponto de vista arquitetural, por esse projeto. Suas características, recursos e funcionalidades foram descritas detalhadamente Capítulo 2 desse texto. A única adaptação necessária foi na interface do usuário através da adição de um *menu* para o acionamento dos recursos de áudio (Figura 4.9). Para desenvolvê-lo foi utilizada uma infra-estrutura disponível no ambiente para a adição de novos componentes na GUI.

Figura 4.9 – Menu de Áudio

O Botão “Start” é o responsável por dar início à conferência de voz. Ao clicar nesse botão o cliente já está disponível tanto para enviar como para receber áudio. Caso não haja ainda clientes conectados ao servidor de áudio, não haverá também a distribuição de pacotes de voz, aguardando, dessa forma, a chegada de outros participantes. O botão “Refresh”, “Reservation”, e o *combobox* são utilizados para estabelecer a conversa reservada. O botão “Stop” é responsável por finalizar a conferência de áudio.

Para o desenvolvimento da GUI apresentada na Figura 4.9 foi utilizada uma estrutura de componentes disponibilizada por Anselmo [8]. “*Component*” é uma classe abstrata da qual todos componente da GUI derivam. A classe “Audio", desenvolvida nesse projeto, é um “Container” (relação de especialização) e contém objetos “Button” e “ComboBox” (relação de associação - composição). Este por sua vez, também é um “Container” composto por objetos da classe “MenuItem” e “LockedButton”.

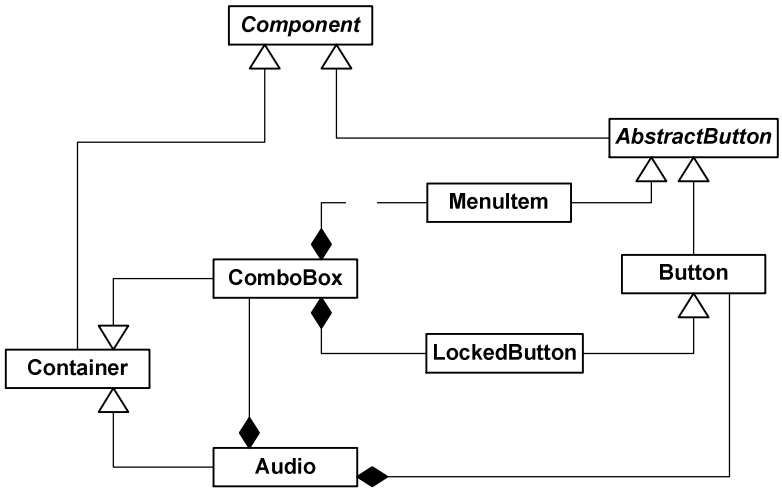


Figura 4.10 – Diagrama de Classes UML apresenta os componentes utilizados para desenvolver o Menu de Áudio, destacado na classe “Audio”

No próximo capítulo, é apresentada a opção de conversa reservada em que apenas dois participantes participam da comunicação por áudio. Nesse caso, o servidor de áudio trabalha de uma forma diferente da que foi apresenta.

5. Arquitetura para Conversa de Áudio Reservada

Com intuito de ampliar as possibilidades de interação entre os participantes do ambiente virtual, foi adicionada uma ferramenta de conversa reservada. Ela permite que dois clientes possam utilizar a voz em uma conversa particular, sem que os demais participantes do ambiente virtual tenham acesso às informações. Esse cenário é útil, considerando o contexto educacional ao qual se propõe o ambiente COGEST, para que alunos possam realizar atividades sem que os demais ouçam o que cada dupla está conversando. Em outro cenário, professores em uma mesma sala virtual podem discutir reservadamente sobre a organização da aula ou algo que não deve ser de conhecimento dos alunos.

A Figura 5.1 apresenta em detalhes a interação que ocorre entre o cliente e o servidor da aplicação para o estabelecimento de uma conversa reservada.

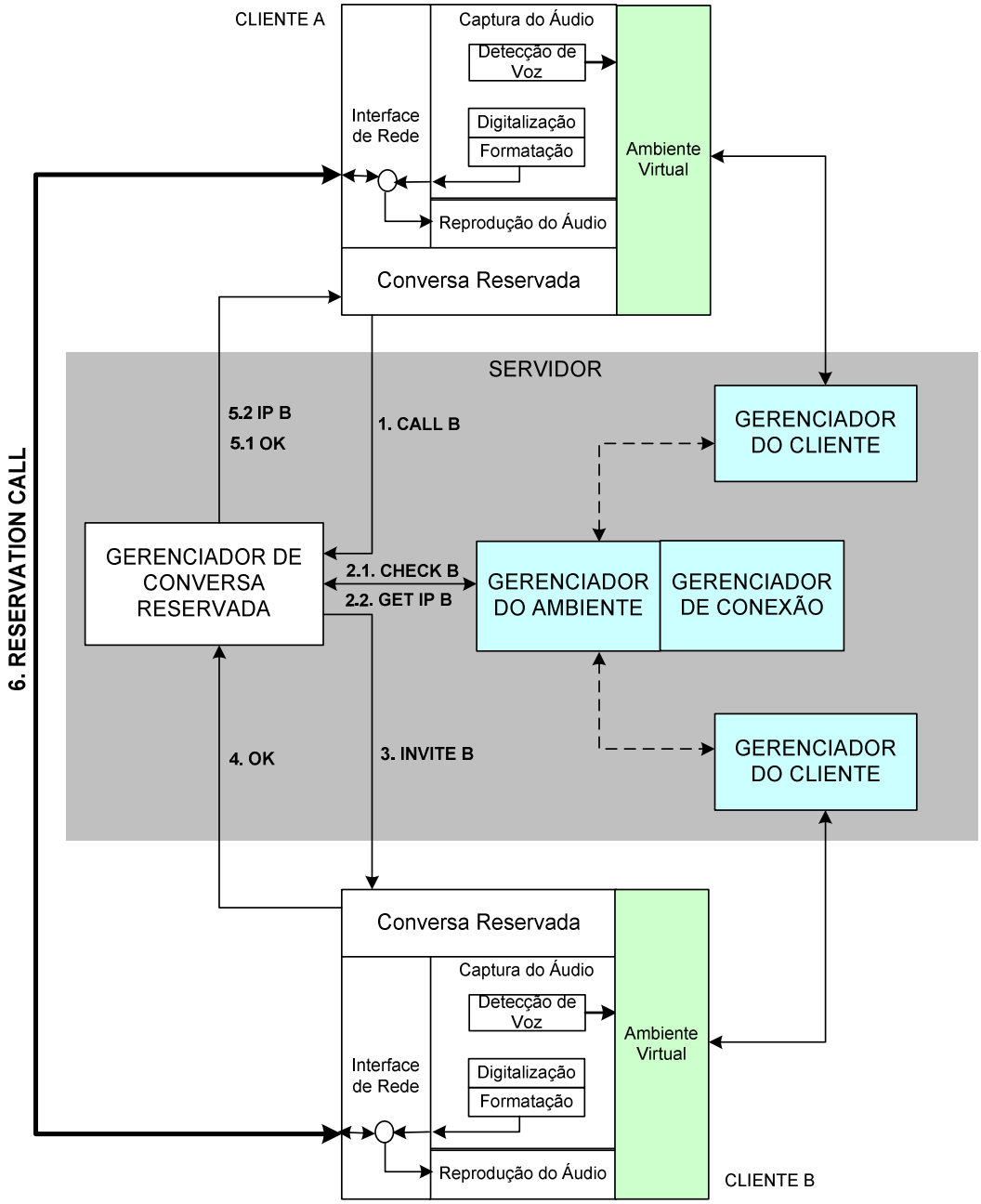


Figura 5.1 – Arquitetura proposta para conversa de áudio reservada

Antes de apresentar o procedimento para que se estabeleça a conexão entre os clientes, serão analisadas as mudanças propostas tanto no cliente como no servidor da aplicação.

5.1 As modificações propostas

Os módulos de software adicionados tanto ao cliente como ao servidor tem a função primordial de estabelecer a conexão entre os clientes que irão participar da conversa reservada. No cliente, esse módulo recebeu o nome de “Conversa Reservada”, já no servidor, o nome escolhido para representação gráfica foi “Gerenciador de Conversa Reservada”.

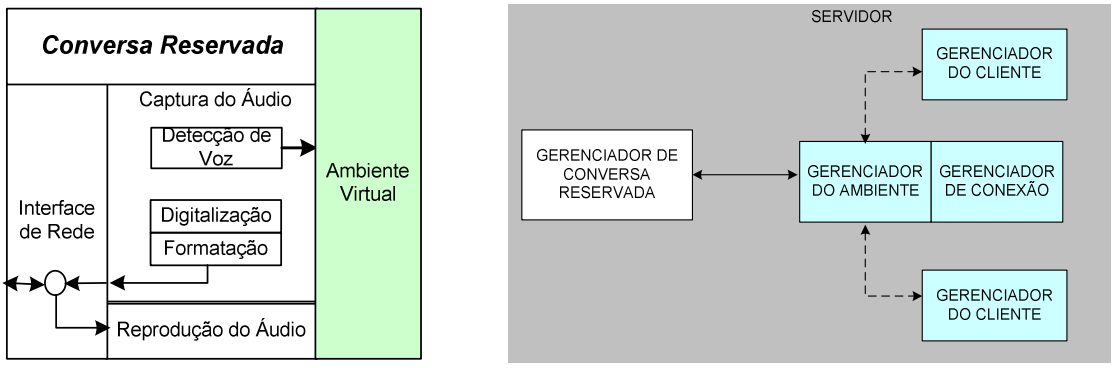


Figura 5.2 – Módulos de software, no cliente e no servidor, responsáveis por estabelecer a conversa reservada

O cliente é responsável por dar início à conversa reservada, enviando a requisição inicial. Além disso, ele monitora as requisições oriundas de outros clientes para o estabelecimento de conexão. O servidor é responsável por intermediar a negociação entre os clientes. Ele recebe o pedido de conexão do cliente, checa se o cliente requisitado está conectado e repassa a requisição ao destino. Mais detalhes do fluxo de mensagens entre o cliente e o servidor para o estabelecimento da conexão serão vistos adiante.

Dessa forma, as alterações propostas representam uma adição de funcionalidade quando comparada à arquitetura apresentada no capítulo anterior.

5.2 O Procedimento de conexão

Como em toda aplicação do tipo cliente/servidor, a interação é iniciada pelo cliente e busca extrair informação do servidor. A decisão de iniciar a conversa parte do botão “Reservation” na Interface Gráfica do Usuário (Figura 5.3). Ao lado desse botão, temos uma lista dos usuários conectados e disponíveis para iniciar a conversa reservada. Escolhido o destino da comunicação, basta agora acionar o botão para dar início ao processo.

A mensagem chamada, na Figura 5.1, de “CALL B” representa o início do procedimento. Após ela ser recebida pelo servidor, o Gerenciador de Conversa Reservada checa juntamente ao Gerenciador do Ambiente, responsável por manter uma lista de usuários conectados, se o usuário de destino está disponível: mensagem “CHECK B”. Caso afirmativo, o Gerenciador de Conversa Reservada obtém o endereço IP do cliente destino (mensagem “GET IP B”). Caso o cliente não esteja mais conectado ao servidor, essa informação é repassada ao cliente (essa mensagem não está destacada na Figura 5.1). Essa situação ocorre tipicamente quando o cliente que origina a requisição possui uma lista defasada de clientes, daí a utilidade do botão “Refresh”, apresentado na figura abaixo, que atualiza a lista de clientes conectados ao servidor de áudio.

Figura 5.3 – Botão “Reservation” para conversa reservada

De posse do IP de destino da comunicação, o Gerenciador de Conversa Reservada inicia uma conexão informando-o sobre uma requisição de chamada (mensagem “INVITE B”). Na tela do cliente B, surgirá uma janela de confirmação com a seguinte questão: “Aceitar ligação do cliente A?”. Caso a resposta confirme a ligação, será encaminhado ao cliente A o IP do cliente B: mensagem “IP B”. De posso do IP da máquina de destino, a conversa pode ser estabelecida diretamente entre os clientes, sem que seja necessária a

atuação do servidor de áudio como no caso de uma chamada de áudio com mais de dois participantes, apresentada no capítulo anterior.

Como o servidor não atua no processo de distribuição dos dados de voz, podemos garantir que os demais clientes não terão acesso às informações trocadas. Outra vantagem dessa arquitetura é diminuir a sobrecarga no servidor devido ao encaminhamento de pacotes aos clientes da conversa de áudio. Como pode ser destacado pelo fluxo de mensagens “RESERVATION CALL” (Figura 5.1) o servidor não atua na distribuição dos dados de voz.

Destaca-se na Figura 5.4, abaixo, a ordem temporal das mensagens trocadas para o estabelecimento da conversa reservada de voz. Na Figura 5.1 temos uma visão estrutural da aplicação como um todo e o papel de cada módulo de software no estabelecimento da comunicação.

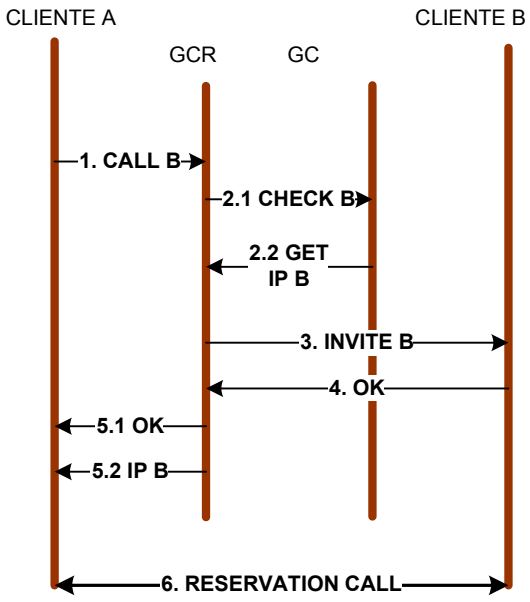


Figura 5.4 – Seqüência temporal das mensagens para conexão de áudio reservada. Legenda: GCR = Gerenciador de Conversa Reservada e GC = Gerenciador de Conexão

5.3 Considerações finais

A arquitetura para a inclusão de comunicação por áudio apresentadas nos Capítulos 4 e 5, respectivamente para os modos de conferência e de conversa reservada, foram avaliadas em cenários experimentas e encontram-se funcionais no ambiente COGEST. No Capítulo 6, são apresentados estes cenários de testes.

6. Cenários de Teste

Com intuito de apresentar as novas funcionalidades desenvolvidas para o ambiente COGEST, foram definidos alguns cenários que destacam a utilização do ambiente virtual juntamente com os recursos de áudio adicionado por esse projeto.

Em cada cenário é destacado qual módulo de software está sendo testado. Tomando como referência as Figuras 4.3 e 5.1, os módulos são: “Gerenciador de Áudio” e “Gerenciador de Conversa Reservada” no Servidor; “Interface de Rede”, “Captura Áudio”, “Tocar Áudio” e “Conversa Reservada” no Cliente.

Em todos os cenários, os clientes estavam inseridos em uma rede doméstica estabelecida a partir de um roteador wireless. Cabe ressaltar que os mesmos resultados podem ser obtidos usando-se uma configuração de rede que permita o roteamento dos pacotes para todos os clientes. Os equipamentos utilizados são computadores portáteis que disponibilizam microfone e alto-falantes utilizados nos testes.

Esse projeto não adiciona restrições à utilização dos recursos do ambiente virtual: quadro virtual e compartilhamento de aplicações. Dessa forma, é possível a utilização simultânea com os recursos de áudio.

6.1 Primeiro Cenário

Objetivo: Testar a distribuição do áudio no contexto da conferência;

Número de Participantes: Três;

Módulos de Software Testados: “Gerenciador de Áudio” no Servidor, “Interface de Rede”, “Captura Áudio”, “Tocar Áudio” no Cliente.

Descrição do cenário: Depois de conectados ao ambiente virtual, ocorre o convite (via *chat*) para iniciar uma conferência de voz. Todos os clientes acionam o botão na Interface Gráfica do Usuário para dar início à conversa. Estabelecida a conexão, todos os clientes podem falar e também podem ouvir o que os demais participantes estão falando.

O teste para esse cenário foi realizado durante aproximadamente 10 min.

Resultados Obtidos: Não houve prejuízo da comunicação quando dois clientes falam ao mesmo tempo, ou seja, enquanto o Cliente A escuta, ele também pode falar e, mesmo assim, os clientes B e C podem ouvir perfeitamente, ou seja, a comunicação é dita *full-duplex,* semelhante ao que ocorre no serviço telefônico convencional e diferente da conversa via rádio.

6.2 Segundo Cenário

Objetivo: Testar a conversa de áudio reservada em conjunto com a conferência;

Número de Participantes: Cinco;

Módulos de Software Testados: “Gerenciador de Conversa Reservada” e “Gerenciador de Áudio” no Servidor, “Interface de Rede”, “Captura Áudio”, “Tocar Áudio” e “Conversa Reservada” no Cliente.

Descrição do cenário: Ampliando o cenário anterior, dois novos clientes se conectam ao ambiente virtual. Os novos clientes, após alguma interação por meio do *chat,* decidem estabelecer uma conversa privativa. O Cliente D faz a ligação ao seu destino (Cliente E) pela seleção no *combobox* e posterior acionamento do botão “Reservation”. Os outros três clientes permanecem em conferência conforme apresentado no cenário anterior. Teste realizado durante aproximadamente 10 min.

Resultados Obtidos: Os três participantes iniciais da conversa (Clientes A, B e C) percebem a entrada de novos avatares (Cliente D e E) no ambiente virtual e a movimentação de suas faces. Entretanto, não escutam o que eles dizem, ou seja, os Clientes D e E estão dialogando, porém não participam da conferência. Os dois fluxos de comunicação (reservada entre os clientes D e E; e a conferência entre os clientes A, B e C) ocorre sem uma interfira na outra, sem que haja perda de privacidade por parte dos clientes em conversa reservada.

6.3 Terceiro Cenário

Objetivo: Testar a finalização da conferência e o estabelecimento da conversa reservada.

Número de Participantes: Quatro;

Módulos de Software Testados: “Gerenciador de Conversa Reservada” e “Gerenciador de Áudio” no Servidor, “Interface de Rede”, “Captura Áudio”, “Tocar Áudio” e “Conversa Reservada” no Cliente.

Descrição do cenário: Os quatro clientes iniciam a conferência e começam a dialogar. Em determinado instante, dois dos quatro clientes decidem manter o sigilo da conversa e iniciam uma conversa reservada. Os clientes que abandonam a conferência fazem isso pela utilização do botão “Stop”. Teste realizado durante aproximadamente 10 min.

Resultados Obtidos: Inicialmente os quatro clientes estão conversando normalmente e todos ouvem o que cada cliente fala. Quando os dois clientes abandonam a conferência, a mesma prossegue normalmente com os outros dois e está livre para receber novas conexões. Os clientes que deixaram a conferência não recebem mais os dados de áudio, passando a estabelecer agora uma conversa reservada. Essa conversa se processa normalmente, como apresentado no segundo cenário, sem que haja quebra de sigilo na comunicação.

6.4 Considerações finais

Os três cenários de testes escolhidos representam bem todas as funcionalidades de áudio disponíveis aos usuários do ambiente virtual colaborativo em situações reais de utilização. No transcorrer dos testes, os cenários não apresentaram comportamento inesperado.

O limiar para detecção do áudio apresentou variação em alguns clientes. Como se utiliza apenas a amplitude das amostras de áudio ocorreu casos em que a intensidade de voz do usuário não foi suficiente para que houvesse o reconhecimento da voz e a posterior transmissão dos dados de voz. Outro aspecto observado foi o da sensibilidade do microfone, ou seja, mesmo em situações em que a voz do usuário não é baixa, o microfone é responsável por diminuir o nível de intensidade sonora prejudicando a análise do algoritmo.

7. Conclusões

Neste trabalho foi apresentado o desenvolvimento de um sistema de distribuição de áudio (voz) integrado ao ambiente virtual do COGEST. Dois subsistemas foram implementados: um permite o diálogo em conferência entre vários usuários do sistema; o outro oferece suporte à privacidade da comunicação por meio de uma conversa reservada.

O principal desafio desse projeto foi estender uma aplicação existente pela adição de novas funcionalidades, sem adicionar restrições especiais ou acoplamento entre os diferentes módulos de sua arquitetura. Os módulos de software para manipulação de áudio foram adicionados de forma a utilizar alguns recursos já desenvolvidos por Anselmo [8], notadamente as funcionalidades para movimentação facial, sem que houvesse alteração no comportamento original do ambiente virtual.

A adição de recursos auditivos ao ambiente virtual colaborativo COGEST e a representação da origem da interlocução através de movimentos faciais do *avatar* amplia o potencial de interação entre os participantes da conferência a distância. Observa-se que a adição desses recursos reduz as limitações da dinâmica do trabalho em grupo ocasionadas pela perda de informação devido ao afastamento geográfico dos participantes, principal problema desse tipo de ambiente colaborativo.

Uma sugestão de trabalho futuro seria ajustar o limar para a detecção da voz em cada usuário. O desenvolvimento de um mecanismo que permita a cada usuário ajustar o valor mais adequado a sua situação de uso do aplicativo eliminaria os casos em que o usuário está falando, porém, como não há reconhecimento de voz, os dados não são transmitidos, prejudicando a comunicação. Além disso, a introdução de um sistema de reconhecimento fonético acrescentaria mais realismo ao ambiente virtual, uma vez que a animação facial seria plenamente coerente com que o usuário fala.

8. Referências

[1] SOARES, J. M.; HORAIN, P.; BIDEAU, A. Sharing and immersing applications in a 3d virtual inhabited world. Laval Virtual 5th virtual reality international conference, Laval, France, p. 2731, 2003.

[2] SOARES, J. M. Contribution à la communication gestuelle dans les environnements virtuels collaboratifs. Tese - Institut National de Télécommunications, Evry, France, 2004.

[3] Disponível em: http://idgnow.uol.com.br/mercado/2010/09/09/pnad-numero-de-lares-com-internet-cresceu-71-em-quatro-anos/.

[4] SANTOS, J. V. dos. RECOLLVE: Um Ambiente Virtual Voltado para a Representação de Atividades Colaborativas. Tese - Universidade Federal do Rio de Janeiro, Março 2009.

[5] VALENTE, C.; MATTAR, J. Second Life e WEB 2.0 na Educação. [S.l.]: novatec, 2007.

[6] GUYE-VUILLÈME, A.; CAPIN, T. K.; PANDZIC, I. S.; P, I. S.; THALMANN, N. M.; THALMANN, D. Nonverbal communication interface for collaborative virtual environments. Virtual Reality, v. 4, n. 1, p. 4959, March 1999.

[7] MPK20. Sun's Virtual Workplace. 2009. Disponível em: <http://research.sun.com/projects/mc/mpk20.html>.

[8] ANSELMO, F. J. M. Uma Insfraestrutura para a Construção de Ambientes Virtuais Colaborativos com Suporte à Comunicação Gestual. Dissertação - Universidade Federal do Ceará, Fortaleza, Brasil, 2010.

[9] SCHMIDT, K. The problem with 'awareness': Introductory remarks on 'awareness in cscw'. The Journal of Collaborative Computing, v. 11, n. 3-4, p. 285 289, 2002.

[10] ISO/IEC 14496-2: Information technology. [S.l.], 2009. Generic coding of audio-visual objects.

[11] PERAYA, D. Théories de la communication et technologies de l'information et dela communication. un apport réciproque. Revue européenne des sciences sociales, Mémoire et savoir à l'ère informatique, v. 14, n. 111, p. 171188, 1998.

[12] Humanoid Animation Working Group. 2009. ISO/IEC FCD19774:200x, Web3D Consortium. Disponível em: <http://h-anim.org>.

[13] Disponível em: http://download.oracle.com/javase/tutorial/sound/index.html

[14] Disponível em: http://www.oracle.com/technetwork/java/javase/tech/index-jsp-140239.html

[15] Disponível em: http://www.faqs.org/rfcs/rfc3261.html

[16] Disponível em: http://www.faqs.org/rfcs/rfc1889.html

[17] MORIMOTO, C. E. Tutorial VNC. 2009. Guia do Hardware. Disponível em: <http://www.guiadohardware.net/tutoriais/vnc/>

[18] ANSELMO, F. J. M.; SOARES, J. M.; CORTEZ, P. C.. Gestos Corporais e Expressões Faciais como Suporte Perceptivo em Ambientes Virtuais Colaborativos. In: 8th International Information and Telecommunication Technologies Symposium, 2009, Florianópolis. Atas do I2TS 2009, 2009. v. 1.

[19] JMF – Atribuições. Disponível em: <http://www.oracle.com/technetwork/java/ javase/attributions-139785.html>

[20] SOARES, J. M.; ANSELMO, F. J. M.; DOURADO JUNIOR, C. M. J. M.; MARCELINO, P. A.; BARROSO, G. C.; CORTEZ, P. C.. Uma Infra-Estrutura para a Colaboração à Distância com Suporte à Comunicação Gestual. In: SEMISH - XXXIII Seminário Integrado de Software e Hardware, 2006, Campo Grande. XXVI Congresso da Sociedade Brasileira de Computação. p. 418-432.

[21] Disponível em: < http://introcs.cs.princeton.edu/97data/FFT.java.html>

[22] < http://www.mathworks.com/products/matlab/>