

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Engenharia de Computação e Automação Industrial

Um Sistema de Video-conferência para Educação a Distância Baseado em Padrões Abertos

Raquel Cristina Bosnardo

Dissertação submetida à Faculdade de Engenharia Elétrica e de Computação da UNICAMP como parte dos requisitos exigidos para obtenção do título de Mestre em Engenharia Elétrica.

Banca Examinadora:

Prof. Dr. Eleri Cardózo (Orientador) – FEEC/UNICAMP
Prof. Dr. Waldomiro P.D.C. Loyolla – PUCCAMP
Prof. Dr. Marco Aurélio A. Henriques – FEEC/UNICAMP
Prof. Dr. Ricardo R. Gudwin – FEEC/UNICAMP

Campinas, Julho de 2001

UNICAMP
BIBLIOTECA CENTRA
SEÇÃO CIRCULANTE

Este exemplar corresponde à redação final da tese	
defendida por	Raquel Cristina
Bosnardo	aprovada pela Comissão
Julgada em	30/07/01
	Eleri Cardózo
	Orientador

UNIDADE 10 C
N.º CHAMADA:
T/UNICAMP
B652s
V. Ex.
TOMBO BC/ 47146
PROC. 16-837/02
C ☐ D ☒
PREÇO R\$ 11,00
DATA 12-01-2002
N.º CPD

CM00162410-3

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

B652s

Bosnardo, Raquel Cristina

Um sistema de viodeoconferência para educação
distância baseado em padrões abertos / Raquel Cristina
Bosnardo.--Campinas, SP: [s.n.], 2001.

Orientador: Eleri Cardozo.

Dissertação (mestrado) - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e
Computação.

1. Videoconferência. 2. JAVA (Linguagem de
programação de computador. 3. Ensino a distância. 4.
Processamento eletrônico de dados – Processamento
distribuído. 5. Programação orientada a objetos
(Computação). I. Cardozo, Eleri. II. Universidade
Estadual de Campinas. Faculdade de Engenharia Elétrica
e de Computação. III. Título.

Resumo

Bosnardo, R. C., Um Sistema de Educação a Distância Baseado em Padrões Abertos. Campinas: DCA/FEEC/UNICAMP, 2001. (Dissertação de Mestrado)

Este trabalho descreve o desenvolvimento de um sistema de videoconferência para redes de alto desempenho, que pode ser integrado em sistemas de educação a distância os quais necessitam desta funcionalidade.

O sistema é portátil, não proprietário e distribuído e permite a interação entre professor e estudante (em um ambiente de educação a distância) através de transmissões em áudio e vídeo, e através de notificações por eventos.

Para a implementação, foram utilizadas a arquitetura CORBA (Common Object Request Broker Architecture) e a linguagem Java. Dois serviços CORBA foram empregados: A/V Streams para o transporte da mídia contínua e o Serviço de Eventos para a notificação de eventos. O projeto e desenvolvimento utiliza tecnologias modernas tais como UML (Unified Modeling Language), componentes de *software*, e orientação a objetos.

Palavras-chaves: Videoconferência, CORBA, A/V Stream, Java, Serviço de Eventos, Evento, Educação a Distância.

Abstract

Bosnardo, R. C., A Distance Learning System Based on Open Standards. Campinas: DCA/FEEC/UNICAMP, 2001. (Master's Thesis)

This dissertation describes the development of a videoconference system targeted to high speed networks that can be integrated into distance learning systems lacking of this functionality.

The system is designed to be portable, open, and distributed, allowing the integration among teacher and students (in a distance learning environment) through live audio and video, and event notifications.

The implementation relies on the CORBA (Common Object Request Broker Architecture) architecture and the Java language. Two CORBA services were employed: A/V Streams for continuous media transport and Event Service for event notification. The design and development employ modern technologies such as UML (Unified Modeling Language), software components, and object-orientation.

Keywords: Videoconference, CORBA, A/V Stream, Java, Event Service, Event, Distance Learning.

Aos meus pais
Maria Inês e José Domingos
e meus irmãos Sandra e Paulo

Agradecimentos

A Deus

À minha família pelo apoio. Vocês representam um exemplo de amor e união.

Ao professor Eleri pela oportunidade, paciência e dedicação a orientação da tese.

A José Roberto e Rodrigo pela amizade e apoio durante a tese.

E a todos os colegas Érico, Jane, Faina, Rossano, Christian, Vinícios, Paulo Bueno, Verena, Rafael, Antônio Gomes, Tadeu, Luis, Lizet, José Soto, Franciraldo, Maria Angélica, Ellen, José Antônio, Luciana, Cláudio, Aurenice, Lavoisier, Ruan Carlos, Nádia, e outros, pela amizade e carinho durante o mestrado.

Índice

Listas de Figuras.....	viii
Lista de Tabelas.....	x

Introdução da Dissertação	1
1.1 Motivação	2
1.2 Objetivo da Tese	2
1.3 Contribuições	2
1.4 Conteúdo.....	2

Educação a Distância	3
2.1 Introdução	3
2.2 Histórico	4
2.3 Definições.....	5
2.4 Algumas Características da Educação a Distância	6
2.5 O Papel dos Atores da Educação a Distância.....	7
2.6 Recursos do Sistema de Educação a Distância.....	8
2.7 Educação a Distância: Classificação.....	9
2.8 Sistemas Educacionais Suportados por Computador	10
2.9 Cenários Síncronos	11
2.9.1 Cenário Utilizando a <i>Internet</i> Pública	11
2.9.2 Cenário Utilizando uma Rede Privada.....	13
2.9.3 Combinando os Dois Cenários	15

Tecnologias para Educação a Distância	17
3.1 Introdução	17
3.2 CORBA.....	18

3.2.1 ORB	18
3.2.2 Serviço de Eventos do CORBA	20
3.2.3 O Serviço A/V Streams.....	23
3.3 Java	26
3.3.1 JMF.....	27
3.3.2 A Comunicação Remota no Java.....	27
Java com CORBA	27
Java com RMI	28
3.4 WWW	29
3.5 Descrição de um Cenário	29
3.6 Tecnologias & Educação a Distância.....	31

Modelagem do Sistema de Videoconferência para Educação a Distância 32

4.1 Introdução	32
4.2 Integração dos Atores do Sistema.....	33
4.3 Representação das Classes	34
4.4 Descrição do Comportamento do Sistema	38
4.5 Responsabilidades das Classes	41
4.6 Diagramas de Estado	42

Projeto do Sistema de Videoconferência para Sistemas de Educação a Distância 44

5.1 Introdução	44
5.2 Projeto da Arquitetura.....	44
5.2.1 Infra-estrutura.....	44
5.2.2 Comunicação entre os Módulos	47
5.3 Projeto da Interface	49
5.3.1 Interface do Professor	50
5.3.2 Interface do Aluno	52
5.4 Projeto da Lógica do Controle	55

5.4.1 O Canal de Eventos “EventLogin”	56
5.4.2 O Canal de Eventos “EventRequest”	57
5.4.3 O Canal de Eventos “EventAllow”	58
5.4.4 O Canal de Eventos “EventStop”	59
5.5 Projeto da Comunicação Multimídia	60
5.5.1 Fluxo de Áudio e Vídeo entre Professor e Estudante	61
5.5.2 Controle dos FEPs	64
5.6 Caso de Teste.....	66
 Conclusão	 69
6.1 Introdução	69
6.2 Avaliação e Contribuições	69
6.3 Trabalhos Futuros	71
 Bibliografia	 73

Lista de Figuras

Figura 2.1 - Cenário síncrono utilizando a <i>Internet</i> como meio de acesso	12
Figura 2.2 - Cenário síncrono utilizando uma rede com QoS de serviço como meio de acesso.....	14
Figura 2.3 - Cenário síncrono utilizando a <i>Internet</i> e uma rede de alto desempenho como meio de acesso	15
Figura 3.1 - Cliente invoca objeto no Servidor via ORB	19
Figura 3.2 - Modelo <i>pull</i> - consumidor solicita eventos através da interface <i>pull</i> definida no produtor.....	21
Figura 3.3 - Modelo <i>push</i> - produtor envia eventos na interface <i>push</i> definida pelo consumidor	21
Figura 3.4 - Modelo <i>push</i> com mais de um produtor	21
Figura 3.5 - Modelo <i>push</i> com vários consumidores	22
Figura 3.6 - modelo <i>push</i> com mais de um produtor e consumidor.....	22
Figura 3.7 - Os componentes da especificação A/V Streams com as interfaces	24
Figura 3.8 - Transmissão de vídeo utilizando apenas as interfaces: Conexão de fluxo e Terminais de fluxo	25
Figura 3.9 - Duas conexões de fluxo de A para B, uma para o áudio e outra para o vídeo.....	25
Figura 3.10 - Transmissão em vídeo em um canal <i>multicast</i>	26
Figura 3.11 - Cenário para o sistema de educação a distância.....	29
Figura 4.1 - Atores e principais atividades	33
Figura 4.2 - Diagrama de classes apresentando uma “visão geral” do funcionamento do sistema	34
Figura 4.3 - Diagrama de classes da utilização de eventos no sistema	35
Figura 4.4 - Diagrama de classes das transmissões de mídia no sistema	36
Figura 4.5 - Diagrama de pacotes do sistema	37
Figura 4.6 – Transmissão da aula do professor e acesso dos alunos no sistema	38
Figura 4.7 - Aluno cancela seu pedido de palavra	39
Figura 4.8 - Pedido da palavra do aluno é aceito	39
Figura 4.9 - Transmissão do aluno com a palavra.....	39

Figura 4.10 - Aluno devolve a palavra	40
Figura 4.11 - Professor retoma a palavra.....	40
Figura 4.12 - Retirada dos alunos e término da aula.....	40
Figura 4.13 - Cartão CRC da classe Teacher.....	41
Figura 4.14 - Cartão CRC da classe Student	41
Figura 4.15 - Cartão CRC da classe UnicastVideoConnection	42
Figura 4.16 - Cartão CRC da classe EventChannel.....	42
Figura 4.17 - Diagrama de estados da classe Student	43
Figura 4.18 - Diagrama de estados da classe Teacher.....	43
Figura 5.1 - Infra-estrutura utilizada no sistema de educação a distância	46
Figura 5.2 – Diagrama de distribuição UML para o sistema AVE.....	48
Figura 5.3 - Telas de acesso do professor e do aluno	49
Figura 5.4 - Telas dos alunos presentes em aula, no computador do professor.....	50
Figura 5.5 - Tela do professor de manuseio da aula.....	51
Figura 5.6 - Janela de confirmação	51
Figura 5.7 - Tela de controle do áudio.....	52
Figura 5.8 - Tela de manuseio da aula, com as telas de áudio, vídeo e controle referentes a transmissão do professor	52
Figura 5.9 - Tela de manuseio do aluno com botão para pedir a palavra.....	53
Figura 5.10 - Tela de manuseio do aluno com botão para cancelar o pedido.....	53
Figura 5.11 - Tela de manuseio do aluno com botão para retorno da palavra.....	54
Figura 5.12 - Tela de controle do vídeo (acima) e do áudio.....	55
Figura 5.13 - Canal de eventos EventLogin.....	56
Figura 5.14 - Canal de eventos EventRequest	57
Figura 5.15 - Canal de eventos EventAllow quando o professor autoriza a palavra do aluno.....	58
Figura 5.16 - Canal de eventos EventAllow, quando o professor retoma a palavra	59
Figura 5.17 - Canal de eventos EventStop.....	59
Figura 5.18 - Transmissão <i>multicast</i> de vídeo, do professor para os alunos	62
Figura 5.19 - Transmissão <i>multicast</i> de áudio, do professor para os alunos	62
Figura 5.20 - Transmissão <i>multicast</i> de áudio, do aluno para o professor e demais alunos	62
Figura 5.21 - Transmissão <i>unicast</i> de vídeo, do aluno com a palavra para o professor.....	63

Figura 5.22 - Controle do FEP na transmissão <i>multicast</i> do áudio	65
Figura 5.23 - Controle do FEP na transmissão <i>multicast</i> do vídeo	65

Lista de Tabelas

Tabela 2.1 – Classificação dos vários tipos de educação a distância	6
---	---

Abreviações

ATM : Asynchronous Transfer Mode

CORBA: Common Object Request Broker Architecture

FEP: Flow End Point

IDL : Interface Definition Language

IIOP: Internet Inter-ORB Protocol

IP: Internet Protocol

JMF: Java Media Framework

LANE: Local Area Network Emulation

MMDevice: Multimedia Device

MPOA: Multiprotocol Over ATM

ORB: Object Request Broker

RTP: Real Time Protocol

SEP: Stream End Point

StreamCtrl: Stream Control

TCP: Transmission Control Protocol

UDP: User Datagram Protocol

UML: Unified Modeling Language

VDev: Virtual Multimedia Device

CRC: Class Responsibility Collaboration

Capítulo 1

Introdução

Nesta dissertação de mestrado descreve-se o desenvolvimento do sistema de videoconferência denominado “AVE” (Áudio, Vídeo e Eventos). O sistema AVE pode ser integrado em sistemas de educação a distância que necessitam comunicar professor e estudante com transmissões de áudio e vídeo.

No ambiente de educação a distância, o sistema de videoconferência permite a interação, quase que em tempo real, entre professor e estudante (além de possibilitar a notificação entre os mesmos).

A dissertação apresenta o desenvolvimento de um sistema de videoconferência, aberto (não proprietário), multiplataforma, distribuído, para redes de alto desempenho, que pode ser utilizado em sistemas de educação a distância. A necessidade de se utilizar redes de alto desempenho é devido, principalmente, a transmissão de vídeo. A opção por criar um sistema aberto, multiplataforma e distribuído permite que o sistema AVE possa ser utilizado em computadores diversificados permitindo assim uma abrangência maior.

O sistema AVE utiliza componentes de *software* gerando uma grande flexibilidade para sua execução. Com os recursos disponíveis atualmente (novas implementações dos Serviços de Eventos e A/V Streams) novas versões do sistema poderão ser totalmente implementados como *applets* Java e, assim, integrados em sistemas de Educação a Distância baseados em WWW.

Para o seu desenvolvimento foi necessário o emprego das tecnologias CORBA e Java. CORBA foi utilizado por apresentar uma série de serviços (como o Serviço de Eventos, A/V Streams), por possuir maturidade, por possuir interoperabilidade com implementações em outras linguagens (que empregam o padrão CORBA) e por poder ser utilizado em ambientes distribuídos e heterogêneos. Já a linguagem Java foi utilizada por ser multiplataforma (portabilidade), orientada a objetos, por possuir a API *Java Media Framework* (JMF) que contém recursos para a transmissão de mídia, e por prover uma implementação da arquitetura CORBA (*Java Interface Definition Language* - Java IDL).

1.1 Motivação e Objetivos

O crescimento de usuários conectados através de redes de computadores proporcionou uma alta demanda para o desenvolvimento de sistemas para a educação a distância [5,16]. Porém, muitos destes sistemas são proprietários.

Com a maturidade dos sistemas abertos, tornou-se possível construir sistemas de educação a distância não proprietários, multiplataforma e distribuídos. O objetivo desta tese é explorar estes tópicos, através do desenvolvimento de um sistema de videoconferência para educação a distância.

1.2 Contribuições

Como contribuição da dissertação tem-se:

- A viabilidade de se utilizar componentes de software no desenvolvimento através dos Serviços CORBA.
- A viabilidade de se desenvolver sistemas de educação a distância não proprietários, distribuídos e multiplataforma utilizando tecnologias recentes (CORBA e Java).
- Devido a sua modularidade, o sistema AVE pode ser aprimorado com a incorporação de novas funcionalidades.

1.3 Apresentação

A organização dos capítulos é a seguinte. Como o sistema AVE pode ser integrado em sistemas de educação a distância, no capítulo 2 faz-se uma apresentação sobre educação a distância. No capítulo 3, descreve-se as tecnologias que melhor se enquadram com as características desejadas para se desenvolver o sistema AVE. No capítulo 4, tem-se a modelagem do sistema AVE. No capítulo 5 descreve-se o desenvolvimento do sistema AVE, e no capítulo 6 tem-se a conclusão e trabalhos futuros.

Capítulo 2

Educação a Distância

Como o sistema AVE pode ser integrado em sistemas de educação a distância, o intuito deste capítulo é introduzir o conceito de educação a distância.

Neste capítulo apresenta-se um breve histórico da educação a distância, algumas definições e classificações que a mesma pode assumir. No capítulo compara-se ainda os principais sistemas educacionais suportados por computador, descreve-se algumas características da educação a distância, determina-se o papel dos atores neste tipo de ensino, bem como os recursos que podem estar disponíveis.

Posteriormente, apresentam-se três cenários de educação a distância: um utilizando a rede *Internet* pública, outro utilizando uma rede de alto desempenho e um terceiro combinando os dois primeiros.

2.1 Introdução

No modelo de educação presencial tradicional, professor e aluno necessitam estar presentes simultaneamente em um mesmo local para viabilizar a aula.

Nesse tipo de educação, o professor é a figura central responsável pelo aprendizado do aluno [1] tendo uma grande carga de trabalho sobre o desenvolvimento da aula. É responsabilidade do professor, determinar o assunto que será estudado, sua preparação e explicação para o aluno [2]. Neste caso, o conhecimento é passado para o aluno através de aulas sequenciais. Quando o aluno falta em uma ou mais aulas da sequência, o seu aprendizado pode ficar comprometido e dificultar o acompanhamento de aulas posteriores, já que é difícil repor uma aula perdida (muito menos uma série das mesmas).

Como é necessário que o aluno frequente as aulas assiduamente, muitas pessoas não conseguem estudar pela falta de condições de compatibilização de horário de trabalho com o horário das escolas convencionais. Outra dificuldade para o aluno que vive em regiões geograficamente distantes dos centros urbanos (onde as escolas se concentram) é percorrer grandes distâncias diariamente.

Percebendo a necessidade de uma adaptação da educação presencial tradicional em um novo contexto que seja capaz de atender o estudante geograficamente distante e/ou com tempo limitado, surgiu a educação a distância. Porém, é importante ressaltar que apesar da educação a distância suprir certas necessidades deste aluno, a mesma não deve ser vista como um substituto da educação presencial tradicional, já que tanto a educação a distância quanto a presencial tradicional procuram atender a características distintas sem o intuito da competição [2].

2.2 Histórico

A educação a distância não é uma idéia recente. Uma das primeiras soluções encontradas para ultrapassar as barreiras causadas pela educação presencial tradicional foi o curso por correspondência. Em 1881, Willian Rainey Harper ofereceu o primeiro curso à distância utilizando a correspondência [3]. Apesar do curso por correspondência atingir com sucesso o estudante remoto, ele era baseado em textos volumosos e deixava a desejar quando comparado ao modo presencial tradicional. O advento do rádio, da TV, do áudio e videocassetes trouxe uma nova fase para o educação a distância [3]. Tanto o rádio quanto a TV puderam transmitir o conhecimento para uma grande quantidade de pessoas ao mesmo tempo.

O rádio permitia que o aluno pudesse ouvir o professor em tempo real, porém assim como o audiocassete, também deixava a desejar em alguns aspectos, pela dificuldade do aluno imaginar situações que não podiam ser vistas, como por exemplo, quando o professor descrevia sobre a localização exata de um país (o que para a televisão não era um problema).

A TV, assim como o videocassete, viabilizaram o uso de recursos gráficos e de animação e permitiram, por exemplo, que os alunos pudessem observar através de um filme o funcionamento do aparelho digestivo. Por outro lado, a televisão era (e ainda é) limitada por um tempo pré-determinado de duração de transmissão [4], o que não torna flexível a questão do tempo do aluno. Já o áudio e o videocassete permitiram essa flexibilidade de horários, pois poderiam ser apresentados a qualquer hora e interrompidos quando necessário [4]. Apesar disso, ainda não existia a interatividade, ou seja, não havia uma participação ativa do estudante no processo de aprendizagem, o que tornava-o um observador passivo.

Em uma nova fase, com os sistemas de telecomunicações digitais e com a amplitude e custo acessível das redes, como a *Internet*, foi possível acrescentar a interatividade à educação a distância, como também a flexibilidade na comunicação entre professor e estudante [3].

Nesta nova fase, a facilidade de acesso a microcomputadores ligados à rede telefônica via *modem*, fez com que os educadores rapidamente percebessem o potencial da *World Wide Web* para prover a educação a distância, pois suas interfaces existem em todas as plataformas, são fáceis de serem usadas e possuem habilidade de manusear vários tipos de mídia [5], permitindo, a interatividade e apresentando facilidades para comunicação entre professor e estudante.

A comunicação telefônica e a instalação maciça de cabos de fibra ótica possibilitou a introdução de meios adequados para a teleconferência e a integração de recursos multimídia em computadores pessoais [1], abrindo espaço para a comunicação de áudio e vídeo em tempo real.

Atualmente, a educação a distância tem sido utilizada no meio educacional em vários países do mundo, principalmente naqueles com grandes dimensões territoriais como o Canadá, Austrália e Estados Unidos, tornando-se sinônimo de economia e flexibilidade [1] [3].

Embora ainda existam formas de educação a distância que utilizem a correspondência, a TV, o rádio, o áudio e videocassete, o assunto em questão utiliza a tecnologia baseada em redes de computadores.

2.3 Definições

Segundo Garcia Aretio [6], “a educação a distância é um sistema tecnológico de comunicação bidirecional, que pode ser maciça e substituir a interação pessoal na sala de aula (professor - estudante) pela ação sistemática e conjunta de diversos recursos didáticos e pelo apoio de uma organização e tutoria que propiciam a aprendizagem independente e flexível dos alunos”.

Não muito diferente da definição anterior, G. Dohmem [6] define a educação a distância como “uma forma sistematicamente e organizada de auto-estudo, no qual o aluno se instrui a partir do material que lhe é apresentado; onde o acompanhamento e a supervisão do sucesso do aluno são conduzidos por um grupo de professores, sendo isto possível através da aplicação de meios de comunicação capazes de vencer essa distância, mesmo longa”.

Seguindo essa idéia de auto-estudo, France Henri [6] relata, em sua definição, que a “educação a distância é o produto da organização de atividades e de recursos pedagógicos dos quais o aluno se serve de forma autônoma, seguindo seus próprios desejos, sem que lhe seja imposto submeter-se a limitações espaço-temporais nem a relações de autoridade da formação presencial tradicional”. Já outros autores como Miguel A. Ramón Martínez [6] mencionam o fato de que “qualquer pessoa, independentemente do tempo e do espaço, pode converter-se em sujeito protagonista de sua própria aprendizagem, graças ao uso sistemático de materiais educativos, reforçado por diferentes meios e formas de comunicação”.

Três dos autores citam que a educação a distância utiliza um meio tecnológico de comunicação. Embora a comunicação entre professor e aluno seja intermediada, alguns sistemas ainda necessitam do contato pessoal [2]. Existem momentos que a presença do indivíduo é importante, como exemplo, para comprovação do aprendizado do aluno [2]. Alguns sistemas necessitam da presença física do estudante apenas quando é realizada uma avaliação do mesmo. Certamente, isto pode vir a ser impraticável em grandes distâncias.

Do mesmo modo, existem exceções na definição de France Henri em que o aluno segue de forma autônoma seus próprios desejos, pois nem sempre é permitido que o aluno escolha determinado assunto primeiro que outros [7]. Por outro lado, é fato que o aluno possui uma responsabilidade maior pelo próprio aprendizado [1,2,5].

Um outro ponto mencionado por Garcia Aretio e G. Dohmem é a organização de apoio ao estudante. Essa organização serve para acompanhar o aluno no decorrer do curso. Porém, esta organização necessariamente não precisa ser composta de professores e pode não existir em todos os sistemas.

Em suma, pode-se dizer que a educação a distância é um sistema de transmissão de informação para o estudante, independente de sua localização geográfica, em que a comunicação entre professor e aluno é mediada. A escolha do meio de comunicação depende do tipo do público alvo, dos custos operacionais e da eficácia da transmissão e recepção [1]. Além disso, vários recursos podem ser empregados na apresentação da aula, sendo que esses recursos podem ser apoiados por uma organização (escola).

2.4 Algumas Características da Educação a Distância

Uma das principais características da educação a distância é a distância geográfica que separa professores e estudantes. Como essa distância passou a ser um problema contornável pela tecnologia, tornou-se possível um acesso maior ao ensino. Nestes termos, pode-se acrescentar que este tipo de educação pode cobrir uma área

geograficamente maior que os sistemas educacionais convencionais [1], obtendo assim um grande número de alunos beneficiados [1,8] e sem o risco de reduzir a qualidade dos serviços oferecidos [9], além do fato de oferecer a oportunidade de se trocar experiências educacionais (colaboração intelectual) [1,5,8]. Outra característica interessante é a grande quantidade de informações disponíveis para pesquisa na *Internet* que indiretamente fortalece o ambiente de estudo do aluno [5,9]. Porém, para que isso seja possível é necessário que tanto o aluno quanto o professor tenham um meio de comunicação disponível, como por exemplo, um computador ligado à rede para receber e transmitir informações.

2.5 O Papel dos Atores da Educação a Distância

Embora o ensino esteja mudando, o papel do professor ainda é o de guia e de detentor do conhecimento [2,5]. Todavia, por indicar o melhor caminho a seguir, o professor passa a ser um facilitador e não mais um orador, pois a apresentação do assunto em estudo passa a contar com o auxílio da tecnologia utilizada. A responsabilidade do professor está em criar situações de aprendizado, prover oportunidades de comunicação entre os participantes, identificar as dificuldades dos estudantes e responder a dúvidas, podendo também elaborar o material didático ou deixar esta tarefa para um terceiro [2].

O aluno, por sua vez, continua desempenhando o papel de aprendiz [2]. Ele pode trabalhar isoladamente ou colaborativamente com outros estudantes, porém, ao invés do aluno ser um receptor passivo como na educação presencial tradicional, ele passa a ter a liberdade de poder trabalhar no seu próprio ritmo, ficando livre para descobrir e explorar, tornando-se responsável por sua própria aquisição de conhecimento [1,2,5].

Além do professor e do estudante outros “atores” podem fazer parte da educação a distância. Nem todos os sistemas possuem todos os tipos de atores e as vezes o papel desempenhado por um, pode ser suprimido por outro. Por exemplo, o professor também pode escrever o conteúdo de um material, desempenhando desse modo, o papel de professor e de autor¹.

Outro ator que é pouco mencionado é o administrador. Ele trata de questões operacionais, como por exemplo, a matrícula dos alunos e o acesso dos indivíduos ao sistema. As vezes, alguns papéis, apesar de diferentes, se confundem. É o caso do professor, do tutor e do monitor. O papel do tutor é o de orientador da aprendizagem do aluno. Ele determina tópicos de estudo, sugere novas leituras ou atividades, auxilia o aluno na construção do seu próprio caminho, corrige provas e exercícios [1]. Este ator

¹ O papel específico do autor é criar e atualizar o conteúdo do material disponível no curso.

pode ser um programa interativo, com instruções pré-determinadas. Já o monitor é aquele que acompanha o aluno no decorrer do curso, podendo ajudar e supervisionar grupos de estudantes em trabalhos em equipe. Certamente os papéis de professor, tutor e monitor podem ser supridos por um só indivíduo, o que geralmente não acontece em sistemas muito complexos.

2.6 Recursos do Sistema de Educação a Distância

Os sistemas de educação a distância podem conter vários recursos como: autoria, escrita colaborativa, aplicações compartilhadas, sistemas de anotação, provas, ambiente de suporte do curso e ferramentas de comunicação.

O recurso de autoria, como o próprio nome indica, permite que os autores tenham meios de criar e acrescentar o conteúdo do material no sistema. Como geralmente o material é apresentado na forma de páginas HTML (Hypertext Markup Language), alguns sistemas utilizam editores HTML comerciais para suporte a autoria [10]. O problema destes sistemas é que o autor deve conhecer tais editores.

Aplicações para escrita colaborativa servem para viabilizar a construção de um texto escrito por vários autores geograficamente dispersos ou não (funciona como um ambiente compartilhado de edição de documentos texto) [11,12].

Os sistemas de anotação [13,14] permitem que o aluno escreva suas anotações no ambiente de aprendizado. Estas anotações podem ser alteradas ou excluídas pelo aluno sem interferir no conteúdo do curso.

O ambiente de suporte do curso [11] usualmente é um *Web site* que abriga o curso. Este *site* pode ser criado e mantido pelo instrutor ou pelo autor do material, ou ser mantido como parte de um sistema integrado servindo um departamento ou faculdade. Como exemplo de ambiente de suporte ao curso, dentre muitas outras pode-se citar Vrije University de Amsterdam, que possui um site onde os estudantes são informados sobre as edições dos cursos, se comunicam entre si e com o instrutor, e publicam seus resultados.

Quanto às ferramentas de comunicação, vários tipos de tecnologia podem ser empregadas. Uma forma conhecida são os *chats* e o correio eletrônico (*e-mail*) para esclarecimento de dúvidas, comentários e trocas de informações (até mesmo entre alunos). Outra forma mais adequada de comunicação para educação a distância, inclui a videoconferência entre professor e alunos.

2.7 Educação a Distância : Classificação

Na tabela 2.1 estão apresentadas diferentes variedades de aplicações para a educação a distância, considerando o tempo e o local (presença física) do professor e do estudante [5].

	Professor e aluno se encontram em um mesmo local	Professor e aluno as vezes se encontram em um mesmo local	Professor e aluno se nunca se encontram em um mesmo local
Assíncrono			A
Parcialmente assíncrono/ síncrono	D	E	B
Síncrono			C

Tabela 2.1 – Classificação dos vários tipos de educação a distância

A - Os alunos podem por exemplo, usar a *Web* para obter o material do curso e se comunicarem (com professor ou outros alunos) através de *e-mails*. Este tipo de sistema permite uma maior flexibilidade para o aluno, por não requerer a comunicação professor e aluno num mesmo espaço de tempo.

B - Os alunos podem por exemplo, usar a *Web* para obter o material do curso e quando precisam ser comunicar (com professor ou outros alunos) podem interagir utilizando um *chat*.

C - Professor e aluno utilizam por exemplo, um sistema de videoconferência onde questões e respostas podem ser obtidas em tempo real.

D - Representa um curso em que aulas presenciais são combinadas com aulas por videoconferência.

E - Situação em que por exemplo, o aluno estuda através de páginas da *Web* e se comunica por *e-mail* ou *chat*, mas com a exceção de algumas vezes, professor e aluno se encontram pessoalmente para fins de avaliação.

O sistema AVE serve para sistemas de educação a distância do tipo B, C, D e E (que utilizam comunicação síncrona).

Com exceção do item C, segundo os componentes definidos por Keegan [6], o estudante fica separado do professor por pelo menos parte do tempo do aprendizado. Desse modo, o conteúdo do curso deverá ser atrativo o suficiente para poder prender a atenção do aluno[2], daí a necessidade da interatividade [1,2,3]. Por exemplo, em uma apresentação sobre o funcionamento de um motor, o assunto é introduzido e explicado por uma

animação em computador. Posteriormente, o aluno poderá estudar o comportamento desse motor simulando efeitos sobre ele e verificando sua reação. Para isso, o conteúdo dos materiais que serão apresentados ao aluno devem ser preparados por uma equipe multidisciplinar que incorpore técnicas de auto-instrução [1,3]. Esta equipe pode ser composta pelo próprio professor e/ou autores.

Visto que o aluno tem que buscar o conhecimento, lendo e interagindo com o sistema, além de poder ter a opção de escolher a hora mais adequada de estudo (ver no quadro anterior os casos A, B e E) ele acaba se tornando mais responsável pelo seu próprio processo de aprendizado [1,2,5].

2.8 Sistemas Educacionais Suportados por Computador

Nem todos os sistemas educacionais podem ser considerados como educação a distância. Por exemplo, um livro “faça você mesmo”, não representa esta modalidade de ensino. É interessante que se tenha a via de comunicação entre o professor e o aluno, assim como um processo pedagógico organizado e contínuo, ou seja, uma metodologia de ensino que se preocupe com o aprendizado do aluno [1]

Para se ter um melhor entendimento sobre os sistemas de educação a distância, a seguir são descritos alguns sistemas suportados por computador. O intuito é ressaltar que nem todos os sistemas suportados por computador podem ser considerados como educação a distância.

O livro *on-line* [1,13] é um texto sobre determinado assunto organizado em uma sequência de páginas disponibilizadas na *Web*. Apesar de ser um sistema suportado por computador, não existe uma forma de comunicação com o professor e não há uma preocupação em acompanhar o aluno para avaliar o seu aprendizado. Portanto não pode ser considerado como educação a distância. Entretanto este sistema pode ser utilizado em conjunto com outros sistemas, por exemplo, *chat* e *e-mail*.

Não muito diferente do livro eletrônico, o *software* educacional [13] é um sistema especializado que cobre um curso, projetado para executar em um computador não necessariamente conectado em rede. Por isso, uma das dificuldades é que também não existe nenhuma conexão com o autor/professor do curso. Sendo assim, o *software* tem que ser autocontido e restrito a um nível de dificuldade. Do mesmo modo que o livro *on-line*, este tipo de sistema não se preocupa em acompanhar e avaliar o aprendizado do aluno, o que também não pode ser considerado como educação a distância.

Em sistemas de educação *on-line*, geralmente o curso é disponibilizado na *Web* para o aluno através de páginas HTML que são previamente preparadas e apresentadas através de *browsers*, podendo ser enriquecidas com Java ou *Javascript* [13]. Como este tipo de

sistema geralmente não requer a presença do professor e do aluno simultaneamente (ver tabela 2.1, itens A, B e E), o acesso às páginas é independente do tempo e da localização, sendo portanto um sistema assíncrono. Neste tipo de curso existe a comunicação entre ambas as partes, professor - estudante, assim como um acompanhamento do aluno, contemplando uma avaliação sobre o que está sendo estudado. Desse modo, este tipo de sistema pode ser considerado como educação a distância.

Em sistemas síncronos de educação *on-line*, câmeras de vídeo e facilidades de áudio permitem que o curso seja transmitido ao vivo através da rede, e alcance lugares distantes onde estudantes sejam capazes de participar remotamente [13].

Serviços disponibilizados em rede permitem que os alunos interajam através do computador, abrindo uma possibilidade de facilidades de comunicação em alta velocidade e não apenas unidirecional [13] (ver tabela 2.1, itens C e D). Dessa maneira, como nos sistemas baseados em *Web*, existe a comunicação entre professor e estudante e uma avaliação do aluno no curso, até mesmo em tempo real. Assim sendo, este sistema também pode ser considerado como educação a distância.

2.9 Cenários Síncronos

Para alcançar o aluno remoto é possível utilizar uma rede de computadores para transmitir a aula e permitir a comunicação entre professor e estudante. Em alguns sistemas de educação a distância essa comunicação exige a presença simultânea de alunos e professor (síncrona) e em outras não (assíncrona). O sistema de videoconferência (como o sistema proposto) é um sistema síncrono, dessa forma, serão apresentados os cenários mais comuns para sistemas síncronos.

2.9.1 Cenário Utilizando a *Internet* Pública

No primeiro cenário, conforme ilustrado na figura 2.1, cada estudante utiliza um computador equipado com microfone e câmera de vídeo, ligado na *Internet* através de um *modem*.

Através do *browser* instalado em seu computador, o aluno poderá acessar a página do curso, ver e ouvir o professor e ter o controle sobre algumas operações do curso, como por exemplo, rever uma aula quando assim necessitar.

O computador do estudante pode estar na própria casa do aluno ou em um lugar acessível. O professor, em outro local, utiliza um computador com uma ou mais câmeras e microfone, com conexão à *Internet*, além de outros recursos como *whiteboard* e páginas *Web*. Em seu computador existe um conjunto de informações armazenadas onde constam dentre outras: as aulas gravadas, os alunos matriculados e as operações de controle do curso, como por exemplo, usar um *Whiteboard* quando achar necessário ou restringir o acesso dos alunos a determinadas partes do curso.

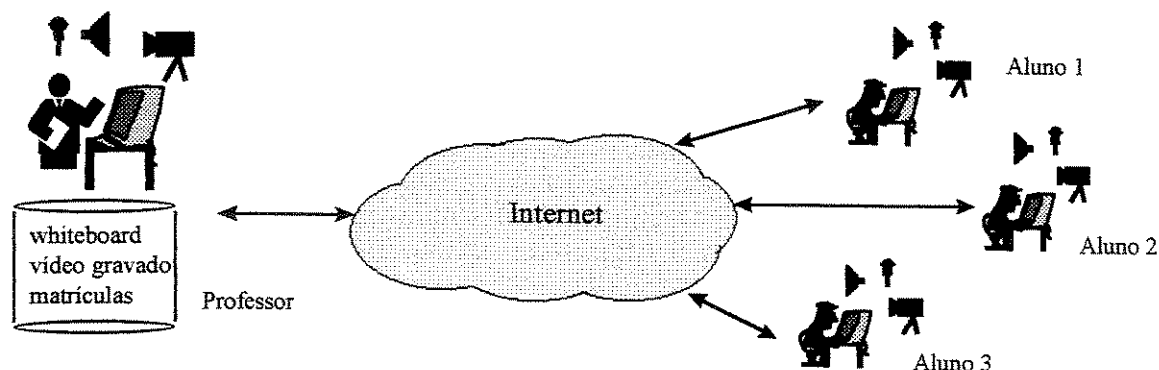


Figura 2.1 - Cenário síncrono utilizando a *Internet* como meio de acesso

Seguido do acesso ao endereço da página do curso, tanto professor quanto estudantes devem se identificar, para fins de segurança. Além dessa questão de segurança, a autenticação também permite a existência de um tipo de acesso diferente para os participantes. Por exemplo, ela pode restringir o acesso antecipado de um teste, somente para os alunos e não para o professor.

O aluno é considerado presente no curso quando ele se identifica e ganha acesso ao sistema. Essa presença é passada diretamente para a tela do professor através da rede, na forma de uma lista de nomes de estudantes. Assim, quando o aluno entrar no curso seu nome será adicionado a essa lista e ao sair seu nome será retirado.

Dependendo do sistema, somente os nomes dos estudantes, sem as suas respectivas imagens, são suficientes para o professor poder acompanhar a classe, por exemplo para organizar equipes ou para distribuir tarefas.

Também existem sistemas que raramente usam a transmissão de vídeo do aluno, já que o envio constante da imagem do aluno aumenta o tráfego da rede e “polui” a tela do professor com uma quantidade excessiva de imagens que podem até ser consideradas desnecessárias (dependendo de como a aula é conduzida).

Quanto a transmissão de áudio, há aplicações que deixam a cargo do professor o controle da permissão da fala dos alunos. Na aula presencial tradicional, quando um aluno quer falar ele sinaliza com a mão e fica a espera do consentimento do professor. No sistema, o pedido de fala é feito pelo aluno acionando um botão ou ícone na tela do computador.

Os pedidos de fala de todos os estudantes são vistos na tela do professor, cabendo ao próprio professor consentir ou não a palavra a um determinado aluno.

É interessante notar que manter o controle da permissão da fala (e de imagens) nas mãos do professor permite que ele controle o acesso da via de transmissão, facilitando assim o processo de comunicação.

De um modo geral, este cenário é bem flexível, principalmente por utilizar a *Internet*. Como ela cobre boa parte do globo, os estudantes mais distantes podem ser alcançados. Além disso, o aluno não precisa se deslocar até a escola (a não ser esporadicamente para avaliação) e permite ainda que o estudante individualmente, possa rever uma aula previamente gravada, bastando que ele acesse a aula referida (o que pode levar muito tempo, dependendo da velocidade da rede).

Por não se tratar de uma tecnologia cara, muitas pessoas podem ter acesso a ela, porém, por oferecer um serviço de melhor esforço [15], não se pode garantir a qualidade da transmissão da aula.

Este cenário é representado pela utilização de ferramentas desenvolvidas no âmbito do MBONE (*Internet Multicast Bone*) tais como VIC, VAT e SD, WB. Estas ferramentas operam sobre a Internet pública, mas requerem a capacidade de comunicação *multicast*.

2.9.2 Cenário Utilizando uma Rede Privada

Conforme pode ser observado na figura 2.2, o segundo cenário tem de um lado o professor em uma sala e do outro estudantes agrupados em um mesmo local assistindo essa aula.

O professor utiliza um computador, com uma ou mais câmeras e microfone, que executa um sistema específico de ensino a distância. Este computador é ligado em uma rede privada de alto desempenho, por exemplo, em uma rede ATM (*Asynchronous Transfer Mode*).

Na sala dos estudantes, os alunos assistem através de telões (ou de aparelhos de TV), o curso transmitido via rede pelo professor. Estes telões são conectados em computadores também equipados com câmeras e microfones, ligados à rede privada.

Dependendo do sistema, um telão projeta a imagem do professor com a apresentação de recursos didáticos complementares. Por exemplo, um *Whiteboard* ou algo semelhante como uma câmera focando um quadro em que o professor expõe um assunto. Um outro telão, conectado em outro computador, apresenta as anotações (como se fossem transparências) do curso.

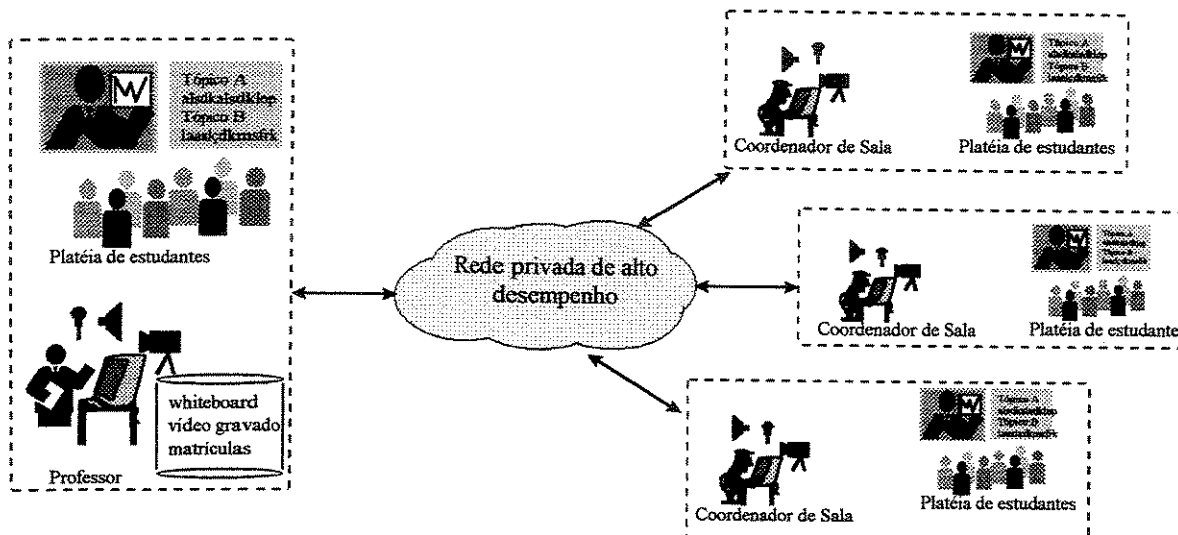


Figura 2.2 – Cenário síncrono utilizando uma rede com QoS de serviço como meio de acesso

Neste cenário, áudio e vídeo provenientes das salas são apresentados no(s) terminal(ais) do professor. Cada aula ministrada é automaticamente gravada pelo sistema em um servidor de vídeo, para que os alunos possam revê-las assim que necessitarem.

A sala dos alunos conta com a ajuda de um coordenador de sala. Dentre as várias tarefas que ele pode executar, o coordenador de sala acessa o sistema para estabelecer a conexão dos alunos com o professor, carrega a aula gravada do servidor de vídeo e as disponibiliza para os alunos no formato de fitas de vídeo e/ou disco ótico.

O professor pode controlar a presença apenas observando a platéia de alunos através do vídeo, ou por um teclado disponível na porta de entrada da sala dos estudantes. Neste caso cada aluno digita uma senha indicando a sua presença, que é transmitida através da rede para o professor. Porém, o método mais comum é o coordenador de sala passar uma lista de chamada para os estudantes e enviá-la para o professor.

Não existe a necessidade de usar a *Web* para a comunicação entre professor e estudantes, porém os computadores podem estar conectados à *Internet* durante a aula, caso o professor queira utilizar a *Web* como uma das fontes de material de aprendizado para apresentar aos estudantes.

Uma vantagem em relação ao cenário anterior é que a transmissão possui qualidade, considerando que, dependendo da tecnologia da rede privada, existe garantia de reserva de recursos.

Na transmissão da imagem vários tipos de câmeras podem ser usadas, tanto fixas quanto móveis. Existe também a possibilidade do professor controlar as câmeras remotamente. Para aplicações que usam câmeras fixas, geralmente é disponibilizada uma câmera para

focar a platéia e uma outra para focar um aluno apresentando uma dúvida diante de um quadro.

Comparando com o cenário anterior, este cenário não é flexível e a tecnologia empregada é cara, porém, a rede privada permite qualidade na transmissão por oferecer garantia de reserva de recursos. Isto faz com que a aula possa ser apresentada com qualidade constante, pois os recursos suficientes para esta transmissão são mantidos durante toda a aula.

Este cenário é típico quando uma rede corporativa é empregada em situações de treinamento interno à corporação.

2.9.3 Combinando os Dois Cenários

Conforme pode ser observado na figura 2.3, a idéia básica deste terceiro cenário é combinar os dois cenários anteriores utilizando uma rede privada de alto desempenho e a *Internet* pública. Desse modo, é possível ter-se em uma sala estudantes assistindo uma aula transmitida via rede privada e, ao mesmo tempo, cada aluno da sala pode dispor de um computador conectado à *Internet* pública. Além de permitir que os alunos acompanhem o curso, o acesso à *Internet* também permite que os alunos pesquisem assuntos relacionados a aula, quando necessário. Já a rede privada transmite o áudio e o vídeo do professor, por oferecer garantia de reserva de recursos.

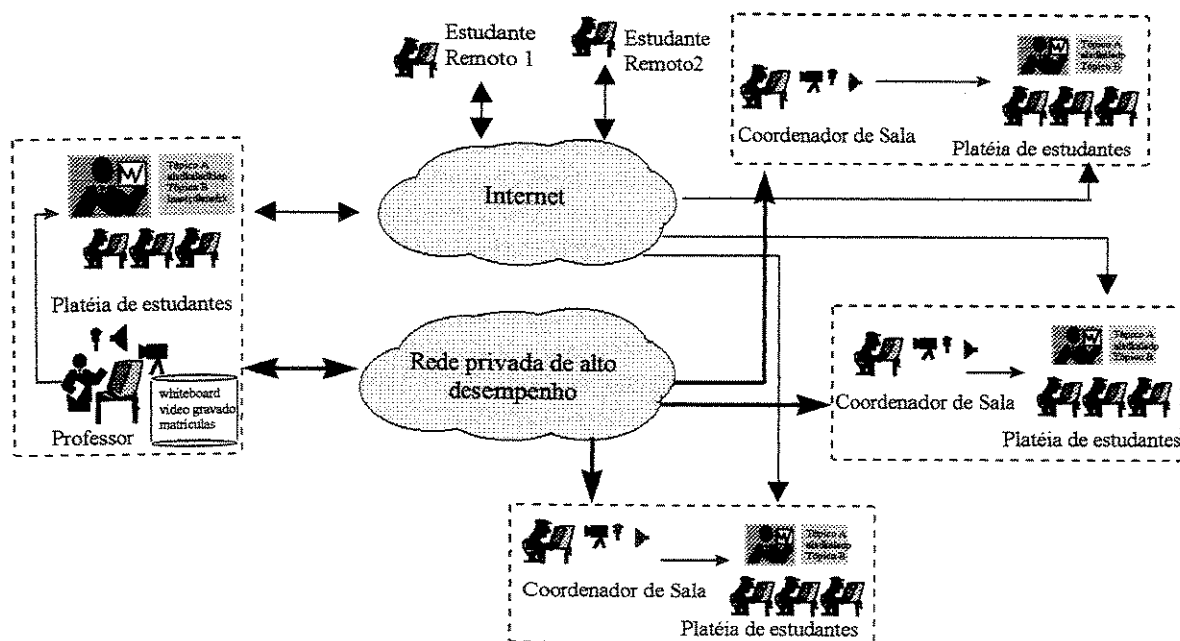


Figura 2.3 - Cenário síncrono utilizando a *Internet* e uma rede de alto desempenho como meio de acesso

Do lado do professor existe um computador equipado com microfone e uma ou mais câmeras para a apresentação da aula. Como pode ser visto na figura 2.3, as conexões através das redes privada e *Internet* funcionam em paralelo. Na sala dos estudantes, o computador do coordenador e dos alunos não precisam se comunicar. O computador do coordenador de sala está conectado à rede privada e é equipado com câmeras e microfones para capturar áudio e vídeo da sala e enviá-los ao terminal do professor.

Os alunos podem expressar suas dúvidas diante de um quadro com uma câmera e um microfone.

Cada aula transmitida é gravada num servidor de vídeo do mesmo modo que o cenário anterior.

Caso o aluno não possa se deslocar até a sala onde exista a platéia de estudantes, o áudio do professor poderá ser transmitido através da *Internet*. Assim, remotamente seria possível acompanhar as anotações do curso e ouvir a aula (neste caso, a transmissão da imagem não seria conveniente pelas condições imprevisíveis da *Internet*). Além disso, este aluno remoto pode tirar suas dúvidas com o professor por *e-mail* ou com um monitor de curso por *chat*.

Pode-se então observar que combinando os dois cenários anteriores tem-se um outro cenário que reúne a flexibilidade da rede *Internet* com a qualidade assegurada pela rede privada. Desse modo, as duas redes juntas podem atingir os alunos em suas casas, como também um grupo reunido em uma sala. A *Internet* pode alcançar os estudantes mais remotos por se tratar de uma rede de alcance mundial, e a rede privada pode compensar a imprevisibilidade e a falta de qualidade da *Internet*, garantindo a reserva de recursos durante toda a transmissão do áudio e vídeo entre professor e alunos.

Capítulo 3

Tecnologias para o Sistema de Videoconferência

Para a criação do sistema de videoconferência é necessário o emprego de algumas tecnologias. Neste capítulo descreve-se as possíveis tecnologias que podem ser utilizadas para o desenvolvimento do sistema AVE. Dentre as tecnologias, descreve-se a arquitetura CORBA, o Serviço de Nomes, o Serviço de Eventos, o Serviço de transmissão de mídia contínua (*A/V Streams*), a linguagem Java, o uso da WWW (*World Wide Web*) e analisa-se as tecnologias que melhor se adaptam à concepção de sistema aberto para ser integrado a educação a distância.

3.1 Introdução

O crescimento de usuários conectados através de redes de computadores proporcionou uma alta demanda de desenvolvimento de *software* distribuído, como por exemplo, softwares para educação a distância [16]. Para o desenvolvimento de um *software* de videoconferência para sistemas de educação a distância, uma série de tecnologias podem ser combinadas para gerar um sistema, não proprietário, em um ambiente distribuído. Dentre as linguagens de programação, as linguagens orientadas a objetos, como Java, são bastante atrativas, pois “ênfatizam a importância do reuso, da manutenção, da flexibilidade e da modularidade no processo de desenvolvimento do *software*” [16]. Para que os objetos possam ser utilizados em ambientes distribuídos, é necessário o emprego de outra tecnologia, como por exemplo, a arquitetura CORBA (*Common Object Request Broker Architecture*), pois as linguagens orientadas a objetos tal como Java, oferecem recursos limitados para gerenciamento de objetos distribuídos. Dessa forma, a integração

de linguagens orientadas a objetos com um padrão como CORBA, viabiliza o desenvolvimento de sistemas de educação a distância, segundo o modelo cliente/servidor em ambientes distribuídos e heterogêneos.

3.2 CORBA

A tecnologia CORBA se destaca dentre outras tecnologias, como por exemplo DCOM² (*Distributed Component Object Model*) da Microsoft, devido a ampla aceitação na indústria, heterogenidade, maturidade, simplicidade e o grande conjunto de serviços de objeto disponíveis [16].

CORBA [17] é patrocinado pelo consórcio OMG³ (*Object Management Group*) e provê interoperabilidade de aplicações distribuídas permitindo que estas aplicações se comuniquem entre si como se estivessem em um único espaço de endereçamento.

Esta arquitetura define um *framework* para desenvolvimento de aplicações distribuídas trazendo consigo técnicas de orientação a objeto e paradigma cliente-servidor. Assim, objetos podem atuar como clientes requisitando serviços ou como servidores provendo serviços. Um objeto que age como servidor pode ainda agir como cliente em relação a outro servidor.

CORBA permite que um objeto cliente invoque transparentemente um método em um objeto servidor sem levar em conta a sua localização ou detalhes internos ao objeto e ao servidor.

3.2.1 ORB

CORBA define uma arquitetura padrão para objetos distribuídos baseada em ORB (*Object Request Broker*). ORB [17] [18] é um componente de *software* que intermedia a transferência de informação para um objeto localizado em um servidor remoto, escondendo a complexidade relativa à distribuição. ORB permite que o cliente tenha acesso à referência de um objeto e que possa invocar métodos nesse objeto. Para tal, o servidor registra o objeto com um nome único em um Serviço de Nomes CORBA, de modo que o cliente possa posteriormente obter uma referência desse objeto consultando o mesmo Serviço de Nomes.

² <http://www.microsoft.com/NTServer/appservice/exec/overview/dcombus.asp>

³ Trata-se de um consórcio para definir uma arquitetura para a distribuição e interação de componentes de *software* que tem como proposta a arquitetura CORBA (<http://www.omg.org>)

Para invocar métodos em objetos servidores, o cliente necessita conhecer a interface do objeto expressa em IDL (*Interface Definition Language*). IDL permite especificar um conjunto de métodos de forma neutra, isto é, independente da técnica de implementação, da linguagem de programação usada, ou do sistema operacional onde os métodos serão executados.

A compilação de interfaces IDL gera classes *stubs* e *skeletons*, respectivamente, para o cliente e o servidor, que são usadas como *proxies* na invocação do objeto remoto. O *stub* contém informações para que o cliente possa fazer a invocação remota do objeto. Ele é transparentemente usado pelo ORB durante a invocação. No servidor, a classe *skeleton* é usada pelo ORB para :

- traduzir a invocação remota em uma invocação local;
- chamar o método a ser invocado;
- retornar o resultado para o cliente.

A figura 3.1 ilustra o modelo. Como pode ser observado na figura, a interface é representada pelo símbolo “”.

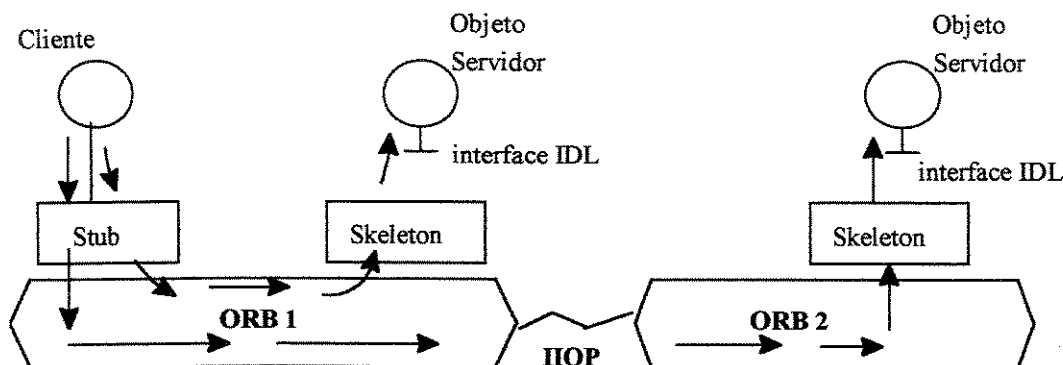


Figura 3.1 - Cliente invoca objeto no Servidor via ORB

As interfaces especificadas em IDL podem ser implementadas em linguagens de programação que tenham o mapeamento IDL, como, por exemplo, Java, C++ e Smalltalk. Isso permite que aplicações CORBA escritas em linguagens diferentes possam interoperar.

Quanto à interoperabilidade⁴ entre diferentes implementações de ORB, a arquitetura CORBA especifica o protocolo IIOP (*Internet Inter-ORB Protocol*) que utiliza o protocolo TCP/IP.

⁴ CORBA também define o protocolo GIOP (*Generic Inter-ORB Protocol*) projetado para trabalhar sobre qualquer protocolo de transporte orientado à conexão.

Além do Serviço de Nomes, CORBA define um conjunto de serviços [18] que permite que as aplicações se comuniquem de maneira padrão. Estes serviços incluem:

- o Serviço de Negociação (*Trade Service*) que permite o cliente localizar referências de objetos baseadas na propriedade desejada de um objeto;
- o Serviço de Transação de Objeto (*Object Transaction Service*) que permite objetos CORBA interagirem usando modelos de processamento transacional;
- o Serviço de Segurança (*Security Service*) que permite programas CORBA interagirem usando comunicações seguras; e
- o Serviço de Eventos (*Event Service*) que permite os objetos notificarem outros objetos sobre a ocorrência de eventos.

Os serviços são especificações que precisam ser implementadas.

3.2.2 Serviço de Eventos CORBA

O Serviço de Eventos CORBA [19] provê um serviço de “notificação”. Objetos que geram notificações, atuam como produtores de eventos e as partes interessadas em receber a notificação atuam como consumidores de eventos. Dessa forma, um evento indica que uma ação ocorreu, como, por exemplo, um botão foi pressionado. O evento é gerado por um objeto produtor e enviado através do serviço para um ou mais objetos consumidores. A informação contida no evento pode ser um objeto qualquer, dependendo da necessidade da aplicação.

O evento se propaga do produtor para o consumidor com a ajuda de um objeto denominado canal de eventos, usado como intermediário entre consumidores e produtores. Como os eventos se propagam com a ajuda do canal de eventos, tanto os produtores quanto os consumidores devem se registrar no canal (para que os eventos possam ser enviados). Assim, o produtor envia o evento para o canal, que por sua vez, distribui o evento para os consumidores. Desse modo, os produtores geram eventos sem conhecer a identidade dos consumidores. Reciprocamente, consumidores recebem eventos sem conhecer a identidade dos produtores.

O produtor e o consumidor usam uma interface IDL padrão com métodos que permitem propagar eventos do produtor para o consumidor. Existem dois modelos de comunicação entre produtores e consumidores, conhecidos como *push* e *pull*.

As figuras 3.2 e 3.3 ilustram os modelos *pull* e *push*, respectivamente.

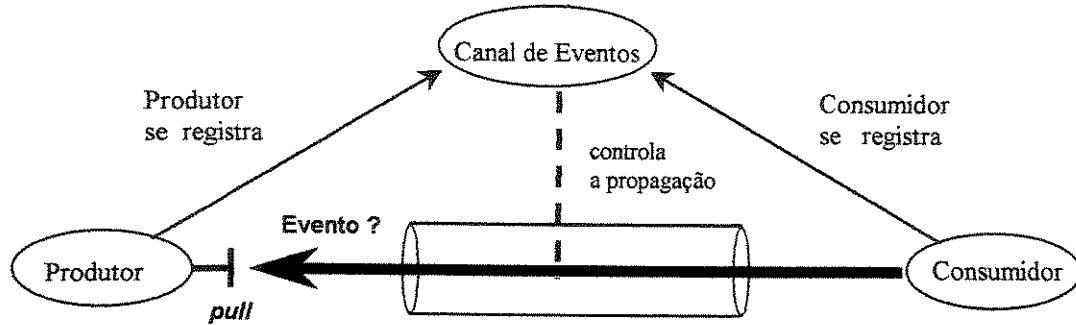


Figura 3.2 - Modelo *pull* - consumidor solicita eventos através da interface *pull* definida no produtor

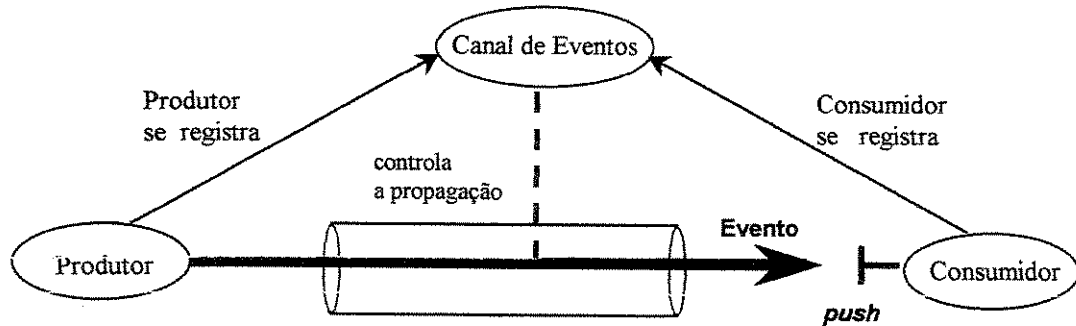


Figura 3.3 - Modelo *push* - produtor envia eventos na interface *push* definida pelo consumidor

Tanto o modelo *push* quanto o modelo *pull* permitem vários produtores enviarem eventos para um consumidor (figura 3.4), ou um produtor enviar eventos para vários consumidores (figura 3.5), ou ainda vários produtores enviarem eventos para vários consumidores (figura 3.6). A quantidade de produtores e consumidores depende da necessidade da aplicação.

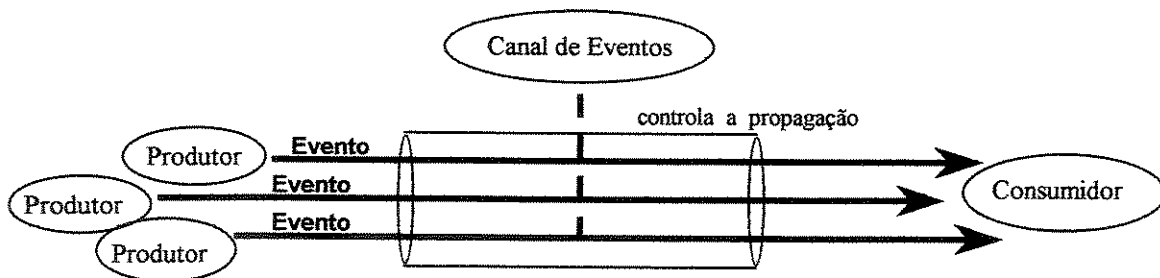


Figura 3.4 – Modelo *push* com mais de um produtor

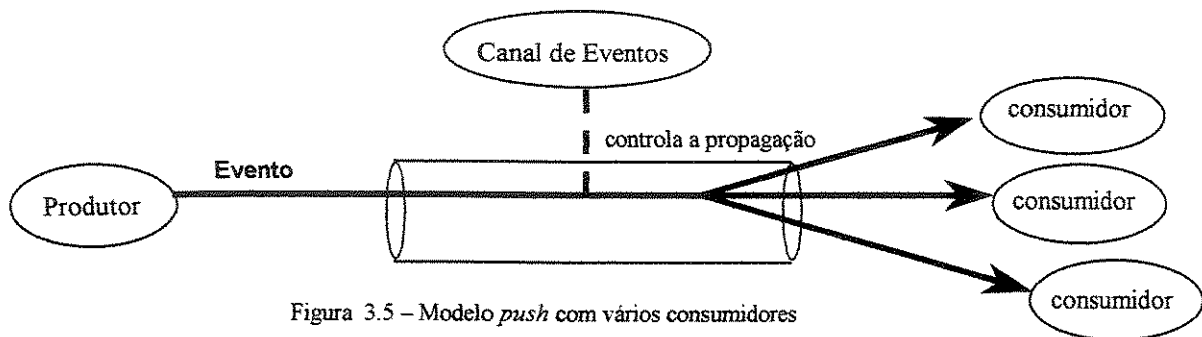


Figura 3.5 – Modelo *push* com vários consumidores

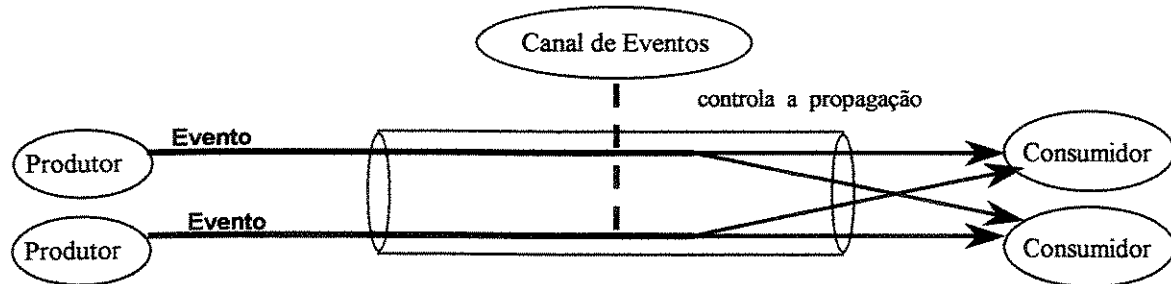


Figura 3.6 – Modelo *Push* com mais de um produtor e consumidor

No modelo *push*, o produtor envia o evento para o consumidor passando-o como parâmetro em uma operação *push* definida pelos consumidores (figura 3.2). No modelo *pull*, o consumidor solicita o evento invocando uma operação *pull* ou *try-pull* definidas pelos produtores (figura 3.3). Na operação *pull* o consumidor fica bloqueado até que o evento esteja disponível ou até que uma exceção ocorra (por exemplo, o produtor se desconectou). O mesmo não acontece com a operação *try-pull*. Nesta operação, se o evento estiver disponível o produtor irá retornar o evento para o consumidor, caso contrário o evento será retornado com um valor indefinido.

A forma de comunicação entre produtor e consumidor é assíncrona, ou seja, o que é enviado pelo produtor pode chegar até o canal “na mesma hora”, porém pode não ser “consumido” pelo consumidor de imediato. Quando o consumidor não “consome” o evento recebido de imediato, o canal mantém o evento por um tempo determinado e posteriormente o passa para o consumidor.

Tanto no modelo *pull* quanto no modelo *push*, o canal de eventos define operações administrativas para adicionar/remover consumidores, para adicionar/remover produtores e para destruir o canal. A operação para adicionar um consumidor ao canal retorna um *proxy* produtor (uma referência do produtor) e a operação para adicionar um produtor no

canal retorna um *proxy* consumidor (uma referência do consumidor). O canal de eventos conecta consumidores ao *proxy* produtor, e produtores ao *proxy* consumidor.

3.2.3 O Serviço A/V Streams

Existem atualmente vários produtos para comunicação multimídia na *Internet*, como, por exemplo, NetShow, RealAudio e RealVideo. Porém, esses *frameworks* utilizam mecanismos proprietários para o estabelecimento e controle de fluxo de mídia (*streams*). Como alternativa não proprietária, existe o Serviço *Control and Management of Audio/Video Streams (A/V Streams)* [20,21], uma proposta baseada em padrões abertos para o estabelecimento e controle de *streams* multimídia. Este serviço é baseado no padrão CORBA e consiste de um conjunto de interfaces para estabelecimento e controle de fluxos de mídia contínua.

Streams são os elementos básicos de suporte à comunicação multimídia e representam um conjunto de fluxos contínuos de áudio ou vídeo transmitidos em uma única direção, do produtor de fluxo de mídia para o consumidor de fluxo de mídia. No caso de áudio, um produtor de fluxo de mídia é um dispositivo que pode ser representado por um microfone, enquanto um consumidor de fluxo de mídia, pode ser representado por um alto-falante. Resumidamente um produtor (microfone) gera um fluxo de mídia para ser recebido por um consumidor (alto-falante).

O Serviço A/V Streams trata de configurações de fluxo ponto-a-ponto e ponto-multiponto, e possui um modelo arquitetural que proporciona as definições dos componentes que constituem um *stream* e as interfaces IDL que disponibilizam as operações de controle. Estas operações são realizadas através do ORB, mas os fluxos de áudio e vídeo em si são transportados por fora do ORB, por meio de um protocolo específico. Isso ocorre devido ao fato de que as implementações CORBA utilizam protocolos confiáveis como TCP/IIOP, enquanto o transporte de fluxos de áudio e vídeo utiliza protocolos sem conexão como RTP/UDP.

O Serviço A/V Streams possui dois perfis para o serviço de fluxo: o *full profile* (completo) e o *light profile* (simplificado). Os componentes do serviço *full profile* e suas respectivas interfaces IDL [22] são:

- Dispositivo Multimídia (*Multimedia Device*)- interface *MMDDevice*;
- Dispositivo Virtual Multimídia (*Virtual Multimedia Device*)- interface *VDev*;
- Controlador de *Stream* (*Stream Control*)- interface *StreamCtrl*;
- Terminal de *Stream* (*Stream End Point*) - interface *StreamEndPoint* (SEP);
- Terminal de Fluxo (*Flow End Point*) - interface *FlowEndPoint* (FEP);
- Conexão de Fluxo (*Flow Connection*) - interface *FlowConnection*.

A figura 3.7 ilustra o modelo *full profile*.

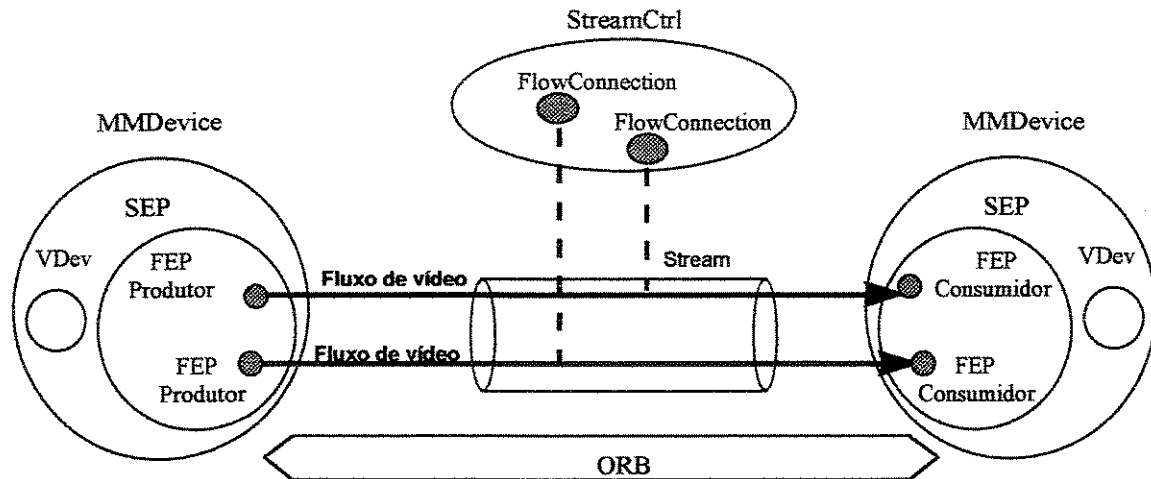


Figura 3.7 - Os componentes do Serviço A/V Streams com suas interfaces

Os Dispositivos Multimídia são conectados por *streams* para estabelecer um fluxo de áudio e/ou vídeo entre produtor e consumidor de mídia. Nos Dispositivos Multimídia existem dois componentes: o Dispositivo Virtual Multimídia e o Terminal de *Stream* que são criados para cada pedido de conexão. O Dispositivo Virtual Multimídia fornece operações relacionadas à configuração do dispositivo como por exemplo, tipo de mídia e taxas de transferência, enquanto o Terminal de *Stream* fornece operações relacionadas ao transporte de rede, como, por exemplo, qualidade de serviço. Este último componente pode conter e controlar vários Terminais de Fluxo que representam o produtor ou o consumidor de fluxo de mídia.

Os fluxos são gerenciados por um componente denominado Controlador de *Stream*. O Controlador de *Stream* fornece operações para o estabelecimento do fluxo entre Dispositivos Multimídia, bem como operações para o controle desses fluxos, como por exemplo, iniciar ou parar o fluxo.

No estabelecimento do fluxo, o Controlador de *Stream* cria uma Conexão de Fluxo para conectar os Terminais de Fluxo produtor e consumidor, verificar a conexão e controlar o fluxo entre os Terminais de Fluxo.

Conforme pode ser observado na figura 3.8, também é possível encontrar no modelo *full profile* apenas os componentes Conexão de Fluxo e Terminais de Fluxo produtor e consumidor.

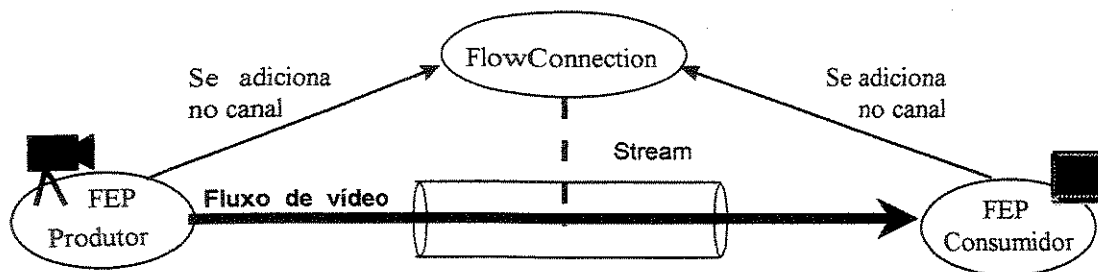


Figura 3.8 – Transmissão de vídeo utilizando apenas as interfaces:
Conexão de fluxo e Terminais de fluxo

Tanto na figura 3.7, quanto na figura 3.8 pode-se observar que a Conexão de Fluxo (*FlowConnection*) controla um único fluxo. Como a Conexão de Fluxo controla apenas um fluxo, caso um produtor A queira transmitir áudio e vídeo para um consumidor B, duas conexões de fluxo serão necessárias, uma para áudio e outra para vídeo, conforme pode ser observado na figura 3.9. Do mesmo modo, se A (produtor) transmitir vídeo para B (consumidor) e B, agora como produtor, também transmitir vídeo para A (consumidor), duas conexões de fluxo serão usadas, uma para o sentido de A - B e outra para o sentido de B - A. Outra condição possível, como pode ser observado na figura 3.10, é ter-se um produtor, por exemplo uma câmera, transmitindo vídeo para várias telas de vídeo (consumidores), em um canal *multicast* (ponto-multiponto), controlado por uma única conexão de fluxo.

Diferentemente do modelo *full profile*, o modelo *light profile* não tem acesso às interfaces do FEP produtor e consumidor e nem à interface do *FlowConnection*, que estão representadas na figura 3.7 em tom cinza.

Os componentes Conexão de Fluxo e Terminal de Fluxo ficam encapsulados no Controlador de *Stream* e no Dispositivo Multimídia respectivamente. Basicamente, os dispositivos de multimídia criam os FEPs para uso, mas mantêm as interfaces desses FEPs escondida. Analogamente, o Controlador de *Stream* cria a Conexão de Fluxo sem que sua interface fique visível.

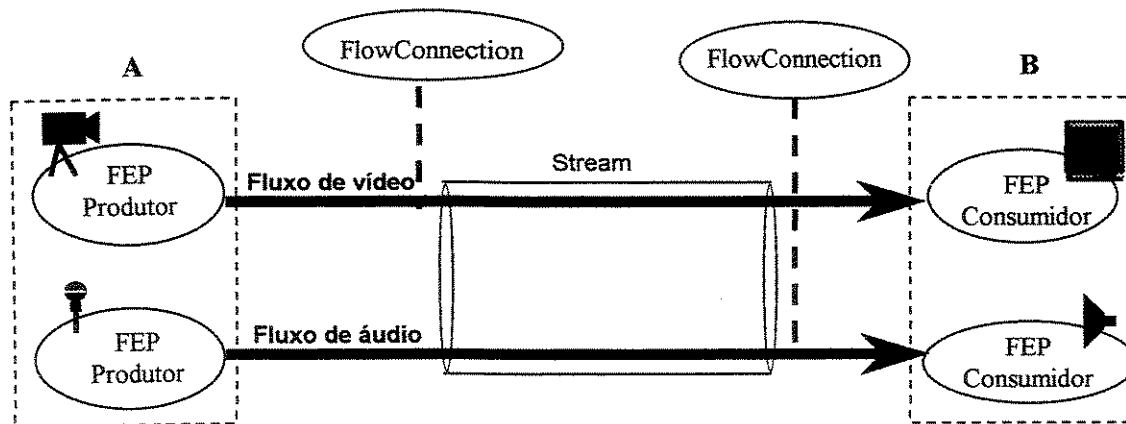


Figura 3.9 - Duas conexões de fluxo de A para B, uma para o áudio e outra para o vídeo.

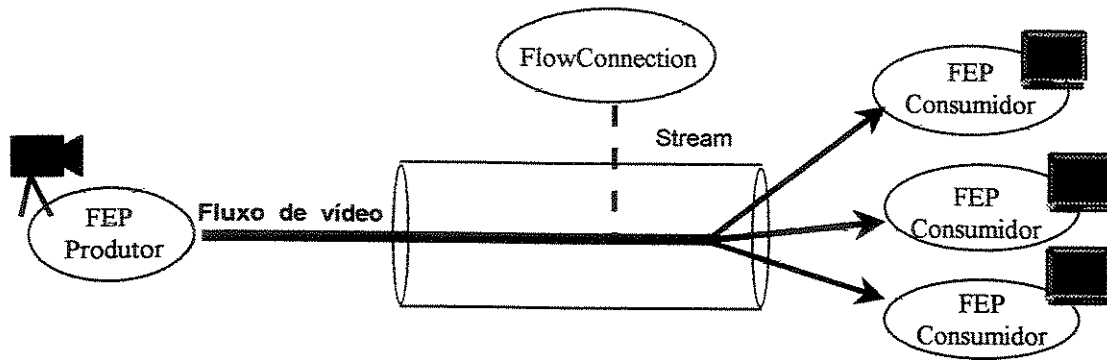


Figura 3.10 – Transmissão em vídeo em um canal *multicast*

A diferença entre os *profiles* está no modo com que cada um pode manusear o *stream*. O *full profile* possui recursos que facilitam o controle individual de fluxos no *stream*. Já o *light profile* possui recursos que facilitam o tratamento do *stream* como um todo.

3.3 Java

Quanto a tecnologia de linguagem de programação, Java [23] é uma das linguagens que melhor se adapta à arquitetura CORBA. De fato, esta tecnologia se destaca por ser apropriada para a implementação de aplicações distribuídas. Por ser portátil e independente de plataforma, permite o uso de estações de trabalho e microcomputadores com qualquer sistema operacional sem a necessidade de adaptação. Além desses aspectos, a linguagem Java é flexível quanto ao modo de execução, pois pode ser executada como um processo, como um *applet*⁵ em um *browser*, ou ainda como um *servlet*⁶ em um servidor. Entre outras vantagens relevantes que a linguagem apresenta pode-se citar a facilidade para o desenvolvimento de interfaces gráficas, o uso de *threads*⁷, o *garbage collection*⁸ e a grande variedade de classes pré-definidas disponíveis. As classes pré-definidas são agrupadas em pacotes que são referidos como biblioteca de classes ou como Interface de Programação de Aplicações (API) [25]. Tanto a interface gráfica quanto a utilização do áudio e vídeo consistem de uma API. A API Java que

⁵ Um *applet* é um programa executado no *browser* do cliente que por segurança possui algumas restrições como ler ou escrever arquivos nos computadores clientes.

⁶ “*Servlets* são para os servidores o que os *applets* são para os *browsers*, mas diferentemente do *applet*, os *servlets* não possuem interface gráfica” [24]. Eles são executados em servidores Web.

⁷ *Thread* é a porção de um código que pode ser executado concorrentemente com outras *threads*.

⁸ Libera o programador de gerenciar a memória.

permite a construção de interfaces gráficas é a API *Java Swing* (mais moderna) e AWT, e a que permite que programas Java incorporem áudio e vídeo é a API *Java Media Framework* (JMF) [26]. JMF exerce um papel fundamental no desenvolvimento de aplicações que necessitam da transmissão de mídia contínua em tempo real.

3.3.1 JMF

Java Media Framework (JMF) é uma API usada para a apresentação, captura e transmissão de mídia em tempo real através da rede. Esta API suporta vários padrões de tipos de conteúdos de mídia tais como MIDI, MPEG, QuickTime, RMF, WAV e AU.

Para poder transportar os fluxos de mídia, JMF usa o protocolo RTP (*Real Time Protocol*) sobre o protocolo UDP (*User Datagram Protocol*) que é um protocolo padrão na *Internet* para o transporte sem conexão. Este protocolo pode ser usado em redes que suportam *unicast* e *multicast* (como vídeo conferências).

Além das facilidades para o manuseio de mídia, JMF permite a criação de uma implementação 100% Java utilizando o padrão *A/V Streams*.

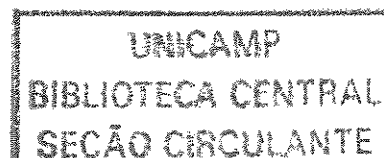
3.3.2 A Comunicação Remota em Java

Java com CORBA

Com relação ao uso do padrão CORBA, a linguagem Java provê o emprego desta arquitetura utilizando *Java Interface Definition Language* (Java IDL) [27]. Esta tecnologia fornece um ORB para suportar a interação entre objetos distribuídos localizados em plataformas diferentes. Desse modo, é possível ter-se a comunicação entre aplicações Java IDL e outras aplicações CORBA mesmo que estas tenham sido escritas em outras linguagens de programação.

Java IDL emprega o modelo cliente/servidor. O servidor fornece um método associado a uma interface e o cliente acessa esta interface através de uma referência de objeto registrada em um Serviço de Nomes CORBA. Para isso, o servidor deve primeiramente registrar o objeto com um nome simbólico no Serviço de Nomes.

A referência de objeto no lado cliente é um *stub* usado pelo ORB para invocar um método no objeto remoto. No lado servidor, o ORB usa *skeletons* para traduzir a invocação e chamar o método no servidor, além de retornar o resultado para o cliente.



Stubs e *Skeletons* são gerados pelo compilador IDL. Se o cliente estiver utilizando um ORB diferente do servidor, o protocolo IIOP irá garantir a interoperabilidade entre ORBs.

Java com RMI

Outra tecnologia similar a Java IDL, RMI (*Remote Method Invocation*) [28], também se encaixa no paradigma cliente/servidor no qual clientes utilizam serviços disponibilizados em servidores. Porém, esta tecnologia provê a comunicação remota entre programas escritos integralmente na linguagem de programação Java. Portanto, RMI assume um ambiente homogêneo Java.

RMI consiste de interfaces e classes. As interfaces definem métodos e as classes implementam os métodos definidos nas interfaces. A interface especifica os métodos que podem ser invocados remotamente pelo cliente, mas não contém o código executável, cuja a implementação fica no servidor.

RMI, assim como Java IDL, possui as classes *stub* e *skeleton*. A classe *stub* atua como *proxy* e serve para interceptar a chamada remota do cliente, destinada a objetos remotos. O *stub* redireciona a chamada para a referência remota, que por sua vez, interpreta e gerencia as referências do cliente passando a informação para o *skeleton*. A classe *skeleton* extrai os parâmetros da chamada do método, faz a chamada ao objeto de implementação do serviço remoto, e retorna o valor para o cliente.

A conexão entre máquinas virtuais Java é feita com um protocolo proprietário JRMP (*Java Remote Method Protocol*) que utiliza o protocolo TCP/IP. Na versão 1.3. Java 2 SDK ⁹ este protocolo foi substituído pelo protocolo IIOP do padrão CORBA. Isto permite aplicações distribuídas que utilizam RMI interagirem com aplicações que utilizam CORBA.

Para acessar objetos remotos, os clientes utilizam um Serviço de Nomes, chamado de registro de RMI (*rmiregistry*). Sua utilização é simples: depois de criar o objeto que implementa o serviço, o servidor registra o objeto utilizando o *rmiregistry* sob um nome público. Posteriormente, RMI cria um serviço de “escuta” para esperar o cliente se conectar e requerer o serviço. O cliente por sua vez, procura o objeto remoto pelo nome registrado no *rmiregistry*, para então, obtida sua referência, invocar um método neste objeto.

⁹ <http://java.sun.com/j2se/1.3/docs/relnotes/features.html#iiop>

3.4 WWW

Uma tecnologia que pode ser usada como um ambiente de execução bastante atrativo para a linguagem Java é a WWW (*World Wide Web*) [22], por permitir o uso da implementação de aplicações cliente / servidor em uma infra-estrutura de objetos distribuídos no ambiente do *browser*. Além desse aspecto, a tecnologia WWW possui uma interface amigável e de fácil manuseio para o usuário. Possui também uma grande independência de plataforma de *hardware* e de *software*, considerando o número diversificado de arquiteturas e sistemas operacionais disponíveis para os *browsers*. Mas a principal característica da WWW é que ela cobre boa parte do globo por utilizar a *Internet* para o transporte de informações.

Quanto ao uso dessa tecnologia não se pode garantir a qualidade de uma transmissão de áudio e vídeo, pois a WWW utiliza a *Internet*, que por sua vez oferece um serviço de melhor esforço.

3.5 Descrição de um Cenário

As tecnologias para o sistema de videoconferência (integrado em um sistema de educação a distância), podem ser empregadas em um cenário como na figura 3.11. Este cenário representa as conexões em rede existentes em uma universidade e fora dela, sendo que as diferentes formas de conexões, conforme pode-se observar na ilustração, são separadas por elipses (regiões), indicando o limite de alcance das redes. A elipse central, referida como “A” na figura, contém a velocidade maior e a elipse exterior a ela, região “B”, possui uma velocidade menor, assim por diante (a velocidade diminui a medida que se afasta do centro). Esta representação serve para mostrar que várias localidades não possuem o mesmo padrão de tecnologia.

A região “A”, representa um campus da universidade. Nela, existe uma rede de alta velocidade entre 155 e 1 Gbps, tipicamente com conexões em ATM (com LANE¹⁰ ou MPOA¹¹) ou *Gigabit Ethernet*. Este tipo de conexão é ideal para a transmissão de áudio e vídeo porque apresenta boa qualidade, dada a alta banda de transmissão.

Dentro da região A, existe o computador do professor que transmite a aula, assim como, platéias de estudantes organizadas em vários auditórios, e/ou estudantes com acessos individuais (*desktop*).

¹⁰ Lan Emulation

¹¹ Multiprotocol Over ATM

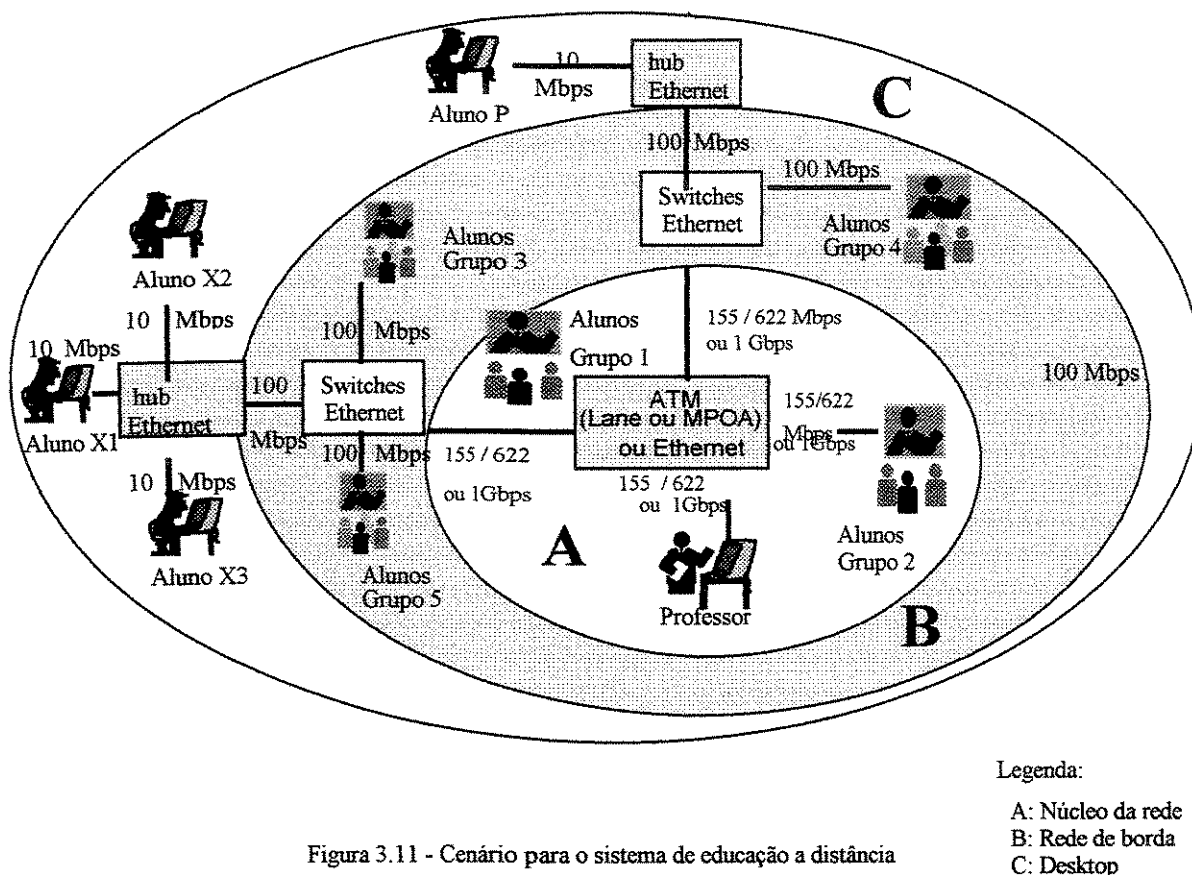


Figura 3.11 - Cenário para o sistema de educação a distância

Na região "B", estão as redes departamentais. Nesta região, o desempenho das redes diminui, uma vez que tipicamente se utiliza *Switches Ethernet* com velocidade de 10 ou 100 Mbps. Desse modo, com essa velocidade os outros departamentos da universidade também podem ter auditórios de alunos e/ou alunos com acessos individuais (*desktop*), recebendo áudio e vídeo do professor, mas com uma qualidade de transmissão inferior à parte central.

Saindo da região "B" para a rede da região "C", que representa redes interconectando *desktops*, usualmente utilizando *hubs Ethernet*, com velocidade compartilhada de 10 Mbps. Assim, neste ponto, a transmissão do vídeo para um auditório de alunos e/ou alunos individuais, é deficiente resultando numa qualidade de recepção mais baixa. Externamente à elipse C encontra-se a *Internet* pública.

3.6 Tecnologias & Educação a Distância

Para um sistema de videoconferência que utiliza objetos distribuídos e fluxos de áudio e vídeo contínuos em um sistema diversificado, tem-se o emprego de algumas das tecnologias descritas anteriormente.

CORBA é uma tecnologia que possui maturidade no tratamento de objetos distribuídos, provê transparência para o cliente durante as operações, fornece uma série de serviços, é voltado para ambientes heterogêneos, estando disponível em várias linguagens de programação. Uma das linguagens que emprega o uso da sua arquitetura é a linguagem Java. Java provê o emprego da arquitetura CORBA através do pacote Java IDL fornecendo um ORB para a interação de objetos distribuídos, possibilitando também a comunicação com outras aplicações CORBA escritas em outras linguagens de programação. Além de Java IDL, Java RMI provê a comunicação remota entre objetos distribuídos para programas escritos integralmente na linguagem Java, assumindo, desse modo, um ambiente homogêneo. Atualmente, no entanto, com o protocolo IIOP do padrão CORBA, as aplicações distribuídas podem também se comunicar com outras linguagens de mapeamento IDL. Na escolha entre Java IDL e Java RMI, prevalece Java IDL, pois o ORB nativo da linguagem Java permite que os componentes dos serviços CORBA interajam através desse ORB.

Com relação ao usuário, a WWW é de fácil instalação e operação e permite acesso global através da *Internet*. Por outro lado, a WWW opera sobre uma rede de melhor esforço, o que compromete o uso de fluxos de mídia contínua, principalmente do vídeo. Consequentemente, é interessante ter um sistema de videoconferência que não utilize a *Internet*, e sim uma rede com qualidade de transmissão, que ofereça garantia de reserva de recursos como por exemplo, uma rede ATM¹². Redes ATM permitem, além da reserva de recursos e da qualidade de serviço, alta velocidade para a transmissão de áudio e vídeo em tempo real. Desse modo, é possível desenvolver um sistema de videoconferência com as facilidades da linguagem Java, adicionadas à transparência de interoperabilidade de aplicações distribuídas em CORBA (Java IDL), que empregue os serviços CORBA e *frameworks* Java e que opere sobre uma rede de alto desempenho para transmissão de fluxos de mídia. Seguindo esta característica, desenvolveu-se um sistema de videoconferência, cuja modelagem é descrita no próximo capítulo.

¹² Apesar da rede ATM utilizada no trabalho suportar qualidade de serviço, a rede foi empregada no modelo "Emulação de LANs" onde qualidade de serviço não é suportada. Entretanto, com a velocidade da rede (155 Mbits/s) e o uso de portas dedicadas, qualidade de serviço foi alcançada através de super-provisão de recursos.

Capítulo 4

Modelagem do Sistema de Videoconferência para Educação a Distância

Neste capítulo utiliza-se diagramas para modelar o sistema de videoconferência, dentre eles, cartões CRC (Class Responsibility Collaboration), casos de uso, diagramas de sequência, diagramas de classe e diagramas de estado.

Os diagramas são baseados na notação UML (Unified Modeling Language), com exceção dos cartões CRC que não fazem parte desta notação [29].

4.1 Introdução

O diagrama de casos de uso é utilizado para indicar a interação do professor e do aluno com o sistema. O diagrama de classes modela as classes mais importantes do sistema apresentando o relacionamento entre as mesmas. Os diagramas de sequência abrangem as operações do sistema e permitem uma idéia mais objetiva do funcionamento do mesmo. Os cartões CRC apresentam as responsabilidades das classes complementando o diagrama de classes. Por último, os diagramas de estado descrevem o comportamento dos subsistemas de interação com o professor e o estudante, que são os elementos com dinâmica mais complexa.

4.2 Interação dos Atores do Sistema

A figura 4.1 apresenta um diagrama casos de uso com as atividades desempenhadas pelos atores professor e aluno. A atividade de condução da aula representa as atividades de início, finalização, concessão e retomada da palavra, e a atividade de intervenção da aula representa as atividades de solicitar, cancelar e devolver a palavra.

Caso de uso:	Conclusão da aula
Atores:	Professor (iniciador), Aluno
Tipo:	Primário
Descrição:	O professor inicia a aula habilitando os canais de áudio e vídeo. Quando o aluno solicita palavra cabe ao professor conceder e posteriormente retomar, caso o aluno não a devolva. Terminada a aula, o professor encerra as transmissões de áudio e vídeo.

Caso de uso:	Intervenção da aula
Atores:	Aluno (iniciador), Professor
Tipo:	Primário
Descrição:	O aluno pode solicitar a palavra. Uma vez concedida, ele faz sua intervenção. Ao término desta, deve devolvê-la. O aluno pode ainda cancelar o pedido da palavra.

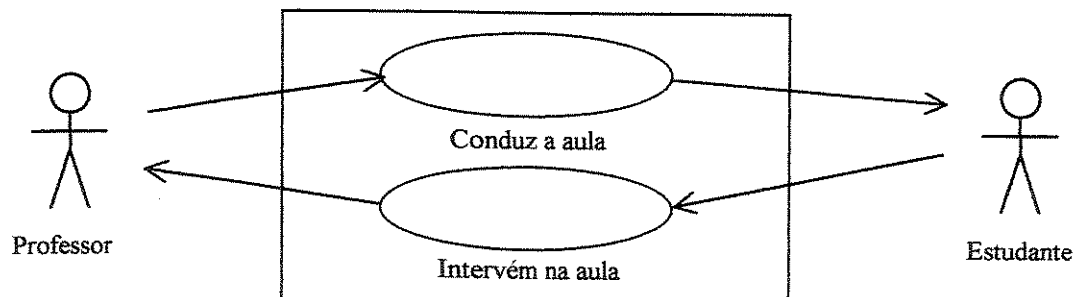


Figura 4.1 - Atores e principais atividades

4.3 Representação das Classes

Para simplificar a modelagem, apenas as principais classes foram representadas no diagrama de classes da figura 4.2.

Observando a figura 4.2, a classe EventEndPoint controla o fluxo de eventos do sistema colaborando com as classes Student e Teacher. Esta classe é utilizada para o envio de informações através de eventos entre professor e estudante.

A classe FlowEndPoint estabelece e controla o fluxo de mídia. Sua colaboração com as classes Student e Teacher se refere a obtenção de informações para a comunicação multimídia entre professor e estudante.

As classes Student e Teacher possuem operações¹³ disponibilizadas, respectivamente, ao estudante e ao professor.

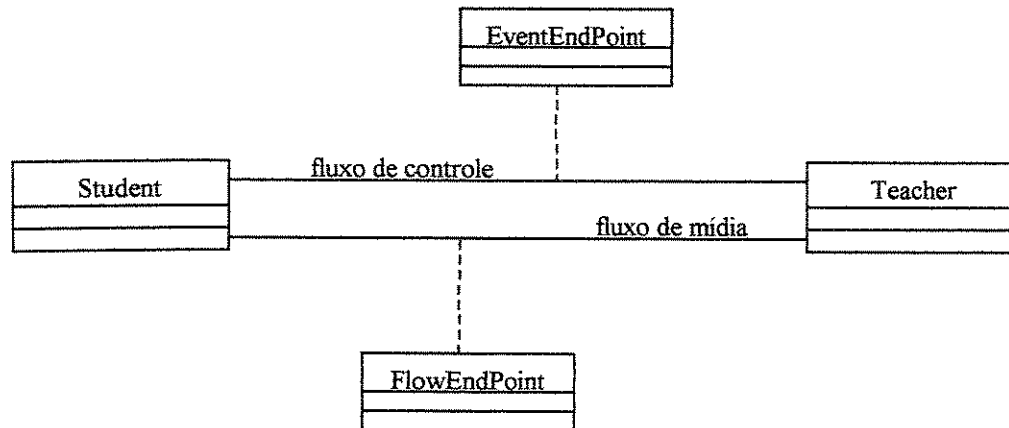


Figura 4.2 – Diagrama de classes apresentando uma “visão geral” do funcionamento do sistema

Na figura 4.3, a classe TeacherEventProducer assim como a classe StudentEventProducer produzem eventos, enquanto as classes TeacherEventConsumer e StudentEventConsumer recebem eventos. As classes PushSupplier e PushConsumer são classes base, respectivamente, para as classes produtoras e consumidoras de eventos. Por exemplo, o evento enviado do professor para o aluno é gerado através da classe TeacherEventProducer e recebido pela classe StudentEventConsumer.

A classe EventChannel controla a emissão ou recepção destes eventos e é utilizada como canal de eventos entre professor e aluno.

¹³ Uma operação, por exemplo, é o professor selecionar um dos alunos para conceder a palavra.

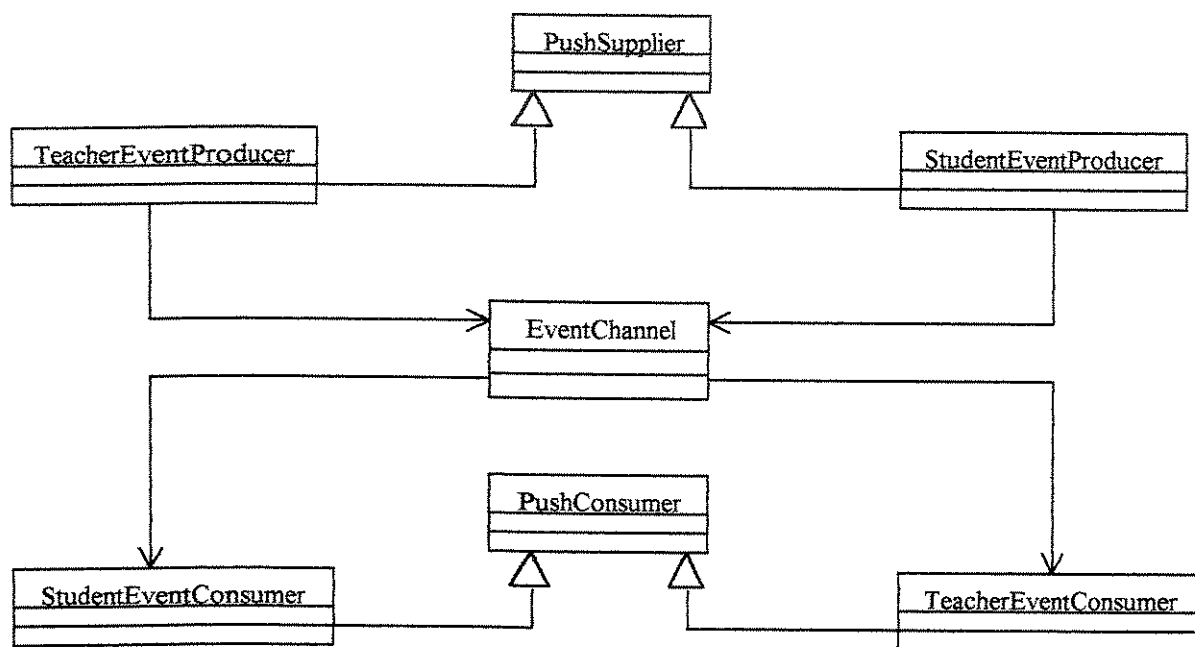


Figura 4.3 – Diagrama de classes da utilização de eventos no sistema

A figura 4.4 representa a transmissão da mídia no sistema. A transmissão da aula apresentada pelo professor é em áudio e vídeo. A transmissão do aluno com a palavra é em áudio e vídeo para o professor e somente em áudio para os demais alunos (conforme justificado no item 5.5.2).

As classes TeacherVideoProducer e StudentVideoProducer são classes produtoras de vídeo, enquanto as classes TeacherAudioProducer e StudentAudioProducer são produtoras de áudio. As classes TeacherVideoConsumer e StudentVideoConsumer são classes consumidoras de vídeo, enquanto as classes TeacherAudioConsumer e StudentAudioConsumer são consumidoras de áudio.

A transmissão de vídeo do professor é do tipo *multicast* e ocorre da classe TeacherVideoProducer para a classe StudentVideoConsumer, sendo a transmissão controlada pela classe MulticastVideoConnection. Analogamente, a transmissão de áudio do professor é do tipo *multicast* e ocorre da classe TeacherAudioProducer para a classe StudentAudioConsumer, sendo controlada pela classe MulticastAudioConnection.

A transmissão de áudio do aluno com a palavra é do tipo *multicast* e ocorre da classe StudentAudioProducer para as classes TeacherAudioConsumer e StudentAudioConsumer (que representa os demais alunos), sendo a transmissão controlada pela classe MulticastAudioConnection.

Já a transmissão de vídeo a partir do aluno é do tipo *unicast* da classe *StudentVideoProducer* para a classe *TeacherVideoConsumer*, sendo a transmissão controlada pela classe *UnicastVideoConnection*.

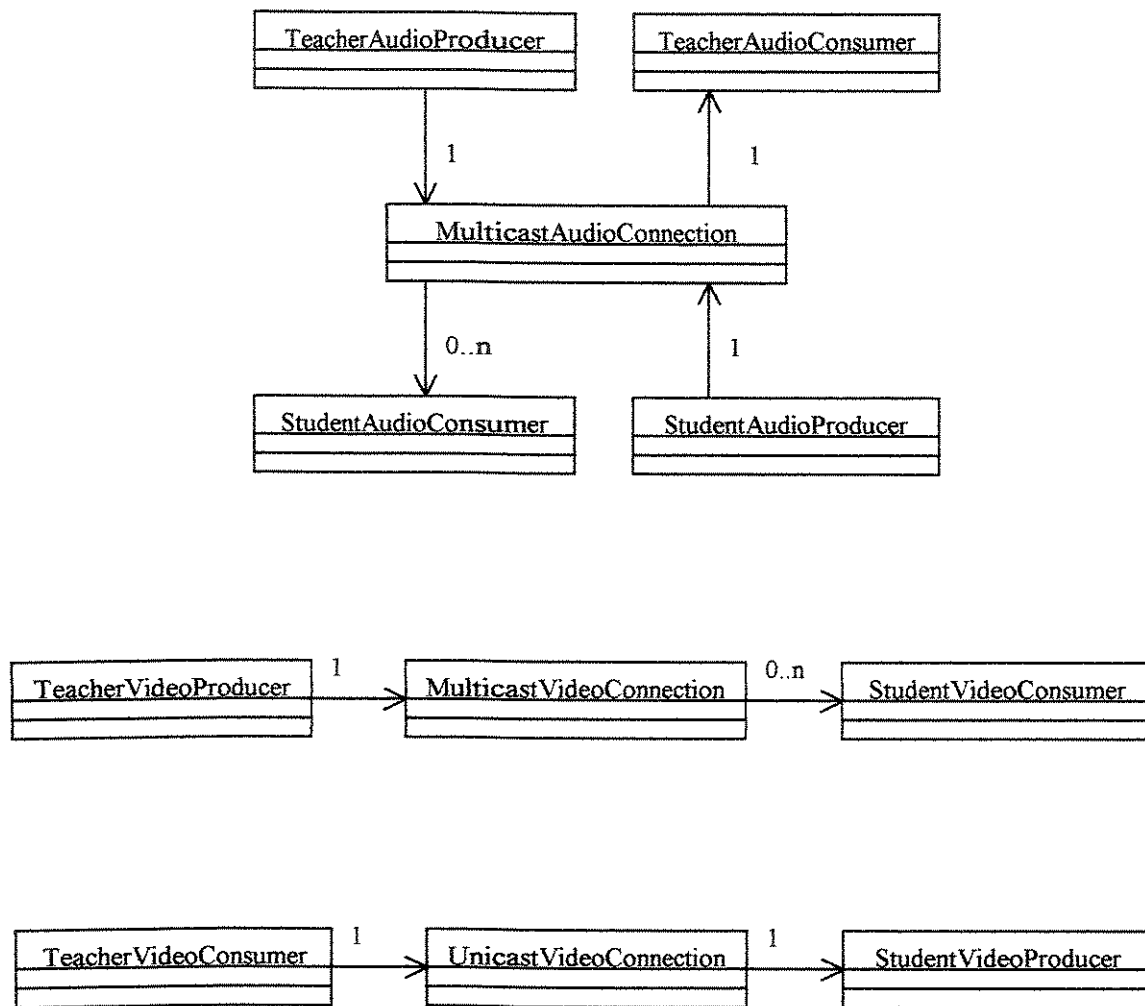


Figura 4.4 – Diagramas de classes das transmissões de mídia no sistema

A figura 4.5 representa um diagrama de pacotes UML para a aplicação.

O pacote Teacher contém a lógica da aplicação e interfaces no lado do professor e o pacote Student contém a lógica da aplicação e interfaces no lado do estudante.

O pacote CORBA A/V Streams possui classes que auxiliam na transmissão de áudio e vídeo (classes não representadas). O pacote EventService possui classes que auxiliam na utilização de eventos (classes não representadas). O pacote CORBA NameService possui o Serviço de Nomes no qual os objetos distribuídos são registrados (classes não representadas).

Os pacotes CORBA A/V Stream, CORBA NameService e CORBA EventService já estavam implementados. Foram desenvolvidos 2 pacotes Teacher (contém por exemplo, as classes TeacherAudioConsumer, MulticastAudioConnection, TeacherEventProducer) e Student (contém por exemplo, as classes StudentAudioProducer e studentEventProducer) para interagir com estes pacotes.

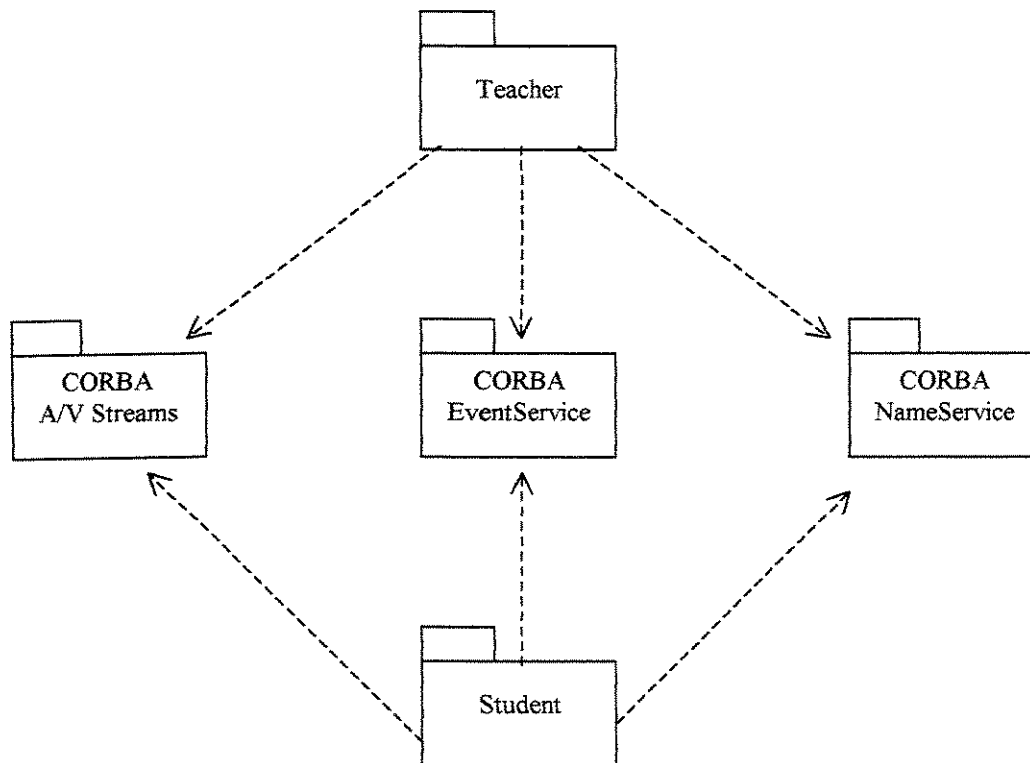


Figura 4.5 - Diagrama de pacotes do sistema

4.4 Descrição do Comportamento do Sistema

Os diagramas de seqüência foram utilizados para modelar a dinâmica do sistema de videoconferência para sistemas de educação a distância. O primeiro diagrama, figura 4.6, referente ao início da aula, demonstra o acesso dos participantes (professor e alunos).

O acesso ou a retirada do aluno gera um evento cujo parâmetro é o *login* do próprio aluno. A notificação do acesso é transmitida do aluno para o professor, a fim de manter o professor informado sobre a presença ou retirada do aluno.

O sistema é flexível quanto ao aspecto do acesso, ou seja, o professor pode ter acesso a sala de aula virtual primeiro que o aluno, ou vice-versa ¹⁴.

Na figura, o aluno representado como st-1, acessa o sistema e espera pela transmissão da aula do professor. O aluno referenciado como st-2, acessa o sistema posteriormente ao início da aula.

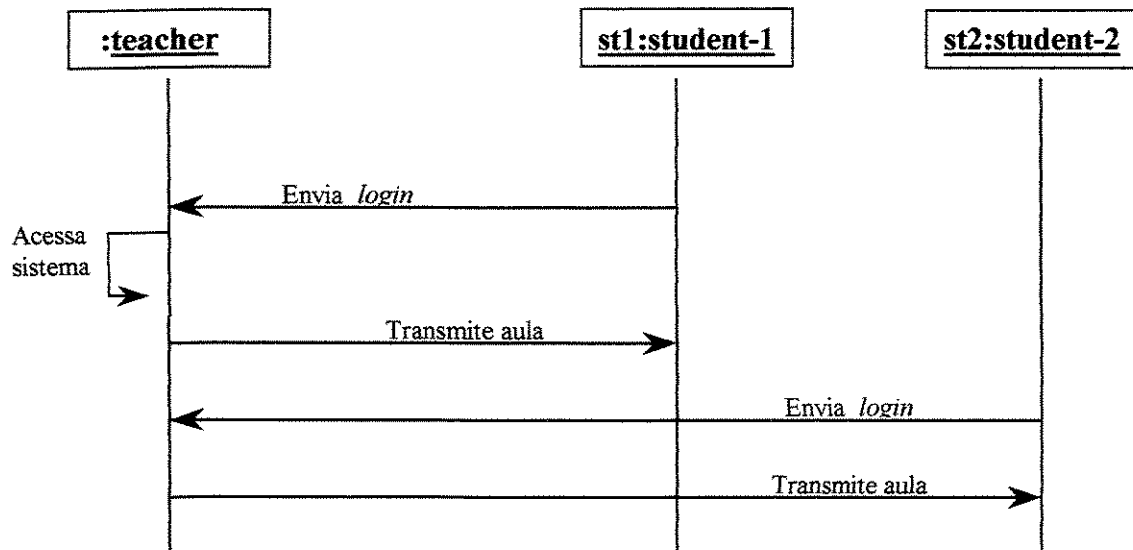


Figura 4.6 – Transmissão da aula do professor e acesso dos alunos no sistema.

¹⁴ A sala de aula virtual é por exemplo, semelhante a uma sala de *chat*, a qual o aluno pode entrar e esperar pelo professor.

O diagrama da figura 4.7 representa o aluno pedindo a palavra e posteriormente cancelando o pedido, antes do professor concedê-la.

Quando o aluno pede a palavra ou cancela o pedido, um evento é enviado para o professor.

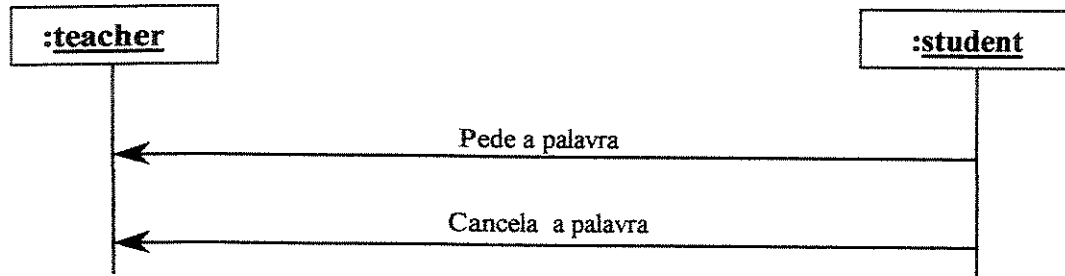


Figura 4.7 – Aluno cancela seu pedido de palavra

Na figura 4.8, o aluno pede a palavra e recebe a concessão do professor por meio de um evento.

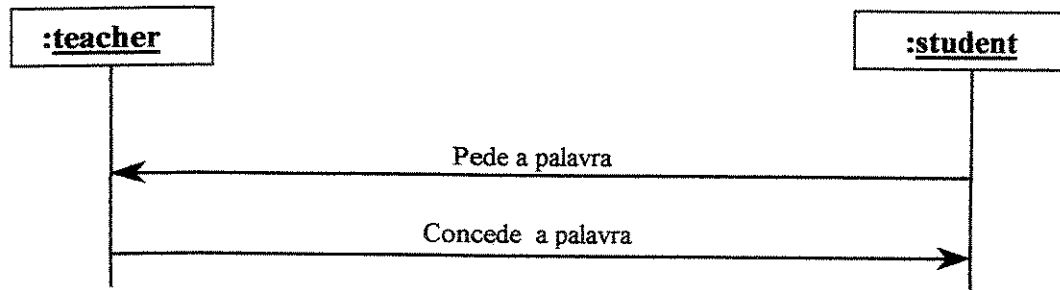


Figura 4.8 – Pedido da palavra do aluno é aceito

Uma vez concedida a palavra, o aluno pode transmitir sua dúvida ou comentário para o professor e demais alunos. A figura 4.9 ilustra este contexto.

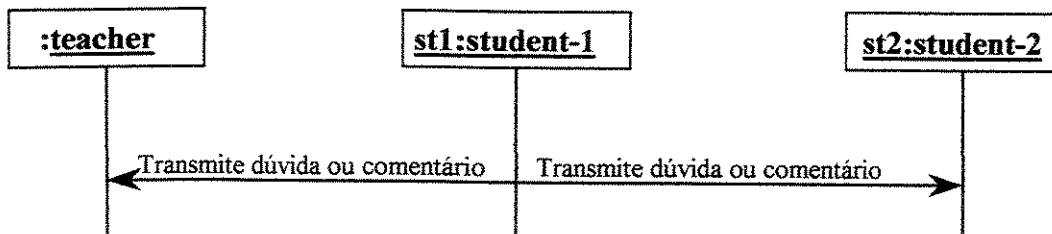


Figura 4.9 – Transmissão do aluno com a palavra

Depois que o aluno apresenta sua dúvida ou comentário, ele deve devolver a palavra para o professor como na figura 4.10, ou a palavra pode ser retomada pelo professor, como na figura 4.11.

Independentemente da opção escolhida, a transmissão do aluno cessa quando a palavra é retornada.

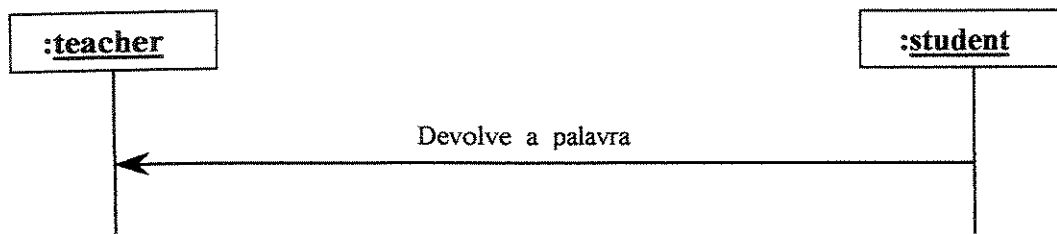


Figura 4.10 – Aluno devolve a palavra

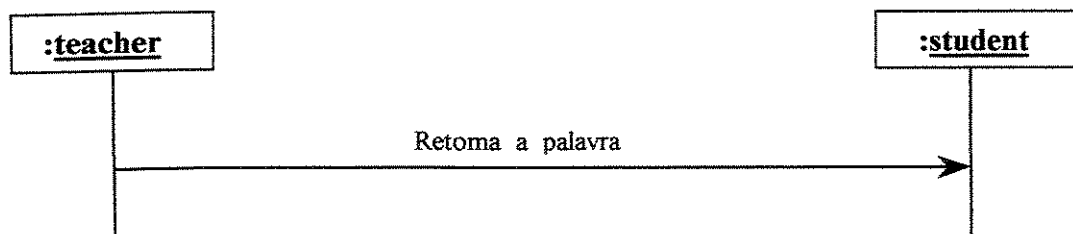


Figura 4.11 – Professor retoma a palavra

A figura 4.12 representa a saída do professor e dos alunos do sistema. O aluno representado como student-1, se retira antes do professor terminar a aula e o aluno representado como student-2, espera o professor encerrar a aula para depois se retirar.

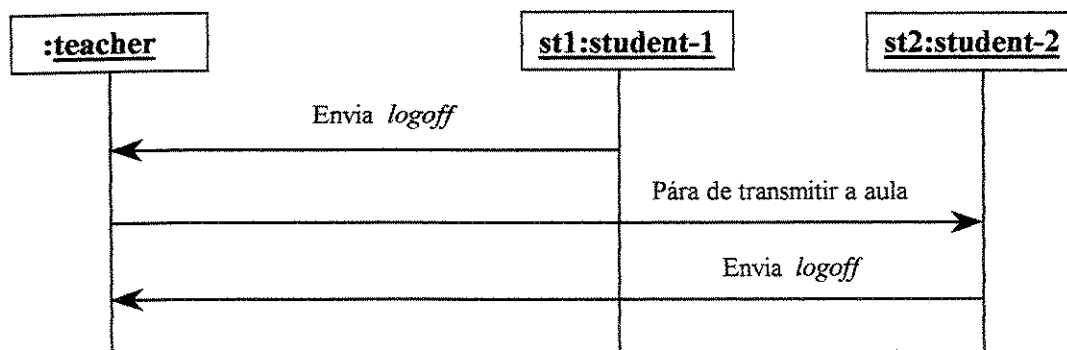


Figura 4.12 – Retirada dos alunos e término da aula

4.5 Responsabilidades das Classes

Para complementar as informações do diagrama de classes foram utilizados os cartões CRC, que apresentam as responsabilidades das classes e a colaboração com outras classes. Estes cartões estão representados nas figuras 4.13, 4.14, 4.15 e 4.16.

Teacher	
Notificar aluno sobre permissão da palavra ou devolução da mesma.	TeacherEventProducer
Receber notificação do aluno (acesso / saída do aluno, pedido / cancelamento da palavra, devolução da palavra)	TeacherEventConsumer
Transmitir o vídeo da aula para os alunos	TeacherVideoProducer
Receber transmissão do vídeo do aluno com a palavra	TeacherVideoConsumer
Transmitir o áudio da aula para os alunos	TeacherAudioProducer
Receber transmissão do áudio do aluno com a palavra	TeacherAudioConsumer

Figura 4.13– Cartão CRC da classe Teacher

Student	
Notificar professor sobre acesso / retirada do aluno, pedido / cancelamento da palavra, devolução da palavra	StudentEventProducer
Receber notificação do professor sobre pedido da palavra ou retomada da mesma	StudentEventConsumer
Transmitir dúvida ou comentário para professor em vídeo	StudentVideoProducer
Receber transmissão do professor em vídeo	StudentVideoConsumer
Transmitir dúvida ou comentário para professor e demais alunos em áudio	StudentAudioProducer
Receber transmissão do aluno com a palavra em áudio	StudentAudioConsumer

Figura 4.14 – Cartão CRC da classe Student

UnicastVideoConnection	
Estabelece ou destrói a conexão para transmissão em áudio e vídeo entre estudante e professor.	StudentVideoProducer, TeacherVideoConsumer

Figura 4.15 – Cartão CRC da classe UnicastVideoConnection

EventChannel	
Estabelece ou destrói o canal de eventos entre professor e estudante	StudentEventProducer, TeacherEventConsumer, StudentEventConsumer, TeacherEventProducer

Figura 4.16 – Cartão CRC da classe EventChannel

Observando as figuras 4.13, 4.14, 4.15 e 4.16 nota-se que as classes TeacherMediaProducer, TeacherMediaConsumer, StudentMediaProducer e StudentMediaConsumer tratam somente da transmissão e recepção da mídia e instanciam objetos que são produtor e consumidor de áudio e vídeo. Já as classes TeacherEventProducer, TeacherEventConsumer, StudentEventProducer e StudentEventConsumer permitem o envio e recepção de eventos e instanciam objetos produtor e consumidor de eventos.

4.6 Diagramas de Estado

Para entender melhor o comportamento das classes Student e Teacher, diagramas de estado foram utilizados.

As figuras 4.17 e 4.18 descrevem, respectivamente, os estados e suas transições para instâncias das classes Student e Teacher.

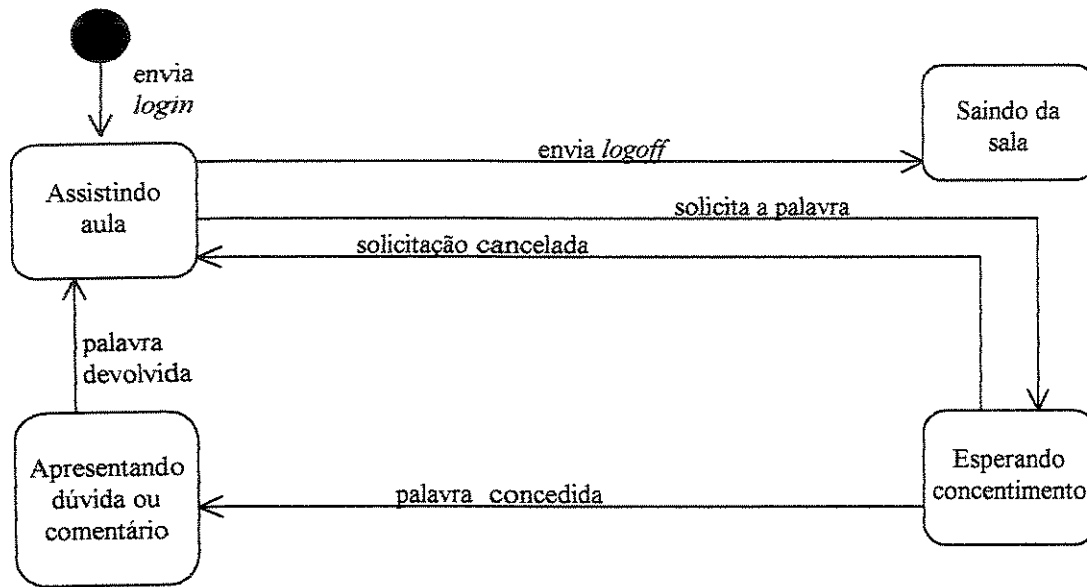


Figura 4.17 – Diagrama de estados da classe Student

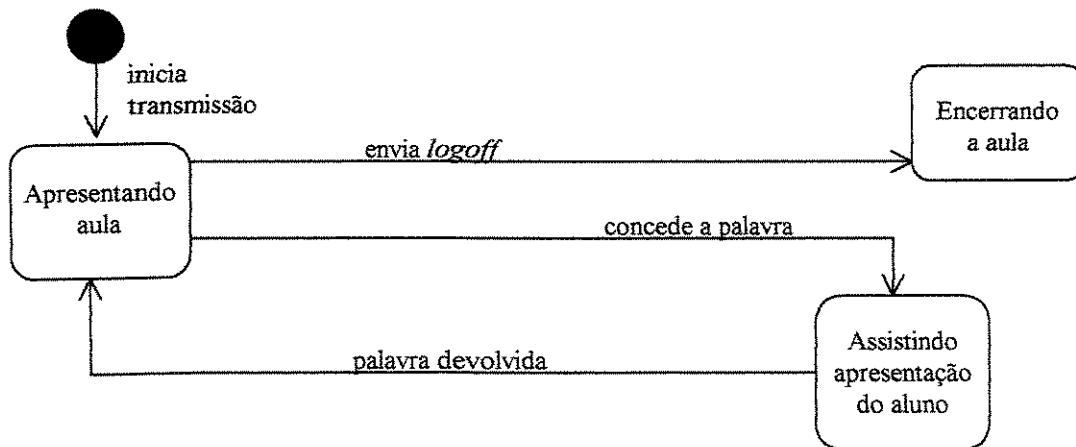


Figura 4.18 – Diagrama de estados da classe Teacher

Os diagramas desse capítulo foram utilizados para dar uma visão do funcionamento da aula e a interação entre professor e aluno. No próximo capítulo é descrito o projeto e implementação do sistema.

Capítulo 5

Projeto do Sistema de Videoconferência para Sistemas de Educação a Distância

Neste capítulo descreve-se o projeto do sistema de videoconferência AVE para sistemas de educação a distância, abordando o uso do *Full profile* da especificação A/V Streams e do Serviço de Eventos, ambos da arquitetura CORBA.

5.1 Introdução

Para um aluno receber as transmissões do professor em áudio e vídeo é necessário o uso de uma rede com qualidade de serviço que suporte a transmissão. Atualmente, poucos ambientes podem dispor dessa tecnologia como universidades e empresas. O sistema de videoconferência AVE foi elaborado para tais ambientes. No futuro espera-se que redes de longo alcance possam oferecer a qualidade necessária para transmissão de áudio e vídeo, tornando o sistema de uso mais disseminado.

5.2 Projeto da Arquitetura

5.2.1 Infra-estrutura

O sistema de videoconferência foi implementado na linguagem Java e é composto de um conjunto de objetos distribuídos sobre uma rede IP/ATM. Basicamente, o sistema AVE

consiste de transmissões em áudio e vídeo e de notificações por meio do Serviço de Eventos. Tanto as transmissões de mídia quanto os eventos ocorrem do professor para o estudante e do estudante com a palavra, para o professor.

O sistema é síncrono, tendo o horário da aula previamente estipulado pelo professor. O aluno acessa um *site* específico na *Internet* que contém um *link* para todos os cursos oferecidos. O aluno seleciona o curso desejado e obtém o dia e a hora da aula.

O sistema de videoconferência AVE é executado em uma rede de alto desempenho, para permitir alta taxa de transmissão de mídia durante a aula.

O aluno, assim como o professor, deve digitar um *login* e uma senha para acessar o sistema AVE. Após o acesso, o aluno espera pela transmissão de áudio e vídeo do professor (pois a aula só tem início quando o professor começa a transmissão).

O professor sabe quem está presente através de uma lista de *logins* de alunos em sua tela. O *login* é enviado para o professor, juntamente com a sua transmissão de vídeo. O vídeo é transmitido e momentaneamente “congelado” até que o professor conceda a palavra para o aluno. Quando a palavra é concedida, o vídeo do aluno é “descongelado”. Posteriormente, quando a palavra é retornada ao professor, o vídeo é novamente “congelado”.

Apenas o *login* dos alunos (sem áudio e vídeo) é o suficiente para o professor, por exemplo, organizar equipes de trabalho. Já a transmissão de áudio e vídeo é importante para o aluno expressar sua dúvida ou comentário.

O vídeo do aluno é “congelado” devido ao tempo dispensado na construção da janela de vídeo. Percebeu-se que o tempo da montagem da janela era extenso¹⁵. Assim sendo, criar a janela apenas uma única vez (congelar o vídeo) e utilizá-la quando necessário (ao descongelar) tornou-se a melhor opção. Esta demora não acontece no áudio em que a estrutura é criada após o aluno ser autorizado a falar. Ao devolver a palavra, o áudio é “congelado” e só será “descongelado” quando o aluno tiver a palavra novamente.

Como a quantidade de alunos presentes pode ser grande, assim como a quantidade de imagens “congeladas”, optou-se por utilizar dois computadores para o professor, ao invés de um único, sendo um para receber a transmissão da mídia dos alunos e outro equipado com câmera de vídeo e microfone para controlar¹⁶ a aula e transmitir áudio e vídeo.

O sistema é flexível quanto ao uso de um ou dois computadores pelo professor. Portanto se o professor preferir pode empregar um único computador¹⁷ para apresentar as imagens do alunos e ter o manuseio da aula.

A aula é ministrada para uma platéia de estudantes reunidos em uma sala, representados por um coordenador de sala, ou para estudantes com acessos individuais (através de

¹⁵ A construção da janela de vídeo depende muito do processamento do computador utilizado. No computador Aracati (ver item 5.6) por exemplo, a construção da janela demorou em torno de 30 segundos.

¹⁶ Por exemplo, conceder a palavra para um aluno.

¹⁷ É uma opção quando se tem poucos alunos, o que resulta em poucas imagens.

computadores *desktops*). No caso da platéia de estudantes, somente o *login* do coordenador de sala é enviado para a lista de presença da tela do professor.

Tanto as platéias das salas de estudantes quanto os alunos com acesso individual possuem um computador equipado com câmera de vídeo e microfone para se comunicarem com o professor. No caso das platéias de estudantes, o computador do coordenador de sala é equipado com câmera de vídeo e microfone e está ligado em um telão para exibir a transmissão provinda do professor.

A quantidade de alunos que participa do curso (tanto os com acesso individual via *desktop* como os agrupados em salas representadas por coordenadores de sala) depende da capacidade do computador do professor em lidar com esse número. O computador deve ser rápido o suficiente para suportar tanto as transmissões e recepções em mídia quanto o manuseio de eventos.

O computador do aluno deve igualmente ser rápido o suficiente para receber e enviar áudio e vídeo com qualidade de transmissão e lidar com os eventos (ver o exemplo de caso de teste item 5.6).

O cenário considerado pode ser observado na figura 5.1.

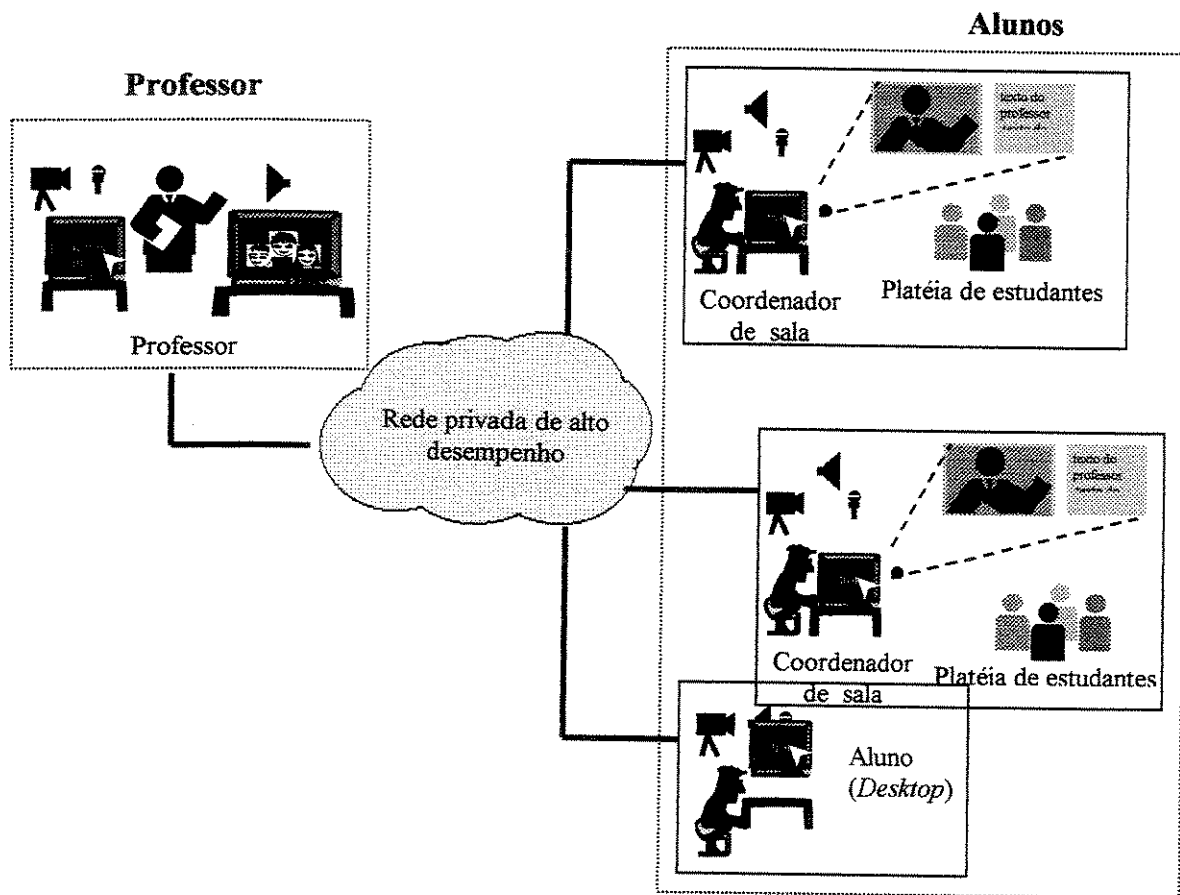


Figura 5.1 – Infra-estrutura utilizada na videoconferência para sistemas de educação a distância

A esquerda na figura 5.1, está o professor com o dois computadores, sendo um utilizado para transmissão da mídia e tela de controle da aula e o outro para receber áudio e vídeo dos alunos. A direita está representado um aluno com acesso via *desktop* e duas salas com platéias de alunos, sendo cada uma representada por um coordenador de sala. A rede utilizada é rede de alto desempenho para poder suportar as transmissões em áudio e vídeo.

O aluno solicita a palavra para o professor quando deseja tirar alguma dúvida ou fazer algum comentário. O pedido pode ser consentido ou não. Caso seja consentido, o aluno com a palavra transmite sua dúvida ou comentário em áudio e vídeo para o professor e somente em áudio para os demais alunos presentes na sala de aula virtual¹⁸. No caso da platéia de estudantes, o pedido é feito via coordenador de sala. Quando o pedido é aceito, o aluno recebe o microfone e, uma câmera na sala de estudantes o focaliza.

Juntamente à tela de manuseio da aula e às transmissões de áudio e vídeo seria possível ter um *whiteboard* para ajudar nas explicações do professor. Com o *whiteboard*, o desenho feito pelo professor é reproduzido no telão da platéia de estudantes (ou na tela de computador do aluno com acesso individual), completando, dessa forma a interação professor-aluno.

5.2.2 Comunicação entre os Módulos

O diagrama de distribuição UML apresentado na figura 5.2, ilustra as relações entre *software* e *hardware* do sistema de videoconferência. A figura representa o laboratório LCA com os computadores conectados em uma rede IP/ATM. A conexão física do laboratório (rede) é representada por uma linha inteira. Os computadores são representados por caixas. As caixas “Teacher Computer” representam os computadores do professor e a caixa “Student Computer” o computador do aluno. O que é executado em cada computador é representado por um conjunto de componentes.

Um dos computadores do professor utilizado para captura de mídia, possui os seguintes componentes: FlowDevice que é utilizado como fábrica de FEP produtor de mídia; EventChannel que representa o canal de eventos; Teacher que concentra operações de controle da aula. NameServer que aparece neste computador é o Serviço de Nomes e será discutido mais adiante.

O computador do professor utilizado para apresentação, possui o componente FlowDevice que é utilizado como fábrica de FEP consumidor de mídia.

Como trata-se de um sistema distribuído, os componentes podem ser executados em qualquer um dos dois computadores do professor (basta que o professor os inicie no computador desejado).

¹⁸ Maiores detalhes no item 5.5

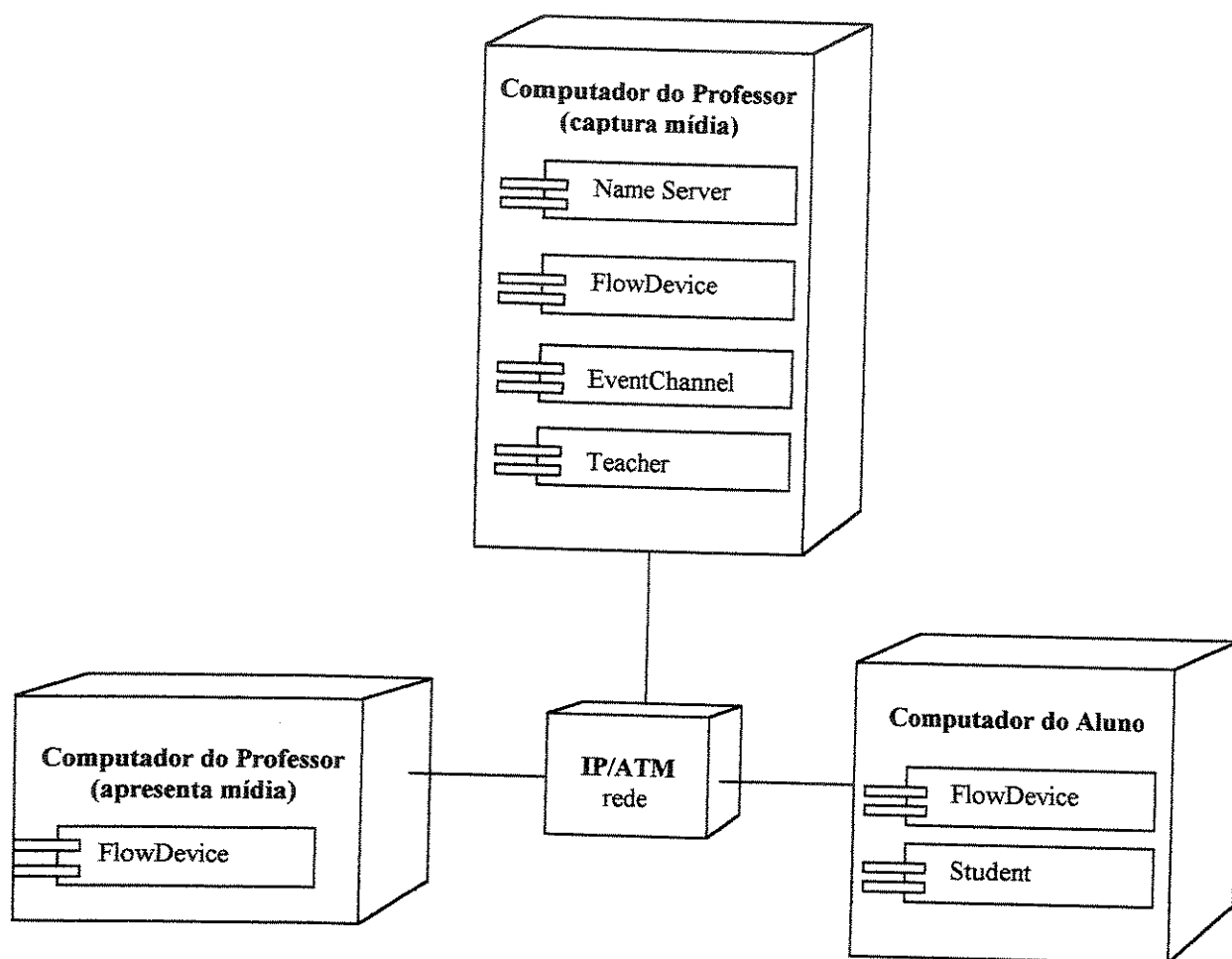


Figura 5.2 – Diagrama de distribuição UML para o sistema AVE

O computador do aluno possui o componente **FlowDevice**, que é utilizado como fábrica de FEP produtor e consumidor de mídia; e o componente **Student**, que concentra as operações de controle da aula, do lado do estudante.

Componentes **FlowDevice** geram FEPs produtores ou consumidores de mídia no computador onde eles se encontram. Assim sendo, no computador do professor (onde o FEP produtor de mídia é criado) é transmitido áudio e vídeo do professor para o computador do aluno (onde o FEP consumidor de mídia referente é criado). Da mesma forma, no computador do aluno é transmitido áudio e vídeo para o computador do professor (onde o consumidor de mídia referente é criado).

Os objetos distribuídos utilizados no sistema são registrados nos Serviço de Nomes. Este serviço pode ser executado em qualquer computador da rede. No sistema desenvolvido, o Serviço de Nomes se encontra no computador do professor.

O nome do computador onde está o Serviço de Nomes deve ser referenciado no computador do aluno (e pelo outro computador do professor) para que os objetos

distribuídos possam ser utilizados. O número da porta do Serviço de Nomes também deve ser informado. Dessa forma, os componentes que compõem o sistema do professor e do estudante devem referenciar a localização do Serviço de Nomes.

Todos os componentes são arquivos Java executados a partir da linha de comando. A localização do Serviço de Nomes (*host* e *port*) é fornecida como argumento para os componentes.

5.3 Projeto da Interface

O sistema é constituído de duas partes, sendo uma executada pelo estudante e a outra executada pelo do professor .

Conforme pode ser observado na figura 5.3, tanto o sistema que atua no computador do professor quanto o que atua no computador do estudante possuem uma janela de entrada com um *login* e uma senha que podem ser confirmados pelo botão “OK”, ou cancelados pelo botão “Cancel”.

O *login* do aluno pode ser o *login* de um coordenador de sala, ou de um aluno com acesso individual. Assim que o aluno, ou coordenador de sala, pressionar o botão “OK”, o *login* é enviado para o professor para mantê-lo informado sobre quem está presente na aula.

Utilizou-se o *login* do aluno ao invés do nome porque o nome pode se repetir e o *login* não (por ser único).

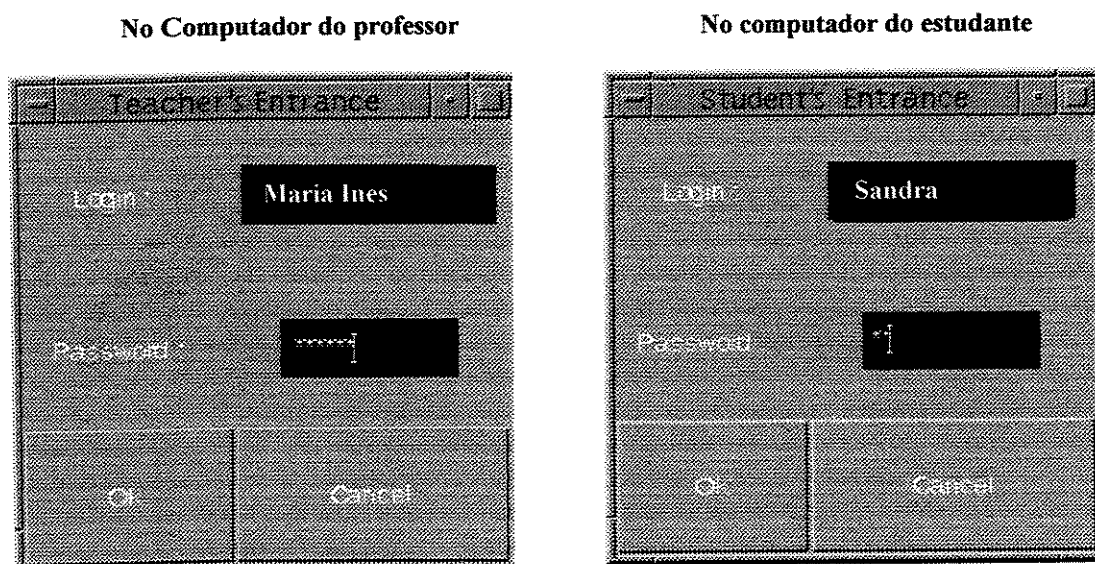


Figura 5.3 - Telas de acesso do professor e do aluno

5.3.1 Interface do Professor

O vídeo dos alunos pode ser exibido em um segundo computador do professor para não sobrecarregar um único computador. A apresentação do vídeo do aluno acontece logo após o acesso do mesmo ao sistema. Cada vídeo é apresentado ao professor e em seguida “congelado”. Quando o aluno se retira do sistema, o vídeo correspondente ao aluno é excluído da tela do professor.

A figura 5.4 apresenta a imagem do vídeo dos alunos que é exibida na tela do professor.

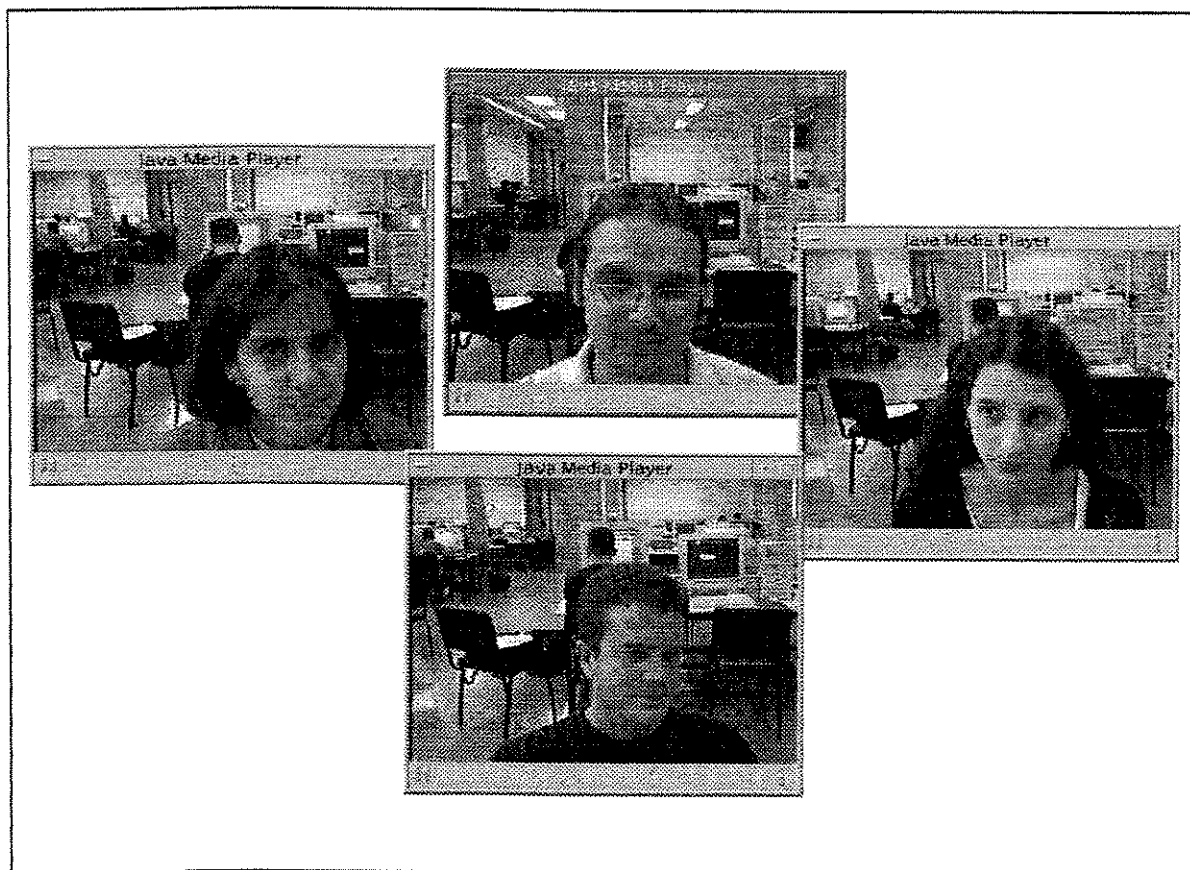


Figura 5.4 - Telas dos alunos presentes em aula, no computador do professor

Após a tela de acesso, conforme pode ser observado na figura 5.5, o professor tem uma tela de controle. Nesta tela existe o botão “Finish class”, o qual permite que o professor termine a aula e saia do sistema. Outro botão, “Begin class”, permite que o professor inicie a aula transmitindo seu áudio e vídeo para os alunos. Na mesma tela ainda existe uma lista de *logins* de alunos (ou coordenadores de sala) conhecida como “Present students” que se refere aos alunos presentes na sala de aula virtual. O *login* do aluno aparece nesta lista quando o aluno ingressa na aula e é excluído da mesma, quando o aluno se retira.

Outra lista intitulada “Who wants to speak” (uma lista de espera) apresenta os *logins* dos coordenadores de sala ou dos alunos com acesso individual que desejam pedir a palavra ao professor. Quando o aluno pressiona o botão “Request speech” em sua tela, um evento é gerado e o *login* desse aluno é inserido na lista.

As listas presentes na tela do professor estão classificadas conforme a ordem de chegada dos alunos.

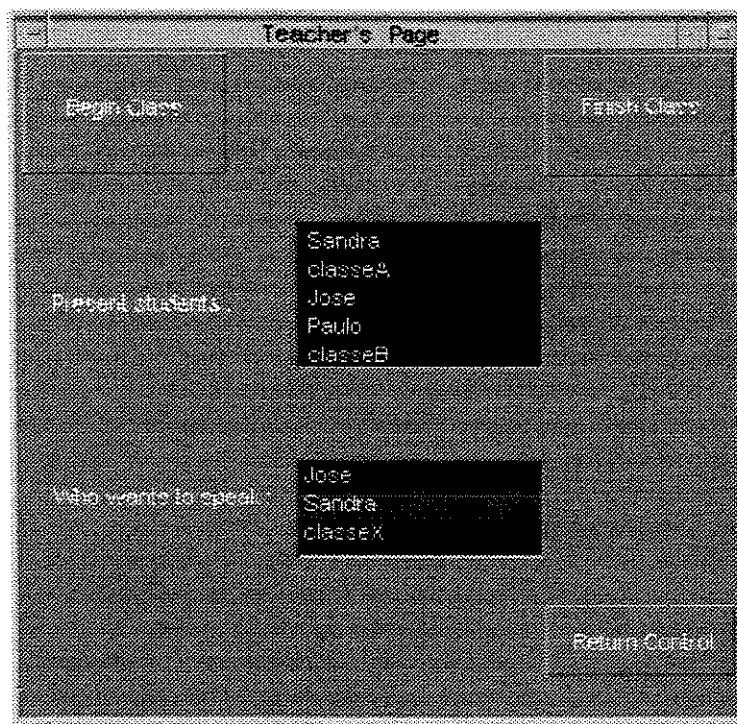


Figura 5.5 – Tela do professor de manuseio da aula

Na segunda lista, o professor pode escolher através do *mouse* um dos *logins* para conceder a palavra. Quando um nome é selecionado uma janela de confirmação é apresentada, como na figura 5.6.



Figura 5.6 - Janela de confirmação

Ao confirmar a seleção, o *login* do aluno selecionado é removido da lista “Who wants to speak” e a imagem do estudante com permissão para falar é “descongelada”. Consequentemente, as telas de controle do áudio, referentes a fala do aluno, irão aparecer na tela do professor e na tela dos demais alunos, como na figura 5.7.



Figura 5.7 – Tela de controle do áudio

Após o professor confirmar a autorização da palavra, a tela do aluno sofrerá uma alteração que será descrita no item 5.3.2.

Se o professor desejar, poderá retomar a palavra pressionando o botão “Return control”. Quando a palavra é devolvida, a imagem do aluno volta a se “congelar” e as telas de controle do áudio referentes ao aluno desaparecem da tela do professor.

O mesmo aluno pode pedir a palavra novamente. Caso isso ocorra, o *login* deste aluno irá para o final da lista “Who wants to speak”.

5.3.2 Interface do Aluno

Posteriormente a tela de acesso ao sistema o aluno possui a tela de intervenção na aula, juntamente às telas de áudio e vídeo referentes à transmissão do professor e suas respectivas telas de controle. A figura 5.8 ilustra o formato destas telas.

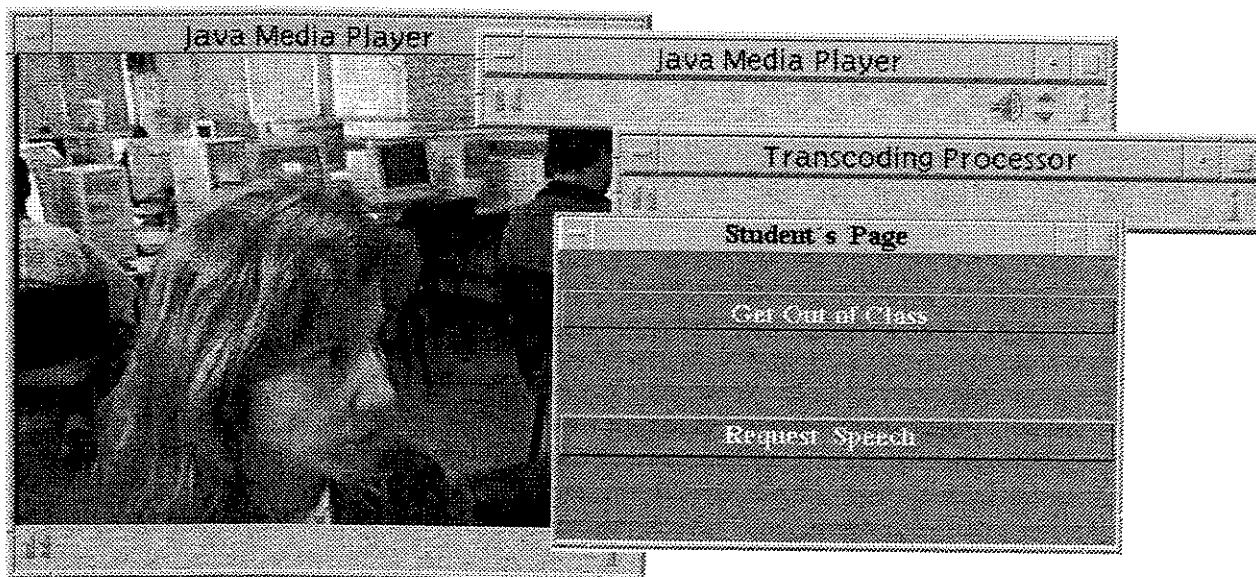


Figura 5.8 – Tela de manuseio da aula, com as telas de áudio, vídeo e controle referentes a transmissão do professor.

Como pode ser observado na figura 5.10, a tela de intervenção da aula tem na barra de títulos o nome “Student’s Page” e é constituída de dois botões : botão “Get Out of Class” e botão “Request Speech”. O botão “Get Out of Class” permite que o aluno se retire da sala de aula virtual. Quando isso ocorre, o *login* referente a esse aluno é removido simultâneamente da primeira e da segunda lista da tela do professor, mesmo que ele ainda não tenha tido a oportunidade de falar. O botão “Request Speech” é utilizado para o aluno pedir a palavra ao professor. Uma vez concedida, o estudante tem seu audio e vídeo transmitidos para o professor.

A figura 5.9 apresenta a tela de manuseio do aluno.

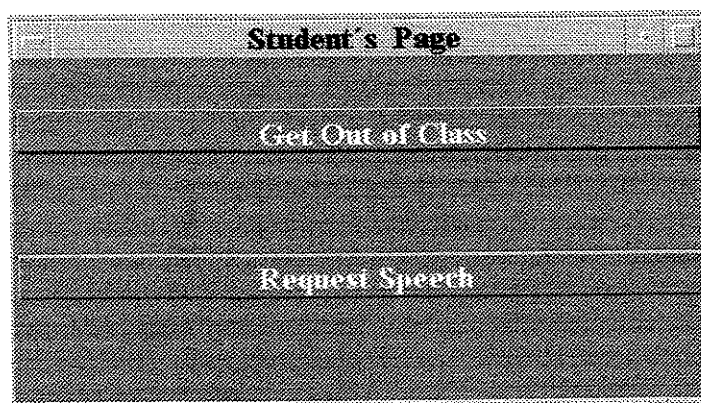


Figura 5.9 - Tela de manuseio do aluno com botão para pedir a palavra

Conforme pode ser visto na figura 5.10, ao ser selecionado, o botão “Request Speech” muda para “Cancel Speech Request”, para indicar ao aluno que, caso deseje cancelar o pedido da palavra, ele deverá pressionar o mesmo botão.

Uma mensagem no canto inferior esquerdo da tela (“waiting for permission...”), avisa o estudante que o pedido já foi enviado, devendo o mesmo esperar pela permissão.

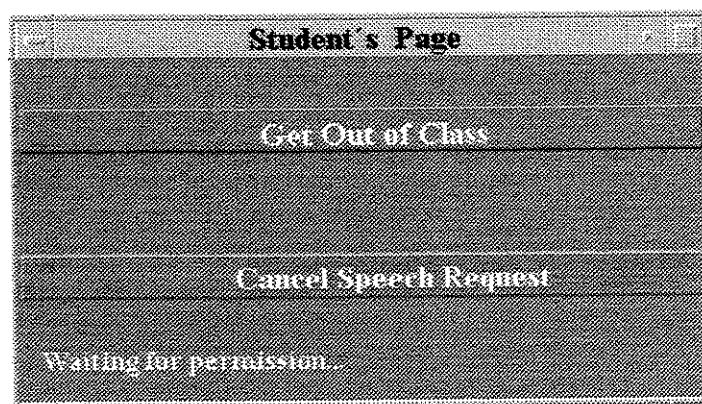


Figura 5.10 - Tela de manuseio do aluno com botão para cancelar o pedido

A tela foi concebida desse modo para forçar o aluno pedir a palavra apenas uma única vez, impedindo-o de enviar a mesma requisição várias vezes, o que sobrecarregaria o sistema desnecessariamente, principalmente se vários alunos resolvessem tomar essa atitude simultaneamente.

Se o botão “Cancel Speech Request” for pressionado, o pedido da palavra será cancelado. O *login* do aluno será então retirado da lista “Who wants to speak” da tela do professor e o nome do segundo botão na tela do aluno, voltará a ser “Request Speech” (como na figura 5.9), permitindo que o estudante possa fazer o pedido novamente caso necessite. Se o botão “Request Speech” for pressionado e o aluno não cancelar o pedido, este pedido ficará na lista de espera “Who wants to speak” da tela do professor até ser atendido. Quando o pedido é aceito o botão da tela do aluno muda de “Cancel Speech Request” para “Return Speech to Teacher”, como na figura 5.11. O botão “Return Speech to Teacher” é pressionado quando o aluno deseja devolver a palavra para o professor.

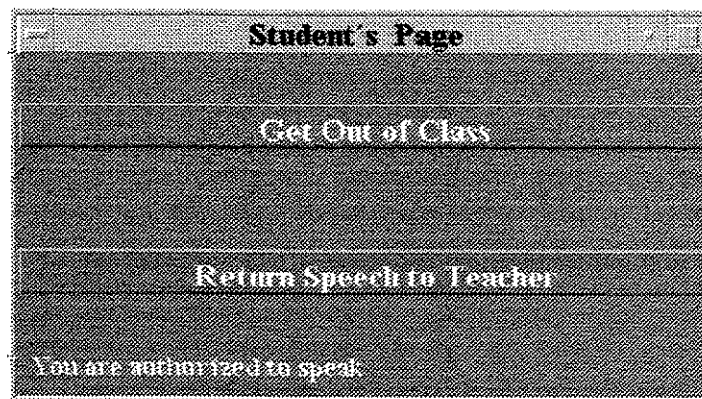


Figura 5.11- Tela de manuseio do aluno com botão para retorno da palavra

De um modo geral, é considerado mais interessante que os botões apareçam somente no momento de uso e que, se alternem conforme a necessidade do aluno, pois além do fato da quantidade de requisições, que podem ser enviadas pelos alunos, a mudança de botões não sobrecarrega a tela de intervenção do aluno e “indica o caminho” a seguir.

A pergunta ou comentário do estudante é transmitida para o professor e demais alunos. Na tela do professor aparecem as telas de controle referentes à transmissão do áudio e do vídeo e na tela dos demais alunos aparece a tela de controle referente à transmissão do áudio. A figura 5.12 ilustra as telas de controle.

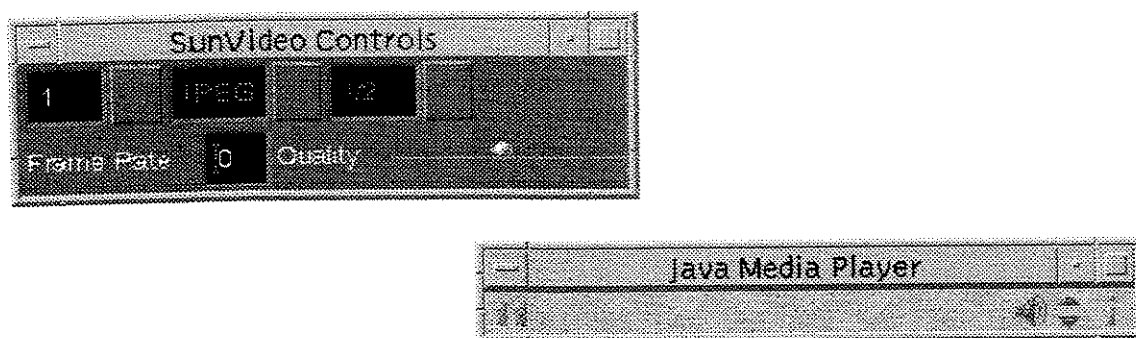


Figura 5.12 – Telas de controle de vídeo (acima) e de áudio

Ao devolver a palavra para o professor, o segundo botão da tela de intervenção do aluno volta a ser “Request Speech” e todo o processo pode voltar a se repetir, caso o aluno queira pedir a palavra novamente.

O *login* do aluno chega até o professor por meio do Serviço de Eventos do padrão CORBA, o qual permite o uso de eventos como se fossem “recados” no formato de *strings*, avisando o professor sobre algum acontecimento (no caso, o ingresso do aluno e o pedido da palavra). Dessa forma, quando o aluno digita o *login* seguido do botão “OK” ou quando pressiona o botão “Request Speech” para pedir a palavra, um evento é gerado e enviado para o professor. De forma análoga, notificações do professor para o aluno são propagadas através do serviço de Eventos.

A transmissão de eventos entre consumidores e produtores são síncronas e ocorrem em paralelo, demandando portanto o uso de *threads*.

5.4 Projeto da Lógica do Controle

No sistema de videoconferência AVE, a transferência de informação entre professor e estudante se dá utilizando o modelo *push* do Serviço de Eventos, pois o evento pode ser enviado pelo produtor assim que gerado, sem que o consumidor necessite verificar a sua existência, como no modelo *pull*.

Foram utilizados quatro canais de eventos no sistema. Três deles para enviar a informação do estudante para o professor e um para enviar a informação no sentido oposto.

Cada canal de eventos possui um nome que é registrado no Serviços de Nomes para que os respectivos produtores e consumidores possam encontrá-lo e nele se registrarem. Os

produtores e consumidores não precisam ser registrados no Serviço de Nomes, porque um produtor não precisa ter a referência de um consumidor e vice-versa. Ambos se registram no canal que propaga o evento entre eles.

A utilização de um único canal exige maior atenção na manipulação das informações de entrada e saída, por isso a melhor opção encontrada foi utilizar quatro canais de eventos, organizando, desse modo, as informações por função. Estes canais foram registrados no Serviço de Nomes como “EventLogin”, “EventRequest”, “EventAllow” e “EventStop”.

Como mais de um evento pode se propagar no canal, foi necessário acrescentar junto a cada evento uma *string* para indicar o que deve ser feito após o consumidor recebê-lo, conforme descrito abaixo.

O Serviço de Eventos provê a separação das informações por tipagem, mas a maneira mais simples encontrada para organizá-las foi criando quatro canais e acrescentando junto ao evento a *string*.

5.4.1 O Canal de Eventos “EventLogin”

O canal de eventos registrado como “EventLogin” no Serviço de Nomes, foi usado para a entrada e a saída dos alunos no sistema, como pode ser observado na figura 5.13.

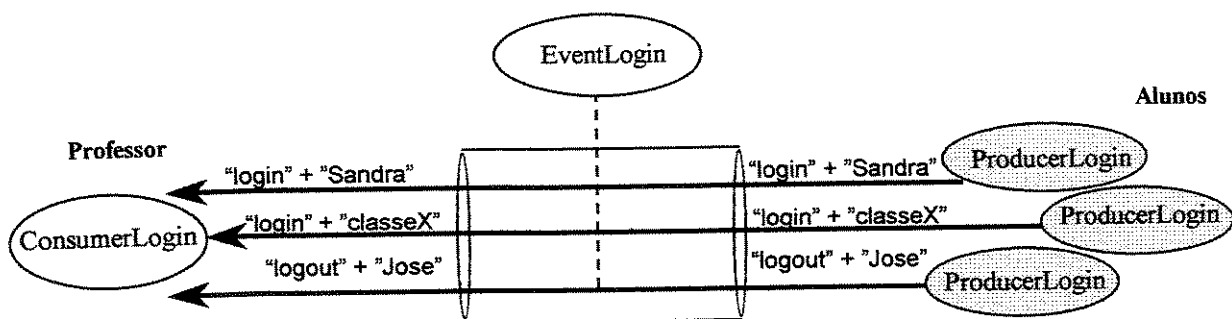


Figura 5.13 – Canal de eventos EventLogin

Ao acessar o sistema, o *login* do aluno é enviado através de um canal até o professor. O *login* é concatenado a uma *string* “login”, que funciona como um código indicando que o aluno está acessando o sistema. Ao sair do sistema, o *login* do aluno é concatenado a uma *string* “logout” para indicar que o aluno está se retirando. Para que ambos os eventos de acesso e saída do aluno possam ser atendidos, o consumidor espera pelo evento desde o início até o fim da aula. Como é utilizado um único canal de eventos para entrada e

saída dos alunos no sistema, quando a *string* concatenada chega até o consumidor do evento, o sistema extrai a referência “login” ou “logout” para saber que medida tomar. Se o aluno estiver acessando o sistema, o nome que segue o *login* será adicionado na lista de alunos presentes da tela do professor. Se o aluno estiver se retirando, o nome que segue o *logout* será eliminado da lista dos alunos presentes e também da lista de alunos que desejam pedir a palavra (caso esteja incluído nesta segunda lista).

5.4.2 O Canal de Eventos “EventRequest”

O canal de eventos “EventRequest” está relacionado ao pedido e cancelamento do pedido da palavra do aluno, conforme pode ser observado na figura 5.14.

Quando o aluno pede a palavra, o produtor de eventos envia uma *string* com o *login* do aluno concatenado a uma *string* “speaker” para o professor. A *string* “speaker” é a referência para o consumidor saber que o aluno deseja pedir a palavra. Da mesma forma, quando o aluno cancela o pedido da palavra, o produtor de eventos envia uma *string* contendo o *login* do aluno concatenado a uma *string* “cancel”. Quando a *string* concatenada chega ao consumidor, o *login* do aluno e a referência “speaker” ou “cancel” são extraídos da *string* concatenada. Se a referência for “speaker”, o *login* do aluno será apresentado na tela de controle do professor na lista “Who wants to speak”. Se for “cancel” o *login* do aluno será retirado desta lista.

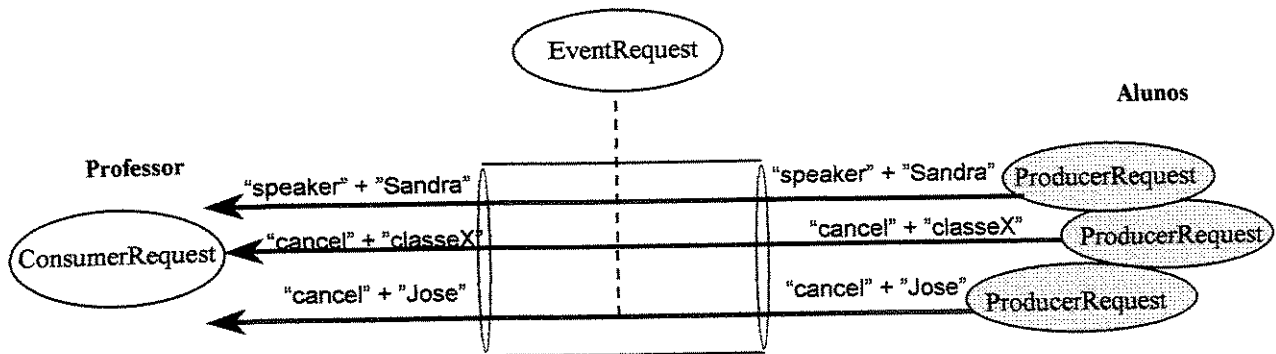


Figura 5.14 – Canal de eventos EventRequest

5.4.3 O Canal de Eventos “EventAllow”

O próximo canal de eventos, ilustrado pela figura 5.15, trata da autorização do professor em conceder a palavra para o aluno. Neste caso, o evento é enviado do professor para o aluno.

Quando o professor seleciona um dos *logins* da lista de alunos que desejam pedir a palavra, o *login* do aluno desta lista de espera é retirado para servir como uma identificação, e concatenado junto a uma *string* “ok” para indicar que aquele estudante tem a permissão para falar.

O professor pode conceder a palavra a qualquer aluno. O problema nesta questão é saber como direcionar o evento para um determinado consumidor, uma vez que o evento se propaga para todos os consumidores registrados no canal. A solução encontrada para que apenas um dos estudantes tenha a autorização, foi fazer com que todos que tenham pedido a palavra recebam o *login* do aluno autorizado a falar. Dessa forma, o evento se propaga a todos os consumidores do canal. O *login* do aluno autorizado é comparado com o *login* do aluno que recebe o evento. Se forem iguais, o aluno terá a palavra.

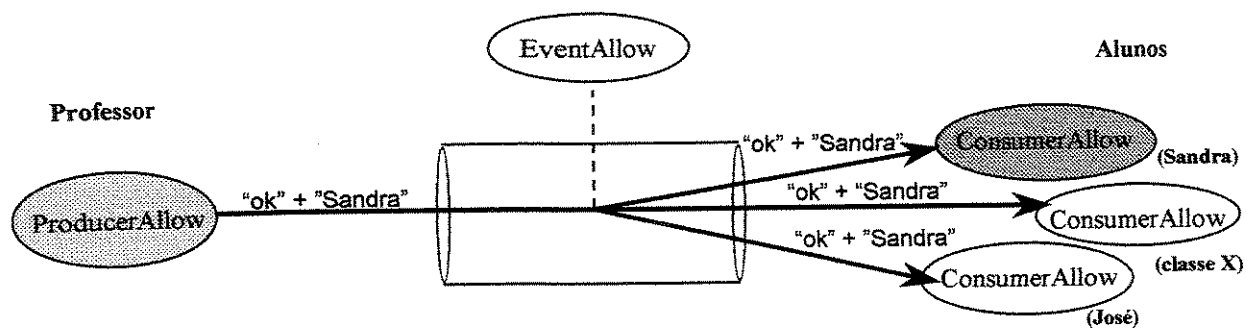


Figura 5.15 - Canal de eventos EventAllow, quando o professor autoriza a palavra do aluno

Caso o professor queira retomar a palavra, deve pressionar o botão “Return Control” em sua tela. Uma *string* “stop” é enviada para todos os consumidores do canal, servindo como um aviso para que o aluno cesse¹⁹ a transmissão. Esta solução faz com que todos os alunos recebam o evento, conforme pode ser visto na figura 5.16, mas que apenas o aluno que está com a palavra seja afetado.

¹⁹ Para fazer a transmissão, o produtor de vídeo do aluno recebe um método *start* e para cessar a transmissão recebe um método *stop*. O procedimento quanto ao uso destes métodos está descrito no item 5.5.1.

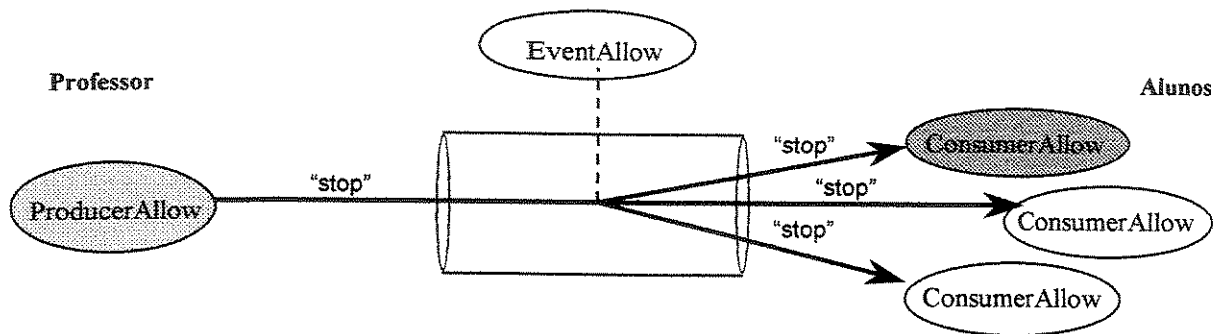


Figura 5.16 – Canal de eventos **EventAllow**, quando o professor retoma a palavra

5.4.4 O Canal de Eventos “EventStop”

O quarto e último canal de eventos, apresentado pela figura 5.17, é usado quando o próprio aluno toma a iniciativa de devolver a palavra para o professor pressionando o botão “Return Speech to Teacher” em sua tela de controle. Quando isso ocorre, a transmissão de áudio e vídeo do aluno pára e uma *string* “stop” é enviada para o professor para indicar que a palavra está sendo devolvida.

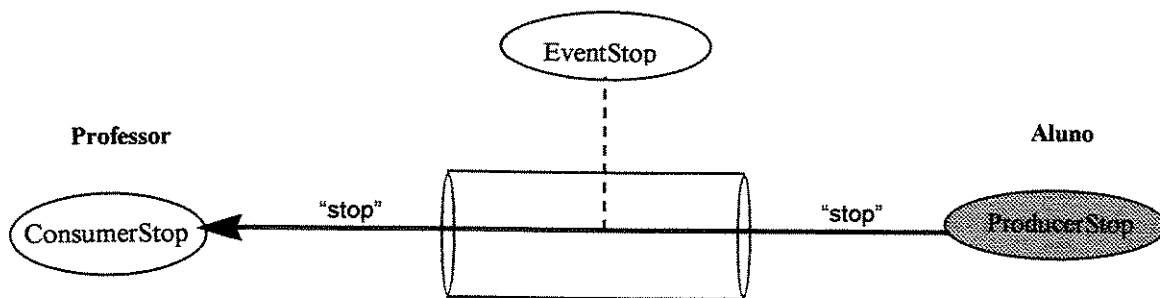


Figura 5.17 – Canal de eventos **EventStop**

A transmissão da mídia entre FEP produtor e consumidor, assim como a de eventos utiliza sincronismo. Desse modo, para que as transmissões de mídia aconteçam em paralelo aos eventos, optou-se pelo uso de *threads*.

Os eventos descritos representam exatamente as interações da seção 4.4.

5.5 Projeto da Comunicação Multimídia

O sistema de videoconferência AVE recebe os *logins* dos alunos que acessam a sala de aula virtual antes ou depois da aula ter início. A aula inicia quando o professor pressiona o botão “Begin Class” (que faz a transmissão do áudio e do vídeo para os alunos) e termina quando o professor pressiona o botão “Finish Class” (em que a transmissão da mídia é encerrada). A transmissão do áudio e do vídeo do professor para os alunos é *multicast* e direcionada a um endereço IP classe D.

Durante a aula o aluno pode requisitar a palavra enviando o *login* para o professor, que por sua vez, pode conceder ou não. Se a palavra for concedida, o áudio e vídeo do aluno é transmitido para o professor.

A transmissão do vídeo do aluno para o professor é *unicast* e do áudio é *multicast* (direcionada também para os demais alunos) A transmissão do áudio é dada do aluno com a palavra para o professor e demais alunos.

Na transmissão de vídeo *unicast*, a imagem do estudante é recebida pelo professor quando o aluno acessa o sistema. Nos testes realizados no laboratório, o tempo para a criação da janela do vídeo do aluno e das telas de controle referentes ao vídeo, foi perceptivelmente²⁰ demorado. Devido a esse fato, optou-se por criar toda essa “estrutura” do vídeo no momento em que o aluno tem acesso ao sistema. Assim, o vídeo que é transmitido para o professor é automaticamente “congelado” até o momento em que o aluno tiver a palavra.

Além do tempo de espera da montagem da janela do vídeo, outro fator importante é que todas as imagens “congeladas” dos estudantes não sobrecarregam tanto o computador do professor, considerando que uma grande quantidade de imagens em movimento utilizam muito mais CPU do que imagens “congeladas”.

A imagem dos alunos vista pelo professor, não só mostra quem está presente na aula, como também permite que o aluno expresse melhor uma dúvida ou comentário.

Quando o professor autoriza a palavra, apenas a imagem do estudante com permissão para falar é “descongelada”, sendo novamente “congelada” quando o aluno terminar de falar. Já as telas que acompanham o áudio do estudante com a palavra são criadas e apresentadas no momento em que o aluno é autorizado a falar, por ser um processo mais rápido do que o do vídeo.

Posteriormente, quando o estudante devolve a palavra para o professor, por uma questão de flexibilidade, o áudio é momentaneamente suspenso²¹ para que suas telas não sejam recriadas quando o estudante for autorizado a falar novamente. Quando o aluno tem novamente a palavra, o áudio, assim como o vídeo, voltam a ser retransmitidos.

²⁰ Uma variação de segundos, conforme o computador utilizado.

²¹ Toda a estrutura do áudio do aluno com a palavra não é destruída. O fluxo é apenas interrompido momentaneamente.

Se fosse utilizado uma transmissão *multicast* de vídeo no lugar de uma *unicast*, cada aluno receberia a transmissão de vídeo dos demais alunos, o que viria a sobrecarregar os computadores dos alunos presentes. Neste caso, alguns computadores (de alunos) podem não suportar uma grande quantidade de imagens, limitando desta forma, a participação destes computadores na videoconferência. A tendência é que no futuro seja possível utilizar uma transmissão de vídeo *multicast*.

Além do aspecto da quantidade de imagens que o computador do aluno pode suportar, o tempo de espera da montagem da janela do vídeo é relevante.

Como a transmissão do áudio do aluno não possui taxas de bits/seg tão alta como a do vídeo é possível permitir que outros estudantes possam ouvir a dúvida ou o comentário do aluno com a palavra.

5.5.1 Fluxo de Áudio e Vídeo entre Professor e Estudantes

Na transmissão e recepção do fluxo de áudio e vídeo entre professor e estudantes, foi utilizada uma implementação na linguagem Java, do Serviço A/V Streams. Assim sendo, os componentes utilizados no sistema para o gerenciamento e controle do fluxo puderam interagir através do ORB nativo da própria linguagem Java. O fluxo de mídia é transmitido externamente ao ORB via protocolo RTP/UDP porque o padrão CORBA utiliza o protocolo TCP/IIOP.

Do serviço, foi escolhido o modo de operação *full profile*, devido a flexibilidade no controle das interfaces produtoras e transmissoras de fluxo.

Quando o professor transmite seu vídeo para os alunos, utiliza um *FlowEndPoint* (FEP) produtor representado por uma câmera, enquanto que o aluno utiliza um FEP consumidor representado por uma tela de vídeo. O controle, por exemplo, é dado diretamente no FEP produtor de vídeo, para o professor iniciar ou interromper sua transmissão.

Na transmissão da mídia do professor para os estudantes criou-se um FEP produtor para o professor e um FEP consumidor para cada aluno.

Como é necessário que o professor transmita seu fluxo de áudio e de vídeo para mais de um aluno ao mesmo tempo, foram utilizados dois fluxos de mídia *multicast* (endereço IP classe D), um para a transmissão do vídeo como representado na figura 5.18, e outro para a transmissão do áudio, como na figura 5.19. Já para a transmissão do aluno com a palavra para o professor e demais alunos, foi utilizado o mesmo canal *multicast* de áudio, como representado na figura 5.20, e um outro canal *unicast* para a transmissão do vídeo, como demonstrado na figura 5.21.

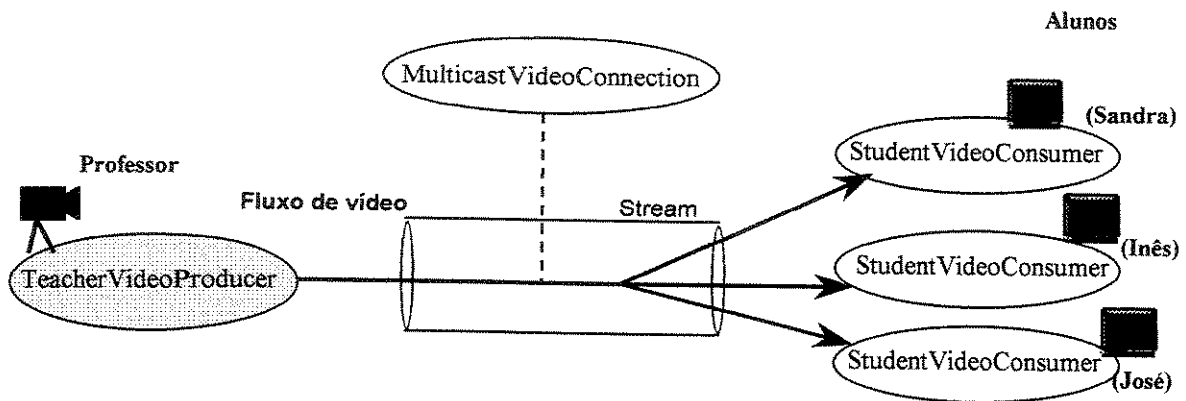


Figura 5.18 – Transmissão *multicast* de vídeo, do professor para os alunos

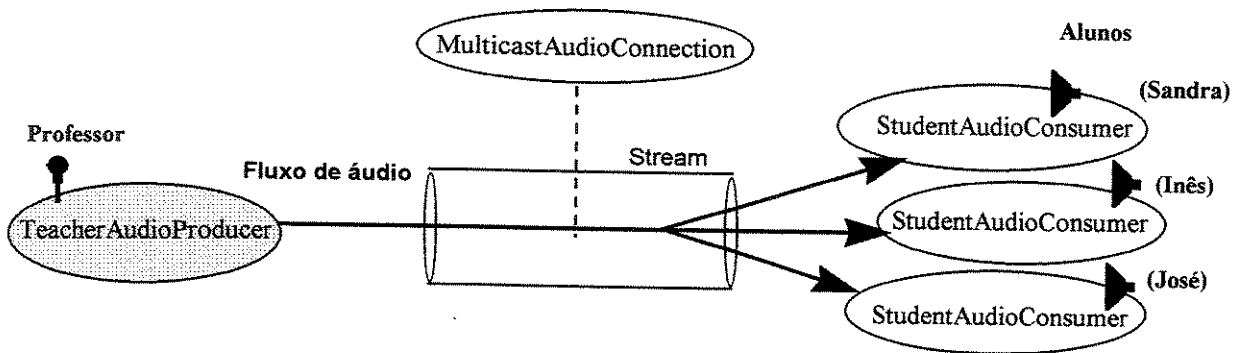


Figura 5.19 – Transmissão *multicast* de áudio, do professor para os alunos

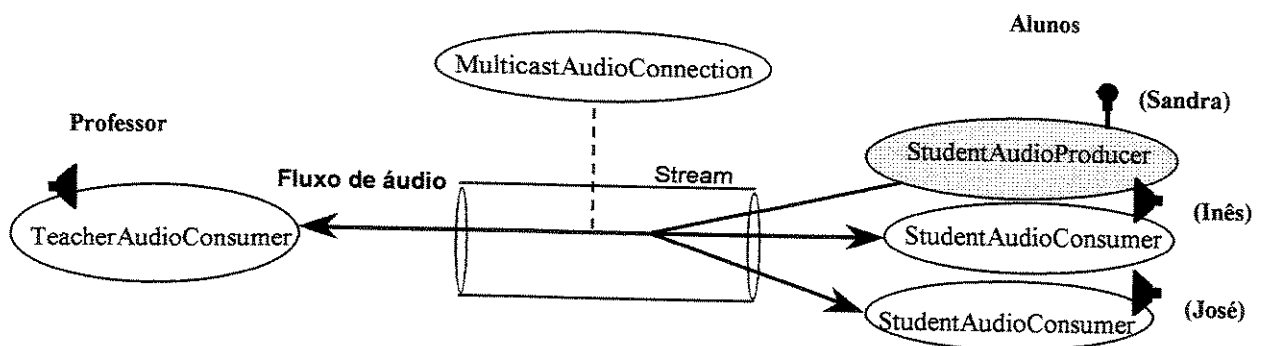


Figura 5.20 – Transmissão *multicast* de áudio, do aluno para o professor e demais alunos

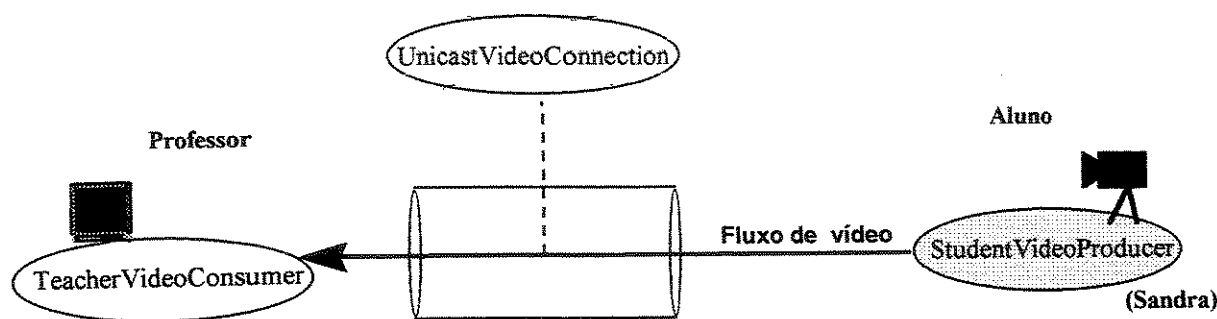


Figura 5.21 – Transmissão *unicast* de vídeo, do aluno com a palavra para o professor

Os FEPs produtor e consumidor podem ser criados sem estarem conectados a um *FlowConnection* (como por exemplo, *MulticastAudioConnection*) ou podem ser criados já conectados a um *FlowConnection*. A primeira opção permite que os FEPs sejam posteriormente conectados e que quando necessário, sejam retirados. Como a segunda opção não tem esta flexibilidade, a primeira foi escolhida para ser utilizada no sistema.

Os FEPs podem conectar-se ou desconectar-se sem influenciar outros FEPs. Desse modo, para que o professor possa fazer sua transmissão em mídia o FEP produtor é adicionado a um *FlowConnection*. Este FEP é retirado da conexão quando o professor termina a aula.

Para que o aluno que acessa a sala de aula virtual possa assistir a transmissão do professor, o FEP consumidor é adicionado ao mesmo *FlowConnection*. O aluno pode acessar a sala de aula virtual, assistir a transmissão e se retirar quando desejar. Dessa forma, o FEP consumidor pode ser adicionado ao *FlowConnection* quando o aluno acessa a aula e pode ser retirado quando o aluno se retira da aula. Já na conexão de vídeo *unicast* (do aluno com a palavra para o professor) a adição dos FEPs no canal não foi utilizada porque não há necessidade do FEP produtor nem do FEP consumidor serem adicionados ou retirados da conexão durante²² a transmissão. Portanto a conexão é criada com os FEPs produtor e consumidor já por ela conectados.

Na conexão *multicast* (tanto no áudio quanto no vídeo) no computador do professor foram criados um *FlowConnection*, um FEP produtor e um FEP consumidor e, no computador do aluno foram criados um FEP consumidor e um FEP produtor.

Dessa forma, para localizar o *FlowConnection* (que se encontra no sistema do professor) o FEP consumidor (que se encontra no sistema do aluno) utiliza o Serviço de Nomes.

Uma vez registrado no Serviço de Nomes, o *FlowConnection* pode ser acessado de qualquer computador participante, ou seja, qualquer aluno participante pode buscar a referência do objeto *FlowConnection* e ser adicionado à conexão.

Na conexão de vídeo *unicast* no computador do aluno foram criados um *FlowConnection* e um FEP produtor e, no computador do professor foi criado um FEP consumidor. O

²² Nem aluno, nem professor se retiram durante a conversação. Isto só acontece quando a conversação é encerrada.

FlowConnection se localiza no lado do aluno para que o computador do professor não fique sobrecarregado com uma quantidade excessiva de conexões de alunos.

5.5.2 Controle dos FEPs

Seguindo o Serviço A/V Streams para o controle da transmissão da mídia, foram utilizados métodos que atuam diretamente sobre os FEPs, iniciando, parando, ou destruindo uma conexão. Respectivamente, estes métodos são representados como *start*, *stop* e *destroy*.

O método *start* faz com que o FEP produtor transmita e o FEP consumidor receba a transmissão. O método *stop* interrompe temporariamente a transmissão da mídia até que seja dado um *start* novamente. Já o método *destroy* elimina a conexão, sendo necessário recriar toda a estrutura (*FlowConnection* e FEPs produtor e consumidor) para poder enviar ou receber a transmissão novamente.

Também é possível iniciar ou parar o *FlowConnection*, porém os FEPs produtores e os consumidores adicionados no *FlowConnection* iniciam ou param simultaneamente. Assim sendo, optou-se por utilizar o controle sobre os FEPs ao invés de utilizar sobre o *FlowConnection*, pois é possível ativar um FEP produtor ou consumidor de cada vez, quando necessário.

No sistema de videoconferência AVE para que o professor possa transmitir o vídeo para os alunos é necessário que o FEP produtor, representado pela câmera de vídeo do professor, esteja no estado *start*. Do mesmo modo, para que o aluno possa receber a imagem do professor, o FEP consumidor, representado pela tela do aluno também deve estar no estado *start*.

Na conexão *multicast* de áudio, conforme pode ser observado na figura 5.22, professor e aluno com a palavra utilizam a mesma conexão para fazer a transmissão. Dessa forma, ambos possuem um FEP produtor (um microfone) e um FEP consumidor (um alto-falante).

Quando o FEP produtor, referente ao professor, estiver no estado *start*, o FEP produtor, referente ao aluno, deve estar no estado *stop*, pois quando existe mais de um FEP produtor em uma mesma conexão, somente um deles deve transmitir (não existe a facilidade de “mixagem”).

Na transmissão do professor para os alunos, o FEP produtor permanece em *start* e o FEP consumidor (professor) em *stop*. O FEP consumidor permanece em *stop* para que o professor não ouça a própria voz (gerando eco). Já o FEP produtor, referente ao aluno, permanece no estado *stop* e o FEP consumidor, em *start*.

Quando o aluno com a palavra transmite áudio para o professor e demais alunos, o FEP produtor (agora do aluno) encontra-se no estado *start* e o FEP consumidor no estado *stop*. Já no professor, o FEP produtor permanece em *stop* e o FEP consumidor em *start*. Os FEPs dos demais alunos não são alterados (FEP produtor em *stop* e FEP consumidor em *start*).

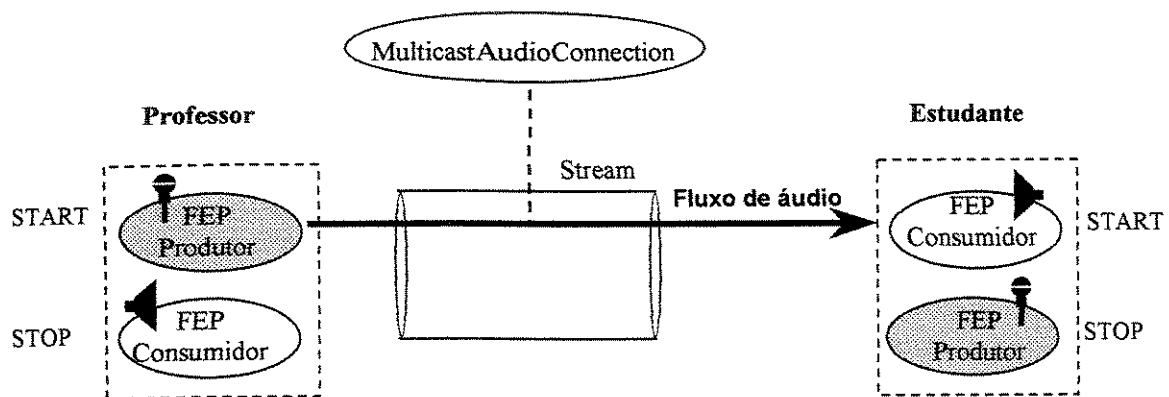


Figura 5.22 – Controle do FEP na transmissão *multicast* do áudio

Na transmissão de vídeo *multicast*, como apresentado na figura 5.23, do professor para os alunos, foi criado o FEP produtor referente ao professor e o FEP consumidor referente ao aluno. O outro sentido da transmissão do aluno para o professor utiliza a conexão *unicast*.

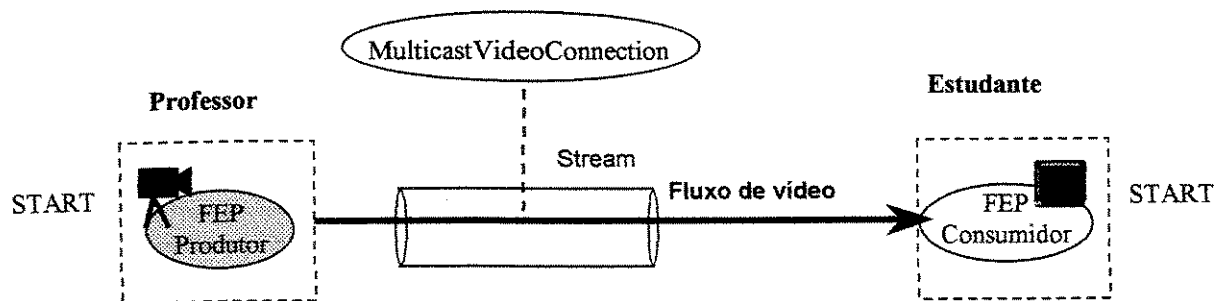


Figura 5.23 – Controle do FEP na transmissão *multicast* do vídeo

Na transmissão de vídeo *unicast* do estudante com a palavra, existem o FEP produtor, referente ao aluno, e o FEP consumidor referente ao professor. Ambos os FEPs permanecem no estado *start*. Como a imagem de vídeo do aluno deve ser recebida pelo professor e “congelada” logo em seguida, os FEPs comutam posteriormente para *stop*.

Nesta transmissão é possível fazer com que o aluno (com a palavra) continue recebendo a imagem de vídeo do professor enquanto a sua imagem é transmitida. Para não sobrecarregar seu computador, optou-se por deixar o FEP produtor do professor da conexão *multicast* em *stop*.

Devido a forma como o protocolo RTP opera, uma vez estabelecido quem é o FEP produtor, o FEP consumidor não consegue mais “esquecê-lo” (isto é, só apresenta a imagem ou o áudio deste produtor). Uma possível solução é destruir os FEPs consumidores e recriá-los todas as vezes em que um novo produtor for adicionado à conexão.

Porém, para a transmissão de vídeo esta solução não foi viável, uma vez que o tempo gasto na criação dos consumidores foi significativo. Esta foi a principal razão para se utilizar uma outra conexão de vídeo, além da conexão *multicast* do professor para os estudantes. Como a outra conexão deve comunicar somente o professor e o aluno com a palavra, foi criada uma conexão de vídeo *unicast*. Nesta conexão, cada aluno autorizado a falar, abre uma conexão de vídeo com o professor. Assim sendo, tanto na conexão *unicast*, quanto na conexão *multicast* o FEP produtor não é alterado. Em ambas conexões, os FEPs consumidores criam a janela do vídeo uma única vez, não existindo a necessidade de serem destruídos e recriados.

Na conexão *unicast* quando o aluno devolve a palavra o FEP produtor (aluno) e o FEP consumidor (professor) da conexão recebem um *stop* até que o aluno tenha a palavra novamente, para que então os FEPs da conexão possam receber um *start*. Já para o áudio, uma única conexão é aproveitada na transmissão do professor para os alunos e do aluno com a palavra para o professor e demais alunos. Nesta conexão, os FEPs consumidores são destruídos e recriados na troca de FEPs produtores (quando professor concede palavra para aluno ou quando o aluno devolve a palavra para o professor). Como a destruição e criação dos FEPs consumidores do áudio é bem mais rápida do que a do vídeo, optou-se por deixar a conexão de áudio alternando seus FEPs produtores, ao contrário da conexão de vídeo.

5.6 Caso de Teste

Conforme pode ser observado na tabela 5.1, foram utilizados quatro computadores no teste: Botafogo, Buzios, Enseada, e Mocooca.

Nome do computador	Tipo	Câmera de vídeo	Microfone	Memória RAM	Clock
Botafogo	Estação Sun ultra 1 com sistema Solaris	Sim	Sim	196 M	200 MHZ
Buzios	Estação Sun, sparc station 4 com sistema Solaris	Não	Não	64 M	110 MHZ
Enseada	Estação Sun com sistema Solaris	Sim	Sim	64 M	167 MHZ
Mocooca	Microcomputador Pentium II com Windows NT	Não	Sim	64 M	333 MHZ

Tabela 5.1 – Computadores utilizados no caso de teste

Todas essas máquinas estão conectadas na rede do laboratório que está dividida em subredes. As estações Botafogo, Enseada, Buzios e Mocooca estão conectadas em uma subrede IP/ATM. Cada computador está conectado em uma porta do switch ATM, com velocidade de 155 Mbits/s.

Computador transmissor	Computador receptor	Taxa de transmissão do vídeo	Taxa de transmissão do áudio
Botafogo	Enseada	617.000 bps a 1084.000 bps	32.542 bps a 68.435 bps
Enseada	Botafogo	165.000 bps a 861.000 bps	22.928 bps a 48.981 bps

Tabela 5.2 – Taxa de transmissão do teste

Verificou-se que a apresentação do vídeo depende mais do processamento do computador que está emitindo ou recebendo o vídeo do que da própria rede (considerando que a transmissão foi local).

Concluiu-se que o ideal é ter-se um computador rápido o suficiente para o professor transmitir e receber o vídeo da aula. Por isso, o computador Botafogo representou o computador utilizado pelo professor. Para o aluno, o ideal é um computador rápido o suficiente para receber e transmitir vídeo.

Para a recepção do vídeo no computador do aluno foi perceptível a diferença entre os computadores Enseada e Mocooca. No computador Mocooca a resolução da imagem foi melhor e a velocidade de apresentação dos quadros foi mais rápida do que a do computador Enseada. Esta diferença na apresentação do vídeo deve ser pelo fato do Java Media Framework, utilizado no A/V Streams, ter eficiência maior em ambientes Windows. Já para a transmissão ou recepção do áudio, a qualidade independe do computador usado.

Quanto aos eventos, percebeu-se que a notificação do produtor para o consumidor é quase que imediata, como se o procedimento estivesse sendo executado em um mesmo espaço de endereçamento.

Capítulo 6

Conclusão

Neste capítulo, apresenta-se uma retrospectiva da tese abordando as tecnologias empregadas e sua utilização no desenvolvimento do sistema de videoconferência para educação a distância, bem como possíveis trabalhos futuros.

6.1 Introdução

Muitas pessoas não possuem condições para estudar por falta de horário ou por residirem em locais distantes dos centros urbanos, longe das escolas. Uma solução para estas pessoas é fazer uso da educação a distância.

Segundo Garcia Aretio “a educação a distância é um sistema tecnológico de comunicação bidirecional, que pode ser massivo e substituir a interação pessoal na sala de aula (professor - estudante) pela ação sistemática e conjunta de diversos recursos”. A escolha do meio de comunicação depende do tipo do público alvo, da eficácia da transmissão e recepção [1], e dos vários recursos tecnológicos empregados na apresentação da aula.

Atualmente, existe uma grande variedade de ferramentas que podem ser combinadas para a criação de um sistema de educação a distância. Dentre os sistemas síncronos estão as aplicações que utilizam transmissões em áudio e vídeo para a comunicação entre professor e estudante. Porém, muitos desses sistemas são proprietários e não operam em ambientes heterogêneos.

6.2 Avaliação e Contribuições

Como proposta, foi desenvolvido um sistema de videoconferência para educação a distância não proprietário, distribuído e destinado a ambientes heterogêneos para atender

Conclusão

a necessidade da comunicação entre professor e estudante, por meio de transmissões em áudio e vídeo e de notificações por eventos.

O sistema desenvolvido poderá ser utilizado em conjunto com um *whiteboard* e integrado em outros ambientes educacionais como CALM [7], AulaNet [30,31] ou WebCT [32]. O sistema poderá ser implementado como *applet* para ser utilizado no WebCT, por exemplo. Neste caso, o sistema seria considerado uma funcionalidade a mais disponível para o usuário, pois WebCT não permite interação professor-aluno através de áudio e vídeo.

Seguindo o padrão CORBA foi utilizado os Serviços de Eventos e de controle de fluxos multimídia (Serviço A/V Streams). CORBA é uma tecnologia de fácil manuseio²³ que possui maturidade para o tratamento de objetos distribuídos, é própria para ambientes heterogêneos e fornece uma série de serviços. Esta tecnologia empregada junto a linguagem Java propiciou o uso de um ORB nativo da própria linguagem, suportando a interação entre objetos distribuídos em plataformas diferentes. Além desse fato, a linguagem proporciona facilidades para o desenvolvimento de interfaces gráficas, para a incorporação de áudio e vídeo (JMF), para a transmissão de eventos (Serviço de Eventos CORBA) e para o uso de *threads*. *Threads* foram utilizadas para permitir que as transmissões em mídia e os eventos pudessem ser processados concorrentemente no sistema

Com o Serviço de Eventos foi possível notificar os participantes (conectados via rede) de maneira simples e rápida. Quatro canais de eventos foram criados, para organizar os eventos por função, sendo que uma *string* foi concatenada em cada evento para indicar a medida a ser tomada pelo consumidor. O resultado foi satisfatório e atendeu as necessidades do sistema.

Com o uso da especificação A/V Streams foi possível criar um sistema que atendesse as necessidades básicas da transmissão de mídia contínua. Com essa tecnologia foi possível dar flexibilidade (através das fábricas) quanto a escolha do local em que o consumidor receberia o vídeo (o professor pôde escolher o computador em que a imagem do aluno seria apresentada).

A facilidade para adicionar ou remover fluxos no canal, tornou possível a implementação de acesso e retirada dos alunos durante a transmissão do curso. Apesar da sua simplicidade de manuseio, A/V Streams apresentou uma desvantagem quando os consumidores do sistema precisaram ser destruídos e recriados em cada troca de produtor (do mesmo canal de comunicação), dificultando o processo da transmissão de vídeo, devido a necessidade da recriação da janela. Como solução foram criados dois canais de vídeo, um do professor para os alunos e outro do aluno para o professor (a fim de manter o mesmo produtor durante as transmissões).

Para contornar o problema do tempo de espera, para gerar a janela de vídeo do aluno com a palavra, foi necessário fazer com que o computador do professor recepcionasse e

²³ Se comparada a outras tecnologias como por exemplo, RPC.

“congelasse” o vídeo de todos alunos presentes. Assim, quando o aluno tivesse a palavra apenas sua imagem seria “descongelada” e novamente “congelada” ao devolvê-la.

Para não sobrecarregar o computador dos demais alunos com a imagem de cada aluno com a palavra, foi transmitido somente o áudio para estes alunos. Outro aspecto presenciado com o uso do A/V Streams foi a diferença de exibição da imagem transmitida em computadores com o sistema Windows NT e outro com o sistema Solaris. No NT a imagem é mais nítida e a mudança de quadros é mais rápida, provavelmente porque JMf utilizado pelo A/V Streams esteja mais direcionado para uso em ambientes Windows, o que não pode ser considerado como empecilho para computadores com sistemas UNIX, apesar da sua apresentação.

O ideal é que recepção da transmissão do vídeo seja feita em um computador que utilize Windows NT. Já para a transmissão ou recepção do áudio e dos eventos, o computador usado é indiferente.

Outro fator que conta principalmente na transmissão do vídeo é a rede como meio de comunicação. Certamente se o meio de comunicação é uma rede de alto desempenho (como ATM), mas o computador não é rápido o suficiente para gerar uma imagem com qualidade (ou seja, demora para compor a imagem na tela) então pode-se dizer que o cenário é inviável. O mesmo pode ser afirmado, para um cenário em que o meio de transmissão é uma rede que opera em condições de melhor esforço (como a *Internet*), e o computador utilizado para receber as imagens é veloz. Dessa forma, principalmente para o vídeo, é necessária a utilização de uma rede de alto desempenho juntamente com um computador que seja capaz de exibir a imagem na tela com certa rapidez.

As redes de computadores de alto desempenho ainda são muito caras e restritas a locais como empresas e universidades, mas este quadro tende a mudar em um futuro próximo.

6.3 Trabalhos Futuros

O sistema AVE pode ser estendido para incluir um *chat* utilizando o Serviço de Eventos da arquitetura CORBA. O *chat* pode funcionar em paralelo à conexão da aula transmitida no telão. Todos os alunos participantes do curso podem trocar informações durante ou depois da aula (considerando que estes alunos tenham acesso individual). Além dos alunos, um monitor de curso também pode fazer parte do *chat* para responder ou ajudar em questões simples.

Para utilizar o *chat* não há necessidade de se ter uma rede de alto desempenho, uma conexão à *Internet* seria suficiente.

Outra extensão para o sistema de videoconferência AVE é o uso de “transparências”. A “transparência” pode ser apresentada na tela do aluno (telão) junto com a imagem do professor e possuir tópicos mencionados pelo professor durante a aula. Quando o professor muda a página da transparência com um clique do *mouse*, automaticamente as

Conclusão

páginas dos alunos também mudam. Esta notificação da mudança de página poderia ser feita também utilizando o Serviço de Eventos do padrão CORBA.

Outra opção que poderia constar no sistema AVE é a utilização de técnicas de realidade virtual, via por exemplo, padrão VRML. Estas transmissões funcionariam em paralelo as transmissões de áudio e vídeo entre professor e estudante e poderiam ser destinadas a explicações mais complexas do professor, como por exemplo, numa aula de biologia, quando o professor precisar apresentar para os alunos a forma de uma molécula.

Bibliografia

- [1] iBase – Instituto Brasileiro de Análise Sociais e Econômicas, “Conceito de Educação à Distância”, <http://www.ibase.org.br/edist/dist/index.htm>, item Alvo.
- [2] Consuelo Tereza Fernandez Gonçalves, ibase – Instituto Brasileiro de Análise Sociais e Econômicas, “Quem tem Medo do Ensino a Distância”, Revista Educação à Distância, No 7-8, 1996, INED/IBASE, ou <http://www.ibase.org.br/edist/dist/index.html>, item Leitura 1.
- [3] Waldomiro P.D.C. Loyolla, Maurício Prates, “Educação a Distância Mediada por Computador (EDMC) Uma proposta pedagógica”, <http://www.puccamp.br/~prates/edmc.html>
- [4] Wolfram Laaser, ibase - Instituto Brasileiro de Análise Sociais e Econômicas, “Produção e Projeto de Vídeo e TV em Educação a Distância”, <http://www.ibase.org.br/edist/dist/index.html>
- [5] Pamela B. Lawhead, Elizabeth Alpert, Constance G. Bland, Linda Carswell, Dawn Cizmar, Jean DeWitt, Mihaela Dumitru, Eva R. Fahraeus, Kirk Scott, “The *Web* and Distance Learning : What is Appropriate and What is Not”, Report of the ITiCSE’97 Working Group on the *Web* and Distance Learning, 1997.
- [6] EDUCNet – Educação a Distância, “Educação à Distância (EAD) – Conceituação”, <http://www.cciencia.ufrj.br/educnet/EDUEAD.htm>
- [7] Christian M. Adriano, Armando L.N. Delgado, Luiz G. Silveira Jr, Raquel Cristina Bosnardo, “Inquiring the Course Paradigm whith CALM”, Conference on Engineering and Computer Education – ICECE99 at Rio de Janeiro, 1999.
- [8] Eli M. Noam, “Os Eletrônicos e o Futuro Obscuro da Universidade”, Revista Adusp, Abril / 1997
- [9] Elisa Wolvnec, “Transformando Instituições de Ensino em Instituições de Aprendizado”, <http://www.techne.com.br/Institucional/Artigos/default.htm>, item VIII InforH.
- [10] Junichi Azuma, “Creating Educational *Web Sites*”, IEEE Communications Magazine, page 109 –113, March / 1999

- [11] Betty Collins, "Applications of Computer Communications in Education: An Overview", IEEE Communications Magazine, page 82 – 86, March / 1999
- [12] Ursula Wolz, Jacob Palme, Penny Anderson, Zhi Chen, James Dunne, Göran Karlsson, Atika Laribi, Sirkku Männikkö, Robert Spielvogel, Henry Walker. "Computer-Mediated Communication in Collaborative Educational Settings" Report of the ITiCSE'97 Working Group on CMC in Collaborative Educational Settings, 1997
- [13] Andreas Ausserhofer, "Web-Based Teaching and Learning: A Panacea ?" IEEE Communications Magazine, page 92 – 96, March / 1999
- [14] Christian Medeiros, Alberto B. Raposo e Ivan L. M. Ricarte, "Implementação de Metáforas de Anotação e Paradigmas de Interação em Ambientes Educacionais", Simpósio Brasileiro de Informática na Educação – SBIE99, Curitiba, 1999.
- [15] Andrew S. Tanenbaum, "Redes de Computadores", 3ª Edição , Editora Campus, 1996
- [16] Kassem Saleh, Robert Probert, Hassib Khanafer, "The Distributed Object Computing Paradigm: Concepts and Applications", The Journal of Systems and Software 47, 125-131, 1999
- [17] OMG. "The Common Object Request Broker: Architecture and Specification" - version 2.2 Object Management Group, http://www.omg.org/gettingstarted/orb_basics.htm
- [18] Alex Chaffee, Bruce Martin "Introduction to CORBA - Short Course", <http://developer.java.sun.com/developer/onlineTraining/corba/corba.html>
- [19] OMG, "Event Service Specification. Technical Report Document", Object Management Group, <http://www.omg.org/> , 11/December/1997
- [20] Eliane G. Guimarães, OMG, "Um Framework de Transmissão de Áudio e Vídeo para Novos Serviços de Telecomunicações", XVIII Simpósio Brasileiro de Redes de Computadores, Belo Horizonte, MG, Maio/2000
- [21] OMG, "Control and Management of Audio/Video Streams". OMG RFP Submission telecom/ 07/May/1997
- [22] Eduardo J. Oliveira, "Uma Arquitetura para Acesso a Serviços de Telecomunicações Baseada em TINA", Dissertação de Mestrado, FEE-Unicamp, Agosto / 1999.

- [23] SUN, Java home page. <http://java.sun.com/products/> , 1999
- [24] Cynthia Bloch, “Overview of Servlets”, <http://www.javasoft.com/docs/books/tutorial/servlets/overview/index.html>, October / 2000
- [25] Deitel & Deitel, “Java How To Program”, 2nd Edition, Prentice Hall, 1998
- [26] Sun Microsystems. “Java Media Framework Programmer’s Guide (v 0.5)”, <http://java.sun.com/products/java-media/jmf> [JMF 2.0 API Guide], December/1998
- [27] Jim Inscore, IDL, <http://www.javasoft.com/docs/books/tutorial/idl/index.html> , October / 2000
- [28] Ann Wollrath and Jim Waldo, “An Overview of RMI Applications”, <http://java.sun.com/docs/books/tutorial/rmi/overview.html>, October / 2000.
- [29] Martin Fowler with Kendall Scott, “UML Distilled – Applying the Standard Object Modeling Language”, 6th printing, Addison-Wesley, 1998
- [30] Mára Lúcia Fernandes Carneiro, Marie Christine Julie Mascarenhas Fabre, julho/99AulaNet – <http://www.niee.ufrgs.br/~alunospg99/mara/aulanet/tsld001.htm>
- [31] AulaNet - <http://www.puc-rio.br/jornaldapuc/agosto98/aulanet.html>
- [32] WebCT - <http://www.webct.com/>