

Um Servidor de Áudio Distribuído Com Edições Dinâmicas

Marcelo Moscato Ferreira

Luis Carlos Trevelin

e-mail: {ferreira, trevelin}@dc.ufscar.br

Universidade Federal de São Carlos - DC
Grupo de Sistemas Distribuídos e Redes (GSDR)
Rodovia Washington Luiz, Km 235 - Caixa Postal 676
13.5565-905 São Carlos - SP Tel. (016) 274-8233

Resumo

Este trabalho apresenta um Servidor de Áudio Distribuído¹ usado para editar, armazenar, recuperar e apresentar a monomídia áudio em um ambiente distribuído colaborativo. Ele oferece um conjunto de serviços de áudio de alto nível, tais como reprodução, gravação e composição da mídia para a construção de aplicações multimídia. Destaca-se ao longo do artigo a capacidade de edições dinâmicas sobre segmentos de áudio distribuídos, entre elas a mistura de diferentes trechos de áudio em tempo real e a inserção de intervalos de silêncio em pontos específicos do mesmo. O servidor, chamado SAD-SMmD, vem sendo desenvolvido para o projeto SMmD² - Sistemas Multimídia Distribuídos. Utilizando-se da plataforma ANSAware para processamento distribuído aberto, o SAD-SMmD possui heterogeneidade de hardware sobre os quais pode executar, compartilhar e gerenciar uma base de dados de áudio aos usuários.

palavras-chave: sistemas multimídia distribuídos, serviços de áudio. ANSAware, SMmD

Abstract

This paper presents a Distributed Audio Server developed for editing, storage, retrieving and presentation of audio monomedia in a cooperative distributed environment. It offers a set of high level audio services such as audio reproduction, recording and composition to build multimedia applications. It is highlighted throughout this paper the real time mixing capability of different audio streams as well as the insertion of silence intervals in any specific point of the stream. The audio server, named SAD-SMmD is being developed for the SMmD - Sistemas Multimídia Distribuídos project. Using the ANSAware open distributed platform the SMmD supports heterogeneous hardware over witch can run, share and manage an audio data base to the users.

key-words: distributed multimedia systems, audio services, ANSAware, SMmD

1. Introdução

A evolução das pesquisas na área de multimídia, somando-se aos avanços das redes de computadores e plataformas baseadas no processamento distribuído aberto, como CORBA [OMG 1995], DCE [OSF 1994] e ANSAware [APM 1993], tem caminhado para soluções distribuídas. As soluções distribuídas,

¹ Este servidor está rodando em uma plataforma de rede de estações SUN, sob os sistemas operacionais SUNOS-5 e SOLARIS-2.x, do Departamento de Computação da Universidade Federal de São Carlos, SP, Brazil.

² SMmD - Sistemas Multimídia Distribuídos: Projeto financiado pelo CNPq em desenvolvimento conjunto na Universidade Federal de São Carlos, Universidade de São Paulo e Universidade Federal do Alagoas (Brasil).

naturalmente, mantém os dados espalhados pela rede e sua utilização ocorre no momento em que a aplicação, que pode ser executada em equipamentos diversos e simultaneamente, os solicitar.

Nesse contexto, em 1993 começou o projeto *Sistemas Multimídia Distribuídos* (SMmD) [Trevelin 1995]. Em linhas gerais o projeto visa definir e implementar uma arquitetura que permita o desenvolvimento de aplicações multimídia e hipermídia para trabalho cooperativo suportado por computador (CSMHCW - Computer Supported Multimedia/Hypermedia Cooperative Work) em ambientes distribuídos.

Entre as ferramentas para o suporte ao desenvolvimento de aplicações multimídia e hipermídia está o Servidor de Áudio Distribuído-SMmD (SAD-SMmD) apresentado neste artigo. Utilizando-se da plataforma ANSAware [APM 1993] é usado para armazenar, recuperar, editar e apresentar a monomídia áudio em um ambiente distribuído colaborativo.

Entre as principais características do SAD-SMmD estão as facilidades para a manipulação do áudio e a capacidade de executar edições dinâmicas sobre o áudio manipulado. Entende-se por edições dinâmicas a capacidade de modificar a maneira como o áudio é apresentado sem modificar os arquivos físicos de áudio utilizados. As funções de edição dinâmica incluem a inserção de intervalos de silêncio entre trechos de áudio e um mecanismo de mistura (mixing) de diferentes trechos de áudio em tempo real.

2. Servidores de Áudio Distribuídos - SADs

Sistemas de áudio em ambiente distribuído colaborativo tem sido objeto de interesse de diversos grupos de pesquisa em universidades e empresas. O objetivo desses sistemas é apresentar ferramentas para a manipulação e gerenciamento de arquivos de áudio digital e disponibilizar uma API (Application Program Interface) básica para interações com aplicações. Diversos sistemas de áudio para ambientes distribuídos têm sido desenvolvidos. Um sistema conhecido é o *Etherphone* [Terry & Swinehart 1988] que fornece facilidades para gerenciar e compartilhar dados de áudio entre múltiplas workstations e múltiplas redes. Diversos projetos estenderam os princípios do *Etherphone* para incorporar aplicações em vídeo [Furht & Milenkovic 1995]. Outro sistema que apresenta características semelhantes é o *SuitSound* [Riedl & Mashayekhi 1993], que oferece ferramentas convencionais na construção de objetos de aplicações que incluem áudio. Um terceiro SAD, o desenvolvido por Li Ning [Ning 1991], é apresentado em seguida por ter sido objeto de estudo durante a fase de especificação do SAD-SMmD [Coutinho & Trevelin 1996].

2.1 O SAD de Li Ning

O sistema de áudio apresentado por Li Ning foi construído sobre a plataforma distribuída ANSAware. Foi desenvolvido para três diferentes plataformas, SUN-OS, HP e PC-LINUX. O sistema de áudio compõe interfaces de programação para o armazenamento, gerenciamento e para conexão no transporte de áudio. É composto de três módulos: *audio server* (AS), *audio storage server* (ASS) e *voice rope server* (VRS) [Ning 1991]. Eles se interrelacionam de acordo com o modelo cliente/servidor provida pela plataforma ANSAware. Os módulos do SAD de Li Ning são descritos abaixo:

- *Audio Server* (AS) desempenha basicamente as funções de ligar diferentes trechos de áudio e transferir dados de áudio de aplicações clientes para o dispositivo de áudio;
- *Audio Storage Server* (ASS) fornece principalmente as funções de gerenciamento e armazenamento de arquivos, tais como funções de *leitura* e *escrita* de arquivos de áudio através da rede;
- *Voice Rope Server* (VRS) fornece as facilidades de um gerenciador de banco de dados para o relacionamento lógico de diversos *ropes* de áudio. O conceito *rope de áudio* (voice rope) veio do artigo de [Terry & Swinehart 1988]. Um *rope* é uma seqüência de áudio armazenado com um determinado significado lógico, podendo ser constituído de vários arquivos de áudio ou mesmo de outros *ropes*.

2.2 Limitações

O VRS juntamente com o ASS formam uma hierarquia de dois módulos que fornecem todo o gerenciamento para o armazenamento, a recuperação e o compartilhamento de áudio. Embora eles dêem uma grande flexibilidade no manuseio de ropes, pois um rope pode consistir de inúmeros trechos de áudio em diferentes arquivos, o mesmo apresenta algumas limitações. Entre elas destaca-se a incapacidade de reprodução simultânea de diferentes ropes e também da incapacidade de inserção de intervalos de silêncios entre segmentos de áudio que formam um rope. Essas funcionalidades são muito importantes na composição de documentos multimídia e hipermídia onde o áudio deve se relacionar de forma sincronizada com outras mídias ou também com diversos outros tipos de interações. Essas duas características, entre outras funções não apresentadas pelo SAD de Li Ning, estão contidas no SAD-SMmD, que por sua vez objetiva oferecer serviços de mais alto nível que os de Li Ning, caracterizando-se como um servidor de áudio com edições dinâmicas. O SAD-SMmD é descrito na seção seguinte.

3. O SAD-SMmD

O SAD-SMmD é um sistema servidor de áudio distribuído projetado para construir aplicações multimídia. Ele oferece uma série de serviços de áudio que incluem o armazenamento, a recuperação, a edição e a apresentação de arquivos de áudio existentes. Através da indexação de arquivos de áudio disponíveis, pode-se criar, compor e misturar ropes. Uma seqüência armazenada de áudio é referida como um *rope de áudio* (voice rope) e pode-se editá-los, concatená-los, inserir intervalos de silêncio e inserir rótulos em pontos específicos. Construindo um novo rope a partir de outros existentes permite uma total abstração do segmento de áudio lógico manipulado com relação aos arquivos de áudio digital distribuídos pela rede. No contexto do projeto SMmD, o SAD-SMmD foi desenvolvido com o intuito de implementar um módulo de serviço de áudio para suporte a esta mídia, dentro de um ambiente de desenvolvimento de aplicações multimídia e hipermídia. Ele conta hoje com diversas aplicações para ele sendo desenvolvidas, dentre as quais pode-se citar a implementação de uma secretária eletrônica na rede e um serviço de correio eletrônico falado. Uma versão do SAD-SMmD também está sendo desenvolvida para atuar no ambiente Java e disponibilizar os ropes na Web.

3.1 Estrutura

Baseado na estrutura proposta no SAD de Li Ning, o SAD-SMmD foi concebido possuindo as seguintes unidades básicas: ASPU, AES, HLVRs e APLICAÇÃO. As características de cada módulo e seus relacionamentos são descritos a seguir:

ASPU - Audio Server Presentation Unit é um módulo local (isto significa que deve ser *processo* da própria máquina onde se deseja produzir som) que controla os canais de comunicação de dados que emitem e recebem sinais de áudio através do dispositivo de áudio. Ele oferece 64 kbps (padrão PCM) tanto no formato μ -low como no A-low. Os dados de áudio são enviados em pacotes chamados *spurts*. Cada spurt tem a quantidade de 512 bytes transmitidos numa taxa de 16 spurts por segundo. Toda a comunicação é feita via os mecanismos de comunicação do ANSAware 4.1.

A forma como o ASPU disponibiliza seus serviços se dá através da *Linguagem de Definição de Interface* (IDL) fornecida pelo ANSAware. Parte da interface da unidade de apresentação é mostrada a seguir

```

ASPU: INTERFACE =
BEGIN
Spurt :      OPERATION [
                in_status:      Boolean;
                in_delay:      CARDINAL;
                in_samples:     SEQUENCE OF OCTET ];
            RETURNS [
                out_status:     Boolean;
                out_delay:      CARDINAL;
                out_samples:     SEQUENCE OF OCTET ];
END.

```

A interface ASPU, no processo de compilação do ANSAware, indica para o cliente (o módulo AES) as operações que podem ser acessadas. A operação *Spurt*, única nesta interface, cuida do estabelecimento de um canal de comunicação dos seus clientes com o servidor ASPU para receber ou enviar os pacotes de áudio para o dispositivo de áudio local.

AES - *Audio Editing Server* é o módulo responsável pelo armazenamento e recuperação dos arquivos de áudio através do sistema de arquivos NFS-Unix e pelas edições dinâmicas dos segmentos de áudio que compõem um *rope*. Esse módulo pode estar localizado em qualquer ponto da rede, ou seja, não precisa necessariamente ser um *processo* na máquina onde está a aplicação. Os serviços básicos oferecidos por esse módulo são o de leitura, escrita, composição e edição de arquivos de áudio. Ele utiliza os serviços do módulo ASPU para a apresentação, com possibilidades de mistura e inserção de intervalos de silêncio, e gravação dos sinais de áudio. Outras funções básicas como cancelamento de operações, iniciação do módulo, etc., também estão implementadas. O AES é transparente para o usuário pois é apenas usado pelo módulo HLVRS. A sessão 4 descreverá as funções de áudio que o HLVRS disponibiliza;

HLVRS - *High Level Voice Rope Server* é um servidor de *ropes* apresentando serviços de alto nível. Ele disponibiliza para aplicação quase todos os serviços de áudio, sendo portanto o módulo que interage diretamente com a aplicação, fazendo uma ponte entre esta e o AES;

Aplicações basicamente fazem uso da API disponível pelo SAD-SMmD, podendo, contudo, vir a oferecer seus próprios serviços. Isso acontece quando a aplicação implementa os serviços da *interface callback*. A *interface callback* é um sub-módulo que fica em background verificando os rótulos de um *rope* para notificar a aplicação. Isto possibilita que a aplicação execute algum procedimento quando um ponto específico do *rope* for exibido, como, por exemplo, ativar âncoras em um hipertexto.

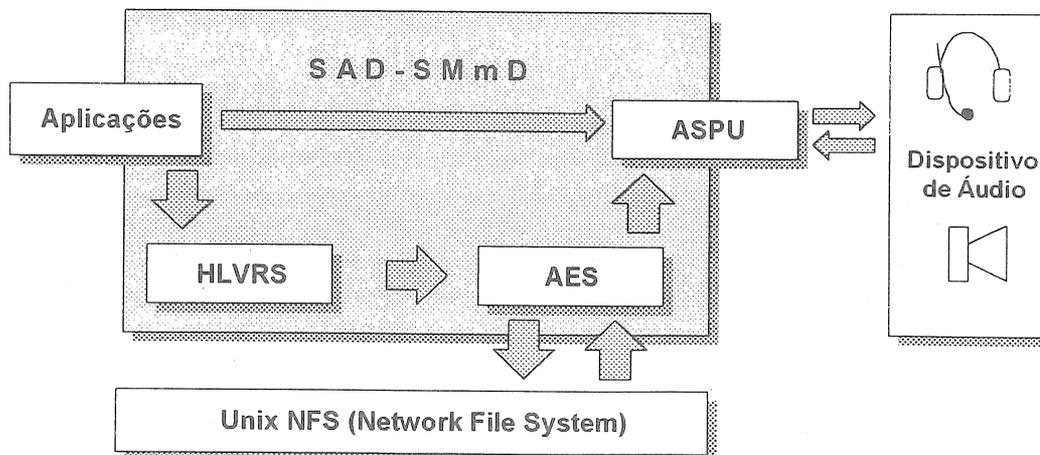


Figura 1: A estrutura do SAD-SMmD: ASPU, AES, HLVRS e Aplicações

3.2 Vantagens

Dentre os vários sistemas de áudio distribuídos para ambiente colaborativo existentes, o SAD-SMmD diferencia-se muito por estar construído sobre uma plataforma distribuída aberta bem completa, o ANSAware 4.1, com ferramentas próprias para a construção e gerenciamento de aplicações. Pode-se citar as seguintes vantagens do ambiente SAD-SMmD:

- o processamento do áudio fica separado das comunicações de rede, dando maior eficiência ao sistema;
- pouco esforço é necessário para incorporar áudio em um ambiente de aplicações multimídia genérico com as interfaces de programação oferecidas;
- O ANSAware permite que o sistema de áudio possa ser usado em WANs (wide area networks) por permitir serviço de “trading” e federação. O ANSAware pode ser instalado numa grande quantidade de hardwares, RISC e computadores pessoais com sistemas operacionais UNIX, VMS ou MS-DOS;
- O ANSAware, como uma plataforma de processamento distribuído aberto, oferece ao programador uma série de ferramentas de programação e bibliotecas que permitem a construção de sistemas distribuídos. A capacidade de *multi-thread*, suporte de sincronização e invocação de interface dinâmica são extremamente úteis para a implementação de um sistema de áudio distribuído.

4. As Funções do SAD-SMmD

As funções do SAD-SMmD podem ser divididas em funções herdadas, modificadas e novas funções em relação ao SAD de Li Ning. Dentre as funções herdadas estão as operações de escrita (gravação) em arquivos de áudio, operações para criar ropes, listá-los e deletá-los, etc. As principais funções como ler e reproduzir um rope, assim como toda a manipulação de segmentos de áudio na construção e edição de ropes foram modificadas para suportarem o mecanismo de mistura e inserções de silêncio entre ropes. Como novas funcionalidades tem-se incorporado no HLVRs os serviços de avanço, retrocesso, reprodução do segmento próximo ou anterior de um rope, etc.

Nesta seção será mostrado principalmente as funções responsáveis pelo mecanismo de mistura e inserções de silêncio entre ropes. O primeiro serviço permitirá que se faça uma mistura dinamicamente, isto é, sem a necessidade de fazer uma edição anterior, o que envolve gasto de tempo para a sua criação, o surgimento de um novo arquivo de áudio e até o uso de outros softwares para a realização da mistura. O segundo serviço, por introduzir intervalos de silêncio dinamicamente em pontos específicos nos ropes, dará grande flexibilidade de sincronismo e interação assim como uma economia de espaço considerável em disco por não representar o silêncio fisicamente nos arquivos.

A figura 2, a seguir, mostra as estruturas mantidas na memória principal para que o SAD-SMmD possa efetuar a maioria de suas operações. As tabelas VRH (voice rope head), VRT (voice rope table), VRS (voice rope segment) e VLT (voice label table) fornecem as informações necessárias para a reprodução, gravação, edição e remoção de ropes. Na tabela VRH encontra-se o cabeçalho de cada rope. Esta contém dois ponteiros, um para a tabela VRT e outro para a tabela VLT, que armazena os rótulos em pontos específicos (offset) do rope. A tabela VLT, por sua vez, é uma tabela de segmentos que possui um ponteiro para o próximo da lista e para a tabela VST, que é a tabela de localização de todos os segmentos de áudio da base de dados distribuída, acessível via NFS.

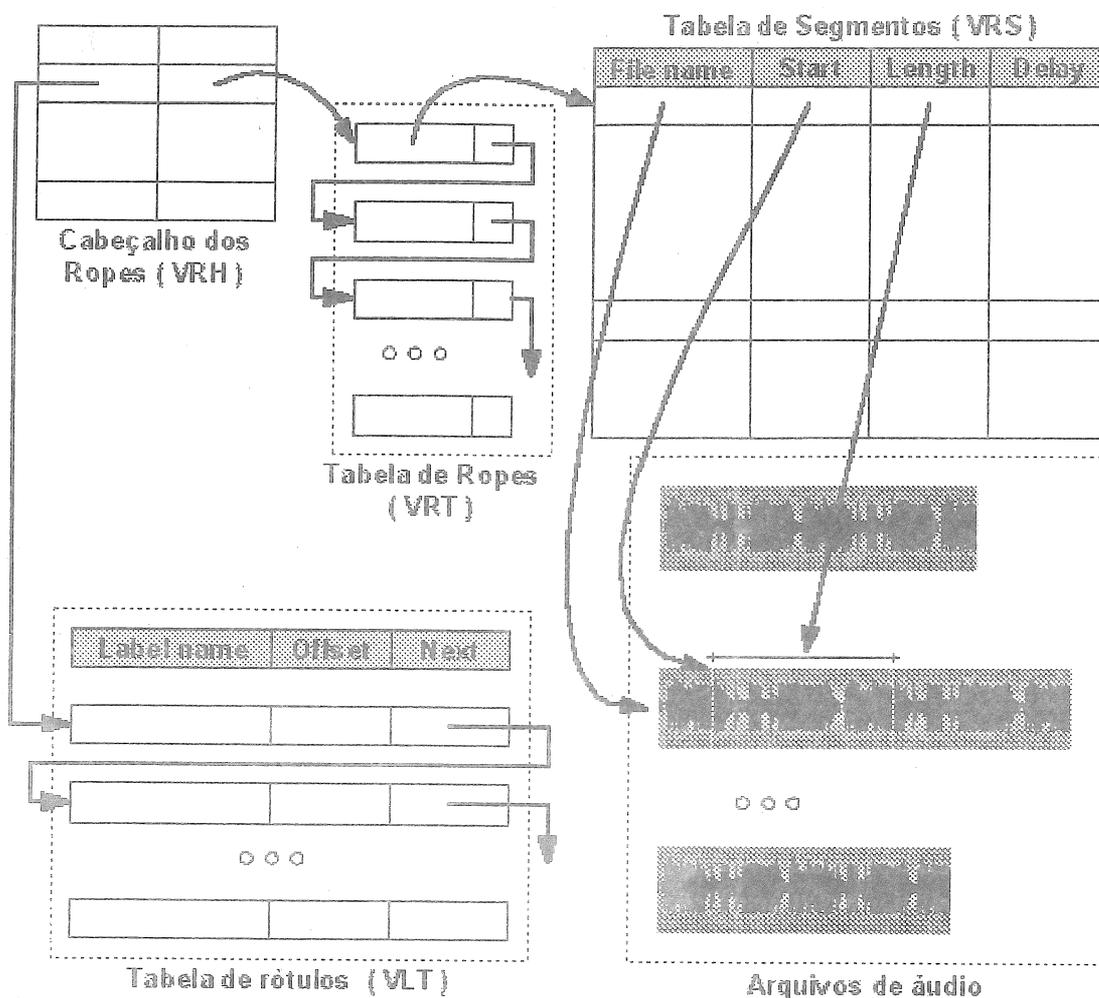


Figura 2: Estruturas de dados usadas no SAD-SMmD

Cada segmento de áudio registrado na VST possui o nome do arquivo físico onde seus dados encontram-se, o ponto de início do segmento no arquivo e seu comprimento. Arquivos físicos de áudio são implementados como ropes de segmento único. Dessa forma, um rope é sempre construído a partir de ropes existentes, numa estrutura única e simples de implementação e manipulação. Obtendo os segmentos desejados de cada rope e eventualmente inserindo intervalos de silêncio entre esses segmentos, através do atributo *delay* na Tabela de Segmentos (VST), faz-se, por exemplo, uma nova trilha sonora sem a necessidade de criar novos arquivos, ou até mesmo que esses silêncios inseridos tornem-se dados de áudio ocupando espaço. Quando nenhum intervalo for desejado (apresentação contínua), o atributo *delay* deve possuir o valor zero.

4.1 Misturando Ropes Dinamicamente

As funções de baixo nível responsáveis pela saída de áudio (output function) não possuem suporte para a saída simultânea de múltiplos canais. Para realizar a tarefa de mistura tem-se então que fornecer ao dispositivo de saída um novo rope que é a mistura equalizada de vários ropes. Ropes são essencialmente vetores de amostras de sinais de áudio (ondas). Para misturá-los os valores das amostras são somadas

juntas dentro de um buffer destino e submetidos para a saída de áudio. Esse buffer destino, que constituirá um novo rope (*MixedRope*) é tipicamente de pequeno tamanho para a obtenção da mixagem em tempo real. Esse tamanho típico é da ordem de 1/8 segundos.

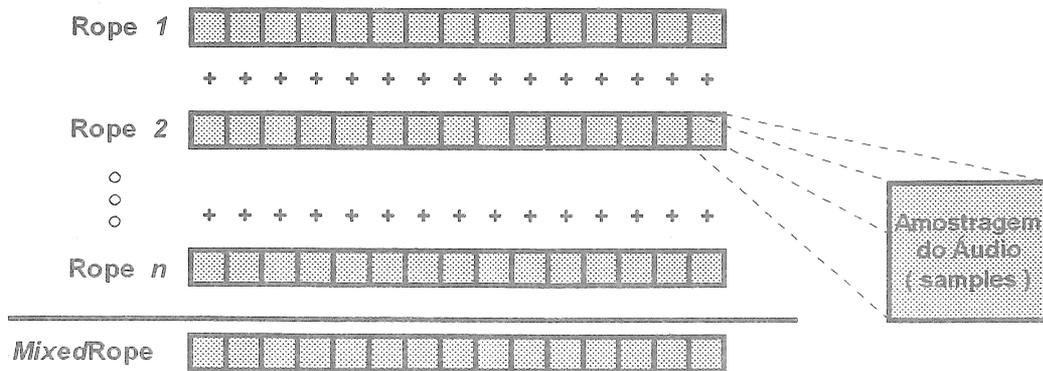


Figura 3: A Mistura de ropes

A figura acima exemplifica a mistura em tempo real fornecida pelo SAD-SMmD. O rope resultante da mistura entre ropes (*MixedRope*) é implementada como a seguir:

$$\text{MixedRope}[i] = \text{Rope1}[i] * \text{fact1} + \text{Rope2}[i] * \text{fact2} + \dots + \text{RopeN}[i] * \text{factN}$$

Cada rope a ser misturado, mais precisamente os segmentos que o compõem, são multiplicados por um fator (*fact1...factN*) que identificará a amplitude desejada para a mistura, isto é, a intensidade de volume específica dos segmentos do rope. Isso permite uma mistura mais equalizada entre diversos ropes, destacando-se convenientemente um determinado rope entre os outros. Por exemplo, uma fala, em dado instante, pode ser reproduzida junto com uma música de fundo que seja tocada com seu volume mais baixo devido ao decréscimo de seu respectivo fator de amplitude. Ao terminar a fala, a mesma música pode ter seu volume aumentado de volta restabelecendo seu fator.

O módulo que implementa o mecanismo de mistura dinâmica, assim como a introdução de silêncios, é o Audio Editing Server (AES). Deve ser lembrado que este módulo é transparente para a aplicação. Nesta interface manipula-se os segmentos de áudio físico, sendo, portanto, o local onde a mistura é realizada. A operação *AesMixedRead* lê um ou mais intervalos de arquivos de áudio, faz a mistura entre eles e reproduz invocando a unidade de apresentação ASPU. Essa mixagem é feita dinamicamente em buffer temporário e pode ser eventualmente gravada em arquivo através do parâmetro *to_filename*. Além da reprodução e mistura dos segmentos de áudio, a operação *AesMixedRead* pode mostrar os níveis dos sinais de áudio em uma interface gráfica do usuário por invocar a interface *display_interface_ref*. A operação *AesWrite* obtém os dados de áudio vindos do dispositivo de áudio e os grava formando um arquivo de áudio. Pelo parâmetro *interval*, *interval.start* recebe sempre o valor 0, que significa que o áudio é sempre gravado a partir do começo do arquivo. Já o parâmetro *interval.length* pode receber um valor arbitrário, o qual refere-se ao comprimento do arquivo. Ambas operações retornam *status* caso bem sucedidas ou não.

O código a seguir é da interface (IDL) AES, através da qual são definidos os tipos de dados utilizados e as operações de leitura e escrita dos arquivos de áudio, sendo omitidas demais operações como *cancelar operações*.

```
Aes: INTERFACE =
BEGIN
-- Type signatures
Interval:      TYPE = RECORD [ start:      LONG INTEGER,
                               length:     LONG INTEGER,
                               delay:      LONG INTEGER];

Intervals      TYPE = SEQUENCE OF Interval;
Segment:       TYPE = RECORD [ filename:  STRING,
                               interval:  Interval,
                               factor:    INTEGER ];

Segments:      TYPE = SEQUENCE OF Segment;
MixedSegments: TYPE = SEQUENCE OF Segments;
LabelPosition: TYPE = RECORD [ offset:   LONG INTEGER,
                               action:   STRING ];

LabelPositions: TYPE = SEQUENCE OF LabelPosition;
MixedLabelPos   TYPE = SEQUENCE OF LabelPositons;
Files           TYPE = SEQUENCE OF STRING;
AesStatus:      TYPE = { OpSuccess, OpFailure, NoMemory };

-- Operation signatures
AesMixedRead:   OPERATION [
audio_interface_ref:  ansa_InterfaceRef;
display_interface_ref: ansa_InterfaceRef;
display_scan_rate:   INTEGER;
files:               Files;
intervals:           Intervals;
mixed_label_pos:     MixedLabelPos;
in_rop:              INTEGER;
callback_interface_ref: ansa_InterfaceRef;
to_filename:         STRING;
preload:             INTEGER;
speed:               INTEGER ];
RETURNS[ AesStatus ];

AesWrite:       OPERATION [
audio_interface_ref:  ansa_interfaceRef;
file_name:           STRING;
interval:            Interval;
callback_interface_ref: ansa_InterfaceRef];
RETURNS[ AesStatus ];

END.
```

Parte da interface High Level Voice Rope Server (HLVRS) é apresentada a seguir, particularmente com as definições dos tipos usados e da função para a mixagem (operação `HLVrsMixedPlay`):

```

HLVrs: INTERFACE =
NEEDS Aes;
BEGIN
-- Type signatures
  RopeSegment:      TYPE = RECORD [   ropename:  STRING,
                                     interval:  Interval,
                                     factor:    INTEGER ];
  RopeSegments      TYPE = SEQUENCE OF RopeSegment;
  MixedRopeSeg      TYPE = SEQUENCE OF RopeSegments;
  Label:            TYPE = RECORD [   labelname: STRING,
                                     offset:    LONG INTEGER ];
  Labels:           TYPE = SEQUENCE OF Label;
  MixedLabels:      TYPE = SEQUENCE OF Labels;
  LabelEvent: TYPE = RECORD [   labelname:  STRING,
                               action:     STRING ];
  LabelEvents:      TYPE = SEQUENCE OF LabelEvent;
  MixedLabelEvents TYPE = SEQUENCE OF LabelEvents;
  Ropes:            TYPE = SEQUENCE OF STRING;
  HLVrsStatus:      TYPE = [ hlvrs_OpSuccess, hlvrs_OpFailure,
hlvrs_NoMemory , ...
-- Operation signatures
  HLVrsMixedPlay OPERATION [
    audio_interface_ref:  ansa_InterfaceRef;
    display_interface_ref: ansa_InterfaceRef;
    display_scan_rate:    INTEGER;
    ropes:                Ropes;
    mixed_play_range:     Intervals;
    mixed_label_events:   mixedLabelEvents;
    callback_interface_ref: ansa_InterfaceRef;
    to_filename:          STRING;
    preload:              INTEGER;
    speed:                INTEGER ];
    RETURNS [ HLVrsStatus ];
END.

```

Primeiramente, através da descrição *NEEDS Aes* na definição da interface *HLVRS*, tem-se que a mesma herda os tipos de dados declarados na interface *Aes*. Os parâmetros *audio_interface_ref*, *display_interface_ref*, *display_scan_rate*, *callback_inteface_ref*, *to_filename*, *preload* e *speed* possuem as mesmas semânticas e uso como na operação *AesMixedRead* descrita atrás.

Na visão do usuário, o mecanismo de mistura existe para reproduzir simultaneamente ropes completos. Para isso é passado como parâmetro em *HLVrsMixedPlay* uma seqüência de ropes e as respectivas posições de início e comprimento para a mistura (*ropes* e *mixed_play_range*).

5. Conclusões

Um servidor de áudio para um ambiente distribuído de aplicações multimídia foi apresentado. O SAD-SMmD apresentou novas e importantes funções agregadas aos serviços de áudio típicos dos servidores existentes. A edição dinâmica de rofes permitindo funções de inserção de silêncio e mixagem em tempo real, no instante da apresentação fornece grande flexibilidade ao servidor para utilização em aplicações multimídia.

A não necessidade do armazenamento do resultado de uma pré-mixagem de rofes caracteriza-se como uma grande vantagem desse servidor. A inclusão de intervalos de silêncio, da mesma forma, permite a composição sincronizada e bastante fácil de documentos multimídia, sem que o usuário tenha que programá-la na aplicação.

A utilização da plataforma ANSAware como base para desenvolvimento dos módulos que compõem o SAD-SMmD permitiu o desenvolvimento de um servidor para sistemas multimídia distribuídos, como é o caso do ambiente do projeto SMmD. Através de técnicas de resolução de nomes e endereços entre diferentes plataformas (Ferreira et ali. 1997), é possível a utilização desses serviços por aplicações escritas em outras plataformas, tais como em CORBA ou Java, como se vem trabalhando no âmbito do projeto SMmD.

Agradecimentos

Este trabalho contou com o apoio do CNPq, nos programas ProTem-CC2, para o projeto SMmD (proc. 680077/94-4), e PIBIC/CNPq para bolsa de Iniciação científica.

Referências

- [APM 1993] ANSAware 4.1: *Application Programming in ANSAware*; Document RM.102.02; Architecture Projects Management Limited; February 1993.
- [Coutinho & Trevelin 1996] Coutinho, R. e Trevelin, L.C.; *SAD-SMmD: Um Servidor de Áudio Distribuído para a Arquitetura SMmD*; Relatório de Projeto; Universidade Federal de São Carlos-UFSCar; 1995.
- [Ferreira et al. 1997] Ferreira, M. M.; Salvador M. R.; Gonçalves A. M. e Trevelin, L.C.; *Achieving Interoperability Between ANSAware and CORBA*; submetido ao CLEI, 1997.
- [Furht & Milenkovic 1995] Furht, B and Milenkovic, M.; *A Guided Tour of Multimedia Systems and Applications*; IEEE Computer Society Press; 1995.
- [Ning 1991] Ning, Li; *A Distributed Audio System*; internal report of Computing Laboratory; University of Kent; Canterbury, UK; 1991.
- [OMG 1995] Object Management Group; *The Common Object Request Broker: Architecture and Specification*; Revision 2.0, July 1995.
- [OSF 1994] Open Software Foundation; *OSF DCE Application Development Guide*; 1994.
- [Riedl & Mashayekhi 1993] Riedl, J. and Mashayekhi, V.; *SuitSound: A System for Distributed Collaborative Multimedia*; IEEE Transactions; Knowledge and Data Engineering; 1993.
- [Terry & Swinehart 1988] Terry, D.B. and Swinehart, D.C.; *Managing Stored Voice in the Etherphone System*; ACM Transactions; Computer Systems; 1988.
- [Trevelin 1995] Trevelin, L.C.; *O Projeto SMmD: uma visão geral*; Anais do I Workshop de Sistemas Hiperídia Distribuídos; São Carlos-SP; Página 26; Julho 1995.