

**UNIVERSIDADE DE SÃO PAULO**

**Sincronização Multimídia:  
Aspectos Básicos**

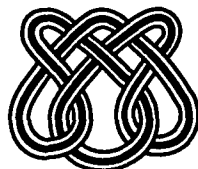
**Laércio Augusto Baldochi Júnior  
Maria da Graça Campos Pimentel**

**Nº 48**

---

**NOTAS DIDÁTICAS**

---



**Instituto de Ciências Matemáticas de São Carlos**

UNIVERSIDADE DE SÃO PAULO  
Instituto de Ciências Matemáticas e de Computação  
ISSN 0103-2585

**Sincronização Multimídia:  
Aspectos Básicos**

**Laércio Augusto Baldochi Júnior  
Maria da Graça Campos Pimentel**

**Nº 48**

NOTAS DIDÁTICAS



São Carlos – SP  
Nov./2000

# **Sincronização Multimídia: Aspectos Básicos**

Laércio Augusto Baldochi Júnior  
Maria da Graça Campos Pimentel

Instituto de Ciências Matemáticas e de Computação  
Universidade de São Paulo  
Caixa Postal 668 - São Carlos-SP - 13560-970

## Sumário

<b>SUMÁRIO</b> .....	<b>I</b>
<b>LISTA DE FIGURAS</b> .....	<b>II</b>
<b>1. INTRODUÇÃO</b> .....	<b>1</b>
<b>2. MULTIMÍDIA: DEFINIÇÕES E CONCEITOS BÁSICOS</b> .....	<b>2</b>
2.1. CONSIDERAÇÕES INICIAIS .....	2
2.2. CLASSIFICAÇÃO DE MÍDIAS .....	2
2.3. REPRESENTAÇÃO ESPACIAL DE MÍDIAS .....	4
2.4. SISTEMAS MULTIMÍDIA .....	5
2.5. REQUISITOS PARA UM SISTEMA DE APRESENTAÇÃO MULTIMÍDIA .....	6
2.6. QUALIDADE DE SERVIÇO EM SISTEMAS MULTIMÍDIA DISTRIBUÍDOS .....	7
2.7. CONSIDERAÇÕES FINAIS .....	9
<b>3. SINCRONIZAÇÃO MULTIMÍDIA</b> .....	<b>10</b>
3.1. CONSIDERAÇÕES INICIAIS .....	10
3.2. NOÇÕES BÁSICAS DE SINCRONIZAÇÃO .....	11
3.2.1. <i>Relações de Sincronização</i> .....	11
3.2.2. <i>Sincronização Intra e Interobjeto</i> .....	12
3.2.3. <i>Unidades Lógicas de Dados</i> .....	13
3.2.4. <i>Especificação de Dependências Temporais</i> .....	15
3.3. REQUISITOS DE APRESENTAÇÃO .....	17
3.3.1. <i>Gap Problem</i> .....	17
3.3.2. <i>Sincronização Labial e Sincronização de Ponteiros</i> .....	19
3.4. MODELOS DE REFERÊNCIA PARA SINCRONIZAÇÃO MULTIMÍDIA .....	21
3.4.1. <i>A Camada de Mídia</i> .....	22
3.4.2. <i>A Camada de Stream</i> .....	23
3.4.3. <i>A Camada de Objeto</i> .....	24
3.4.4. <i>A Camada de Especificação</i> .....	25
3.5. MÉTODOS DE ESPECIFICAÇÃO DE SINCRONIZAÇÃO MULTIMÍDIA .....	26
3.5.1. <i>Especificações Baseadas em Intervalos</i> .....	26
3.5.2. <i>Especificação de Sincronização Baseada em Eixos</i> .....	29
3.5.3. <i>Especificação de Sincronização Baseada em Controle de Fluxo</i> .....	30
3.5.4. <i>Especificação de Sincronização Baseada em Eventos</i> .....	31
3.6. SINCRONIZAÇÃO EM AMBIENTES DISTRIBUÍDOS .....	32
3.7. QUALIDADE DE SERVIÇO .....	33
3.8. CONSIDERAÇÕES FINAIS .....	34
<b>4. CONCLUSÃO</b> .....	<b>35</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	<b>36</b>

---

## Lista de Figuras

---

FIGURA 3.1 - SINCRONIZAÇÃO INTRA-OBJETO ENTRE FRAMES DE UMA SEQUÊNCIA DE VÍDEO [SN95] .....	13
FIGURA 3.2 - EXEMPLO DE SINCRONIZAÇÃO INTEROBJETO [SN95] .....	13
FIGURA 3.3 - VISÃO DAS ULDs DE UMA SINCRONIZAÇÃO LABIAL [SN95] .....	15
FIGURA 3.4 - SOLUÇÃO PARA O GAP PROBLEM ATRAVÉS DO MÉTODO DE BLOQUEIO RESTRITIVO [SN95] .....	18
FIGURA 3.5 - MODELO DE REFERÊNCIA DE QUATRO CAMADAS [MES93].....	22
FIGURA 3.6 - RELAÇÕES TEMPORAIS BINÁRIAS [ALL83].....	27
FIGURA 3.7 - OPERAÇÕES DO MODELO AVANÇADO DE SINCRONIZAÇÃO BASEADO EM INTERVALOS [WR94].....	28

---

## 1. Introdução

---

Na última década houve um crescimento explosivo no uso de dados e aplicações multimídia. Hoje em dia, computadores e redes processam e transmitem muito mais que apenas texto e imagens estáticas. Mídias contínuas, como vídeo e áudio, juntamente com mídias discretas, como gráficos, se tornaram parte de aplicações integradas de computador [Buf94].

As novas aplicações que estão surgindo fazem crescer a cada dia a necessidade de distribuição de dados e de processamento. Em função dessa distribuição, e do conseqüentemente desacoplamento dos dados, os aspectos relacionados à sincronização multimídia se tornam cada vez mais complexos e sujeitos a fatores externos, tais como atraso e *jitter* gerados na transmissão das mídias que compõem uma apresentação multimídia.

Este documento procura abordar todos os aspectos relacionados à sincronização dos elementos envolvidos em uma apresentação multimídia. Inicialmente, no Capítulo 2, é feito um estudo no sentido de identificar e classificar os diversos tipos de mídia existentes, buscando, com isso, caracterizar um sistema multimídia, enfatizando seus requisitos básicos. A partir do conhecimento dos principais tipos de mídia é possível determinar seus possíveis relacionamentos, o que possibilita a definição de métodos de especificação de sincronização, que serão abordados em profundidade no Capítulo 3.

Uma outra preocupação desse trabalho está no estudo de modelos de referência para apresentação multimídia. Os principais modelos utilizados, tais como o modelo de camadas proposto por *Meyer, Effelsberg & Steinmetz* [MES93] serão apresentados e discutidos. Finalmente, os problemas de sincronização e a questão da qualidade de serviço em ambientes distribuídos serão abordados, fechando assim os principais temas relacionados à sincronização de apresentações multimídia.

---

## 2. Multimídia: Definições e Conceitos Básicos

---

### 2.1. Considerações Iniciais

A fim de capacitar o leitor para o devido entendimento da sincronização multimídia, tema central desse trabalho, é necessário definir o que é multimídia e apresentar uma série de conceitos e requisitos relacionados aos sistemas de intercâmbio e apresentação multimídia.

Além da própria definição da palavra multimídia, este capítulo apresenta os vários tipos de mídia existentes e seus critérios de classificação. Além disso, são também abordadas as principais características de um sistema multimídia, dando ênfase aos requisitos necessários a este tipo de sistema, como o suporte a programação *multithreading*, por exemplo. Finalizando, o capítulo aborda os problemas relacionados ao intercâmbio de mídia contínua nos sistemas multimídia distribuídos, destacando a importância dos parâmetros de qualidade de serviço no gerenciamento desses problemas.

### 2.2. Classificação de Mídias

Existem várias formas de se definir a palavra multimídia. De acordo com as definições do latim, pode-se dividir a palavra em dois radicais:

- Multi: muitos; múltiplos.
- Mídia (*medium*): meio; substância ou matéria através da qual alguma coisa é transmitida; meio de comunicação de massa, como jornal, revista ou televisão (do *American Heritage Dictionary*, 1991).

A descrição acima é derivada das formas comuns de interação humana, não sendo, portanto, muito exata, devendo ser adaptada ao processamento computacional. Em computação, mídia é normalmente descrita como sendo um meio de distribuição e apresentação de informação. Exemplos de mídia são texto, gráfico, voz, música, etc. Uma mídia pode ser classificada de acordo com diferentes critérios [SN95], entre eles:

- **Percepção:** Este critério envolve a forma na qual os humanos percebem a informação num ambiente computacional. Uma vez que as formas mais comuns de percepção são a visão e a audição, de acordo com este critério as mídias se dividem em mídias visíveis (texto, imagens e vídeo) e audíveis (música, barulho, voz);
- **Representação:** Aborda a forma na qual as mídias são representadas internamente pelos computadores. Nessa abordagem, os formatos de representação usados servem para classificar as mídias. A codificação ASCII, por exemplo, é usada para representar caracteres de um texto e o formato JPEG é usado para codificar uma imagem;
- **Apresentação:** Refere-se ao uso de ferramentas e dispositivos na entrada e saída de informações. Neste critério, o mais importante é definir através de qual mídia uma informação entra ou sai do computador. Assim, as mídias papel, tela (monitor de vídeo) e alto falantes são usados na saída de informações, enquanto que o teclado, o *mouse*, a câmera e o microfone são usados na entrada de dados;
- **Armazenamento:** Refere-se aos dispositivos usados no armazenamento de informações. No entanto, estes dispositivos não se limitam apenas aos componentes de um computador. Assim, uma folha de papel pode ser considerada uma mídia de armazenamento. Entre as principais mídias de armazenamento estão o disco rígido e o CD-ROM;
- **Transmissão:** Aborda os meios de comunicação (mídias) que possibilitam a transmissão de dados contínuos. Estes meios seriam as redes de computadores, compostas basicamente de fios e cabos de transmissão, como o cabo coaxial e as fibras óticas, além, é claro, do ar — meio de transmissão das comunicações sem fio (*wireless communications*);
- **Intercâmbio de informação:** Envolve todas as mídias usadas na transmissão de dados, ou seja, todas as mídias de armazenamento e transmissão de informações. Esta classificação é usada porque o processo de transmissão de dados muitas vezes usa mídias de armazenamento temporário, que podem ser transportadas para fora de uma rede de computadores até o



destinatário através da transmissão direta entre diversas redes, ou através do uso conjunto de mídias de armazenamento e transmissão, como ocorre no sistema de correio eletrônico.

### 2.3. Representação Espacial de Mídias

Uma característica importante inerente a uma mídia é sua representação espacial. Cada espaço de representação consiste de uma ou mais dimensões de representação. Assim, a tela de um computador possui duas dimensões, não sendo, portanto, adequada para a representação de holografia, pois esta requer uma dimensão a mais. O tempo pode ocorrer dentro de cada espaço de representação como uma dimensão adicional, dividindo as mídias em dois tipos:

- Mídias não dependentes de tempo (discretas): são mídias nas quais a informação consiste exclusivamente de uma seqüência de elementos individuais ou de um *continuum* sem o componente de tempo. Exemplos de mídias discretas são textos e gráficos. O processamento de mídias discretas deve acontecer o mais rápido possível. No entanto, este processamento não é crítico em relação ao tempo uma vez que a validade e a corretude dos dados não depende de condições temporais.
- Mídias dependentes de tempo: são mídias que têm seus valores modificados com o tempo. Neste tipo de mídia, a informação é expressa não somente por valores individuais, mas também pela ordem de ocorrência desses valores. Exemplos desse tipo de mídia são som e vídeo *full-motion*. O processamento dessas mídias é crítico em relação ao tempo porque a validade e a corretude dos dados depende de condições temporais.

As mídias dependentes de tempo são subdivididas em mídias contínuas e não contínuas. Áudio e vídeo são classificados como mídias contínuas porque são constituídos de elementos discretos (*frames* e *samples*) que ocorrem de forma periódica no tempo. Dessa forma, valores de representação que ocorrem de forma aperiódica, como comandos de controle em um sistema de tempo real, não são caracterizados como mídia contínua, mesmo sendo dependentes de tempo.

## 2.4. Sistemas Multimídia

Baseado na definição de multimídia fornecida pelo *American Heritage Dictionary*, um sistema multimídia seria qualquer sistema que suportasse mais de um tipo de mídia. No entanto, esta caracterização baseada apenas em aspectos quantitativos é insuficiente para definir um sistema multimídia. Mais que quantidade, é importante avaliar a qualidade das mídias apresentadas por um sistema antes de dizer se este sistema é ou não multimídia [SN95]. Um sistema multimídia se distingue de outros sistemas através de quatro propriedades principais:

- **Combinação de mídias:** nem toda combinação arbitrária de mídias justifica o uso do termo multimídia. Este termo deve apenas ser usado quando se combinam mídias contínuas e discretas.
- **Independência entre mídias:** diferentes mídias devem ter um nível de independência satisfatório em relação umas às outras. Mídias como áudio e vídeo são, algumas vezes, coletadas de forma simultânea e mantidas numa mesma mídia de armazenamento. Um sistema não pode ser considerado multimídia somente por apresentar estas duas mídias, uma vez que entre elas existe um alto grau de dependência.
- **Integração de mídias suportada por computador:** um sistema multimídia deve ser capaz de combinar várias mídias de forma arbitrária. Para que isso seja possível, o sistema deve suportar relações de sincronização espacial, temporal e semântica. Assim, não basta que um sistema apresente várias mídias simultaneamente para que ele seja considerado multimídia, é necessário também que o sistema permita que estas mídias interajam entre si.
- **Comunicação:** um sistema multimídia deve ser capaz de se comunicar com outros sistemas. Como a maioria dos computadores atualmente estão interconectados, é interessante que o sistema suporte aplicações multimídia distribuídas, como, por exemplo, programas de vídeo conferência. O sistema multimídia deve permitir que uma informação possa ser não somente criada, processada, apresentada e armazenada, mas também distribuída através da rede. Sistemas com estas características são chamados sistemas multimídia distribuídos.

## 2.5. Requisitos para um Sistema de Apresentação Multimídia

A apresentação de uma aplicação multimídia requer a execução simultânea, sequencial e independente de dados heterogêneos. O problema da apresentação multimídia se assemelha muito ao problema da execução de *threads* sequenciais e paralelas em computação concorrente [Buf94]. Muitas linguagens concorrentes, como CSP e Ada, suportam o conceito de *multithreading*. No entanto, a solução presente nas linguagens concorrentes não se adequa ao problema da apresentação multimídia, uma vez que estas linguagens estão direcionadas para a solução de problemas que necessitam de um alto *throughput*, como a solução paralela para a inversão de matrizes. Por outro lado, a apresentação multimídia se preocupa com a apresentação coerente de mídias heterogêneas ao usuário. Assim, existe um limite na velocidade de fornecimento dos dados acima da qual o usuário não consegue assimilar o conteúdo da apresentação. Para os sistemas computacionais comuns é sempre desejável produzir uma solução no menor tempo possível. Os sistemas multimídia, por outro lado, devem priorizar a apresentação, não a computação.

Um outro fator que diferencia os sistemas concorrentes dos sistemas multimídia é a forma de tratar os eventos dependentes de tempo. Nos sistemas concorrentes existe uma grande preocupação com a ordem dos eventos que ocorrem em paralelo. No entanto, não existe preocupação com a taxa de execução desses eventos (a performance da CPU não é considerada). Os dados multimídia dependentes de tempo, por outro lado, devem obedecer limitações de precedência e tempo de execução, ou seja, se um dado multimídia deve ser apresentado por 10 segundos, apresentá-lo por 9 ou por 11 segundos não é permitido.

Um esquema de apresentação multimídia deve capturar a precedência dos componentes e obedecer as limitações de tempo impostas pelas aplicações. Uma metodologia de especificação para apresentações multimídia deve incluir a representação da semântica de tempo real e concorrência e também prover formas de modelagem hierárquica. A natureza da apresentação de dados multimídia também implica outros requerimentos, como a possibilidade de reverter uma apresentação, de permitir acesso aleatório a qualquer ponto da mesma, de especificar temporização de forma parcial ou incompleta, de permitir o compartilhamento de componentes

sincronizados entre aplicações e de prover meios de armazenagem de informações de controle [Buf94].

## 2.6. Qualidade de Serviço em Sistemas Multimídia Distribuídos

A importância da informação multimídia e da comunicação cresce com o aumento de consumidores potenciais dessa informação que estão conectados a alguma rede. Espera-se para um futuro próximo que os sistemas multimídia sejam acessíveis em larga escala e possam prover funções de distribuição similares à rede mundial de telefones e à *Internet*. No entanto, a difusão dos sistemas multimídia distribuídos esbarra nos problemas envolvidos no intercâmbio de mídia contínua [Buf94].

O termo mídia contínua, como já foi dito, refere-se à dimensão temporal de mídias como áudio e vídeo, nas quais a informação se resume a uma seqüência de amostras, cada uma com uma posição específica no tempo. Os requisitos de tempo envolvidos são reforçados durante o processo de apresentação e captura, quando os dados são visualizados pelos usuários. Além disso, dados de mídia contínua possuem geralmente um volume considerável. Estes dois fatores devem ser considerados no projeto de sistemas multimídia, diferenciando estes sistemas dos convencionais, sobretudo nos aspectos de gerenciamento de processos, arquivos e rede.

A fim de atender aos requisitos existentes no intercâmbio de mídia contínua, o gerenciamento de recursos, função principal de um sistema operacional, deve atender a uma nova classe de serviços, na qual restrições de tempo e níveis de serviço negociáveis devem ser satisfeitos. Assim, para a maioria das aplicações de mídia contínua é possível estabelecer níveis de serviço flexíveis, que variam de acordo com a carga do sistema. A flexibilidade do nível de serviço é estabelecida através de um processo de negociação entre a aplicação e o gerenciador de recursos local. O termo qualidade de serviço (*QoS - Quality of Service*) é usado para representar os requerimentos da aplicação a um dado recurso [Buf94].

Parâmetros típicos de qualidade de serviço são: resolução máxima e mínima, taxa de erros permitida, limites de atraso e *jitter* aceitáveis. Resolução máxima e mínima referem-se à

qualidade em que são exibidos objetos visíveis, como texto, imagem e vídeo. Taxa de erros está ligada aos protocolos de transporte de *streams* de vídeo e áudio. O parâmetro atraso refere-se ao tempo gasto no processo de intercâmbio de um dado multimídia entre a fonte e o destino. Por fim, *jitter* refere-se à variação desse atraso. Assim, se cada *frame* de um vídeo demora exatamente 10 ms entre a fonte e o destino, o *jitter* é igual a zero (não há variação no atraso). No entanto, devido a oscilações na carga da rede, quase sempre ocorrem variações no atraso durante o processo de transporte de um *stream*. A fim de oferecer uma boa qualidade de serviço na apresentação de uma mídia contínua, como áudio ou vídeo, o sistema multimídia deve tentar manter o valor do *jitter* constante e o mais próximo de zero possível.

O paradigma para gerenciamento de recursos no contexto de mídia contínua é baseado no fato de que algumas aplicações requerem garantias determinísticas de serviço, enquanto outras requerem garantias probabilísticas, ou ainda, garantias mínimas de serviço. Garantias determinísticas e probabilísticas podem ser asseguradas desde que fatores como carga do sistema, espaço de *buffer* e escalonamento de processos possam ser monitorados e controlados. O que diferencia as duas formas de garantia citadas é que as probabilísticas são mais flexíveis, permitindo que o gerenciador de recursos admita um maior volume de tráfego, o que leva a uma melhor utilização do sistema. Aplicações que requerem garantias de performance mínimas podem ser atendidas na política do melhor esforço (*best effort basis*).

O gerenciador de recursos do sistema controla os níveis de serviço oferecidos. Assim, para uma dada classe de serviços, o gerenciador executa um teste de admissibilidade quando um recurso é solicitado. O teste determina se uma tabela de serviços que atenda os requisitos dos clientes pode ser construída. Em caso afirmativo, a solicitação é atendida. Caso contrário, o gerenciador de recursos pode recomendar classes de serviço de mais baixo nível que possam estar disponíveis. O gerenciador de recursos mantém uma função de escalonamento através da qual os clientes admitidos são atendidos. Durante o ciclo de serviços, o gerenciador monitora o uso que cada aplicação faz dos recursos de modo a evitar que uma sobrecarga inesperada, gerada por um cliente, não cause a deteriorização dos serviços prestados a outros clientes.

A natureza temporal das amostras de áudio e vídeo digital requer que parâmetros de *QoS* como atraso e *jitter* variem o mínimo possível do ponto de geração ou recuperação até o ponto de apresentação. Este requerimento é conhecido como *sincronização intramídia*. Quando vários *streams* de mídia contínua são apresentados em paralelo, geralmente a partir de diferentes pontos de geração ou recuperação, problemas referentes a relacionamentos temporais acontecem. Faz-se necessário então, realizar um ajuste de temporização entre as mídias envolvidas. Este ajuste é conhecido como *sincronização intermídia*. Os aspectos envolvidos na sincronização de mídias inter-relacionadas serão discutidos no próximo capítulo.

## 2.7. Considerações Finais

Neste capítulo foram introduzidos conceitos básicos que serão de grande utilidade para o entendimento dos assuntos apresentados no restante deste documento. Vários critérios de classificação de mídias foram apresentados, dos quais vale lembrar aquele que classifica uma mídia de acordo com as relações temporais entre seus componentes, identificando as mídias que são dependentes de tempo e as que não são. A especificação dessas dependências recebe o nome de sincronização multimídia, assunto do próximo capítulo.

Além dos critérios de classificação de mídias, dois outros assuntos relevantes foram abordados neste capítulo. O primeiro refere-se aos sistemas multimídia e aos requisitos que tais sistemas devem possuir a fim de apresentar corretamente composições multimídia. O segundo refere-se aos problemas relacionados ao intercâmbio multimídia em sistemas distribuídos, abordando o uso de parâmetros de qualidade de serviço no sentido de estabelecer limites máximos para o atraso e o *jitter* no transporte de mídias contínuas.

---

## 3. Sincronização Multimídia

---

### 3.1. Considerações Iniciais

A palavra sincronização refere-se a tempo, e sincronização em sistemas multimídia refere-se às relações temporais entre os objetos de mídia do sistema. Estas relações envolvem objetos de mídia dependentes de tempo (mídias contínuas) e objetos de mídia não dependentes de tempo (mídias discretas). Um exemplo de sincronização de mídias contínuas é a sincronização entre a informação visual e acústica na televisão. Nos sistemas multimídia, uma sincronização semelhante deve ser provida para a apresentação de áudio e vídeo. Um exemplo de relações temporais entre mídias dependentes e não dependentes de tempo é o *slide show*, onde a apresentação de slides (mídia discreta) deve ser sincronizada com um áudio (mídia contínua) que comenta cada slide.

A sincronização de mídias é realizada por vários componentes de um sistema multimídia, como o sistema operacional, o sistema de comunicação, os gerenciadores de bancos de dados e as aplicações. Dessa forma, a sincronização deve ser analisada em vários níveis em um sistema multimídia [SN95].

O sistema operacional e as camadas básicas de comunicação manipulam *streams* de mídia com o objetivo de evitar a ocorrência de *jitter* na apresentação das unidades de cada *stream*. Imediatamente acima do sistema operacional existe uma camada que dá suporte em tempo de execução para a sincronização de múltiplos *streams* de mídia. O objetivo desta camada é manter as relações temporais entre vários *streams* que participam de uma mesma apresentação. Em seguida, existe a camada que dá suporte em tempo real à sincronização de mídias dependentes e não dependentes de tempo, além de também manipular a interação dos usuários. Finalmente, no topo, vem a camada de especificação, que permite a autoria de aplicações multimídia [MES93].

Este capítulo está organizado da seguinte forma: a Seção 3.2 introduz os conceitos básicos sobre sincronização multimídia; a Seção 3.3 aborda problemas relacionados à apresentação síncrona de

objetos de mídia inter-relacionados; a Seção 3.4 comenta modelos de referência para apresentação multimídia, dando maior ênfase ao modelo de camadas proposto por Meyer, Effelsberg & Steinmetz [MES93]; a Seção 3.5 trata dos métodos de especificação de sincronização multimídia; a Seção 3.6 aborda problemas de sincronização em ambientes distribuídos; a Seção 3.7 apresenta os parâmetros de qualidade de serviço (*QoS*) relacionados à sincronização multimídia; e, finalmente, a Seção 3.8 faz algumas considerações a respeito da importância dos assuntos abordados neste capítulo.

## 3.2. Noções Básicas de Sincronização

O processamento de mídias integradas é uma das principais características dos sistemas multimídia. A principal razão da necessidade de integração de mídias é a dependência inerente entre a informação codificada em diferentes objetos de mídia. Estas dependências devem ser mantidas durante os processos de armazenamento, manipulação, comunicação, captura, e sobretudo, apresentação desses objetos. A tarefa de manter as interdependências temporais entre objetos de mídia recebe o nome de sincronização multimídia [SN95].

Nesta seção são apresentados os conceitos básicos de sincronização, abordando as relações de sincronização existentes, as formas de especificação de dependências temporais entre objetos de mídia, as unidades lógicas de dados (ULDs - usadas na sincronização fina), e também a sincronização intra e interobjeto.

### 3.2.1. Relações de Sincronização

A palavra sincronização está diretamente ligada ao fator tempo. No entanto, a palavra sincronização refere-se não somente às relações temporais, mas também às relações espaciais e de conteúdo dos objetos de mídia [SN95].

As *relações de conteúdo* definem a dependência de objetos de mídia a algum dado. Por exemplo, o conteúdo de uma planilha pode estar associado a um gráfico que representa seus dados. Dessa forma, qualquer mudança nos dados da planilha implica uma mudança no gráfico exibido.



Documentos com esta característica são chamados documentos multimídia integrados. A sincronização multimídia é muito importante neste tipo de documentos.

As *relações espaciais*, também conhecidas como relações de *layout*, servem para definir um espaço de apresentação de objetos de mídia em um dispositivo de saída em um certo momento do tempo. Assim, se o dispositivo de saída tem duas dimensões (monitor ou folha de papel), as relações de *layout* especificam a área bidimensional a ser usada em um dado momento da apresentação.

As *relações temporais*, por sua vez, definem dependências de tempo entre diferentes objetos de mídia. Estas relações devem ser consideradas em qualquer apresentação que envolva objetos de mídia dependentes de tempo. Apesar da existência das relações espaciais e de conteúdo, a sincronização em sistemas multimídia refere-se sobretudo aos aspectos temporais do relacionamento entre objetos de mídia, tendo as relações espaciais e de conteúdo normalmente pouca importância dentro desse contexto.

### 3.2.2. Sincronização Intra e Interobjeto

Existem relações temporais entre *as unidades de um objeto de mídia* dependente de tempo e relações temporais entre *diferentes objetos de mídia*. As primeiras definem a sincronização que deve existir entre as componentes de um mesmo objeto, ou seja, sincronização intra-objeto. As últimas, definem a sincronização existente entre objetos de mídia distintos, chamada sincronização interobjeto [SN95].

Um exemplo de sincronização intra-objeto é a relação temporal existente entre os *frames* de uma seqüência de vídeo. Para um vídeo com taxa de 25 *frames* por segundo, cada *frame* deve ser exibido por 40 ms. A Figura 3.1 mostra este exemplo para uma seqüência de vídeo que apresenta uma bola em movimento vertical.

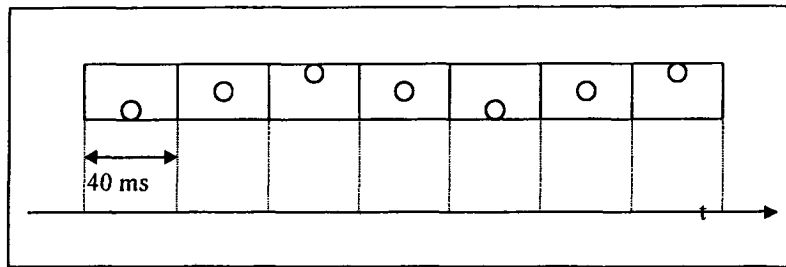


Figura 3.1 - Sincronização intra-objeto entre frames de uma sequência de vídeo [SN95]

A sincronização interobjeto deve ocorrer em toda e qualquer apresentação que envolva dois ou mais objetos de mídia, onde pelo menos um deles é um objeto de mídia contínua. A Figura 3.2 mostra um exemplo de sincronização interobjeto no qual objetos de mídia como áudio, vídeo, animação e imagens estáticas são apresentados. Estes objetos devem ser apresentados de forma ordenada no tempo, mantendo as dependências temporais especificadas.

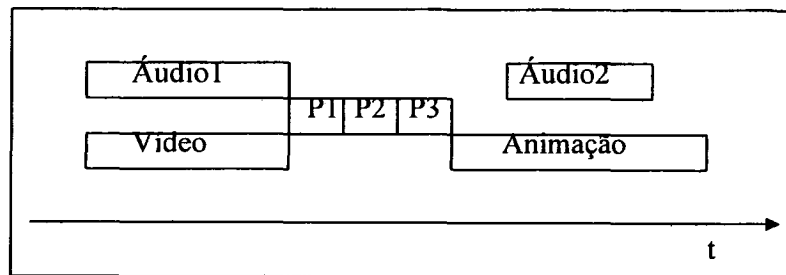


Figura 3.2 - Exemplo de sincronização interobjeto [SN95]

### 3.2.3. Unidades Lógicas de Dados

Objetos de mídia dependentes de tempo geralmente são constituídos de uma sequência de unidades de informação. Estas unidades de informação são chamadas unidades lógicas de dados (ULDs) [SN95].

Na maioria das vezes existem vários níveis de granularidade de ULDs em um objeto. Um exemplo disso é, por exemplo, um objeto de vídeo não comprimido dividido em cenas e *frames*. Cada *frame* pode ser particionado em áreas de  $16 \times 16$  *pixels*. Cada *pixel* é dividido em *luminance values* e *chrominance values*. Todas estas unidades são candidatas a se tornar ULD do objeto.

Os níveis de granularidade implicam uma decomposição hierárquica dos objetos de mídia. Existem dois tipos de hierarquia:

- Hierarquia de conteúdo: gerada pelo conteúdo do objeto de mídia. Assim, uma sinfonia é composta por movimentos, que por sua vez são compostos por notas musicais;
- Hierarquia de codificação: baseada na forma em que os dados estão codificados. Nesta hierarquia, um movimento de uma sinfonia pode ser representado por um objeto de mídia. Cada objeto que representa um movimento pode ser dividido em blocos de amostras. As amostras seriam, então, o nível mais baixo na hierarquia de codificação.

As ULDs são ainda classificadas em abertas e fechadas. ULDs fechadas são aquelas que têm uma duração pré-definida. Exemplos de ULDs fechadas são as ULDs que compõem objetos de mídia contínua armazenados, como áudio e vídeo. ULDs abertas, por sua vez, são aquelas que não possuem duração pré-definida. As ULDs abertas ocorrem, por exemplo, nos objetos de mídia que envolvem interação do usuário.

### **Classificação das Unidades Lógicas de Dados**

Para vídeo digital, geralmente os *frames* são selecionados como ULDs. Por exemplo, para um vídeo com 30 *frames* por segundo, cada ULD tem a duração de 1/30 segundos.

Em alguns casos, a ULD é muito pequena para ser manipulada. É o que acontece com os *streams* de áudio, nos quais a duração de cada unidade física tem uma dimensão muito reduzida, o que faz com que cada ULD seja formada por um bloco de amostras (geralmente composto de 512 amostras).

A importância das ULDs pode ser melhor observada quando se deseja realizar sincronização labial (*lip synchronization*). A Figura 3.3 mostra um exemplo de sincronização labial entre um objeto de áudio e um objeto de vídeo. Através do uso de ULDs, é possível especificar que o *frame n* do objeto de vídeo deve ser apresentado simultaneamente à amostra *m* do objeto de áudio.

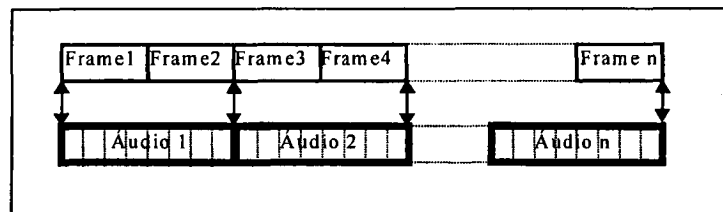


Figura 3.3 - Visão das ULDs de uma sincronização labial [SN95]

### 3.2.4. Especificação de Dependências Temporais

O processo de autoria de apresentações multimídia consiste basicamente em se especificar relações, a maioria dependentes de tempo, entre os componentes da aplicação. O sistema multimídia, por sua vez, deve interpretar estas especificações e prover a apresentação adequada. Interpretar especificações significa processar cada item da apresentação de acordo com os parâmetros de sincronização estabelecidos no processo de autoria. Portanto, os modelos de tempo são importantes tanto no processo de autoria (especificação de relações temporais) quanto no processo de apresentação (identificação de relações temporais). Existem diversos tipos de dependências temporais, que variam de acordo com a forma em que os dados são capturados ou apresentados [Buf94]. A Tabela 3.1 mostra algumas definições de dependências temporais.

Dependências de tempo presentes no momento da captura dos dados são chamadas *naturais* ou *implícitas*. Um exemplo de dependência *natural* é aquela existente entre objetos de vídeo e áudio capturados simultaneamente. Estes *streams* de dados são descritos como *contínuos*, uma vez que os elementos de dados gravados são exibidos de forma contígua no tempo. Por outro lado, dados

também podem ser capturados sem nenhuma ordem específica (por exemplo, um conjunto de fotografias). Sem uma dependência de tempo, estes dados são ditos *estáticos*. Dados *estáticos* podem ter relações temporais  *sintéticas*, ou seja, relações temporais criadas artificialmente.

Tabela 3.1- Definições de dependências temporais [Buf94]

Tipo de mídia	Dependência Temporal
Estática	Nenhuma dependência temporal
Discreta	Um único elemento
Transiente	Efêmera
Natural ou implícita	Dependência temporal do mundo real
Sintética	Dependência temporal criada artificialmente
Contínua	Execução contígua no tempo
Persistente	Mantida em um banco de dados
Ao vivo	Dados gerados em tempo real
Dados armazenados	Dados pré-gravados e armazenados

A combinação de dependências temporais *naturais* e *sintéticas* descrevem os requerimentos temporais de qualquer apresentação multimídia pré-orquestrada [Buf94]. No momento da apresentação, os dados podem manter suas dependências temporais naturais ou podem ser “forçados” a participar de relacionamentos temporais sintéticos. Um relacionamento *sintético* possui uma dependência temporal fabricada de acordo com as necessidades da aplicação. Por exemplo, um filme (longa metragem) consiste de uma seqüência de cenas gravadas. Estas cenas são gravadas naturalmente, mas arranjadas de forma artificial durante o processo de edição do filme. De forma similar, uma animação é uma ordenação sintética de dados estáticos. Uma fonte de dados “*ao vivo*” é uma fonte gerada dinamicamente e em tempo real, contrastando com as fontes de dados *pré-armazenados*. As relações sintéticas são válidas apenas para dados armazenados, uma vez que não é possível reordenar ou conhecer os valores futuros de uma fonte gerada “*ao vivo*”.

Dados multimídia podem ainda ser classificados de acordo com a duração de sua existência e de sua apresentação. Um objeto *persistente* é aquele que existe durante toda a execução da aplicação. Um objeto *não persistente* é criado dinamicamente e descartado quando obsoleto.

Com relação à apresentação, um objeto *transiente* é definido como um objeto que é apresentado por uma curta duração de tempo.

É importante observar que as definições de dependências temporais não são mutuamente exclusivas, ou seja, um mesmo objeto de mídia pode se enquadrar em duas ou mais dessas definições.

### 3.3. Requisitos de Apresentação

A fim de apresentar corretamente dados multimídia na interface do usuário, a sincronização é fundamental. Não é possível, no entanto, mensurar objetivamente a sincronização a partir do ponto de vista subjetivo da percepção humana, uma vez que esta varia de pessoa para pessoa. Portanto, apenas critérios heurísticos podem determinar se a apresentação de um *stream* está sincronizada ou não [SN95]. Nesta seção são apresentados experimentos extensivos que abordam a percepção humana da sincronização em diferentes mídias.

Os requisitos de apresentação incluem, para sincronização intra-objeto, a precisão relacionada aos atrasos na apresentação das ULDs de um *stream* qualquer e, para sincronização interobjeto, a precisão na apresentação paralela de objetos de mídia.

Um outro requisito de apresentação que deve ser considerado é o problema gerado pelo bloqueio de um *stream* que está sendo apresentado de forma síncrona com outros objetos de mídia. Este problema é conhecido como *gap problem*, e será abordado a seguir.

#### 3.3.1. Gap Problem

Uma apresentação multimídia pode ser composta por um ou mais *streams* de dados que, na maioria das vezes, são capturados e armazenados separadamente em servidores de mídia contínua. A aplicação encarregada de mostrar esta apresentação deve, então, solicitar aos servidores de mídia os *streams* requeridos e sincronizá-los, apresentando-os em paralelo. Mas, o que acontece se um dos *streams* for interrompido, mesmo que temporariamente, antes do seu término normal? Nesse caso, a aplicação deve prover artifícios para controlar o problema,

minimizando ao máximo o efeito causado na apresentação como um todo. O problema causado pela interrupção de um *stream* é chamado *gap problem*.

Duas soluções são propostas para o *gap problem* [SN95]. A primeira, conhecida como bloqueio restritivo, consiste em se repetir a última ou as últimas amostras do *stream* recebidas antes de sua interrupção, ou ainda prover um *stream* alternativo que possa ser usado caso ocorra um *gap*. A outra forma baseia-se na realização de uma reamostragem de um ou mais *streams* com o propósito de restabelecer a sincronização perdida devido a um *gap*. A solução mais indicada para o problema varia de acordo com a duração do *gap*, uma vez que o mesmo pode durar desde alguns milissegundos até vários minutos. Cabe à aplicação (e não ao sistema multimídia) optar pela melhor solução. As soluções para este problema são detalhadamente abordadas a seguir.

### Método do Bloqueio Restritivo

O método do bloqueio restritivo, como já foi dito, consiste em se repetir a última ou as últimas amostras de um *stream* interrompido. Assim, para um *stream* de vídeo, o último *frame* recebido é mostrado como uma imagem estática. Uma outra forma de solução do problema através de bloqueio restritivo consiste na especificação de uma apresentação alternativa, que é apresentada toda vez que um *gap* excede um determinado limite de tempo. A Figura 3.4 mostra esta solução. Nela pode-se observar que, quando ocorre um *gap* no *stream* de áudio, uma apresentação alternativa é acionada, cobrindo a falta do *stream* interrompido.

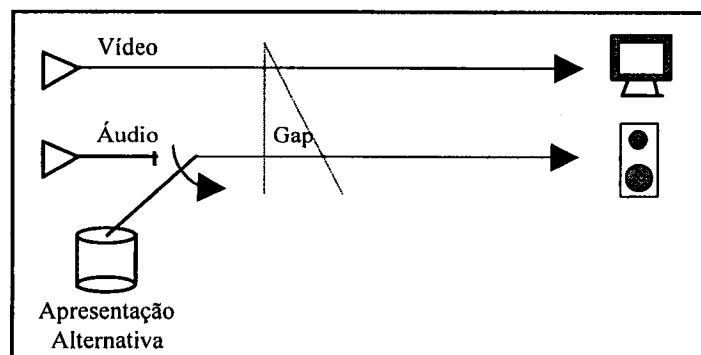


Figura 3.4 - Solução para o Gap Problem através do Método de Bloqueio Restritivo [SN95]

### Método da Reamostragem de um *Stream*

O problema gerado por um *gap* em um *stream* pode ser também solucionado através da reamostragem desse *stream*, desde que o *gap* tenha uma duração curta (de no máximo alguns segundos). A solução consiste em aumentar ou diminuir a velocidade em que o *stream* é apresentado, até que se alcance a sincronização desejada. Este tipo de solução é conhecida como reamostragem *on-line*. A reamostragem de um *stream* pode também ser usada para sincronizar mídias capturadas a partir de dispositivos diferentes, que podem, e muitas vezes têm, uma precisão insuficiente em seus cristais de *clock*. Um exemplo disso ocorre quando se captura um concerto musical com dispositivos de áudio e vídeo diferentes. A duração da apresentação dessas duas mídias de acordo com a taxa de amostragem das sequências de áudio e vídeo armazenadas pode variar. É necessário então, fazer a reamostragem de um dos *streams* de forma a se obter a sincronização requerida. Este processo recebe o nome de reamostragem *off-line*, uma vez que os *streams* se encontram armazenados (não são gerados ou recebidos via rede em tempo real).

A percepção humana da reamostragem de *streams* depende muito do tipo de mídia utilizado. No caso de vídeo, a reamostragem através da inserção ou remoção de *frames* pode ser feita sem muitos problemas — é o que acontece nas conversões NTSC/PAL. Por outro lado, a reamostragem de *streams* de áudio é bem mais complexa. Um usuário certamente se sentirá desconfortável ao ouvir um trecho de áudio que possua blocos repetidos ou removidos. Além disso, mudanças na taxa de *playback* podem ser facilmente percebidas pelo usuário, uma vez que essa mudança causa uma variação instantânea na frequência do áudio. Todos estes problemas inviabilizam a reamostragem *on-line* de *streams* de áudio [SN95].

#### 3.3.2. Sincronização Labial e Sincronização de Ponteiros

A sincronização labial (*lip synchronization*) e a sincronização de ponteiros referem-se aos relacionamentos temporais entre *streams* de áudio e vídeo que possuem um alto grau de acoplamento, isto é, são fortemente inter-relacionados.



## Sincronização Labial

A sincronização labial deve ser observada no caso de apresentações que envolvam uma pessoa falando. Se o usuário percebe uma falta de sincronismo entre os movimentos dos lábios do interlocutor e as palavras que estão sendo olvidas, então diz-se que os *streams* não estão em sincronismo. A diferença de tempo entre ULDs de áudio e vídeo relacionados é chamada *skew* (variação). Quando  $skew = 0$ , a apresentação está em sincronismo, quando  $skew < 0$ , o áudio está atrasado em relação ao vídeo e, finalmente, quando  $skew > 0$ , o vídeo está atrasado em relação ao áudio.

Experimentos realizados no *IBM European Networking Center* [SE93] mostram que pequenas variações na sincronização dos *streams* (para mais ou para menos) são praticamente imperceptíveis à maioria dos usuários. Dessa forma, variações de -80 ms até +80 ms não são percebidas pela maioria das pessoas, portanto, variações dentro desse intervalo são consideradas aceitáveis. É importante destacar, no entanto, que a percepção humana da sincronização labial é extremamente subjetiva. O experimento mostra casos de pessoas que não percebiam a falta de sincronismo entre áudio e vídeo mesmo com variações inferiores a -160 ms e superiores a +160 ms.

## Sincronização de Ponteiros

Com o advento do Trabalho Cooperativo Suportado por Computador — CSCW (*Computer-Supported Cooperative Work*) — câmeras e microfones passaram a ser incorporadas às estações de trabalho dos usuários, permitindo que discussões de trabalho sejam feitas *on-line* através do computador. Neste tipo de trabalho, um mesmo objeto pode ser manipulado por vários usuários remotos. Cada usuário deve ter um *feedback* das suas ações e perceber as ações dos outros participantes em tempo real.

Um problema inerente ao ambiente CSCW é a sincronização entre a voz de um usuário e suas ações, realizadas com o *mouse*. Assim, se um usuário aponta um determinado objeto e fala algo sobre ele, os demais usuários devem ver o ponteiro se posicionar sobre o referido objeto no

mesmo instante em que ouvem a referência àquele objeto. O problema da sincronização de voz e ponteiro não é tão drástico quanto o problema da sincronização labial, e variações de até 500 ms (para mais e para menos) não são perceptíveis para a maioria dos usuários.

### 3.4. Modelos de Referência para Sincronização Multimídia

Os modelos de referência para sincronização multimídia são necessários para [SN95]:

- entender os diversos requisitos relacionados à sincronização multimídia;
- identificar e estruturar mecanismos de *run-time* que suportem a execução de sincronização;
- identificar a interface existente entre esses mecanismos de *run-time*;
- comparar soluções para o problema de sincronização multimídia.

Algumas classificações foram propostas na tentativa de se estabelecer um modelo de referência. *Little & Ghafoor* [LG90], por exemplo, propuseram a classificação de objetos multimídia de acordo com a sincronização intermídia e intramídia utilizada. *Ehley, Furth & Ilyas* [EFI94] propuseram uma classificação baseada no controle remoto ou local da sincronização de objetos de mídia. Por fim, *Meyer, Effelsberg & Steinmetz* [MES93] propuseram uma classificação baseada em camadas de sincronização.

A classificação de *Meyer, Effelsberg & Steinmetz* possui quatro camadas, como mostra a Figura 3.5. A primeira, chamada camada de mídia, é responsável pela sincronização *intrastream* entre mídias dependentes de tempo. A segunda, conhecida como camada de *stream*, realiza a sincronização entre *streams* de mídia (sincronização *interstream*). A terceira, chamada camada de objeto, é responsável pela apresentação de mídias dependentes e não dependentes de tempo. Finalmente, a quarta camada, conhecida como camada de especificação, permite a autoria de aplicações multimídia [MES93]. Cada uma das quatro camadas do modelo será detalhadamente abordada a seguir.

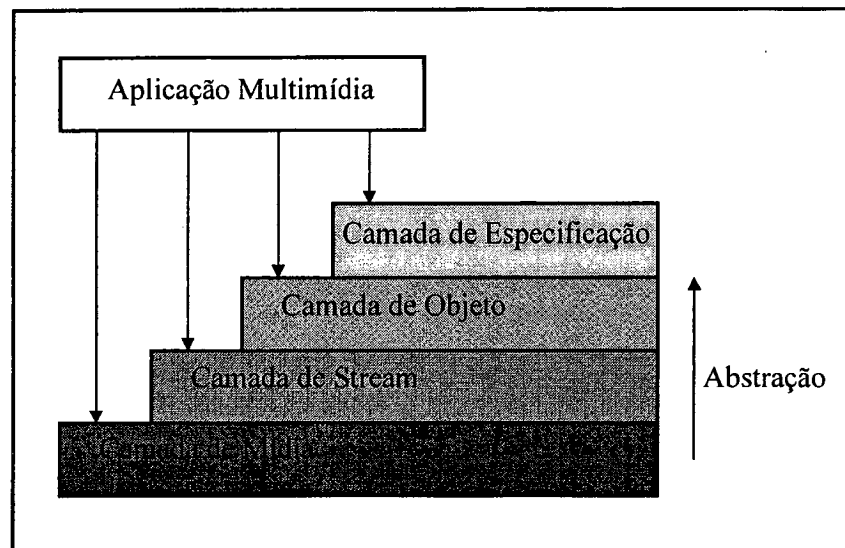


Figura 3.5 - Modelo de referência de quatro camadas [MES93]

### 3.4.1. A Camada de Mídia

Na camada de mídia, uma aplicação opera sobre um único *stream* de mídia contínua, o qual é tratado como uma seqüência de ULDs. A abstração oferecida nessa camada consiste de uma interface independente de dispositivo com operações como:

*read(deviceHandle, ULD)* ou *write(deviceHandle, ULD)*.

Usando esta camada, a própria aplicação é responsável pela sincronização *intrastream*. Neste caso, a sincronização é feita através do uso de mecanismos de controle de fluxo entre um dispositivo produtor e um dispositivo consumidor de *streams*. No caso especial de sincronização labial, a sincronização *interstream* pode ser provida facilmente através do entrelaçamento (*interleaving*) de *frames* simultâneos de áudio e vídeo dentro da mesma ULD. Assim, as implementações da camada de mídia podem ser classificadas em implementações simples e implementações que provêm acesso a *streams* de mídia entrelaçados.

### 3.4.2. A Camada de *Stream*

A camada de *stream* opera sobre *streams* de mídia contínua provendo sincronização *intra**stream*, e também sobre grupos de *streams* de mídia provendo sincronização *inter**stream* quando a apresentação paralela de *streams* é necessária. A abstração oferecida nesta camada é a noção de *stream* com parâmetros de temporização relacionados à qualidade de serviço (*QoS*) para sincronização *intra* e *inter**stream*.

Mídias contínuas são vistas pela camada de *stream* como um fluxo de dados que possui condições temporais implícitas; ULDs individuais não são visíveis. Os *streams* são executados em um ambiente de tempo real, onde todo processamento é regido por especificações temporais bem definidas. Por outro lado, aplicações que usam os serviços da camada de *stream* são executadas em um ambiente chamado NRTE (*Non Real-Time Environment*), no qual o processamento de eventos é controlado pela política de escalonamento do sistema operacional.

Alguns exemplos de operações invocadas a partir do NRTE por aplicações que desejam manipular *streams* e grupos de *streams* são:

- *start(stream)*;
- *stop(stream)*;
- *create\_group(list\_of\_streams)*;
- *start(group)* e
- *stop(group)*.

A interação de *streams* com mídias não dependentes de tempo e com o usuário é feita através da junção de eventos aos *streams* de mídia contínua. A operação abaixo, por exemplo, realiza uma junção deste tipo:

*setcuepoint(stream/group, at, event)*.

Assim, quando um *stream* em processo de *playback* encontra um evento de interação, este evento é enviado à aplicação, que, nesta camada, é responsável pelo processamento de objetos de mídia não dependentes de tempo e por interações com o usuário.

As implementações da camada de *stream* podem ser classificadas de acordo com o suporte dado à distribuição (fim-a-fim ou local), de acordo com os tipos de garantia de serviço oferecidos (melhor esforço ou determinístico) e de acordo com o tipo de *stream* suportado (analogico e/ou digital) [SN95].

Uma aplicação usando a camada de *stream* é responsável por iniciar, finalizar e agrupar *streams*, e, além disso, deve definir os parâmetros de *QoS* que serão utilizados pela camada de *stream*. A aplicação também é responsável pela sincronização de *streams* com objetos de mídia não dependentes de tempo.

### 3.4.3. A Camada de Objeto

A camada de objeto opera sobre todos os tipos de mídia, oferecendo às aplicações a abstração de uma apresentação sincronizada. A camada de objeto recebe uma especificação de sincronização multimídia como entrada e gera uma apresentação sincronizada como saída. As funções localizadas na camada de objeto são responsáveis pela execução de roteiros completos de apresentação que incluem a apresentação de objetos de mídia não contínua e chamadas à camada de *stream*. Além disso, a camada de objeto é responsável por iniciar a preparação de ações que são necessárias a fim de se obter uma apresentação corretamente sincronizada. A camada de objeto não manipula a sincronização *intrastream* e *interstream*. Para este propósito ela usa os serviços da camada de *stream*.

Um exemplo de implementação da camada de objeto é a especificação MHEG [ISO97]. Uma possível implementação dessa camada pode ser feita através de um sistema *run-time* MHEG, conhecido como MHEG *engine*. A função do MHEG *engine* é avaliar o *status* de objetos e realizar operações (ações) como *prepare*, *run*, *stop* e *destroy* sobre estes objetos. No caso de objetos de mídia dependentes de tempo, a operação *run* pode ser mapeada para a inicialização de

um *stream* de mídia na camada de *stream*. No caso de um objeto de mídia não dependente de tempo, a operação *run* faz com que o objeto seja diretamente apresentado.

A fim de apresentar corretamente um objeto, o MHEG *engine* necessita de um tempo para a preparação do mesmo (operação *prepare*). Isto ocorre porque, no caso de objetos de mídia contínua, o *engine* deve contactar a camada de *stream* para que ela apresente o referido objeto. No caso de objetos de mídia discretos, a preparação é necessária a fim de que os dispositivos de saída sejam preparados para a apresentação do objeto.

Neste contexto, a tarefa de uma aplicação que deseja utilizar a camada de objeto é fornecer uma especificação de sincronização adequada.

#### 3.4.4. A Camada de Especificação

Esta camada é uma camada aberta, que não oferece nenhuma interface específica. Ela inclui aplicações e ferramentas que permitem a criação de especificações de sincronização. Exemplos dessas aplicações e ferramentas são editores de sincronização, editores de documentos multimídia, sistemas de autoria e ferramentas de conversão, capazes de converter especificações para um formato aceitável pela camada de objeto, como, por exemplo, um formatador de documentos multimídia que produz especificações MHEG.

Uma outra função da camada de especificação é mapear os requerimentos de *QoS* do nível de usuário para os serviços oferecidos pela interface da camada de objeto.

Os métodos de especificação de sincronização podem ser classificados em quatro categorias principais [SN95]:

- Especificações baseadas em intervalos, que permitem a especificação de relações temporais entre intervalos de tempo das apresentações de objetos de mídia;
- Especificações baseadas em eixos, que relacionam eventos de apresentação com eixos que são compartilhados pelos objetos da apresentação;

- Especificações baseadas em controle de fluxo, nas quais o fluxo da apresentação é sincronizado em pontos de sincronização pré-estabelecidos;
- Especificações baseadas em eventos, nas quais a ocorrência de eventos durante a apresentação das mídias dispara a execução de ações.

Cada um desses métodos de especificação de sincronização é detalhadamente descrito a seguir.

### **3.5. Métodos de Especificação de Sincronização Multimídia**

Para a especificação de sincronização de múltiplos objetos de mídia, incluindo interação do usuário, métodos sofisticados de especificação devem ser usados. Nesta seção, os principais métodos de especificação de sincronização multimídia são abordados.

#### **3.5.1. Especificações Baseadas em Intervalos**

Através do conceito de instante e intervalo é possível estabelecer um modelo de tempo conceitual aplicável ao problema de sincronização multimídia [Buf94]. Em primeiro lugar, é importante definir o que é um instante e o que é um intervalo.

Um instante é um momento de comprimento zero no tempo, como por exemplo 4:00 da tarde. Por outro lado, um intervalo de tempo é definido como sendo o comprimento (duração) entre dois instantes. Assim, dados dois instantes  $a$  e  $b$ , sendo  $b > a$ , o comprimento do intervalo é dado por  $b - a$ , ou seja, a diferença entre os instantes final e inicial. Uma outra forma de especificar intervalos usa comparações ao invés de pontos. Nesta forma de especificação, um intervalo é definido em função de outro ou outros intervalos, o que possibilita a geração de relações temporais.

Segundo Allen [All83], existem treze formas nas quais dois intervalos podem se relacionar no tempo. Estas relações estão representadas graficamente na Figura 3.6. É importante observar que as treze relações são representadas por sete casos, uma vez que seis deles são inversos. Por

exemplo, *before* é a relação inversa de *after*, ou de forma equivalente,  $after^{-1}$  é a relação inversa de *after*; além disso, *a equals b* é o mesmo que *b equals a*.

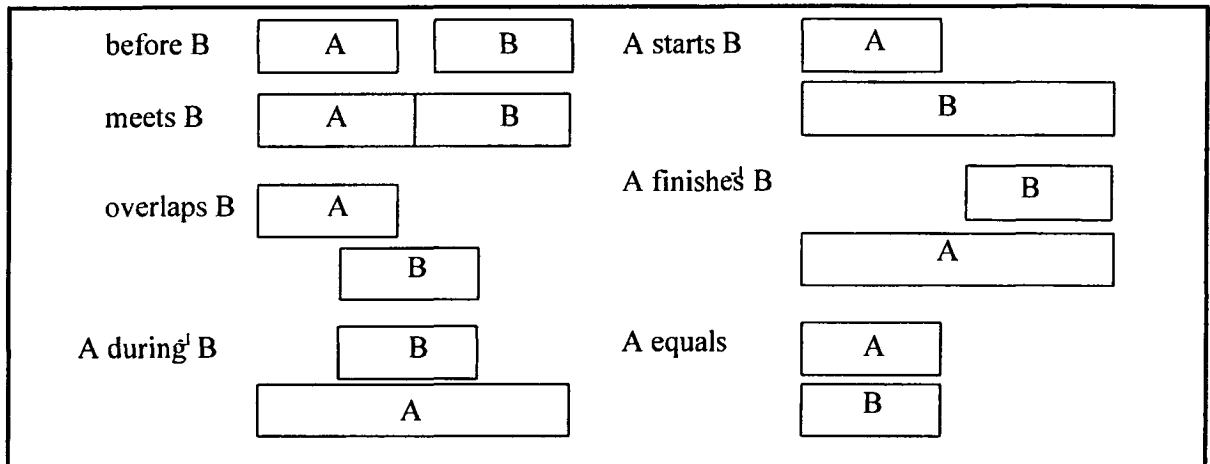


Figura 3.6 - Relações temporais binárias [All83]

É possível estender as 13 relações temporais binárias básicas através do modelo avançado de sincronização baseado em intervalos, proposto por *Wahl e Rothermel* [WR94]. Neste modelo, 29 relações relevantes à apresentação multimídia são obtidas a partir de disjunções das relações básicas entre intervalos mostradas na Figura 3.6. A fim de simplificar a especificação de sincronização, dez operadores foram definidos para manipular as relações entre os intervalos. As operações básicas do modelo estendido são mostradas na Figura 3.7.

Este modelo permite que se defina tanto a duração de objetos de mídia dependentes de tempo quanto a duração de objetos de mídia não dependentes de tempo. Dessa forma, é possível definir, por exemplo, a duração de uma interação com o usuário. A facilidade de manipular interações com o usuário é um dos pontos fortes deste modelo.

Usando o modelo avançado de sincronização baseado em intervalos pode-se especificar, por exemplo, a apresentação de um *slide show*, onde  $Slide_i$  ( $1 \leq i \leq n$ ) e um objeto de áudio possuem as seguintes relações temporais:



*Slide<sub>i</sub> co-início(0) Áudio*

*Slide<sub>i</sub> antes(0) Slide<sub>i+1</sub> ( $1 \leq i \leq n-1$ )*

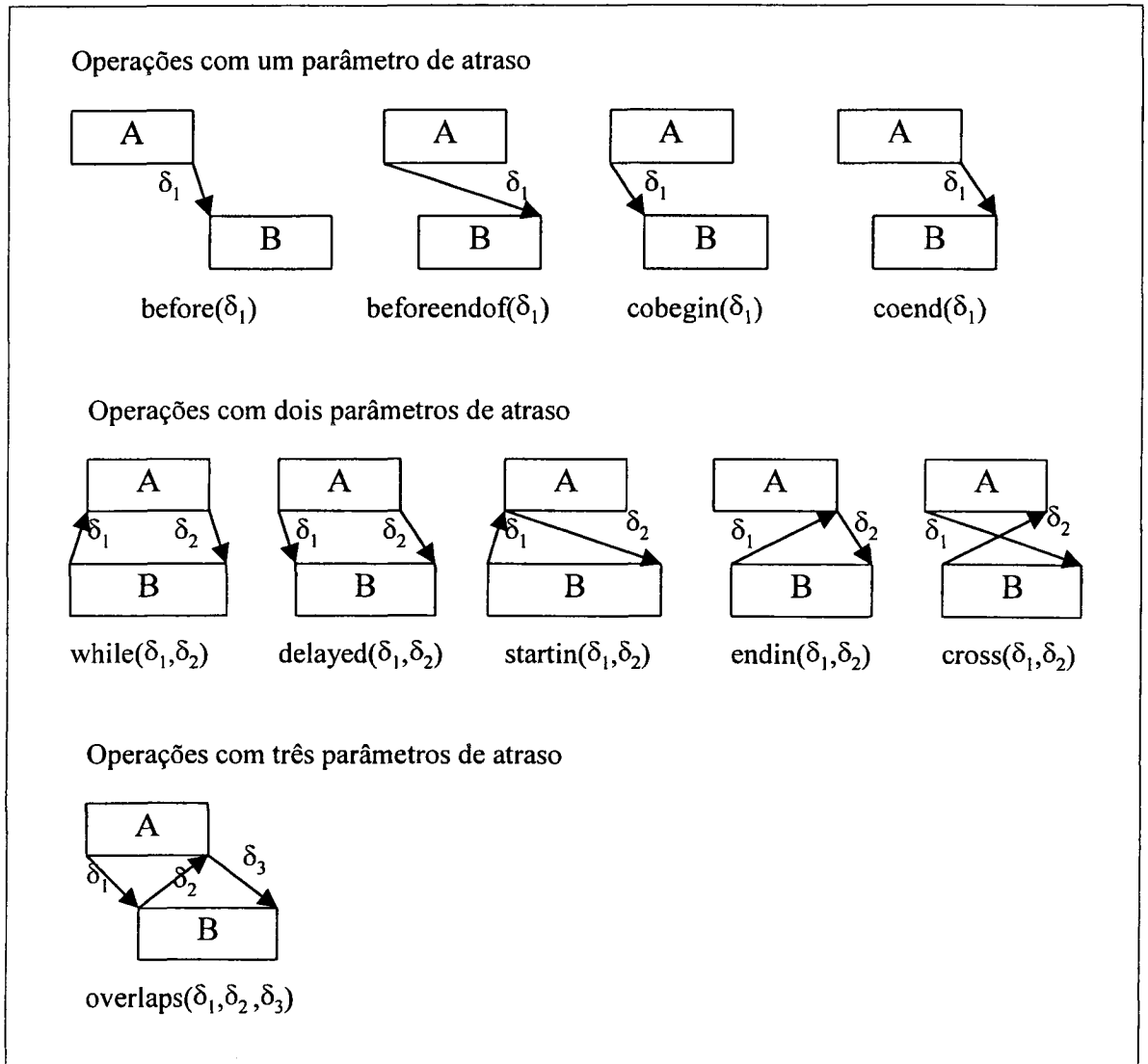


Figura 3.7 - Operações do modelo avançado de sincronização baseado em intervalos [WR94]

Um outro ponto forte deste modelo é a simplicidade da especificação. Para expressar a sincronização labial entre um objeto de vídeo e um objeto de áudio, basta apenas a seguinte especificação:

### *Áudio while(0,0) Video*

As principais desvantagens deste modelo estão ligadas à falta de mecanismos que permitam a especificação de relações temporais diretamente entre as subunidades (ULDs) de cada objeto. Além disso, este modelo não suporta especificações de parâmetros de *skew*.

#### **3.5.2. Especificação de Sincronização Baseada em Eixos**

Em uma especificação baseada em eixos, os eventos de uma apresentação, como o início e o fim da apresentação de uma mídia, são mapeados para eixos que são compartilhados por todos os objetos da apresentação. Existem duas formas de se especificar sincronização através do uso de eixos:

- Sincronização baseada em um temporizador global: Neste método, cada objeto de mídia é colocado em um eixo de tempo que representa uma abstração do tempo real. Este método de especificação é usado, por exemplo, no projeto *Athena Muse* [HSA89], no qual a sincronização é feita colocando-se todos os objetos em um eixo de tempo. Como os objetos são independentes uns dos outros, a remoção de um objeto não afeta a sincronização de outro. Com algumas modificações, este tipo de especificação é também usado no modelo de mídias ativas [TGD91]. Neste modelo, um tempo global é mantido, o qual é acessível a todos os objetos. Cada objeto pode mapear este tempo global para seu tempo local. Quando a distorção entre os tempos global e local excede um dado limite, é necessário fazer resincronização com o tempo global.
- Sincronização baseada em eixos virtuais: Neste método, ao invés de se usar um único eixo que represente uma abstração do tempo real, são usados diversos eixos, criando um espaço de coordenadas virtual. Esta forma de especificação é útil quando um objeto de mídia já possui uma forma de representação própria no tempo, como as notas musicais de uma canção [SN95]. Nesta especificação, um conjunto de eixos virtuais são mapeados para eixos reais durante o processo de execução da apresentação. Um dos pontos fortes dessa especificação é que ela apresenta uma abstração adequada para conteúdos de mídia, facilitando o

entendimento do usuário. O padrão HyTime [ISO92] utiliza este método para especificar sincronização entre dois ou mais componentes de uma apresentação multimídia.

### 3.5.3. Especificação de Sincronização Baseada em Controle de Fluxo

Nas especificações baseadas em controle de fluxo, o fluxo da apresentação de *threads* concorrentes é sincronizado em pontos pré-definidos da apresentação. Existem três formas de se especificar sincronização através do controle de fluxo [SN95]:

- **Especificação hierárquica básica:** Descrições de sincronização hierárquica são baseadas em duas operações de sincronização básicas: sincronização serial de ações (seqüência de slides, por exemplo) e sincronização paralela de ações (apresentação síncrona de áudio e vídeo, por exemplo). Em uma especificação de sincronização hierárquica, os objetos multimídia são representados através de uma árvore, na qual os nós denotam apresentação serial ou paralela de suas sub-árvores. A especificação hierárquica é largamente usada devido à facilidade de compreensão de sua estrutura básica. No entanto, esta especificação apresenta uma série de limitações; uma delas, e talvez a mais grave, é o fato de que cada ação só pode ser sincronizada em seu início ou seu fim. Isto significa, por exemplo, que a apresentação de subtítulos em partes de uma seqüência de vídeo requer que a seqüência seja dividida em vários componentes consecutivos.
- **Especificação de sincronização através do uso de pontos de referência:** Neste tipo de especificação, cada objeto de mídia é considerado como uma seqüência de unidades lógicas de dados (ULDs). O início e o fim de cada objeto de mídia, bem como o início de cada sub-unidade dependente de tempo desse objeto são considerados pontos de referência. A sincronização entre objetos é então definida pela conexão de pontos de referência desses objetos. Um conjunto de pontos de referência conectados é chamado ponto de sincronização. A apresentação das sub-unidades que participam de um mesmo ponto de sincronização deve ser iniciada ou finalizada quando este ponto de sincronização é alcançado. Este método de sincronização especifica relações temporais entre objetos sem fazer referência explícita ao tempo. Esta especificação permite que sejam feitas sincronizações em qualquer instante

durante o processo de apresentação de um objeto. Além disso, a apresentação de objetos de duração desconhecida pode ser facilmente integrada. Esta especificação é também muito fácil de ser usada por ser extremamente intuitiva.

- *Especificação de sincronização através de Redes de Petri*: É possível especificar sincronização através do uso de *Redes de Petri*. Para tanto, basta estender os conceitos das *Redes de Petri* com especificações de duração [LG92], criando uma Rede de Petri temporizada. As regras para uma *Rede de Petri* temporizada são as seguintes:
  - uma transição ocorre se todos os locais de entrada contêm um *token* não bloqueante;
  - se uma transição ocorre, um *token* é removido de cada local de entrada e um *token* é adicionado a cada local de saída;
  - um *token* adicionado a um novo local fica bloqueado durante a duração que é atribuída a este local.

Para objetos de mídia dependentes de tempo, cada local na *Rede de Petri* representa uma ULD. Dessa forma, a sincronização fina pode ser modelada através da conexão de ULDs apropriadas.

#### 3.5.4. Especificação de Sincronização Baseada em Eventos

No caso da sincronização baseada em eventos, as ações de apresentação são iniciadas por eventos de sincronização. Ações típicas de apresentação são:

- inicie uma apresentação;
- interrompa uma apresentação;
- prepare uma apresentação.

Os eventos que iniciam ações de apresentação podem ser externos (gerados por um *timer*, por exemplo) ou internos à apresentação, gerados por um objeto de mídia dependente de tempo que alcançou uma ULD específica. Este modelo de especificação é facilmente estendido a novos

tipos de sincronização. O padrão MHEG-5 utiliza este modelo de sincronização para apresentação de objetos de mídia que possuem relações temporais.

### 3.6. Sincronização em Ambientes Distribuídos

A sincronização em um ambiente distribuído é mais complexa que em um ambiente local. Isto ocorre devido ao armazenamento distribuído das informações de sincronização e dos objetos de mídia envolvidos em uma apresentação. O processo de comunicação entre os locais de armazenamento e apresentação introduz atrasos e *jitter* adicionais, que devem ser considerados no processo de apresentação.

No local de apresentação, os componentes do sistemas precisam ter disponíveis as informações de sincronização de um objeto de mídia no momento em que este objeto deve ser apresentado. Existem três formas básicas de se enviar informações de sincronização a um local de apresentação, são elas [SN95]:

- Envio de toda a informação de sincronização antes do início da apresentação: Esta técnica é freqüentemente usada no caso de sincronização sintética. Neste caso, a aplicação que está sendo executada no local de apresentação acessa a camada de objeto passando a especificação previamente recebida, ou uma referência a ela, como parâmetro. Esta técnica tem como vantagem a facilidade de implementação e como desvantagem o atraso gerado antes do início da apresentação (tempo gasto no recebimento da especificação de sincronização).
- Uso de um canal adicional de sincronização: Este método é usado quando se deseja sincronizar objetos de mídia que estão sendo gerados em tempo real a partir de uma única fonte. Neste caso, a informação de sincronização de um dado objeto não é previamente conhecida, estando geralmente disponível apenas no momento em que o objeto deve ser apresentado, o que traz maior complexidade ao processo de sincronização. A vantagem dessa técnica é que nela as informações de sincronização não são responsáveis pelo atraso na apresentação de objetos multimídia.

- Uso de streams de dados multiplexados: Esta técnica baseia-se na multiplexação de dois ou mais *streams* de dados em um único canal de comunicação. A vantagem da multiplexação é que as informações de sincronização são enviadas juntamente com as unidades de mídia. Este procedimento não gera nenhum atraso adicional e dispensa o uso de um canal extra para a informação de sincronização. Por outro lado, é difícil usar este método quando os objetos de mídia que devem ser sincronizados estão armazenados em locais diferentes. Um exemplo de multiplexação de *streams* é o padrão MPEG [Key94], onde um único *stream* combina áudio, vídeo e informações de sincronização entre as duas mídias.

### 3.7. Qualidade de Serviço

Na Seção 2.6, os aspectos genéricos a respeito de qualidade de serviço (*QoS*) foram apresentados. Nesta seção, o assunto *qualidade de serviço* é abordado com relação à sincronização intermídia. A qualidade de serviço está presente tanto na sincronização intramídia, envolvendo objetos de mídia isolados, quanto na sincronização intermídia, onde dois ou mais objetos de mídia possuem relações temporais.

A especificação de *QoS* para um objeto de mídia inclui a qualidade que cada uma de suas ULDs deve apresentar, bem como a precisão das relações temporais entre as ULDs desse objeto, caso o mesmo seja um objeto de mídia dependente de tempo. Assim, se uma especificação de *QoS* para um *stream* de vídeo determina que sejam mostrados 30 *frames* por segundo, então cada *frame* deve ser exibido por exatamente 1/30 segundos a fim de se atender à especificação de qualidade de serviço requisitada.

Parâmetros de *QoS* são também importantes na sincronização intermídia. Estes parâmetros são usados, por exemplo, para indicar o máximo *skew* permitido na apresentação síncrona de duas mídias. Dessa forma, para uma apresentação de áudio e vídeo que envolva sincronização labial, o parâmetro *skew* não deve ser superior a 80 ms. A Tabela 3.2 mostra alguns valores de *QoS* usados na sincronização de pares de mídia.

Tabela 3.2 - Qualidade de Serviço para propósitos de sincronização [SN95]

Mídias	Modo, Aplicação	QoS	
Vídeo	Animação	Correlacionado	+/- 120 ms
	Áudio	Sincronização Labial	+/- 80 ms
	Imagem	Sobreposto	+/- 240 ms
		Não-sobreposto	+/- 500 ms
	Texto	Sobreposto	+/- 240 ms
		Não-sobreposto	+/- 500 ms
Áudio	Animação	Correlação de eventos (dança)	+/- 80 ms
	Áudio	Fortemente acoplados ( <i>stereo</i> )	+/- 11 $\mu$ s
		Fracamente acoplados (modo de diálogo com vários participantes)	+/- 120 ms
		Fracamente acoplados (música em <i>background</i> )	+/- 500 ms
	Imagem	Fortemente acoplados (música com respectivas notas musicais)	+/- 5 ms
		Fracamente acoplados ( <i>slide show</i> )	+/- 500 ms
	Texto	Anotação textual	+/- 240 ms
	Ponteiro	Áudio relacionado ao item que o ponteiro aponta	+/- 500 ms

### 3.8. Considerações Finais

Este capítulo apresentou e discutiu os principais problemas envolvidos na sincronização dos elementos de uma apresentação multimídia, deixando clara a importância das relações intra e intermídia na obtenção da sincronização desejada. Além disso, apresentou e discutiu modelos de referência para a apresentação multimídia e métodos de especificação de sincronização. Finalmente, tratou dos problemas de sincronização em ambientes distribuídos, onde o desacoplamento das mídias pode gerar diversos problemas de sincronização no momento da apresentação.

---

## 4. Conclusão

---

A apresentação simultânea de dois ou mais objetos de mídia em um sistema que suporte programação *multithreading* é uma tarefa relativamente simples, desde que não existam dependências temporais entre tais objetos. No entanto, não faz sentido falar em uma apresentação multimídia que não possua relações temporais entre seus componentes. Logo, todo e qualquer sistema que se proponha a apresentar objetos multimídia deve ser capaz de interpretar e executar especificações de sincronização.

Além de interpretar e executar especificações intermídia, um sistema apresentador de objetos multimídia deve também ser capaz de atender especificações de sincronização intramídia. Tais especificações são também chamadas de parâmetros de qualidade de serviço, uma vez que através delas pode-se especificar, por exemplo, a qualidade da apresentação de um *stream* de áudio ou de vídeo.

Este trabalho procurou apresentar e discutir estes dois tipos de sincronização, mostrando a importância de ambos na obtenção da sincronização desejada. Aspectos básicos, tais como a definição de ULD's, foram apresentados de forma a permitir o entendimento do conceito de dependência temporal, fundamental para compreensão da sincronização multimídia. Aspectos mais avançados, tais como modelos de referência e métodos de especificação de sincronização foram descritos e discutidos em profundidade. Finalmente, o impacto da distribuição das mídias no processo de sincronização foi abordado, mostrando que, através de parâmetros de qualidade de serviço, é possível especificar condições mínimas de serviço capazes de atender aos requisitos das apresentações multimídia.



---

## Referências Bibliográficas

---

- [All83] J. F. Allen; *Maintaining Knowledge about Temporal Intervals*. Communications of the ACM, pp. 832-843, novembro, 1983.
- [Buf94] J. F. K. Buford et al.; *Multimedia Systems*. ACM Press, 1994.
- [EF194] L. Ehley, B. Furth, M. Ilyas; *Evaluation of Multimedia Synchronization Techniques*. In Proceedings of the International Conference on Multimedia Computing and Systems, Boston, MA, pp.110-119, maio, 1994.
- [HSA89] M. E. Hodges, R. M. Sasnett, M. S. Ackerman; *Athena Muse: A Construction Set for Multimedia Applications*. IEEE Software, pp. 37-43, janeiro, 1989.
- [ISO92] ISO 10744, *Hypermedia/Time-Based Structuring Language - HyTime*, 1992.
- [ISO97] ISO/IEC 13522-5: *Information Technology – Coding of Multimedia and Hypermedia Information – Part 5: Support for Base Level Interactive Applications*, Março, 1997.
- [Key94] J. Keyes; *Multimedia Handbook*. McGraw-Hill, 1994.
- [LG90] T. D. C. Little, A. Ghafoor; *Network Considerations for Distributed Multimedia Objects Composition and Communications*. IEEE Network Magazine, pp. 32-39, novembro, 1990.
- [LG92] T. D. C. Little, A. Ghafoor; *Scheduling of Bandwidth-Constrained Multimedia Traffic*. Computer Communication, pp. 381-387, julho, 1992.
- [MES93] T. Meyer, W. Effelsberg, R. Steinmetz; *A Taxonomy on Multimedia Synchronization*. In Proceedings of the 4th International Workshop on Future Trends in Distributed Computing Systems, Lisbon, Portugal, setembro, 1993.
- [SE93] R. Steinmetz, C. Engler; *Human Perception of Media Synchronization*. Technical Report 43.9310, IBM European Networking Center Heidelberg, Heidelberg, Germany, 1993.
- [SN95] R. Steinmetz, K. Nahrstedt; *Multimedia: Computing, Communications & Applications*. Prentice Hall, 1995.
- [TGD91] D. Tschritzis; S. Gibbs; L. Dami; *Active Media*. In *Object Composition* (D. Tschritzis, ed.). Universite de Geneve, Centre Universitaire d'Informatique, Geneve, pp. 115-132, junho, 1991.
- [WR94] T. Wahl, K. Rothermel; *Representing Time in Multimedia Systems*. In Proceedings of the International Conference on Multimedia Computing and Systems, Boston, MA, pp. 538-543, maio, 1994.