

Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia

Programa de Pós-Graduação em Ciência da Computação

**Componentes para manipulação de objetos
multimídia utilizando o padrão MPEG-7**

Leandro Donaires Figueira

São Carlos

Junho/2007

Universidade Federal de São Carlos

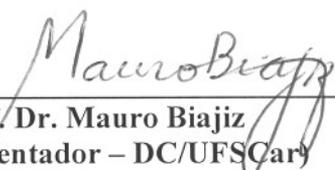
Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

“Componentes para Manipulação de Objetos Multimídia utilizando o Padrão MPEG-7”

LEANDRO DONAIRES FIGUEIRA

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Membros da Banca:



Prof. Dr. Mauro Biajiz
(Orientador – DC/UFSCar)



Prof. Dr. César Augusto Camillo Teixeira
(DC/UFSCar)



Prof. Dr. Luiz Cantolesi Junior
(UNIMEP)

São Carlos
Junho/2007

**Ficha catalográfica elaborada pelo DePT da
Biblioteca Comunitária da UFSCar**

F475cm

Figueira, Leandro Donaires.

Componentes para manipulação de objetos multimídia
utilizando o padrão MPEG-7 / Leandro Donaires Figueira. --
São Carlos : UFSCar, 2007.

84 f.

Dissertação (Mestrado) -- Universidade Federal de São
Carlos, 2007.

1. Banco de dados multimídia. 2. MPEG-7. 3.
Componentes de software. Título.

CDD: 003 (20^a)

Para minha família, especialmente meus pais Flávio e Vanilde, a quem devo por tudo o que sou. Espero sempre ser um motivo de orgulho na vida de vocês e trazer felicidade.

Para minha amada Carina, cujo apoio foi substancial na reta final desse trabalho.
Você sabe o quão especial é na minha vida.

Para meus amigos, que sempre me apoiaram e torceram por mim, que tornaram mais alegre os meus dias e me proporcionaram experiências únicas. Mesmo com uma eventual distância, vocês sempre estarão no meu coração.

Agradecimentos

A Deus, que olha por mim e me provê a saúde e paz necessárias para atingir os meus objetivos e aprender cada dia um pouco sobre a vida e sobre os assuntos que me interessam.

Ao meu orientador, Prof. Mauro, pela dedicação, amizade e o grande aprendizado que me proporcionou.

Aos meus colegas do Departamento de Computação, por toda ajuda e companheirismo.

À CAPES, que me auxiliou no desenvolvimento deste trabalho.

À Bell Labs pela sua criação revolucionária, o transistor.

À id, Valve, Blizzard, SNK e Capcom por suas criações.

Muito agradecido a todos vocês.

Resumo

Este trabalho apresenta uma camada de manipulação de mídias baseada no padrão de descrição de dados multimídia MPEG-7. Esta camada realiza algumas partes dos MDS (*Multimedia Description Schemes*) do padrão MPEG-7, e através de sua construção por componentes, forma uma base para a implementação de quaisquer aplicações que tenham requisitos de manipulação de mídias, desde as mais simples até um sistema de banco de dados multimídia. Nesse último caso, pode ser um sistema que execute as várias operações possíveis em um banco de dados, como mineração de dados multimídia e indexação por características de baixo-nível, tudo apoiado sobre o padrão. Esta camada foi feita com a intenção de unificar as necessidades de manipulação de mídias por várias aplicações, integrando-as e agindo dessa forma como mediadora. Ela provê interfaces para que sejam feitas inserções de mídias junto com as anotações semânticas, além de consultas sobre elas através desses metadados semânticos provindos das anotações. Foi proposto um modelo simplificado de anotação e consulta semântica para a fácil utilização dessas interfaces, sendo que tanto as anotações como as consultas são feitas de maneira semelhante. Finalmente, essa camada possibilita que seja feita a exportação dos metadados da consulta no formato usado pelo padrão MPEG-7, para intercâmbio de dados com outros sistemas.

Abstract

This work presents a layer for the manipulation of media based on the multimedia content description interface, MPEG-7. This layer realizes some parts of the MPEG-7 MDS (Multimedia Description Schemes), and being built by components, makes a base for the implementation of any application that has media manipulation requisites that range from the simplest ones to a database multimedia system. In this last case, it can be a system that have many operations being executed in a database, like multimedia data mining and indexing by low level features, all supported by MPEG-7. This layer was built to unify the necessities on manipulating media of many applications, integrating them and acting as a mediator. It provides interfaces so that the insertion of media together with their semantic annotation is possible, along with queries based on these semantic metadata derived from these annotations. A simplified model for the semantic annotation and queries for the ease of use of these interfaces was proposed, being the annotations and queries made in a similar manner. Finally, this layer allows the exporting of the query metadata on the format used by MPEG-7 for interchange between systems.

Índice de Figuras

Figura 1 – O modelo de dados do VIDEX [TUSCH; KOSCH; BÖSZÖRMÉNYI, 2000].....	10
Figura 2 – O processamento de uma consulta SQL+D	13
Figura 3 – Arquitetura do sistema multimídia distribuído [KOSCH <i>et al.</i> , 2001b]	19
Figura 4 – O buscador com a especificação de uma restrição temporal [KOSCH <i>et al.</i> , 2001b] .	20
Figura 5 – A interface do anotador de vídeos [KOSCH <i>et al.</i> , 2001b]	21
Figura 6 – O gerenciador de apresentação [KOSCH <i>et al.</i> , 2001b]	22
Figura 7 – O editor temporal da ferramenta MAE	24
Figura 8 – Arquitetura do ambiente AMMO.....	25
Figura 9 – Panorama dos <i>MDS</i> do MPEG-7 [SALEMBIER; SMITH, 2001].....	33
Figura 10 – <i>Description Schemes</i> Temporais [SALEMBIER; SMITH, 2001]	34
Figura 11 – DSs para descrição de aspectos conceituais [SALEMBIER; SMITH, 2001]	37
Figura 12 – Variações de um material áudio-visual [SALEMBIER; SMITH, 2001]	39
Figura 13 – Descrição MPEG-7 e unidades de acesso [AVARO; SALEMBIER, 2001]	51
Figura 14 – Modelo conceitual do LMS [BRESSAN; FRANCO; RUGGIERO, 2005].....	54
Figura 15 – Esquema conceitual de banco de dados das ferramentas síncronas	55
Figura 16 – Arquitetura do sistema	59
Figura 17 – Processo de conversão do XML <i>Schema</i> ao banco de dados relacional	60
Figura 18 – Cenário típico de inserção/anotação da mídia.....	62
Figura 19 – Cenário típico de consulta da mídia	63
Figura 20 – Exemplo de construção da <i>string</i> do dado semântico	71
Figura 21 – Processo de inserção/anotação da mídia	75
Figura 22 – Processo de consulta da mídia.....	76
Figura 23 – Parte semântica do XML exemplo	77
Figura 24 – Parte automaticamente obtida da mídia	78

Índice de tabelas

Tabela 1 – Características dos BDs e BDMMs. [SUBRAMANYA, 2000].....	6
Tabela 2 - Exemplo de parâmetros da consulta [GAVIOLI; BIAJIZ; MOREIRA, 2005].....	27
Tabela 3 – Facetas providas pelo XML <i>Schema</i> [HUNTER, 2001]	45

Sumário

1	INTRODUÇÃO	1
1.1	MOTIVAÇÃO E OBJETIVOS	1
1.2	ORGANIZAÇÃO DA MONOGRAFIA	2
2	BANCO DE DADOS MULTIMÍDIA	3
2.1	BD VERSUS BDMM – A NATUREZA DOS DADOS MULTIMÍDIA	3
2.2	MODELOS DE DADOS MULTIMÍDIA	7
2.3	LINGUAGENS DE CONSULTA MULTIMÍDIA	11
2.3.1	SQL/MM e consultas sobre o padrão MPEG-7	13
2.3.2	Busca baseada em conteúdo	16
2.4	SISTEMAS DE BANCO DE DADOS MULTIMÍDIA	18
2.4.1	SMOOTH	19
2.4.2	MAE (<i>Multimedia Authoring Environment</i>)	23
2.4.3	MIFLIR (<i>Metric Indexing and Fuzzy Logic-based Image Retrieval System</i>)	25
2.4.4	MARS (<i>Multimedia Analysis and Retrieval System</i>)	27
2.5	CONSIDERAÇÕES FINAIS	28
3	O PADRÃO MPEG-7	29
3.1	DESCRIPTORS E (MULTIMEDIA) DESCRIPTION SCHEMES	31
3.1.1	Elementos básicos	33
3.1.2	Gerenciamento e descrição de conteúdo	35
3.1.3	Navegação e acesso, interação do usuário e organização do conteúdo	38
3.2	LINGUAGEM DE DEFINIÇÃO DE DESCRIÇÕES MPEG-7	41
3.2.1	Componentes estruturais do XML Schema	42
3.2.2	Tipos de dados do XML Schema	45
3.2.3	Extensões específicas do MPEG-7	47
3.2.4	Exemplo de uso da DDL	48
3.3	O FORMATO BiM	49
3.4	CONSIDERAÇÕES FINAIS	52
4	O AMBIENTE DE APRENDIZADO ELETRÔNICO TIDIA-AE	53
4.1	CONSIDERAÇÕES FINAIS	577
5	A CAMADA DE MANIPULAÇÃO DE MÍDIAS	58
5.1	DADOS SEMÂNTICOS, INSERÇÕES E CONSULTAS	61
5.1.1	Modelo de representação semântica em interfaces e detalhes técnicos	66
5.2	DESCRIÇÃO DAS INTERFACES	73
5.2.1	Exemplo de utilização das interfaces	74
5.3	CONSIDERAÇÕES FINAIS	79
6	BIBLIOGRAFIA	81

1 INTRODUÇÃO

1.1 MOTIVAÇÃO E OBJETIVOS

O avanço da infra-estrutura de telecomunicações torna cada vez mais viável o uso de recursos que antes não podiam ser aproveitados devido à escassez de largura de banda e baixa qualidade de serviço (QoS – *Quality Of Service*). Os recursos em questão são as transmissões de mídias (por exemplo: música, áudio, vídeo, imagem e animação), que se encaixam nos mais diversos contextos.

Apenas para exemplificar alguns dos possíveis contextos onde a transmissão de mídias representa avanços, há o campo da telemedicina, que possibilita consultas e até a execução de cirurgias remotamente, e o campo de educação à distância, que habilita o aprendizado remoto. Porém, para assegurar que a transmissão da mídia seja feita sem provocar insatisfação por parte do usuário ou até danos (no caso da transmissão de uma cirurgia), há de se avaliar a necessidade de qualidade de serviço e largura de banda, ou seja, verificar se a conexão por onde passarão os dados atendem os requisitos mínimos de latência e largura de banda. Devem existir mecanismos que ajam em favor da menor redução possível de qualidade no ato de congestionamento da rede, garantindo qualidade de serviço. A largura de banda determina em que resolução, ou *bit rate*, pode-se transmitir a mídia.

Dentro desse cenário, pode haver a necessidade de se armazenar as mídias para uma posterior consulta, e esse armazenamento não é feito de acordo com os moldes tradicionais, pois a natureza dos dados multimídia é diferente da natureza dos dados convencionais. Esse fato é detalhado na seção 2.1. Além do desafio de armazenar mídias eficientemente, a busca eficiente (rapidez) e eficaz (resultados relevantes) é outro grande desafio em se tratando de dados multimídia. Pode haver, ainda, a necessidade de intercâmbio de informações entre sistemas, gerando a necessidade de uso de um padrão de descrição de dados multimídia.

Com tudo isso, o objetivo deste trabalho é prover uma camada multimídia genérica e reutilizável que possa satisfazer os requisitos de armazenamento e busca das mais variadas aplicações que trabalhem com dados multimídia. Utilizando um padrão para descrição de dados multimídia, o MPEG-7 (descrito no capítulo 3), deseja-se obter interoperabilidade com outros

sistemas que tenham conformidade ao padrão. O ponto inicial, ou a grande motivação para o desenvolvimento da camada proposta se dá para realizar a unificação das necessidades de manipulação de dados multimídias das ferramentas do projeto TIDIA-Ae (discutido no capítulo 4), que fazem uso intensivo de material multimídia devido às novas possibilidades da Internet com banda irrestrita.

Com isso, a intenção dessa camada é de servir como mediadora para integrar essas diversas aplicações distintas em se tratando dos seus requisitos multimídia, podendo ser elas num primeiro momento as ferramentas do TIDIA-Ae. Dessa forma, todas as aplicações têm à disposição uma infra-estrutura unificada para a inserção e consulta de mídias, além de contarem com metadados baseados no padrão MPEG-7, provendo interoperabilidade com outros sistemas. Interoperabilidade é um ponto muito importante, pois permite que sistemas distintos possam trocar informações, enriquecendo-os. Os sistemas de banco de dados multimídia expostos no capítulo 2 são isolados, não trocam dados entre si, e essa camada provê essa possibilidade de intercâmbio não presente nos outros sistemas.

Com a preocupação de manter as interfaces enxutas e simples de serem usadas pelos autores das ferramentas, foi proposto um modelo de anotação semântica baseado em *strings* (cadeias de caracteres) apoiados sobre os metadados do MPEG-7. Esse mecanismo de consulta de mídias compreende uma das contribuições deste trabalho, e conforme explicitado na subseção 2.3.1, contempla o método de consulta que não é definido pelo MPEG-7. Esse modelo (ou método) de anotação semântica é detalhado na subseção 5.1.1.

1.2 ORGANIZAÇÃO DA MONOGRAFIA

Esta monografia está organizada do seguinte modo: no capítulo 2 são apresentadas características dos dados multimídia e dos sistemas de banco de dados multimídia, para contextualizar o escopo multimídia com a camada proposta. No capítulo 3 é apresentado o padrão MPEG-7 de descrição de dados multimídia, que é a base para a construção da proposta. No capítulo 4 são apresentadas características e necessidades do projeto TIDIA-Ae, ambiente onde a camada foi aplicada. No capítulo 5 é apresentada a camada proposta, em detalhes.

2 BANCO DE DADOS MULTIMÍDIA

Este capítulo apresenta características dos dados multimídia, modelos de dados multimídia e dos sistemas de banco de dados multimídia, para contextualizar o escopo multimídia com a camada proposta.

2.1 BD VERSUS BDMM – A NATUREZA DOS DADOS MULTIMÍDIA

O conceito de Banco de Dados (BDs) define uma coleção de dados operacionais armazenada em um meio computacional de acordo com um conjunto de regras específicas. A criação e a manutenção de um BD são realizadas por um conjunto de software denominado Sistema Gerenciador de Bancos de Dados (SGBD). Um SGBD é implementado de acordo com um paradigma computacional e visa suprir as necessidades de manipulação de grandes volumes de dados em diferentes aplicações, provendo consistência, concorrência, integridade, segurança e disponibilidade dos mesmos.

SGBDs provêm recursos para a fácil manipulação, consulta e recuperação dos dados relevantes desse grande volume armazenado, sendo que a recuperação é feita abstraído-se os detalhes de acesso aos mesmos. Convencionalmente, os tipos de dados que as aplicações usualmente tratam são textuais e numéricos, direcionando o desenvolvimento e a otimização dos SGBDs para gerenciá-los.

Nos dias de hoje é crescente o uso de dados não convencionais, tais como, dados multimídia. De acordo com o que é definido em [SUBRAMANYA, 2000], dados multimídia geralmente referem-se a imagens digitais, músicas, áudio, vídeo, animações e gráficos – junto com dados textuais – semanticamente relacionados e conjuntamente manipulados. BDs e respectivos SGBDs que fornecem suporte para esses tipos de dados vêm sendo denominados Bancos de Dados Multimídia (BDMM).

Em se tratando de dados multimídia baseados em tempo, é definido em [GIBBS; BREITENEDER; TSICHRITZIS, 1994] a noção de mídias contínuas e não-contínuas. Mídias contínuas são fluxos contínuos onde há um elemento único para cada valor de tempo. Nesses fluxos, dado um instante s_i e uma duração d_i , $s_{i+1} = s_i + d_i$, para $i = 1, \dots, n-1$, sendo áudio e vídeo

exemplos desse tipo de mídia. Mídias não-contínuas apresentam intervalos e/ou sobreposição de elementos, resultando que $s_{i+1} \neq s_i + d_i$, sendo exemplos desse tipo de mídia animações e músicas, onde na animação pode haver um intervalo entre um objeto e outro, e em uma música, que é representada por notas, uma sobreposição das mesmas para a formação de acordes.

Bancos de Dados Multimídia têm que lidar com o uso crescente de dados multimídia por diversas aplicações. Para tanto, devem prover todas as funcionalidades de um BD além de novos recursos para o gerenciamento desses dados. Os tipos de informações que provêm de dados multimídia que devem ser gerenciados por um BDMM podem ser classificados em:

- Dados da mídia. Este é o conjunto de *bytes* que representa mídia. Pode ser o resultado da captura, processamento, compressão e armazenamento da imagem, áudio e vídeo. Por exemplo: a captura de um feixe de luz por uma câmera digital, o processamento desse feixe de luz para a transformação no padrão de cores RGB, a compressão da imagem para reduzir o espaço de armazenamento ocupado, e por fim o próprio armazenamento dos *bytes* resultantes em memória.
- Formato de dados da mídia. Contém informação pertencente ao formato dos dados da mídia inserida em memória. Por exemplo, informação sobre taxa de amostragem, resolução, quadros por segundo e sistema de codificação.
- Palavras-chave dos dados da mídia. Contêm descrições relacionadas à geração dos dados da mídia. Por exemplo, para um vídeo, podem ser a data e o lugar da sua criação, a pessoa que o gravou, e a cena que foi gravada.
- Características dos dados da mídia. São derivados da própria mídia e caracterizam o conteúdo da mesma. Por exemplo, distribuição de cores (histograma), texturas e formas presentes na imagem.

Os últimos três tipos são chamados de metadados, pois são informações que descrevem vários aspectos da mídia. O formato de dados da mídia é usado na apresentação do resultado da busca, enquanto os dois últimos tipos são usados como índices no processo de busca.

Algumas características inerentes a esses dados multimídia têm impacto no projeto e desenvolvimento de BDMMs, como seu tamanho, natureza temporal (no caso de áudio, vídeo e animações), riqueza de conteúdo, complexidade de representação e subjetividade de interpretação. O impacto do tamanho das mídias, ou seja, a quantidade de *bytes* necessários para representar o conteúdo dessas mídias, se reflete nas questões de armazenamento, recuperação e

transmissão; a natureza temporal em aspectos de transmissão, sincronização e apresentação; e a riqueza de conteúdo em complexidade de representação e indexação, além da subjetividade de interpretação na formulação de consultas. Das atividades realizadas por um BDMM, algumas são descritas a seguir:

- No armazenamento de dados, o dado é processado, codificado e então armazenado em discos. Devido à grande quantidade de informação, o armazenamento geralmente consiste em hierarquias de armazenamento secundário (discos magnéticos) e terciário (*juke boxes* de mídias ópticas).
- Na construção de índices, os dados são analisados manualmente ou por ferramentas de extração automática de características para derivar os metadados, que por sua vez são organizados em estruturas adequadas para suportar tanto a busca baseada por palavras-chave quanto a busca baseada em conteúdo.
- Na formulação de consultas, a interface ao usuário provê ferramentas e facilidades para a formulação de consultas baseadas em conteúdo que são então convertidas internamente em consultas estruturadas, podendo ser manipuladas pelo sistema de busca.
- Na consulta em si, o processo de busca procura pelos melhores resultados para uma dada consulta usando critérios de busca e medidas de similaridade apropriadamente definidas, usando índices, e então filtra e organiza pela relevância os resultados dessa busca.
- Na sincronização e apresentação, quando diferentes *streams* (fluxos de dados) estão envolvidos na resposta de uma consulta, eles têm que ser recuperados do depósito de dados e sincronizados apropriadamente antes de serem apresentados ao usuário pela interface, para que tenham significado e sejam fáceis de serem compreendidos (Por exemplo, um vídeo combinado com um áudio diferente do original deixa o usuário confuso).

A tabela 1 mostra as principais diferenças entre um BD e um BDMM:

Tabela 1 – Características dos BDs e BDMMs. [SUBRAMANYA, 2000]

Função	Banco de Dados	Banco de Dados Multimídia
Aquisição de dados	Dados são normalmente fornecidos para o BD através de um terminal.	Além dos meios convencionais, dados podem ser fornecidos ao BD por dispositivos como scanners, microfones e câmeras.
Formatos de dados	Dados crus são geralmente armazenados como arquivos binários ou ASCII.	Há muitos formatos, dentre eles, JPEG e GIF para imagens, MPEG e QUICKTIME para vídeo, e MPEG e WAV para áudio.
Armazenamento de dados	Os dados são armazenados sem compressão. Em casos que disponibilidade e tolerância à falhas são requisitos, os dados são replicados em diferentes discos.	Por ocuparem muito espaço, são armazenados com compressão. Várias formas de <i>striping</i> ¹ e outros mecanismos são usados para eficiente acesso aos dados.
Organização de índices	Os índices são organizados em estruturas de dados adequadas: árvores B+ e suas variantes sendo as mais comuns.	A organização dos índices requer estruturas de dados multi-dimensionais, como por exemplo árvores R.
Consultas	A consulta geralmente é em SQL ou variantes, que são baseadas na álgebra relacional e/ou cálculo relacional.	Consultas por palavra-chave podem ser inadequadas para dados multimídia. Mecanismos de buscas por exemplo e buscas por conteúdo são requeridas.
Procura e recuperação	A procura é geralmente baseada na chave dada como parte da consulta. Ela é exata ou por intervalo.	A procura é mais adequada por similaridade. O feedback do usuário para o motor de busca, baseado nos resultados da procura, é requerido.
Transmissão	Geralmente não há restrições rígidas de tempo-real, pois as transferências de dados pela rede são rápidas o suficiente para prover um tempo de resposta aceitável às consultas.	Há restrições mais estritas de tempo-real, QoS e sincronização na transmissão, para que os resultados da consulta sejam significativos devido à natureza temporal do áudio e vídeo.
Apresentação	Os dados constituem-se primariamente de textos, com gráficos ocasionais como de barras e de tendências.	Novos dispositivos devem ser integrados ao sistema, como alto-falantes para áudio e monitores de alta resolução para imagens e vídeo. A apresentação tem que lidar com diferentes mídias.

¹ Striping é o nome do mecanismo que consiste na utilização de vários discos em paralelo para o armazenamento e recuperação em faixas paralelas (stripes), aumentando assim a velocidade de acesso aos dados.

2.2 MODELOS DE DADOS MULTIMÍDIA

Mídias baseadas em tempo contínuo como áudio e vídeo envolvem noções de fluxo de dados, temporização, composição temporal e sincronização. Essas noções são bem diferentes das noções dos dados tradicionais e, por consequência, os modelos de dados tradicionais não são adequados para modelar BMMs. Assim, o problema principal é a descrição de uma estrutura com restrições temporais para mídias em uma forma apropriada para consulta, atualização, recuperação e apresentação.

Uma das primeiras abordagens de modelos de dados multimídia foi introduzida em [GIBBS; BREITENEDER; TSICHRITZIS, 1994]. Nela, o modelo de dados inclui noções de objetos de mídia, elementos de mídia e fluxos temporais. Três mecanismos gerais de estruturação são utilizados: interpretação, derivação e composição de mídias. Esse tipo de modelo de dados é especializado em mídias contínuas como áudio e vídeo.

O modelo de dados chamado de vídeo algébrico é proposto em [WEISS; DUDA; GIFFORD, 1995], sendo usado para compor, localizar e reproduzir apresentações de vídeos digitais. É um modelo de dados importante no sentido de que muitas de suas características influenciaram modelos posteriores. Distingue entre o arquivo de vídeo físico e os segmentos lógicos, permitindo aos usuários comporem apresentações de vídeos (evitando assim replicação de dados físicos) suportados pela álgebra de vídeo. O modelo usa combinações algébricas de segmentos de vídeos (nós) para criar novas apresentações.

Esse modelo de dados consiste de composições hierárquicas de expressões de vídeos criadas e relacionadas umas às outras por quatro operações da álgebra: criação (cria a apresentação), composição (combina vídeos), resultante (define características de saída) e descrição (associação de metadados). Uma vez que a coleção de nós é definida e persistida, os usuários podem consultar, navegar e localizar as apresentações de vídeos armazenadas. Cada expressão algébrica de vídeo denota uma apresentação de vídeo. Porém, esse modelo tem algumas limitações: a dificuldade de adicionar métodos automáticos de segmentação de vídeo e a necessidade de se utilizar motores de RI (recuperação de informação) para a execução de buscas, devido ao uso exclusivo de descrições textuais.

Um esquema geral de modelagem de objetos de vídeos é apresentado por [ZHONG; CHANG, 1997], que incorpora características visuais de baixo nível e agrupamento hierárquico.

Esse esquema provê um framework para extração de objetos do vídeo, indexação e classificação, além de apresentar novos algoritmos de segmentação de vídeo e rastreamento de objetos, baseado nas características mais notáveis de cor e movimento. Por objetos de vídeo, os autores referem-se a objetos de interesse, como regiões da imagem de mais destaque (cor uniforme, padrões de texturas), objetos se movendo em primeiro plano e grupos de objetos que satisfazem a restrições espaciais e temporais (por exemplo, partes distintas de uma pessoa). Podem ser automaticamente extraídos em vários níveis através de mecanismos de segmentação e rastreamento de objetos, sendo assim inseridos em um repositório para consultas futuras.

Objetos de vídeos podem ser agrupados e semântica pode ser associada a esses grupos. Um problema disso é que as características de baixo nível provêm poucas ligações com os conceitos semânticos, e assim foi proposto um modelo hierárquico de representação de objetos nos quais objetos em vários níveis podem ser indexados, localizados e agrupados a conceitos de alto nível. O modelo proposto suporta busca baseada em conteúdo (CBR – *Content Based Retrieval*), porém essas buscas podem apenas recuperar segmentos de vídeo baseados em algumas características espaciais e temporais dos objetos de vídeo consultados.

Um modelo de dados para vídeos é proposto em [JIANG; MONTESEI; ELMAGARMID, 1999], chamado de *VideoText*. É baseado em anotações livres de texto que são associadas aos segmentos lógicos de vídeo, com uma linguagem de consulta correspondente. Diferencia o arquivo físico do vídeo do segmento lógico, sendo que cada segmento pode ter uma ou mais anotações associadas (pode haver também uma anotação para vários segmentos de vídeo). Nesse modelo de dados, os autores consideram um sistema genérico de recuperação de informação com resultados ordenados pela relevância e com a possibilidade de se expressar consultas lógicas e temporais. As consultas são definidas através de variáveis, termos e por um conjunto de operadores (AND, OR, NOT, ADJ).

É proposto em [JIANG; ELMAGARMID, 1998] um modelo de dados de vídeo chamado de *Logical Hypervideo Data Model* (LHVDM), que estende o modelo de dados *VideoText* e é capaz de representar abstrações multi-nível de vídeos. Essas abstrações são representações dos objetos de vídeo de interesse, chamados de *hot objects*, e podem ter relações semânticas com outras abstrações além dos próprios *hot objects*. Os autores consideraram que pouco trabalho foi feito em cima do problema de modelar e acessar objetos de vídeo baseados nas suas características espaciais e temporais, descrições de conteúdo semântico, e associações com outros

clipes de vídeo que estão entre os objetos em si. Os usuários podem querer navegar em bancos de dados de vídeos de uma maneira não seqüencial, assim como nos documentos hipertexto. Assim, o modelo proposto suporta associações semânticas chamadas de *hyperlinks* de vídeo e dados de vídeo lidando com tais propriedades. Desse modo, definiu-se a noção de *hypervideo*.

No conjunto, é proposto um framework para definir, avaliar e processar consultas de vídeos usando restrições temporais, espaciais e semânticas, baseadas no modelo de dados de *hypervideo*. A granularidade de uma consulta pode ser de vídeos lógicos, segmentos de vídeo lógicos, ou *hot objects*, considerando que no último, os resultados são sempre apresentados como segmentos lógicos de vídeo nas quais esses ocorrem. É importante notar que o modelo usa exclusivamente um motor de RI para realizar as consultas.

Na perspectiva de que os BDMMs lidem com os dados multimídia de uma maneira estruturada e considerando que os motores de MM-RI (recuperação de informação multimídia) são usados para buscas através de palavras-chave, foi proposto em [TUSCH; KOSCH; BÖSZÖRMÉNYI, 2000] um modelo de dados multimídia de indexação genérica chamado de VIDEX. Esse modelo introduz meios para estruturar trechos de vídeos e faz a generalização do processo de indexação. Descreve o mundo narrativo como um conjunto de classes e relacionamentos semânticos, incluindo relacionamentos espaciais e temporais entre as classes e segmentos de mídia.

O modelo de indexação define classes que são a base para um sistema de indexação, considerando que classes específicas da aplicação são adicionadas declarando-se subclasses, que são chamadas por classes de conteúdo, herdando das classes base. Assim, o VIDEX introduz conceitos para a estruturação detalhada dos trechos de vídeo e para relacionar as instâncias das classes semânticas com os segmentos de mídia. O sistema de banco de dados multimídia distribuído SMOOTH [KOSCH *et al.*, 2001a] [KOSCH *et al.*, 2001b] [BACHLECHNER *et al.*, 2000] implementa a parte de alto nível do modelo de indexação de vídeo VIDEX e é exposto com mais detalhe na subseção 2.4.1.

A figura 1 mostra as partes de indexação genérica de baixo e alto nível do VIDEX em um diagrama de classes feito na notação da linguagem UML (*Unified Modeling Language*). Nesse diagrama está o conjunto básico de classes em alto nível, que são: *Events*, *Objects*, *Persons* e *Locations*. Essas classes são subclasses de *ContentObject*, que fazem a interface entre as partes de indexação de baixo e alto nível com a classe *MotionObject*.

O VIDEX provê meios para segmentar trechos de vídeo em diferentes granularidades, como *Shot*, *Scene* e *Sequence*. A classe *Frame* denota o ponto de entrada de acesso a características de baixo nível do vídeo, consistindo de uma ou mais regiões, concretizadas através da classe abstrata *Region* que pode ser tanto *BasicRegion* como *VisualObject*. A classe *Region* denota objetos espaciais, enquanto que *MotionObject* denota objetos espaciais e temporais. Isso permite que sejam expressas relações temporais, espaciais e a combinação das duas entre diferentes objetos de baixo e alto nível.

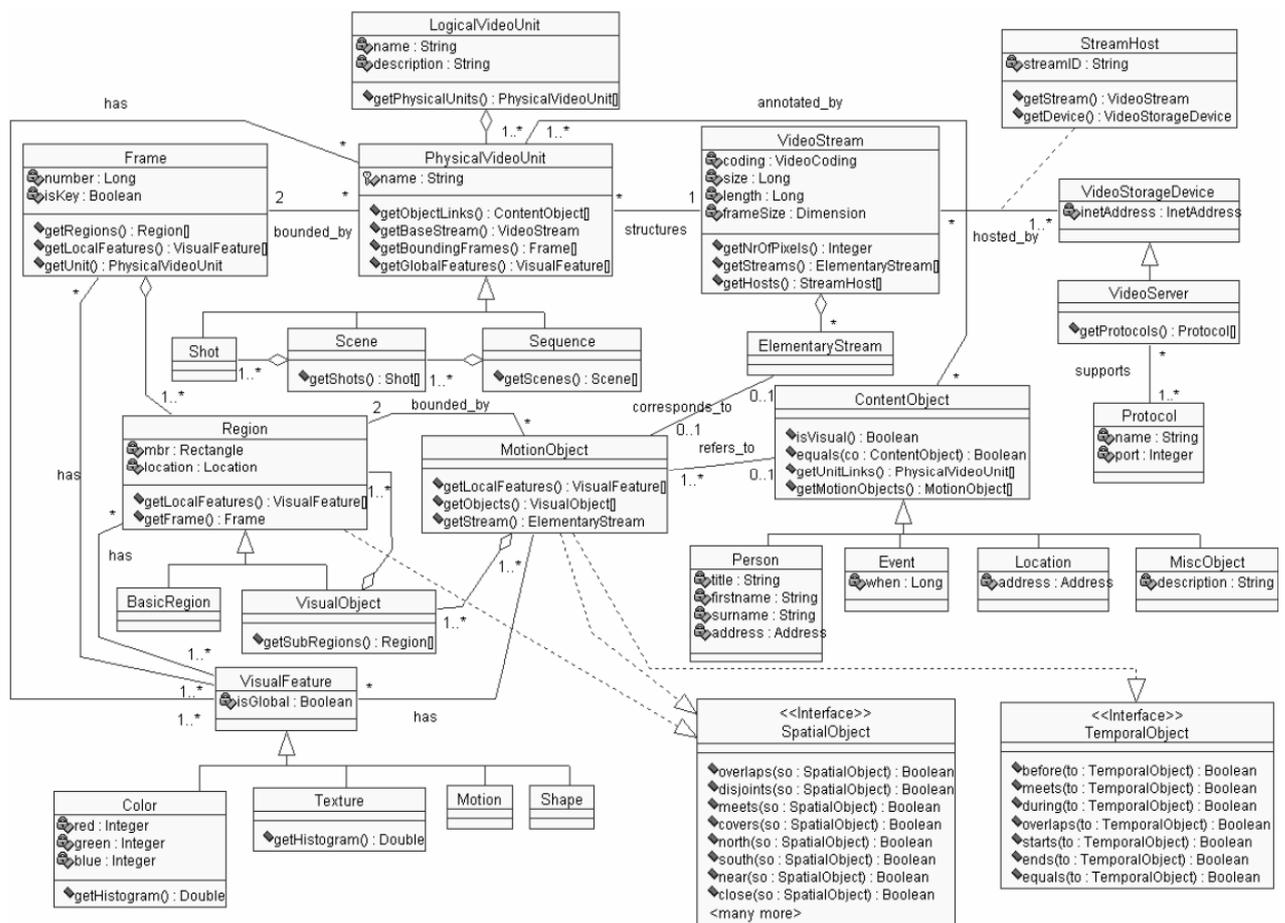


Figura 1 – O modelo de dados do VIDEX [TUSCH; KOSCH; BÖSZÖRMÉNYI, 2000]

2.3 LINGUAGENS DE CONSULTA MULTIMÍDIA

Para buscar dados multimídia descritos por um modelo de (meta)dados, o sistema de banco de dados multimídia deve prover uma linguagem de consulta multimídia. Essa linguagem deve ser capaz de lidar com consultas em que sejam especificadas condições espaciais e temporais, palavras-chave, e também termos subjetivos e objetivos de objetos multimídia. Nesse contexto, foi definido em [GUDIVADA; RAGHAVAN; VANAPIPAT, 1996] um conjunto de classes de consulta genéricas, sendo elas:

- Consulta por atributos subjetivos ou objetivos: a consulta de atributo objetivo consiste de uma maneira estruturada de busca que é semelhante a consultas baseadas em tipos de dados pré-definidos, usados em contextos na qual a forma sintática da consulta pode ser especificada com precisão. Essa forma precisa vem em contraste a situações onde o usuário submete uma consulta que faz uso de lógica nebulosa (*fuzzy*). A consulta de atributo subjetivo é composta de atributos cuja interpretação varia de um usuário a outro. Por exemplo, uma consulta do tipo “dê-me todos os rostos que apresentam um nariz bonito” é subjetiva, portanto depende do gosto do usuário a classificação do nariz.
- Recuperação por navegação: empregado por usuários cuja estrutura e conteúdo de um sistema de banco de dados não lhe sejam familiares. A navegação também é usada para consultas exploratórias.
- Consultas baseadas em conteúdo (CBR): permite consultar características de baixo nível das mídias, como por exemplo, cor, formas e textura. Esse tipo de consulta é detalhado na subseção 2.3.2.
- Consultas correlatas: usadas para reaver mídias que são similares a um objeto de consulta em termos espaciais, temporais ou a combinação dos dois.

É proposto em [BARAL; GONZALEZ; NANDIGAM, 1998] a combinação das consultas multimídia e das linguagens de apresentação, isto é, uma expressão de consulta consistindo em dois componentes: uma linguagem de consulta para descrever quais informações a serem recuperadas e uma linguagem de apresentação para especificar como mostrar os resultados da consulta.

Os autores introduziram o SQL+D, que é uma extensão da linguagem objeto-relacional SQL para multimídia e apresentação, que permite aos usuários especificarem dentro de uma consulta SQL um *layout* onde serão exibidos os resultados. Por exemplo, a tabela “PRAIA” contém informações de áudio e vídeo sobre as praias mundiais: PRAIA (nome: string; país: string; a: audio; v: video). Se um usuário deseja visualizar os vídeos que mostram praias localizadas no Brasil junto com seus trechos correspondentes de áudio, a consulta pode ser formulada de acordo com a sintaxe do SQL+D:

```
SELECT a,v FROM PRAIA
WHERE país ='Brasil'
DISPLAY panel main
WITH a AS audio A, v AS video V
ON main.Center(Overlay),
SHOW V,A
```

O processamento da consulta é mostrado na figura 2. O interpretador do SQL+D recebe a consulta e a divide na parte referente ao SQL e na parte referente à especificação do *DISPLAY*. A seguir, o interpretador requisita uma conexão ao banco de dados, que é obtida pela interface com o mesmo. Essa interface é responsável por iniciar o processamento da consulta SQL. Então, as especificações do *DISPLAY* e o resultado do processamento são enviados como entradas para o controlador de vídeo, que mostra o *layout* multimídia. Há botões que permitem a navegação sobre o conjunto multimídia que resulta da consulta.

Nota-se que o uso de dados multimídia requer a introdução de novos tipos de dados e métodos no SQL+D. Como consequência, há de se empregar o uso de um SGBD relacional-estendido (ou objeto-relacional) para suportar o uso dessa extensão.

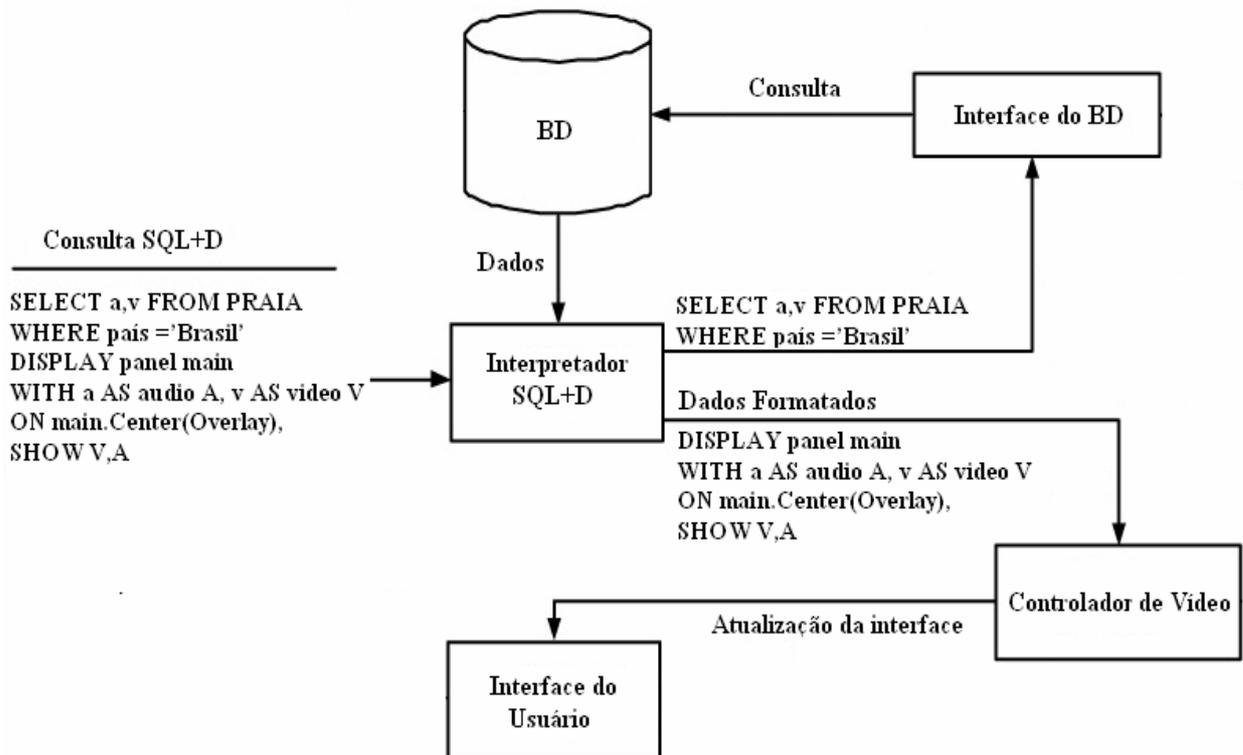


Figura 2 – O processamento de uma consulta SQL+D

2.3.1 SQL/MM e consultas sobre o padrão MPEG-7

A organização ISO/IEC (*International Standards Organization – International Electrotechnical Commission*) publicou em 2001 (há uma versão mais nova do documento de resumo, de 2004 [MARTÍNEZ, 2004]) o padrão MPEG-7. Esse padrão especifica uma maneira padronizada de descrição de diferentes tipos de conteúdo multimídia. As instâncias do conteúdo multimídia são representadas em documentos XML (*Extensible Markup Language*) seguindo as regras definidas pela DDL (*Data Definition Language*), baseado em XML-Schema. A DDL provê meios para definir a estrutura, conteúdo e semântica dos documentos XML. No capítulo 3, o padrão é detalhado.

Nesse mesmo contexto, a ISO/IEC desenvolveu o padrão SQL/MM – extensões multimídia para o SQL [MELTON; EISENBERG, 2001]. O SQL/MM é dividido em partes, sendo que a parte 1 é referente ao framework, a parte 2 ao padrão *Full-Text*, a parte 3 aos dados

espaciais, a parte 5 à imagens e por fim a parte 6 à mineração de dados (*Data-Mining*). A parte 4 que trataria de facilidades de propósitos gerais foi revogada.

Ambos MPEG-7 e SQL/MM introduzem um modelo de dados conceitual multimídia que pode ser usado em sistemas de banco de dados multimídia. O MPEG-7 usa uma extensão do XML-Schema, e o SQL/MM estende o conceito da linguagem objeto-relacional SQL-99 [EISENBERG; MELTON, 1999]. Para ilustrar as diferenças e semelhanças dos dois modelos, é preciso analisar as diferenças do que eles permitem especificar sobre as mídias.

O SQL/MM permite especificar basicamente características de baixo nível de mídias. A seguir serão abordadas as maneiras pelas quais cada modelo lida com imagens. *SI_StillImageType* em SQL/MM é definido da seguinte forma:

```
CREATE TYPE SI_StillImage
AS (
    SI_content BINARY LARGE OBJECT(SI_MaxContentLength),
    SI_contentLength INTEGER,
    SI_reference DATALINK SI_DLLinkControl SI_DLIntegrityOption
        SI_DLReadPermission SI_DLWritePermission
        SI_DLRecoveryOption SI_DLUnlinkOption,
    SI_format CHARACTER VARYING(SI_MaxFormatLength),
    SI_height INTEGER,
    SI_width INTEGER
)
INSTANTIABLE
NOT FINAL
```

SI_MaxContentLength define o tamanho binário máximo de *SI_StillImage* no atributo *SI_content*. *SI_MaxFormatLength* define o tamanho máximo da representação em caracteres de um formato de imagem. Esse formato contém toda informação sobre a codificação, compressão, e assim por diante. *SI_height* é a altura da imagem e *SI_width* a largura, ambos em *pixels*. Os atributos *SI_DLReadPermission*, *SI_DLWritePermission* e *SI_DLRecoveryOption* são relativos ao comportamento interno do SGBD.

Além das definições da imagem, tipos para definir características na comparação de imagens são propostos. O tipo *SI_FeatureList* disponibiliza uma lista de todas características de baixo nível disponíveis (os quatro tipos descritos a seguir). Quanto à cor, é proposto um valor de histograma pelo tipo *SI_ColorHistogram* e duas características de cor: a cor média por *SI_AverageColor* e uma lista de cores dominantes por *SI_PositionalColor*. Quanto à textura, o tipo *SI_Texture* contém valores que representam as características da textura da imagem.

A cor média é uma característica similar, mas não idêntica, à cor predominante de uma imagem. A média é definida como a soma dos valores da cor de todos os *pixels* da imagem dividido pelo número de *pixels* da mesma. Por exemplo, se uma metade dos *pixels* de uma imagem é de cor azul e a outra metade é de cor vermelha, a cor média dessa imagem é roxa.

O histograma de cores da imagem é especificado como uma distribuição de cores na imagem medida sobre um espectro de N cores (geralmente 64 a 256). Valores posicionais de cor indicam a média dos valores de cor dos *pixels* em uma área específica da imagem. A textura da imagem é usada na busca por um padrão, e é medida sobre três fatores: aspereza, contraste e direção da imagem. Uma imagem de um martelo, por exemplo, tem uma direção predominantemente vertical, enquanto que uma imagem de areia na praia não tem uma predominância de direção.

Comparado ao SQL/MM, o modelo conceitual de dados do MPEG-7 é mais rico. Além de representar características de baixo nível das mídias, há uma parte da DDL apresentada a seguir que aponta elementos que não são cobertos pelo SQL/MM. Essa parte se refere à descrição semântica e à decomposição de imagens:

```
<element name="TextAnnotation" minOccurs="0" maxOccurs="unbounded">
...
</element>
<choice minOccurs="0" maxOccurs="unbounded">
  <element name="Semantic" type="mpeg7:SemanticType"/>
  <element name="SemanticRef" type="mpeg7:ReferenceType"/>
</choice>
...
<element name="SpatialDecomposition"
  type="mpeg7:StillRegionSpatialDecompositionType"
  minOccurs="0" maxOccurs="unbounded"/>
...
```

O SQL/MM, ao contrário do MPEG-7, se integra bem nos SGBDs, propondo meios de especificar permissões de acesso, opções de recuperação de falhas, e assim por diante. O MPEG-7 depende da implementação do SGBD para armazenar e indexar os documentos XML, e deve ser associado a mecanismos proprietários de consulta e processamento para ser tão funcional quanto o SQL/MM. Porém, o SQL/MM não provê meios para a troca de dados e metadados multimídia.

2.3.2 Busca baseada em conteúdo

A funcionalidade principal em um banco de dados multimídia trata de como consultar informações multimídia contínuas e não-contínuas com eficácia. No sentido mais amplo, consultas são feitas sobre a extração de características das mídias, que podem ser características baseadas em texto (palavras-chave, anotações) ou características visuais (cor, textura, forma). A primeira classe também é denominada de características de alto nível, e a segunda de características de baixo nível. Visto que já existe literatura ampla no que se refere à extração de características de alto nível, o principal foco de pesquisa é na extração de características de baixo nível. [RUI; HUANG; CHANG, 1997]

Um método amplamente usado, a busca baseada em conteúdo (CBR) de objetos multimídia, se apóia no uso de características de baixo nível extraídas das mídias. Em contrapartida, na busca baseada em características de alto nível, a interpretação semântica dos objetos é adicionada. Por exemplo, objetos podem ser identificados como uma pessoa com nome, e assim por diante. Essa interpretação semântica pode ser adicionada pelo usuário ou obtida através de um processo de indexação semi-automático.

Devido ao uso cada vez mais amplo do CBR e a crescente necessidade de indexação e consulta de dados semanticamente ricos, os sistemas de banco de dados multimídia devem prover meios para realizar CBR assim como complexas consultas estruturadas. Isto implica que a próxima geração de BDMMs deve prover motores genéricos e reutilizáveis para diferentes propósitos de processamento, e devem ser construídos preferencialmente por cima dos motores de banco de dados existentes.

Existem alguns sistemas de CBR de imagens existentes, como os propostos em [SMITH; CHANG, 1997], [DAS; RISEMAN; DRAPER, 1997], [GAVIOLI; BIAJIZ; MOREIRA, 2005] e [HUANG; MEHROTRA; RAMCHANDRAN, 1996], sendo estes dois últimos abordados nas subseções 2.4.3 e 2.4.4. Além desses, há o sistema QBIC (*Query by Image and Video Content*) da IBM (<http://www.qbic.almaden.ibm.com>) [FLICKNER *et al.*, 1995] que suporta CBR de imagens e vídeos, se embasando em vetores extraídos das características de cor, textura, formas, histogramas e movimento de objetos no caso de vídeos. O QBIC está incluso nas extensões de imagem do SGBD comercial DB2 da IBM. Também é usado em CBR de aplicações Web, e um

exemplo disso é um sistema baseado em Web para achar ilustrações na coleção digital do *Hermitage Museum* (<http://hermitagemuseum.org/>).

A tecnologia do QBIC permite a busca de imagens e vídeos baseada em seus conteúdos. Usando esse mecanismo de busca, no caso de imagens, o usuário pode especificar características de conteúdo, como valores de cores e outra imagem a ser usada como entrada para a busca. Essas características são comparadas com as imagens armazenadas no banco de dados, e uma pontuação é atribuída para cada imagem. A pontuação é um valor de ponto flutuante de dupla precisão entre zero e um que indica o quão distante os aspectos de uma imagem estão dos aspectos especificados na consulta. Quanto menor a pontuação, maior a semelhança. Os aspectos da imagem que podem ser usados em consultas são: cor média, distribuição de cores do histograma, valores posicionais de cores e a textura da imagem. Estão condizentes com o padrão SQL/MM descrito na subseção 2.3.1, porém, uma sintaxe diferente de consulta foi definida pela IBM.

Quanto a vídeos, as consultas podem ser feitas nos objetos (ache trechos de vídeo com um objeto redondo e vermelho), nos quadros do vídeo como se fossem imagens (ache imagens com trinta por cento de vermelho e quinze por cento de azul), *shots* (ache trechos de vídeos cuja câmera esteja se movendo da esquerda para a direita) ou qualquer combinação. A identificação dos objetos pode ser feita de modo manual, semi-automático ou automático, sendo este último o mais complexo, eficiente apenas para domínios específicos.

Também dando suporte a CBR de vídeo, há o sistema VideoQ (<http://www.ee.columbia.edu/dvmm/researchProjects/MultimediaIndexing/VideoQ/VideoQ.htm>) [CHANG *et al.*, 1997]. Além dos métodos tradicionais de busca por palavras-chave, incorpora meios para realizar a busca de vídeos baseada em um rico conjunto de aspectos visuais, como cor, textura, forma, movimento e relacionamentos espaciais e temporais. O sistema VideoQ extrai regiões e objetos dos *Shots* de vídeos usando informações de cores e bordas, e isto possibilita acompanhar o movimento. As consultas são feitas em uma interface gráfica com a possibilidade de especificar aspectos de cor, textura e movimento de objetos assim como seus relacionamentos espaciais e temporais com outros objetos.

Em geral, CBR de áudio consiste em achar em um banco de dados ocorrências semelhantes a um padrão de sons. Uma maneira natural de fornecer esse padrão na consulta é por *humming*. Um sistema que suporta consulta por *humming* foi proposto em [GHAS *et al.*, 1995].

Nesse sistema, o QBH (*Query By Humming*), o usuário faz um som no microfone (esse é o ato de *humming*) e o sistema busca sons no banco de dados com um padrão semelhante.

As ferramentas de melodia do MPEG-7 também podem ser usadas na consulta por *humming*. Em um cenário simples, o áudio resultante do *humming* de uma pessoa é analisado pelo contorno e a batida (uso do *MelodyContour DS*) e então comparado com as melodias do banco de dados. Por fim, a melodia com contorno e batida de maior similaridade é selecionada do banco de dados.

Mais recentemente, outras formas de CBR de áudio atraíram atenção. Em um cenário hipotético onde um trecho de uma música é gravado em um dispositivo ubíquo e enviado a um servidor para tentar achar uma música de padrão semelhante, o *AudioSignature DS* do MPEG-7 é um bom representante do conteúdo do banco de dados e do padrão da consulta, já que ele provê uma “impressão digital” única para cada mídia. Alguns dos trabalhos de identificação de áudio baseado no conteúdo que utilizam o padrão MPEG-7 foram feitos por [CRYSANDT; WELLHAUSEN, 2003] e [ALLAMANICHE *et al.*, 2001].

Ambos os trabalhos utilizam o descritor de baixo nível *AudioSpectrumFlatness D* para efetuar a identificação do áudio, já que esse descritor foi feito especificamente para suportar a equiparação robusta de áudios. Ele é estatisticamente resumido no *AudioSignature DS* como uma representação condensada de um sinal de áudio, com o objetivo de prover uma identificação única do conteúdo para o fim de realizar a identificação automática de sinais de áudios.

2.4 SISTEMAS DE BANCO DE DADOS MULTIMÍDIA

Alguns sistemas de banco de dados multimídia que provêm suporte à manipulação de dados multimídia são apresentados nas próximas seções: o SMOOTH, MAE (*Multimedia Authoring Environment*), MIFLIR (*Metric Indexing and Fuzzy Logic-Based Image Retrieval*) e MARS (*Multimedia Analysis and Retrieval System*).

2.4.1 SMOOTH

O SMOOTH é um sistema de banco de dados multimídia distribuído cujo protótipo integra um framework de consultas, navegação e anotação em vídeos, guiado pelo conteúdo de um meta-banco de dados. Esse meta-banco de dados implementa a parte semântica e estrutural do modelo de dados VIDEX, apresentado na seção 2.2.

A figura 3 mostra a arquitetura geral do SMOOTH. O servidor de vídeo provê acesso seletivo aos trechos físicos de vídeo. O projeto usa o servidor de vídeo da Oracle com os protocolos UDP (*User Datagram Protocol*) e RTP (*Real-Time Transport Protocol*).

Quanto ao meta-banco de dados, seus componentes principais são as classes bases: *Events*, *Objects*, *Persons* e *Locations*. Essas classes são subclasses de *ContentObject*, que pode referenciar características de baixo nível de *MotionObject* (cor, textura, etc). Classes específicas da aplicação podem ser adicionadas declarando-se subclasses (classes de conteúdo) das classes base. Ademais, o meta-banco de dados contém classes para uma estruturação detalhada dos trechos de vídeos e para relacionar as instâncias das classes semânticas aos segmentos das mídias.

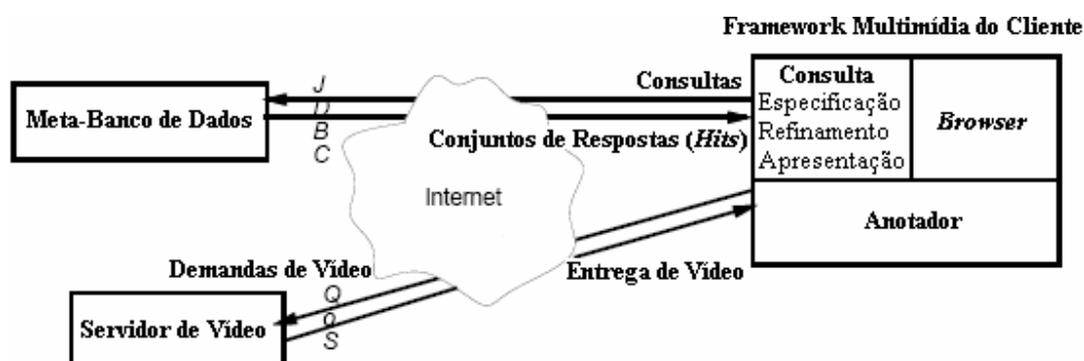


Figura 3 – Arquitetura do sistema multimídia distribuído [KOSCH *et al.*, 2001b]

A interface provê meios para anotar, consultar e navegar sobre os vídeos. Os componentes principais do cliente incluem o anotador (*annotator*), buscador (*querier*), gerenciador de apresentação (*presentation manager*) e navegador (*navigator*). O anotador permite a segmentação do vídeo e a anotação de objetos de alto nível. O buscador segue uma técnica de consulta estruturada baseada em texto, permitindo a construção de consultas de vídeos especificando-se condições sobre os objetos de alto nível e relações temporais ou semânticas a outros objetos. No

gerenciador de apresentação, o usuário pode compor apresentações com restrições temporais a partir de resultados da consulta. O navegador permite a navegação através do conteúdo do meta-banco de dados.

A figura 4 mostra uma tela da interface do buscador. As classes criadas especificamente para a aplicação são detectadas pelo sistema, como a classe *Player* (tipo *PLAYER_T* de *Person*), específica da aplicação de futebol. Nessa interface o usuário pode colocar condições na consulta através do preenchimento de campos e seleções dos menus.

Por exemplo, nessa aplicação de futebol, uma consulta do tipo “Ache todos os trechos de vídeo onde o jogador Silva marca um gol após receber um passe de mais de trinta metros” pode ser especificada. Para a construção dessa consulta, é necessário declarar duas condições, uma para a subclasse *Player* de *Person* e uma para as subclasses *Goal* e *Pass* de *Event*. A última condição é uma restrição temporal, onde o evento de *Pass* tem que anteceder o evento *Goal*, sendo especificado no diálogo localizado no centro da figura 5.

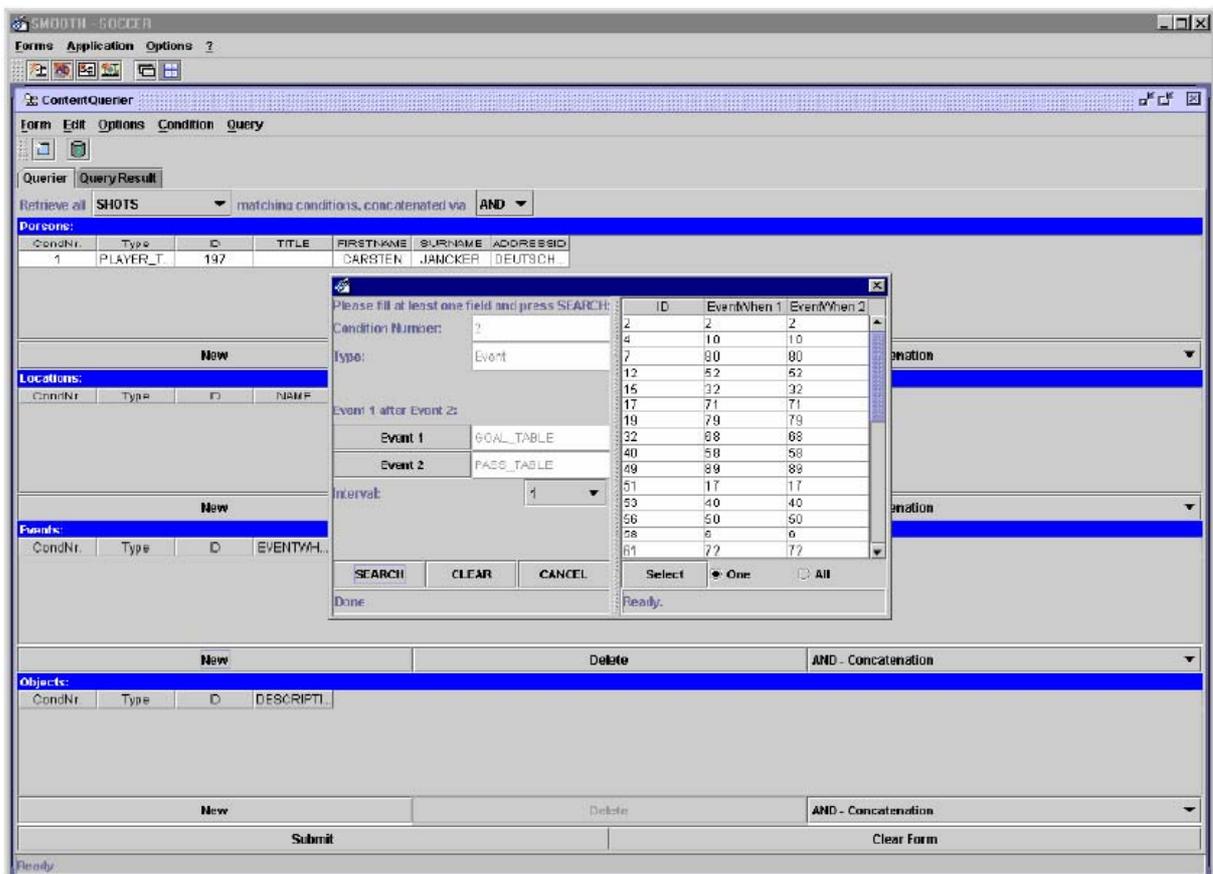


Figura 4 – O buscador com a especificação de uma restrição temporal [KOSCH *et al.*, 2001b]

A figura 5 mostra uma tela da parte de anotação de vídeos do SMOOTH, que implementa a expressividade do modelo de dados genérico VIDEX e isso se traduz em anotações manuais complexas. Contudo, a interface é responsável por ajudar o usuário a anotar os vídeos e a referenciar vídeos já anotados para reutilização de informações. Nessa tela é apresentado um cenário típico de anotação de vídeo: na esquerda, os segmentos já anotados são exibidos, permitindo a definição de segmentos maiores, como *Scenes*, feitos a partir de dois *Shots*. Além disso, para definir as informações dos eventos são usados campos de texto e seleções dos menus.

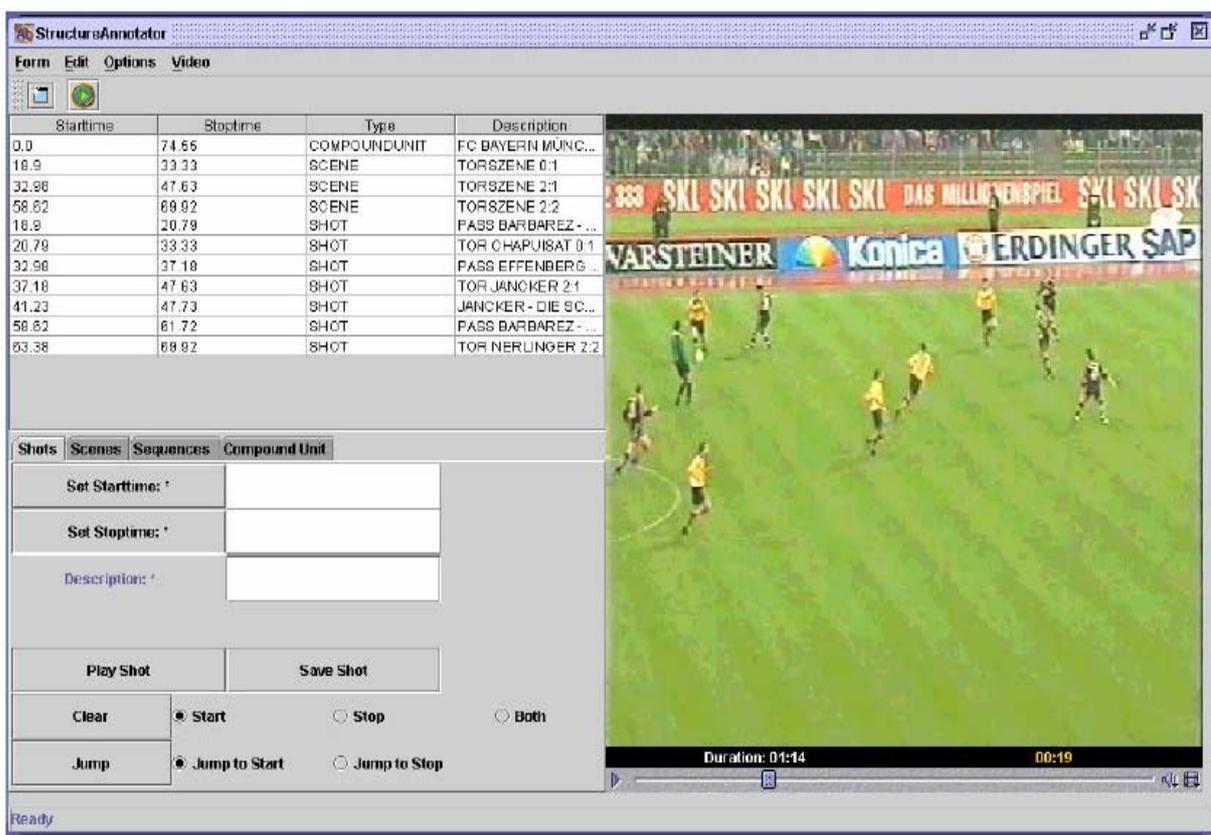


Figura 5 – A interface do anotador de vídeos [KOSCH *et al.*, 2001b]

Há também um framework incluso para a adição de algoritmos de detecção automática de *Shots* ou *Scenes* de vídeo, permitindo uma pré-seleção de segmentos na tentativa de auxiliar na anotação. O gerenciador de apresentação tem a interface disposta da maneira mostrada na figura 6, permitindo que os resultados de uma consulta sejam compostos de forma a organizar uma apresentação de vídeos.

Apresentações complexas podem ser definidas pelos usuários através da imposição de restrições temporais entre vídeos. Essas restrições compreendem a ordenação da seqüência de apresentação dos vídeos e o tempo de espera entre eles. No exemplo da figura 6 está especificado que o segundo vídeo começa a ser apresentado oito segundos após o começo da apresentação do primeiro vídeo. Assim, os dois vídeos rodam em paralelo até que o primeiro termine (A figura 6 é a captura de uma tela onde os dois vídeos estão rodando em paralelo).

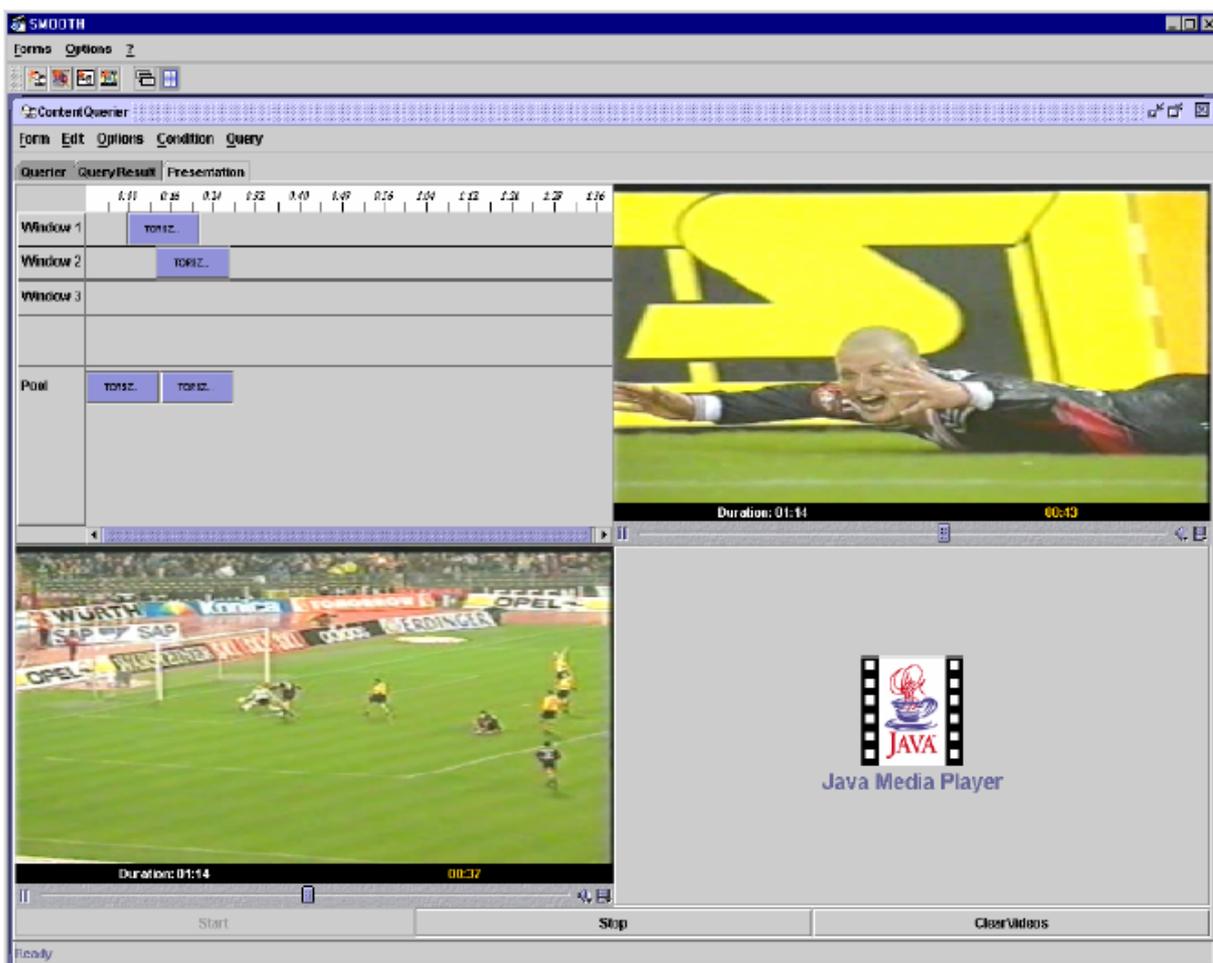


Figura 6 – O gerenciador de apresentação [KOSCH *et al.*, 2001b]

2.4.2 MAE (*Multimedia Authoring Environment*)

De acordo com a ferramenta de autoria multimídia MAE proposta em [SANTOS; VIEIRA; FIGUEIREDO, 2002], a principal característica desse tipo de ferramenta é fornecer funcionalidades que permitam a integração de várias mídias de forma a construir uma aplicação que pode ser mostrada através de uma ferramenta de apresentação. O maior diferencial da ferramenta MAE é o fato dela permitir a utilização de mais de uma aplicação, como, por exemplo, na reutilização de partes de aplicações já desenvolvidas. Para o uso e reutilização de aplicações, a ferramenta MAE habilita o armazenamento em um SGBD, característica ausente nas outras ferramentas de autoria.

A criação e alteração das apresentações multimídia são baseadas no padrão SMIL (*Synchronized Multimedia Integration Language*), que é um padrão complementar ao MPEG-7 para representar, descrever e transmitir apresentações multimídia [ZHOU; GEDEON; JIN, 2005]. O SMIL é um padrão proposto pela organização W3C (*World Wide Web Consortium*) e tem por objetivo proporcionar uma maneira fácil de criar apresentações audiovisuais interativas. Os documentos SMIL são feitos em XML, assim como os documentos MPEG-7.

O objetivo da ferramenta é auxiliar, via Web, a construção, manipulação, armazenamento e recuperação (através da estrutura semântica do ambiente AMMO) de aplicações multimídia baseadas no padrão SMIL, sendo que é oferecido um ambiente visual para a realização dessas operações. Ela é composta pelos módulos: editor de propriedades, editor textual, editor espacial, árvore de objetos, editor temporal e editor de navegação. Enquanto a aplicação está sendo manipulada, a ferramenta mantém a árvore de objetos em memória até o momento da aplicação ser salva em um documento SMIL e/ou no SGBD. Essa árvore mostra através de uma hierarquia os objetos que compõem uma aplicação.

O editor de propriedades mostra as propriedades do objeto selecionado na árvore de objetos, editor espacial ou temporal. O editor espacial possibilita efetuar operações sobre as mídias que fazem parte da aplicação, como incluir, apagar, recortar, colar, copiar e alinhar. O editor temporal permite especificar como será a apresentação das mídias. O padrão SMIL foi determinante na definição do modelo de edição temporal na MAE, e como utiliza os conceitos de blocos paralelos e seqüenciais na apresentação, eles foram adotados na edição temporal da ferramenta.

A figura 7 mostra a interface do editor temporal, com a árvore de objetos no canto superior esquerdo (com a aplicação “*Application 1*” carregada), tendo o editor de propriedades logo abaixo. A orientação horizontal no editor temporal denota seqüência temporal de apresentação das mídias, e a vertical denota paralelismo (vários níveis) na apresentação das mesmas. As figuras 7(a) e 7(b) indicam os dois blocos configurados para a apresentação, sendo o primeiro paralelo e o segundo seqüencial (tem apenas um nível). Há um intervalo de um segundo entre os dois blocos, no instante do 19º segundo. Mídias podem aparecer em vários blocos, como é o caso de *image2* e *text3*.

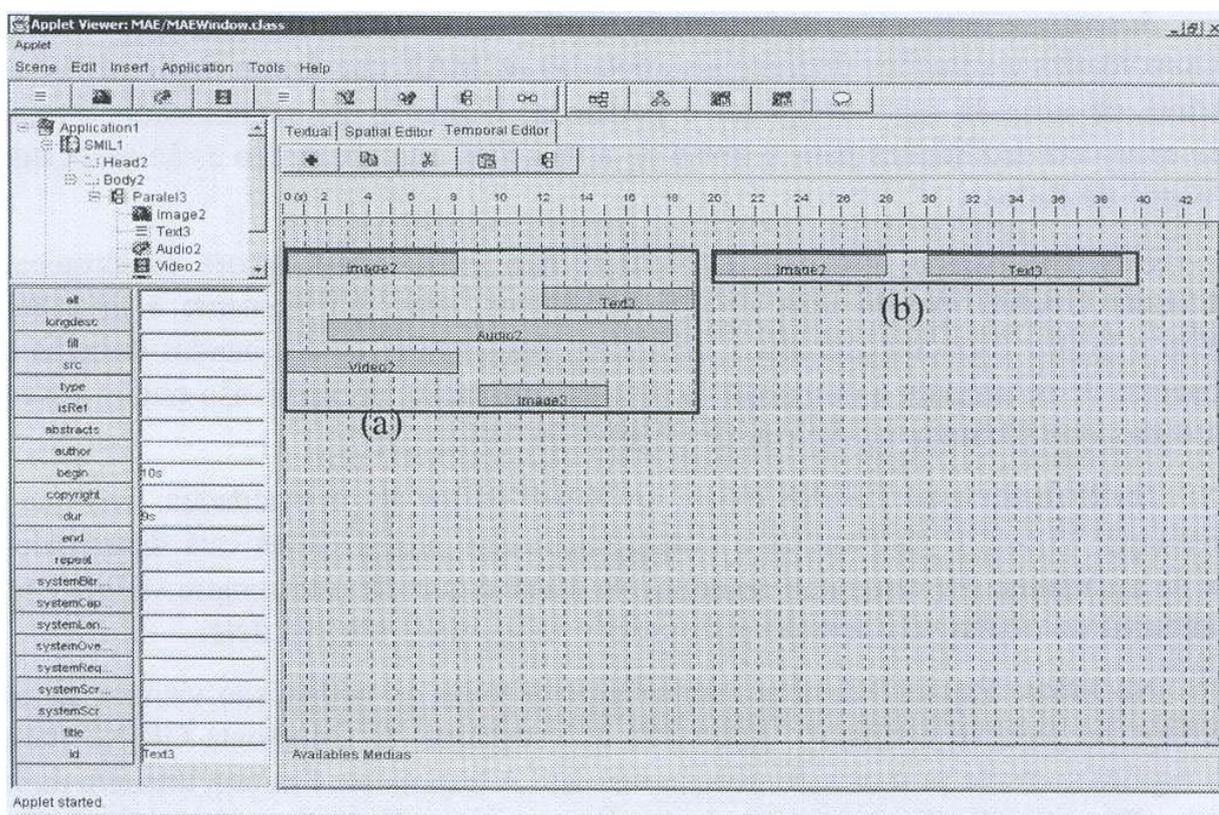


Figura 7 – O editor temporal da ferramenta MAE

Essa ferramenta está inserida no ambiente AMMO (*Authoring and Manipulation of Multimedia Objects*) que é desenvolvido pelo grupo de banco de dados do departamento de computação da universidade federal de São Carlos. Esse ambiente provê suporte à autoria, armazenamento e manipulação de aplicações multimídia. A arquitetura desse ambiente é ilustrada na figura 8, possuindo os seguintes componentes:

- Multimedia Objects Server (MmOS), responsável por gerenciar as aplicações multimídia armazenadas em um banco de dados multimídia.
- Multimedia Application WebBuilder (MAW), provê o suporte à autoria de aplicações multimídia na Web. Os módulos do MAW foram implementados em Java, com o objetivo de serem executados em qualquer navegador Web. É nesse módulo que a ferramenta MAE está inserida.
- Interface Web, provê uma interface para realizar consultas ao banco de dados multimídia e construir aplicações através da Web.
- Sistema Gerenciador de Banco de Dados Orientado a Objetos (SGBDOO), gerencia os objetos mantidos no banco de dados multimídia.

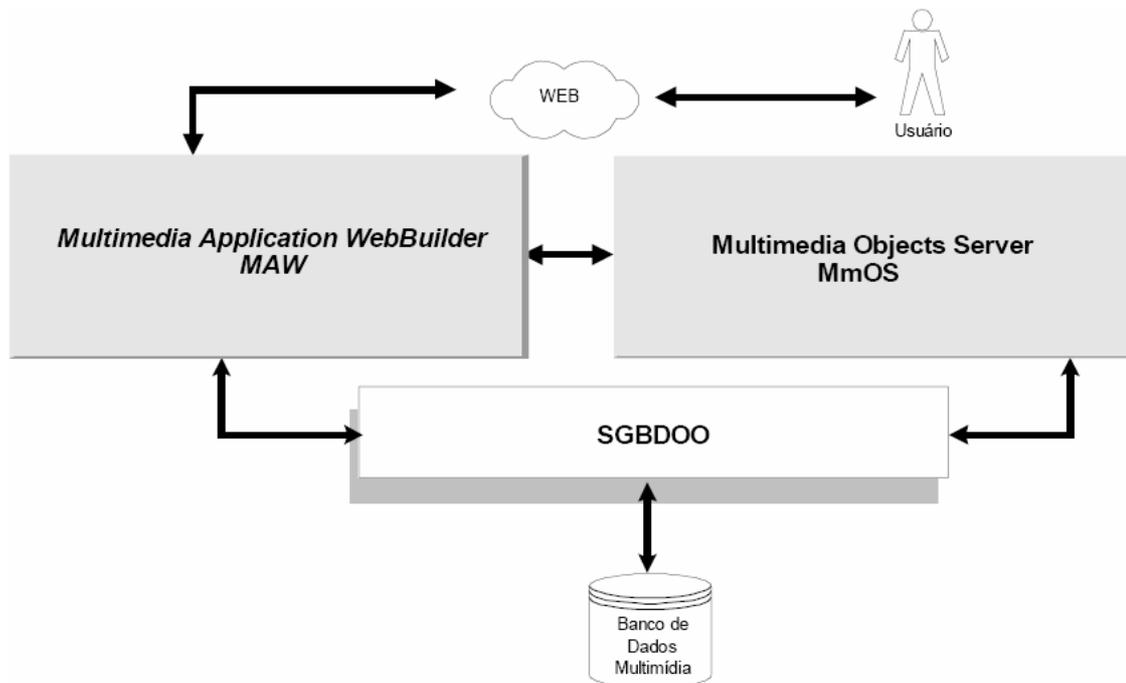


Figura 8 – Arquitetura do ambiente AMMO

2.4.3 MIFLIR (*Metric Indexing and Fuzzy Logic-based Image Retrieval System*)

O trabalho proposto em [GAVIOLI; BIAJIZ; MOREIRA, 2005] trata de um sistema para realizar busca de imagens baseada em conteúdo (CBIR – *Content Based Image Retrieval*). Especificamente, esse sistema trata de imagens de faces humanas vistas de frente que são

indexadas e buscadas por predicados *fuzzy*. Esses predicados são gerados pela combinação de cinco medidas: comprimento do rosto, do queixo, do nariz, e largura da boca e do rosto, e pelos conjuntos de qualificadores *fuzzy* (termos lingüísticos) dessas medidas. Por exemplo, o conjunto de qualificadores para comprimento de rosto pode ser $Q = \{\text{muito curto, curto, médio, longo, muito longo}\}$ e assim, um predicado *fuzzy* que poderia ser usado é “o comprimento da face é muito longo”.

Esses predicados *fuzzy* reduzem a lacuna semântica que existe entre elementos de alto nível (termos usados por serem humanos para indicar suas percepções visuais) e de baixo nível (medidas geométricas das imagens) relacionados a conjuntos de imagens de faces humanas. Isso ajuda a melhorar a eficiência de busca das imagens, como por exemplo, diminuindo o *overhead* da busca.

O sistema pode ser dividido em três módulos. O módulo 1 foi criado para permitir uma tarefa fundamental para o uso de predicados *fuzzy*: a inserção de funções membro que correspondem às cinco características geométricas que são usadas para gerar predicados. Com o uso dessas funções, esse módulo torna possível o cálculo automático dos valores “verdade” dos predicados *fuzzy* usados para representar cada uma das imagens.

O módulo 2 permite que o usuário defina consultas *fuzzy* pelas medidas geométricas. Um exemplo de consulta seria: “(*Face Length: Long or Face Width: Wide or Chin Length: Short*) and (*Mouth Width: Narrow and Nose Length: Short*)”. Através do uso de operadores lógicos e sub-expressões, o usuário tem alta flexibilidade para formular consultas. Ele inclusive pode indicar a relevância da sub-expressão, numa escala de 1 a 100 (que indica, na realidade, a porcentagem de relevância).

Esse módulo também permite escolher o tipo de consulta por similaridade a ser aplicado a cada predicado. Os tipos de consulta por similaridade podem ser *range* e K-NN. O primeiro se baseia na distância entre o elemento consultado e os demais num dado raio, e o segundo no número K de elementos mais próximos do consultado. A tabela 2 ilustra esse cenário:

Tabela 2 - Exemplo de parâmetros da consulta [GAVIOLI; BIAJIZ; MOREIRA, 2005]

Predicado <i>Fuzzy</i>	Tipo da consulta	Raio do <i>range</i> / K-NN vizinhos
<i>Face Length: Long</i>	<i>range</i>	0.25
<i>Face Width: Wide</i>	K-NN	5
<i>Chin Length: Short</i>	K-NN	3
<i>Mouth Width: Narrow</i>	<i>range</i>	0.49
<i>Nose Length: Short</i>	K-NN	6

O módulo 3 é a parte do sistema que corresponde à implementação dos algoritmos responsáveis por efetuar consultas *fuzzy*, incluindo o gerenciamento de índices. Os algoritmos desse módulo usam a *HCS-tree* para indexar os predicados que representam as imagens de faces humanas, gerando instâncias que são usadas para buscar as imagens que satisfazem às consultas formuladas pelo módulo 2.

2.4.4 MARS (*Multimedia Analysis and Retrieval System*)

O MARS [HUANG; MEHROTRA; RAMCHANDRAN, 1996] [MEHROTRA et al., 1997] é um projeto que junta vários pesquisadores de várias áreas com o objetivo de desenvolver um sistema de banco de dados multimídia eficaz. O primeiro passo desse projeto foi a implementação de um sistema de consulta de imagens, cujo protótipo suporta consultas por similaridade e conteúdo (CBIR), este último tipo baseado na combinação de cores, texturas e formas.

Histogramas 2D sobre as coordenadas HS do espaço de cores HSV são utilizados para representar as cores. Para representar a forma são empregados descritores de Fourier, e para a textura são usados valores de espessura, contraste e direcionalidade.

Esse sistema permite a utilização de operadores booleanos para a elaboração de consultas complexas, mas sem o uso conjunto de graus de relevância. As características empregadas durante a construção dessas consultas podem ser especificadas por meio da seleção de imagens-exemplo ou escolhendo-se, por exemplo, texturas de um conjunto de padrões disponíveis ou cores de uma paleta. Tudo isso é feito através de uma interface gráfica que é provida ao usuário do sistema.

Um dos métodos para a recuperação de imagens emprega o conceito de conjunto nebuloso, já que, para uma determinada característica, as distâncias entre uma imagem de consulta e aquelas de uma base de dados são interpretadas como graus de pertinência ao conjunto nebuloso dos objetos que satisfazem tal característica. No entanto, o outro método de recuperação, denominado *probabilistic boolean retrieval*, baseia-se na medição de probabilidades no lugar de graus de pertinência.

2.5 CONSIDERAÇÕES FINAIS

Nesse capítulo foram mostrados conceitos de banco de dados multimídia e sistemas de banco de dados multimídia para contextualizar com a camada proposta. Essa camada pode servir de infra-estrutura para novos sistemas de banco de dados multimídia, sendo que alguns existentes foram descritos nesse capítulo. Também foram discutidos modelos de dados multimídia, que subsidiaram a escolha do MPEG-7 como proposta de modelo multimídia a ser empregado nessa camada. Finalmente, foram discutidas consultas sobre dados multimídia, mais precisamente o SQL/MM e consultas sobre o padrão MPEG-7. Essa discussão foi feita para contextualização com o modelo de consultas proposto nessa camada (detalhado na subseção 5.1.1), feito com base no MPEG-7, que não define a forma de consultar os dados.

3 O PADRÃO MPEG-7

Com o crescimento cada vez maior da disponibilidade de conteúdo digital multimídia proveniente de várias fontes, como *scanners*, mídias ópticas, câmeras digitais, gravadores de áudio e vídeo digitais, redes de difusão e bibliotecas na Internet, achar o que é preciso se tornou mais difícil. Para lidar com esse desafio, os avanços nas áreas de anotação, indexação e processamento de linguagem natural possibilitaram que motores de busca mais sofisticados fossem desenvolvidos para facilitar a procura por conteúdo multimídia. Porém, com algumas exceções, grande parte da procura é centrada em texto, e assim dependente das palavras-chave usadas para indexar o conteúdo multimídia e dos detalhes intrínsecos aos motores de busca.

Mesmo com um bom conhecimento do motor de busca, os usuários ainda encontram dificuldades em localizar o conteúdo que procuram, ou recebem muitas respostas devido a ambigüidades, exigindo assim um tempo considerável para afinar a procura por conteúdo multimídia. Além disso, lidar com os detalhes de diferentes motores de busca para que consultas eficientes possam ser conduzidas requer um esforço considerável. Uma solução que permita busca centrada em multimídia é requerida.

Busca e navegação são essencialmente aplicações denominadas *pull applications*. Filtragem de conteúdo multimídia, como o de redes de difusão, é outro tipo de aplicação, denominado *push application*. Aqui, o termo difusão inclui TV aberta, TV a cabo, TV via satélite, e outras aplicações onde o conteúdo multimídia é fornecido ao usuário. Guias de televisão com uma breve descrição textual ou visual de certos programas existem, porém, enquanto os usuários têm acesso a centenas de canais de TV, os recursos disponíveis para filtrar e selecionar o conteúdo de interesse são muito limitados, exigindo esforço manual.

Para aprimorar a busca e filtragem, o conteúdo multimídia precisa ser descrito efetivamente. Para essa finalidade, descritores de metadados são requeridos para expressar várias características do conteúdo multimídia. Além disso, a padronização das características se mostra importante para permitir que dispositivos ou sistemas de diferentes fabricantes sejam interoperáveis. Mesmo quando as aplicações de diferentes fabricantes são similares, as características e descritores que eles suportam podem ser um pouco diferentes devido à falta de padronização.

Considerando a situação descrita, reconhecendo a demanda da indústria e a necessidade da disponibilidade de uma tecnologia adequada, o grupo MPEG (*Moving Picture Experts Group*) da organização ISO iniciou o trabalho no MPEG-7, seu quarto padrão, em 1998. O MPEG-7 foi denominado formalmente de interface de descrição de conteúdo multimídia, lidando com dados sobre os dados multimídia, diferentemente dos padrões MPEG anteriores como o MPEG-1, MPEG-2 e MPEG-4 que lidam com codificação de áudio e vídeo.

Das publicações feitas na época em que se tornou padrão internacional, é introduzido em [CHANG; SIKORA; PURI, 2001] os objetivos, contexto, componentes e sua relação com outros padrões de metadados existentes; [SIKORA, 2001], [BOBER, 2001], [MANJUNATH *et al.*, 2001] e [JEANNIN; DIVAKARAN, 2001] tratam da parte visual; [QUACKENBUSH; ADAM LINDSAY, 2001], [CASEY, 2001] e [CHARLESWORTH; GARNER, 2001] tratam da parte de áudio; [SALEMBIER; SMITH, 2001] trata dos *Description Schemes* para conteúdo multimídia; [AVARO; SALEMBIER, 2001] apresenta uma visão geral do desenvolvimento dos sistemas MPEG-7 e [HUNTER, 2001] introduz uma linguagem padronizada pelo MPEG-7 denominada DDL (*Description Definition Language*).

Formalmente, o padrão MPEG-7 é referido como ISO/IEC 15938, sendo organizado nas seguintes partes (ou componentes):

- 1) ISO/IEC 15938-1: Sistemas MPEG-7
- 2) ISO/IEC 15938-2: Linguagem de Definição de Descrições MPEG-7
- 3) ISO/IEC 15938-3: Parte visual do MPEG-7
- 4) ISO/IEC 15938-4: Áudio do MPEG-7
- 5) ISO/IEC 15938-5: *Description Schemes* Multimídia do MPEG-7
- 6) ISO/IEC 15938-6: *Software* de referência ao MPEG-7
- 7) ISO/IEC 15938-7: Conformidade com o MPEG-7
- 8) ISO/IEC 15938-8: extração e uso dos descritores MPEG-7

A parte 1 especifica funcionalidades ao nível de sistema, como a preparação dos descritores MPEG-7 para o transporte e armazenamento eficientes, sincronização de conteúdo e descrições, e desenvolvimento de decodificadores em conformidade ao padrão. A parte 2 especifica a linguagem de definição de descrições (DDL), que define os *Descriptors* (Ds) e *Description Schemes* (DSs), além de permitir criar, estender ou modificar os existentes. A parte 3 define um conjunto de Ds e DSs para lidar com aspectos visuais e a parte 4 para lidar com os

aspectos do áudio. A parte 5 define um framework de alto nível que permite descrever genericamente todos os tipos de conteúdo multimídia, incluindo dados visuais, textuais e de áudio. Esse framework vem em contraste às descrições específicas definidas pelas partes 3 e 4. A parte 6 visa prover uma referência de implementação provinda de grandes contribuições da comunidade, e é conhecido como modelo de experimentação (*XM – eXperimentation Model*). A parte 7 visa prover guias e procedimentos para testar a conformidade das implementações com o MPEG-7, e finalmente, a parte 8 provê informação na extração e uso de algumas ferramentas de descrição, dando um discernimento do *software* de referência.

As próximas seções estão organizadas da seguinte maneira: a seção 3.1 engloba as partes 3, 4 e 5, dando mais ênfase para a última; a seção 3.2 a parte 2; e a seção 3.3 um subconjunto da parte 1.

3.1 *DESCRIPTORS E (MULTIMEDIA) DESCRIPTION SCHEMES*

O quarto padrão especifica *Descriptors* (Ds) e *Description Schemes* (DSs) padronizados para áudio e vídeo, assim como conteúdo multimídia. Especifica também uma linguagem de definição de descrições, a DDL, que possibilita a criação de novos Ds e DSs. Além disso, especifica uma camada de sistema que permite a sincronização e acesso a descrições de metadados MPEG-7 com ou sem os dados multimídia correspondentes. Essas descrições abrangem áudio, vídeo, gráficos, imagem, fala e suas combinações. O MPEG-7 complementa a suíte dos padrões MPEG e visa ser aplicável a muitos formatos existentes, incluindo formatos não-MPEG e formatos que não utilizam compressão.

O objetivo principal é prover interoperabilidade entre sistemas e aplicações usados na geração, gerenciamento, distribuição, e consumo de descrições áudios-visuais. Essas descrições de mídias armazenadas ou transportadas em tempo real ajudam os usuários ou aplicações a identificar, buscar e filtrar informações áudios-visuais. O uso dessas descrições resulta em um framework flexível e escalável para desenvolver serviços que possam ser acessados de uma variedade de terminais, como dispositivos ubíquos e estações de trabalho.

Os Ds definem a sintaxe e semântica de aspectos do conteúdo áudio-visual. Diferentes níveis de abstração são abordados pelo padrão, sendo que em um nível baixo, Ds incluem forma, textura, cor e movimento de câmera e objetos para imagens e vídeos, e para áudio, energia,

harmonia e timbre. Em um nível alto de abstração, Ds incluem eventos, conceitos abstratos, e assim por diante. Ds de áudio e vídeo representam características específicas relacionadas a esses conteúdos, enquanto que Ds genéricos representam aspectos genéricos.

É importante notar que os Ds foram desenvolvidos com o objetivo principal de representar características de baixo nível, enquanto que os DSs foram desenvolvidos com caráter primário de representação de aspectos de alto nível. Os DSs permitem a construção de descrições complexas através da especificação da estrutura e semântica dos relacionamentos entre os Ds e DSs constituintes. Por exemplo, um DS relacionado a segmentos de vídeo especifica a sintaxe e semântica dos elementos que os compõem como decomposição do segmento, atributos individuais (duração, anotações textuais) e relacionamentos entre segmentos. Assim como os Ds, as categorias dos DSs são: áudio, vídeo e genérico. Os genéricos geralmente representam meta-informações genéricas relacionadas a todos os tipos de mídias.

Além de haver Ds e DSs que são derivados intrinsecamente do conteúdo, o MPEG-7 também inclui Ds e DSs relacionados à criação, produção, gerenciamento e acesso de conteúdo áudio-visual. Esses metadados incluem informações sobre o esquema de codificação usado para a compressão do conteúdo (JPEG, MPEG-2), o tamanho dos dados, condições para acesso ao material (informação de direitos de propriedade intelectual), classificação (faixa etária recomendada, e classificação do conteúdo em um número pré-definido de categorias), e *links* para materiais relevantes.

Os *Multimedia Description Schemes* (MDS), como já mencionados na descrição da parte 5 do MPEG-7, especificam um framework de alto nível que permite a descrição genérica de todos os tipos de mídias. Geralmente, os MDS descrevem conteúdos que consistem da combinação de áudio, dados visuais e possivelmente textuais, enquanto que DSs visuais e de áudio se referem especificamente a características especiais do domínio visual e de áudio, respectivamente. A figura 9 mostra um panorama dos níveis e relacionamentos entre níveis na hierarquia dos MDS.

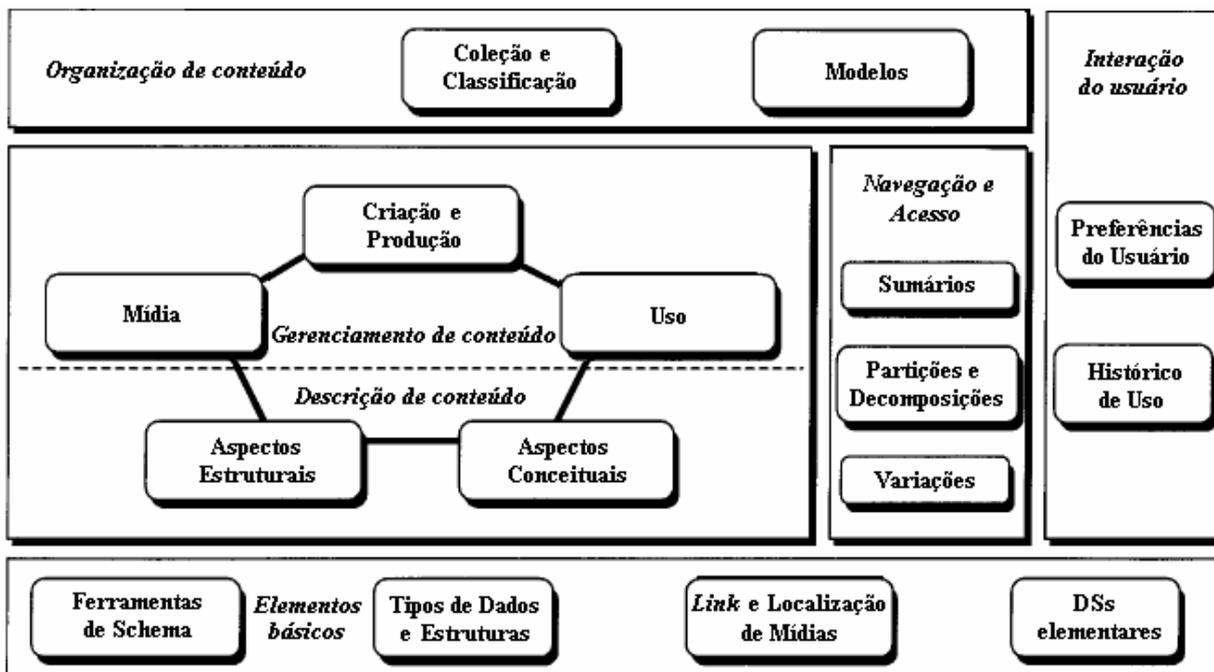


Figura 9 – Panorama dos MDS do MPEG-7 [SALEMBIER; SMITH, 2001]

3.1.1 Elementos básicos

O nível mais baixo, dos elementos básicos, constitui elementos fundamentais para a definição dos DSs. Esse nível consiste de um conjunto estendido de tipos de dados, estruturas matemáticas (vetores e matrizes), ferramentas de localização e ligação (*linking*) de mídias, e DSs elementares que descrevem tempo, local, pessoas, grupos, organizações, e outras anotações textuais. Resumidamente, a seguir são abordados meios para descrever tempo e anotações textuais.

Time DS e *MediaTime DS* descrevem informações temporais no mundo real e nas mídias, respectivamente. Ambos seguem a mesma estratégia que está descrita na figura 10. A figura 10(a) ilustra a maneira mais simples de se descrever um instante e um intervalo. O instante t_1 pode ser descrito usando-se o *TimePoint*, e o intervalo $[t_1, t_2]$ pelo *TimePoint* inicial t_1 com a duração (*Duration*) $t_2 - t_1$. Uma maneira alternativa de descrever um instante é mostrada na figura 10(b), que se baseia no *TimePoint* relativo (*RelTimePoint*). O instante t_1 é descrito por um deslocamento temporal com relação a uma referência t_0 , chamado de *TimeBase*. O objetivo do

TimePoint relativo é definir um instante temporal t_1 , e não um intervalo de duração, como representado na figura 10(a).

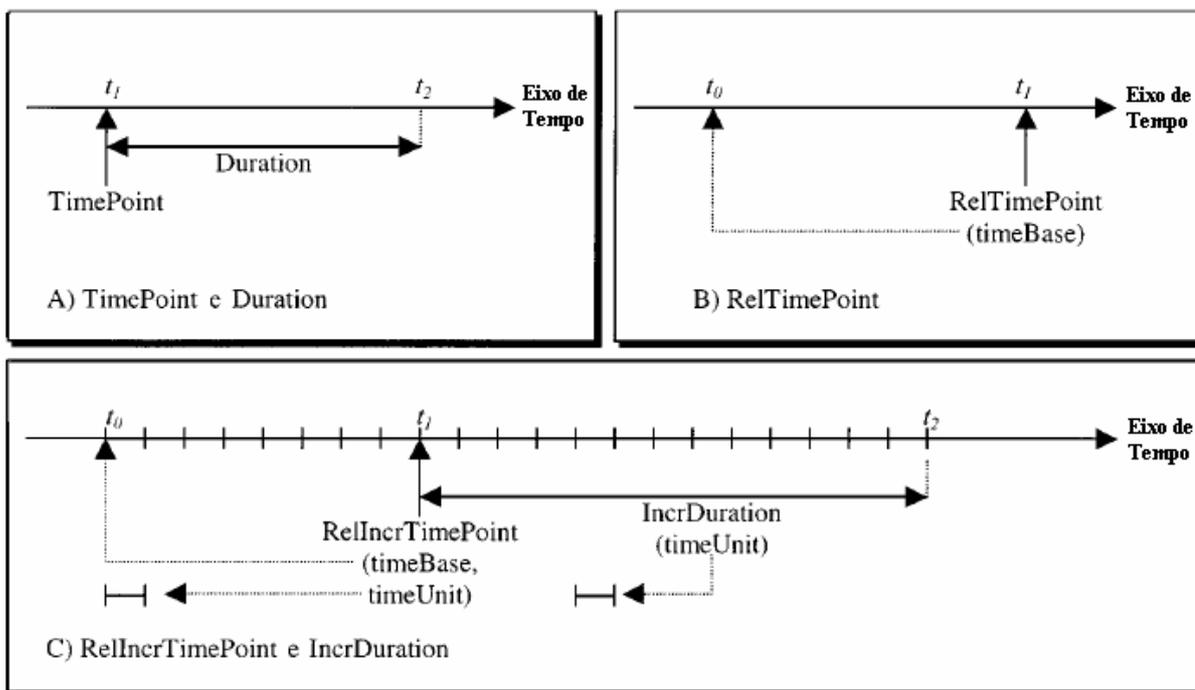


Figura 10 – *Description Schemes* Temporais [SALEMBIER; SMITH, 2001]

Finalmente, a figura 10(c) ilustra a especificação de tempo usando um intervalo pré-definido chamado de *TimeUnit*, contando o número de intervalos. Essa especificação é particularmente eficiente para representar sinais temporais periódicos ou amostrados. Como a estratégia consiste de contar os *TimeUnits*, a especificação do instante tem que ser feito relativo a um *TimeBase*. O instante t_1 é definido com um *TimePoint* relativo incremental (*RelIncrTimePoint*), contando-se oito *TimeUnits* a partir de t_0 . O intervalo $[t_1, t_2]$ também pode ser definido por contagem. Na figura, a duração incremental (*IncrDuration*) é usada para contar treze *TimeUnits* e definir o intervalo $[t_1, t_2]$.

Anotação textual é um componente importante de muitos DSs. O padrão provê construtores básicos para essas anotações, sendo que o construtor mais flexível é o tipo de dados para texto livre. Texto livre permite a formação de uma cadeia arbitrária de texto, que opcionalmente inclui informação sobre a linguagem. O padrão provê também uma ferramenta para realizar anotações textuais mais estruturadas pela inclusão de campos específicos,

correspondentes às perguntas “Quem? Qual objeto? Qual ação? Onde? Quando? Por quê? Como?”. Além disso, anotações textuais mais complexas podem ser definidas através da descrição explícita da dependência sintática entre elementos gramaticais que formam as sentenças (por exemplo, relação entre sujeito e verbo).

Anotações textuais complexas são particularmente úteis em aplicações cujas anotações sejam processadas automaticamente. Por último, o MPEG-7 provê construtores para esquemas de classificação e termos controlados. O primeiro provê um conjunto independente de termos que formam um vocabulário para uma aplicação ou domínio, e o segundo é usado em descrições que fazem referência às entradas nos esquemas de classificação. Permitir que termos controlados sejam descritos por esquemas de classificação oferece vantagens sobre a padronização de vocabulários fixos para diferentes aplicações e domínios, pois é provável que esses vocabulários para aplicações multimídia evoluam com o tempo.

3.1.2 Gerenciamento e descrição de conteúdo

O próximo nível, denominado de gerenciamento e descrição de conteúdo, baseia-se no nível mais baixo. Descreve o conteúdo de vários pontos de vista: criação e produção, aspectos das mídias, uso do conteúdo, aspectos estruturais e aspectos conceituais. Os três primeiros elementos lidam com informação relacionada ao gerenciamento de conteúdo, enquanto que os dois últimos são responsáveis pela descrição de informações perceptíveis (descrição de conteúdo).

No gerenciamento de conteúdo, as informações de criação e produção descrevem a própria criação e classificação do conteúdo áudio-visual. Essas informações provêm um título, que pode ser um texto ou outro material áudio-visual, anotações textuais, informação de autores, locais de autoria e datas. As informações de classificação descrevem em que categorias o material áudio-visual é classificado, como gênero, objetivo, língua, e assim por diante. Também incluem informações guias como classificação etária, orientação aos pais e uma revisão subjetiva. Por fim, as informações de materiais relacionados descrevem se há outros conteúdos áudios-visuais que são relacionados ao que está sendo descrito.

As informações relativas aos aspectos das mídias descrevem o formato, compressão e codificação do conteúdo áudio-visual. O *MediaInformation DS* identifica a mídia mestra, que é a fonte original da qual diferentes instâncias do material áudio-visual são produzidas. Essas

instâncias são referidas como perfis da mídia, que são versões obtidas da mídia mestra possivelmente usando diferentes esquemas de codificação ou formatos diferentes de armazenamento e entrega. Cada perfil é descrito individualmente em termos de parâmetros de codificação, informações de armazenamento e localização.

As informações de uso do conteúdo descrevem aspectos como direitos de uso, registro de uso e dados financeiros. Os direitos de uso não são explicitamente incluídos nas descrições MPEG-7, ao invés disso, referências são fornecidas aos detentores dos direitos. Os DSs de direitos autorais fornecem essas referências na forma de identificadores únicos que estão sob tutela de autoridades externas. Os DSs de registros de uso provêm informações relacionadas ao uso do conteúdo como em redes de difusão, entrega sob demanda, venda de CDs, e assim por diante. Por fim, o DS financeiro provê dados relacionados a custos de produção e à renda resultante do uso do conteúdo.

Na descrição do conteúdo, que aborda aspectos estruturais e conceituais (semânticos), as ferramentas (DSs) estruturais descrevem a estrutura do conteúdo áudio-visual em termos de segmentos de vídeo, quadros, regiões estáticas e em movimento, e segmentos de áudio. As ferramentas semânticas descrevem os objetos, eventos e noções do mundo real que são capturadas pelo conteúdo áudio-visual.

Quanto aos aspectos estruturais, o *Segment DS* descreve o resultado de partições espaciais, temporais ou a combinação dos dois sobre o conteúdo áudio-visual. Esse DS pode descrever uma decomposição recursiva ou hierárquica em segmentos que formam uma árvore de segmentos. O *SegmentRelation DS* descreve relacionamentos espaciais e temporais adicionais entre segmentos.

Para aplicações onde os aspectos estruturais são de pouco ou nenhum uso e o usuário está interessado na semântica do conteúdo, uma abordagem alternativa é provida pelo *Semantic DS*, onde a ênfase muda dos segmentos para os eventos, objetos, lugares e tempo em mundos narrativos. Esse mundo narrativo se refere a um contexto para uma descrição semântica, ou seja, é a “realidade” na qual a descrição faz sentido. A hierarquia de DSs para descrição semântica é apresentada na figura 11.

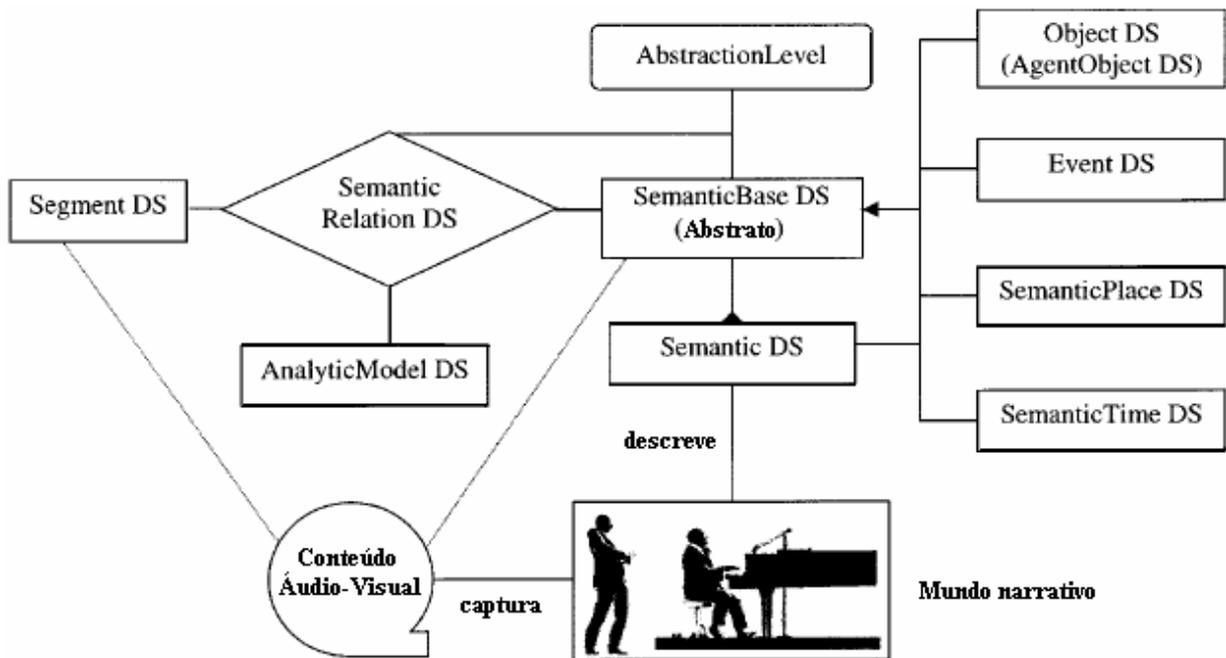


Figura 11 – DSs para descrição de aspectos conceituais [SALEMBIER; SMITH, 2001]

O *SemanticBase DS* descreve mundos narrativos e entidades semânticas nesses mundos. Além disso, DSs especializados são derivados dele, que descrevem tipos específicos de entidades semânticas, como objetos, agentes de objetos, eventos, lugares e tempo. O *Semantic DS* descreve mundos narrativos que são representados ou relacionados a conteúdos áudios-visuais. Na prática, esse DS tem o objetivo de encapsular a descrição do mundo narrativo. O *Object DS* descreve um objeto concreto ou abstrato, sendo que um objeto concreto é uma entidade que existe, ou seja, tem uma extensão temporal e espacial nesse mundo narrativo (O piano de Tom). Já um objeto abstrato é o resultado de aplicar uma abstração a um objeto concreto (qualquer piano). Em essência, isso gera um modelo de objetos.

O *AgentObject DS* estende o *Object DS*. Descreve uma pessoa, grupo, organização ou objetos personalizados (uma xícara falante em um desenho animado). O *Event DS* descreve um evento concreto ou abstrato. Um evento concreto é uma relação dinâmica que envolve um ou mais objetos que ocorrem em uma região, em um tempo e espaço de um mundo narrativo (Tom tocando o piano). Um evento abstrato é o resultado de aplicar uma abstração a um evento concreto (qualquer um tocando o piano). Aqui também se gera um modelo de eventos. Finalmente, *SemanticPlace DS* e *SemanticTime DS* descrevem, respectivamente, um local e um tempo em um mundo narrativo.

3.1.3 Navegação e acesso, interação do usuário e organização do conteúdo

Além dessas descrições relacionadas a conteúdo providas pela parte de gerenciamento e descrição de conteúdo, ferramentas também são definidas para navegação e acesso. Os elementos de sumário, decomposição e variação permitem a adaptação de diferentes apresentações multimídia para as capacidades dos terminais dos clientes, condições da rede e preferências do usuário. Algumas ferramentas são definidas para configurar as preferências do usuário e histórico de uso para melhorar a experiência de interação. O último conjunto de ferramentas lida com a organização do conteúdo por coleções e classificação, e pelo uso dos modelos.

Para sumarização, os principais DSs são: *Summarization DS*, *HierarchicalSummarization DS* e *SequentialSummary DS*. Os sumários do MPEG-7 permitem uma navegação rápida e eficiente por abstrair as informações relevantes do conteúdo áudio-visual. O *Summarization DS* contém *links* para o material áudio-visual ao nível de segmentos e quadros. Pode descrever múltiplos sumários do mesmo material para prover diferentes níveis de detalhamento ou realçar características específicas, objetos, eventos ou semântica. Incluindo *links* para o material nos sumários, são possíveis que sejam gerados e armazenados múltiplos sumários sem armazenar múltiplas versões do material áudio-visual.

O *HierarchicalSummary DS* descreve a organização dos sumários em múltiplos níveis para descrever diferentes níveis de detalhes temporais e o *SequentialSummary DS* descreve um sumário que consiste de uma seqüência de imagens ou quadros de vídeo que podem estar sincronizados com áudio. Esse DS pode também conter uma seqüência de clipes de áudio. O material áudio-visual que faz parte desse sumário seqüencial pode ser armazenado separadamente do material original para permitir navegação e acesso rápidos. Alternativamente, esse DS pode fazer a ligação direta com o material original para reduzir o espaço de armazenamento utilizado.

Quanto às decomposições, o *View DS* descreve a estrutura, partição ou decomposição de um sinal visual ou de áudio em termos de espaço, tempo e frequência. Em se tratando da variação, o *Variation DS* descreve variações do material áudio-visual, como versões em baixa resolução ou em que se utilizou compressão, sumários, diferentes línguas, e diferentes modalidades como áudio, vídeo, imagem, texto, e assim por diante. Uma das funcionalidades providas por esse DS é permitir que um servidor ou *proxy* selecionem a variação mais adequada

do material áudio-visual para efetuar a entrega de acordo com as capacidades de *hardware* dos terminais, condições de rede e preferências do usuário.

O *Variations DS* descreve as diferentes alternativas de variação. Essas variações podem se referir ao conteúdo áudio-visual recentemente feito ou corresponder a conteúdos derivados de outra fonte. O valor de fidelidade da variação indica a qualidade da variação comparada ao original. O atributo de tipo da variação indica o tipo da mesma, como sumário, mudança de modalidade (áudio, vídeo, texto e imagem), tradução da língua, redução de cores, redução espacial, redução de fluxo, compressão, e assim por diante.

A figura 12 ilustra os conjuntos de variações de um material áudio-visual. O exemplo mostra o vídeo original no canto inferior esquerdo (A) e oito variações. Essas variações têm diferentes modalidades: duas são de vídeo (E, H), três são de imagens (B, F, I), duas são de texto (C, G) e uma é de áudio (D). Cada uma das variações tem um valor de fidelidade especificado que indica a fidelidade da variação em relação ao conteúdo original.

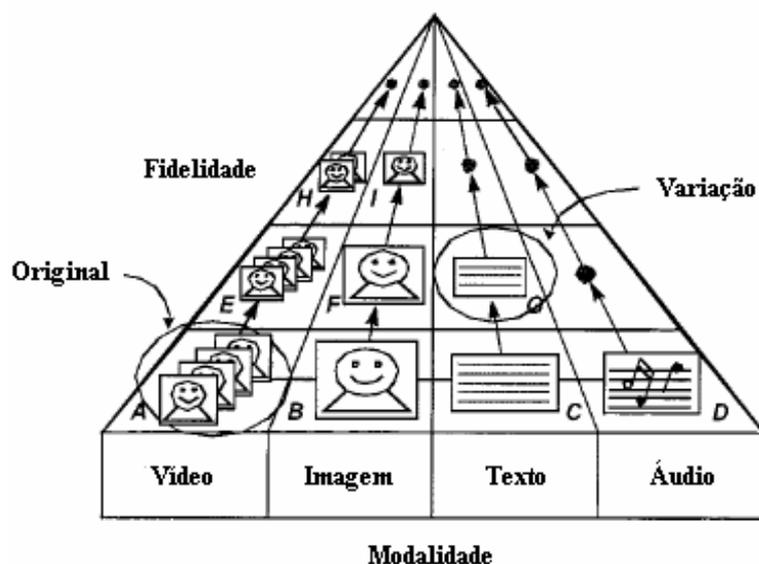


Figura 12 – Variações de um material áudio-visual [SALEMBIER; SMITH, 2001]

Quanto à interação do usuário, o *UserInteractions DS* descreve as preferências dos usuários relativas ao consumo do material áudio-visual, assim como o histórico de uso. Os descritores de conteúdo áudio-visual do MPEG-7 podem ser associados aos descritores de preferências para selecionar e personalizar o conteúdo para acesso, apresentação e consumo mais eficientes e efetivos.

O *UserPreference DS* descreve preferências para diferentes tipos de conteúdo e modos de navegação, incluindo dependência de contexto em termos de tempo e local. Esse DS também descreve o peso da importância de diferentes preferências, as características de privacidade das preferências e se elas são sujeitas à atualização, como por um agente que “aprende” automaticamente através da interação com o usuário. O *UsageHistory DS* descreve o histórico das ações do usuário em um sistema multimídia. As descrições de histórico podem ser trocadas entre usuários, seus agentes, provedores de conteúdo e dispositivos, além de poderem ser usados para determinar as preferências do usuário em relação a um conteúdo áudio-visual.

Na organização do conteúdo, o *Collection DS* descreve coleções do material áudio-visual, de descritores do material, de conceitos semânticos, ou ainda coleções mistas do material e estruturas de coleção em termos dos relacionamentos entre as próprias coleções. Um exemplo seria uma organização de coleções em uma estrutura de coleção. Cada coleção consiste de um conjunto de imagens com propriedades em comum, como por exemplo, cada uma representando eventos similares em um jogo de futebol. Dentro de cada coleção, os relacionamentos entre as imagens podem ser descritos, como o grau de similaridade entre elas. Entre coleções, relacionamentos adicionais também podem ser descritos, como o grau de similaridade das próprias coleções.

O *Model DS* descreve modelos parametrizados de conteúdo áudio-visual, descritores ou coleções. Os modelos podem ser expressos em termos estatísticos ou probabilísticos associados com os atributos das coleções do conteúdo. O *ProbabilityModel DS* descreve diferentes funções estatísticas e estruturas probabilísticas, que podem ser usadas para descrever amostras do conteúdo áudio-visual e classes de descritores usando aproximação estatística.

O *AnalyticModel DS* descreve uma coleção de exemplos do conteúdo áudio-visual ou *clusters* de descritores que são usados para prover um modelo para uma classe semântica em particular. Por exemplo, uma coleção de imagens de arte rotuladas com etiquetas indicando que as pinturas são exemplos do período impressionista forma um modelo analítico. Esse DS também pode opcionalmente descrever a confiança nas quais rótulos semânticos são aplicados. O *Classifier DS* descreve diferentes tipos de classificadores que são usados para associar os rótulos semânticos ao conteúdo áudio-visual ou coleções.

3.2 LINGUAGEM DE DEFINIÇÃO DE DESCRIÇÕES MPEG-7

Nesse trabalho, será adotado por simplicidade o termo DDL (*Description Definition Language*) para designar a linguagem de definição de descrições do MPEG-7. Essa linguagem provê as regras sintáticas para criar, combinar, estender e refinar os *Descriptors* e *Description Schemes*. Para contemplar os interesses de interoperabilidade, que é um dos objetivos do padrão MPEG-7, a linguagem XML *Schema* da W3C adicionada de certas extensões específicas para o MPEG-7 foi selecionada para ser a DDL.

A DDL faz parte do núcleo do padrão MPEG-7 e não é uma linguagem de modelagem como a UML. É uma linguagem de *schema* para representar os resultados de modelagem de dados áudio-visuais (Ds e DSs) como um conjunto de restrições sintáticas, estruturais e de valor. De acordo com os requerimentos da DDL, ela tem que ser capaz de expressar relacionamentos estruturais, de herança, espaciais, temporais e conceituais entre elementos dentro de um DS ou entre DSs, além de prover um rico modelo de *links* e referências entre um ou mais descritores e os dados que eles descrevem. Deve ser independente de plataforma e aplicação, legível por máquinas, e de preferência por humanos.

Também deve ser capaz de especificar tipos de dados para descritores, ambos primitivos (inteiro, texto, data) e compostos (histogramas, tipos enumerados). Em adição a isso, um *parser* (analisador gramatical) capaz de validar a sintaxe dos DSs (conteúdo e estrutura) e tipos de dados dos descritores é requerido. Dado uma descrição MPEG-7, o *parser* também deve ser capaz de verificar a conformidade da descrição com as regras expressas pelos DSs e Ds correspondentes.

Como a linguagem XML *Schema* não foi desenvolvida especificamente para o padrão MPEG-7, algumas extensões específicas para o padrão foram adicionadas na linguagem. Como conseqüência, a DDL pode ser dividida nos seguintes componentes, apresentados nas seções 3.2.1, 3.2.2 e 3.2.3:

- 1) Componentes estruturais do XML *Schema*;
- 2) Tipos de dados do XML *Schema*;
- 3) Extensões específicas do MPEG-7.

3.2.1 Componentes estruturais do XML *Schema*

Os componentes estruturais do XML *Schema* compõem a parte 1 das duas partes da especificação da linguagem. Provêm facilidades para descrever a estrutura e aspectos de restrição do conteúdo de documentos XML, sendo que um *parser* pode usá-los durante o processo de validação. Por exemplo, quais elementos podem ocorrer e com que frequência, que atributo pertence a quais elementos e assim por diante, podem ser especificados. O XML *Schema* provê meios para definir a estrutura de documentos XML. Esses documentos são descritos por um documento *schema* em particular e são chamados de instâncias de documento, sendo válidos apenas se eles estiverem de acordo com todas as restrições especificadas pelo *schema*.

O XML *Schema* tem como características principais: tipos de dados simples e complexos; derivação de tipos e herança; restrições de ocorrência de elementos e declarações de elementos e atributos a par de um *namespace*. Por ter essas características, é possível definir e compelir regras concisas e estritas em relação aos conteúdos de elementos e atributos. Por exemplo, podem-se declarar novos tipos de dados ou reutilizar e derivar de tipos existentes. Similar a documentos de instância, todo documento *schema* consiste de um elemento raiz, que inclui declarações para todos os elementos e atributos que são permitidos ocorrerem em documentos de instância válidos.

Documentos XML basicamente compreendem elementos aninhados. Desse modo, o elemento é uma das mais importantes declarações em um documento de *schema* típico. O exemplo a seguir mostra um simples *schema*:

```
<?xml version="1.0" encoding="ISO-8859-1?">
<schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <element name="Vídeo" type="string"/>
</schema>
```

Essa amostra de *schema* declara um elemento global denominado “Vídeo”. Um elemento global é definido como um filho imediato do elemento *schema*, e apenas esses elementos podem aparecer como raízes em documentos de instância.

Para adicionar informações úteis a um elemento, é possível designar atributos a ele. Por exemplo, o elemento “Vídeo” pode se beneficiar de ter um atributo adicional de linguagem chamado “lang” cujo propósito é especificar a linguagem do vídeo. Essa declaração pode se parecer com o seguinte trecho de documento XML:

```

<element name="VÍdeo">
  <complexType>
    <simpleContent>
      <extension base="string">
        <attribute name="lang" type="language" use="optional"/>
      </extension>
    </simpleContent>
  </complexType>
</element>

```

Com a introdução de tipos como um novo conceito dentro do XML *Schema*, torna-se possível atribuir tipos a declarações de elementos, o que não era possível com os DTDs (*Document Type Definition*). Definições de tipos definem componentes internos do *schema*, que podem ser usados em outros componentes de *schema* como as declarações de elemento ou atributo, ou ainda outras definições de tipo. Há dois tipos de componentes de definição de tipos: tipos simples e tipos complexos.

Tipos simples são usados em elementos que apenas contêm texto, o que significa que não podem incluir elementos aninhados. Os atributos devem sempre ser declarados com tipos simples, pois não podem conter elementos aninhados ou subelementos. Tipos simples podem ser divididos em três categorias: atômicos ou embutidos, tipos de lista e tipos *union*, sendo que os dois últimos são tipos simples agregados.

Tipos atômicos são, em geral, tipos que podem ser encontrados em linguagens de programação ou sistemas gerenciados de banco de dados, como "string" ou "boolean". Tipos de lista são compostos por seqüências de tipos atômicos e separam suas entradas por espaços em branco, e por fim tipos *union* permitem que valores de elementos ou atributos sejam instâncias de um tipo formado pela união de vários tipos atômicos e de listas. Tipos de lista e *union* são discutidos na subseção 3.2.2.

Tipos complexos permitem elementos filhos no seu conteúdo e podem ter atributos, ao contrário dos tipos simples. Definições de tipos complexos provêm restrições na aparição e conteúdo dos atributos, restrições nos elementos dos filhos e derivações dos tipos complexos de outros tipos simples ou complexos através de extensão ou restrição. Tipos complexos podem ser definidos usando-se o elemento *complexType* e tais definições geralmente contêm um conjunto de declarações de elementos e atributos. Por exemplo, o tipo complexo *Organization*, que é definido a seguir, contém três declarações de elementos e uma declaração de atributo.

```

<complexType name="OrganizationType">
  <sequence>
    <element name="OrgName" type="string"/>
    <element name="ContactPerson" type="IndividualType"
      minOccurs="1" maxOccurs="unbounded"/>
    <element name="Address" type="PlaceType"
      minOccurs="1" maxOccurs="1"/>
  </sequence>
  <attribute name="id" type="ID" use="required"/>
</complexType>
<element name="ProdComp" type="OrganizationType"/>

```

Essa definição diz que cada elemento *ProdComp* que apareça em um documento de instância deve ter um elemento *OrgName*, um ou mais elementos *ContactPerson* e um elemento *Address*. O primeiro desses elementos conterá uma *string*, o segundo conterá o tipo complexo *IndividualType*, e o terceiro conterá o tipo complexo *PlaceType*. Qualquer elemento cujo tipo seja *Organization* deve também ter um atributo chamado *id*, que deve ser um identificador único. Um exemplo de uma instância válida é mostrado a seguir.

```

<ProdComp id="UFSCar754">
  <OrgName>Departamento de Computação</OrgName>
  <ContactPerson>Leandro Donaires Figueira</ContactPerson>
  <Address>Universidade Federal de São Carlos</Address>
</ProdComp>

```

As restrições de ocorrência *minOccurs* e *maxOccurs* dizem quantas vezes o elemento pode ocorrer em um determinado local do documento. Quanto à colocação dos elementos, há três maneiras de ser feita. A estrutura *sequence* especifica que os elementos devem estar presentes no documento de instância na mesma ordem em que aparecem no documento *schema*. A estrutura *choice* especifica que apenas um elemento da lista deve aparecer no documento de instância, e a estrutura *all* especifica que todos os elementos devem aparecer, mas em uma ordem arbitrária.

Se uma declaração de tipo complexo é embutida dentro de uma declaração de elemento, esse tipo é chamado de tipo anônimo em vez de um tipo nomeado. Tipos nomeados são necessários quando é preciso compartilhar tipos de dados entre múltiplos elementos diferentes. Tipos complexos são criados derivando-se por extensão ou por restrição. Derivar por extensão resulta que o tipo derivado herda todas as declarações do super tipo. Derivar por restrição faz com que partes das declarações do pai sejam restringidas, como por exemplo limitar o número de ocorrências de um elemento.

3.2.2 Tipos de dados do XML Schema

Os tipos de dados do XML Schema compõem a parte 2 da especificação da linguagem. Provêm um conjunto de tipos de dados primitivos embutidos, um conjunto de tipos de dados derivados e mecanismos nas quais usuários podem definir seus próprios tipos de dados derivados, sendo eles os tipos de dados de facetas, listas e *union*. Esses mecanismos podem ser usados para restringir os possíveis valores dos descritores MPEG-7 dentro das instanciações.

Os seguintes tipos de dados primitivos estão incluídos na linguagem XML Schema: *string*, *boolean*, *float*, *double*, *decimal*, *timeDuration*, *recurringDuration*, *binary*, *uriReference*, *ID*, *IDREF*, *ENTITY* e *QName*. Os seguintes tipos de dados derivados de tipos simples também estão incluídos: *CDATA*, *token*, *language*, *IDREFS*, *ENTITIES*, *NMTOKEN*, *NMTOKENS*, *NCName*, *Name*, *NOTATION*, *integer*, *nonPositiveInteger*, *positiveInteger*, *nonNegativeInteger*, *NegativeInteger*, *long*, *unsignedLong*, *int*, *unsignedInt*, *short*, *unsignedShort*, *byte*, *unsignedByte*, *time*, *timeInstant*, *timePeriod*, *date*, *month*, *year*, *century*, *recurringDate* e *recurringDay*.

Um tipo de dados derivado é definido aplicando-se facetas de restrição a um tipo de dados primitivo ou outro tipo de dados derivado. A tabela 3 lista as facetas disponíveis para gerar novos tipos de dados.

Tabela 3 – Facetas providas pelo XML Schema [HUNTER, 2001]

Facetas	Nome
De limites	minInclusive, minExclusive, maxInclusive, maxExclusive
Numéricas	precision, scale
De tempo/data	duration, period
De padrões	pattern
De enumeração	enumeration
De tamanho	length, minLength, maxLength
De codificação	Encoding (hex ou base64)
De espaços em branco	whitespace

O exemplo a seguir ilustra a aplicação da faceta *maxLength* a um tipo de dados de caractere, que restringe o tamanho máximo da *string*. Nesse exemplo se define um novo tipo simples que é derivado do tipo embutido “string” e não deve conter mais do que quarenta caracteres.

```

<simpleType name="StringExemplo">
  <restriction base="string">
    <maxLength value="40"/>
  </restriction>
</simpleType>

```

O exemplo a seguir ilustra o uso da faceta *pattern*. Elementos do tipo *PhoneNum* são restritos a *strings* de três dígitos, seguido por um hífen e terminando com quatro dígitos, formando um padrão.

```

<simpleType name="PhoneNum" base="string">
  <pattern value="\ d{3}-\ d{4}" />
</simpleType>

```

Tipos de lista são compostos de seqüências de tipos atômicos, separados por espaços em branco. O XML *Schema* tem três tipos embutidos de lista: *NMTOKENS*, *IDRGS* e *ENTITIES*. Além disso, podem-se criar novos tipos de lista por derivação dos tipos atômicos. Não se podem criar listas a partir de outros tipos de lista ou de tipos complexos, porém facetas podem ser usadas (*length*, *minLength*, *maxLength* e *enumeration*) para derivar novos tipos de lista.

Como exemplo, segue a definição e uma instância válida de uma lista de valores inteiros:

Definição:

```

<simpleType name="VetorDeInteiros" >
  <list itemType="integer"/>
</simpleType>

```

Instância:

```

<VetorDeInteiros>1 2 3 4</VetorDeInteiros>

```

Tipos *union* permitem que valores de elementos ou atributos sejam de um tipo feito pela união de vários tipos atômicos ou de lista. No exemplo a seguir, o elemento *Unsigned6OrDirection* pode ter um valor que seja tanto do tipo *Unsigned6* como do tipo *Direction*.

```

<element name="Unsigned6OrDirection">
  <simpleType>
    <union memberTypes="unsigned6 directionType" />
  </simpleType>
</element>

```

3.2.3 Extensões específicas do MPEG-7

Foi necessária a adição das seguintes características para satisfazer os requisitos do padrão MPEG-7: tipos de dados de *array* e matriz, tipos de dados derivados como *MimeType*, *CountryCode*, *RegionCode*, *CharacterSetCode*, *CurrencyCode*, *basicTimePoint* e *basicDuration*, e referências com estruturas de tipos.

O MPEG-7 requer um mecanismo para restringir o tamanho dos *arrays* e matrizes para um valor pré-definido ou atribuído no momento da instanciação (parametrizado). Uma nova faceta, a *mpeg7:dimension*, que é uma lista de inteiros positivos, é disponibilizada para permitir a especificação das dimensões de um *array* ou matriz de tamanho fixo. Para manter a compatibilidade com o XML Schema, os *parsers* ignoram extensões específicas do MPEG-7. Assim, *parsers* do próprio MPEG-7 os validam, e por isso a anotação *appinfo* é requerida. O exemplo a seguir ilustra a definição e a instanciação de uma matriz de inteiros com três linhas e quatro colunas:

Definição:

```
<simpleType name="IntMatrix2D">
  <list itemType="integer">
    <annotation><appinfo>
      <mpeg7:dimension value="unbounded unbounded" />
    </appinfo></annotation>
  </list>
</simpleType>
<simpleType name="IntMatrix3x4">
  <restriction base="IntMatrix2D">
    <annotation><appinfo>
      <mpeg7:dimension value="3 4" />
    </appinfo></annotation>
  </restriction>
</simpleType>
<element name="IntMat3x4" type="IntMatrix3x4" />
```

Instância:

```
<IntMat3x4>
5 8 9 4
7 6 1 2
1 3 5 8
</IntMat3x4>
```

O atributo especial *mpeg7:dim* também é disponibilizado para suportar tamanhos de *arrays* e matrizes parametrizados. Esse atributo é uma lista de inteiros positivos e especifica as dimensões a serem usadas a um tipo lista no momento da instanciação. Segue um exemplo de instanciação usando esse atributo:

```
<IntegerMatrix mpeg7:dim="2 4">
1 2 3 4
5 6 7 8
</IntegerMatrix>
```

A faceta *mpeg7:refType* provê uma maneira de verificar o tipo de um elemento referenciado. O tipo do elemento referenciado deve ser do tipo especificado pelo valor de *refType*, ou um tipo derivado dele. No exemplo a seguir, o valor do elemento *SegmentRef* deve ser um *IDREF* para um elemento *SegmentType*:

```
<simpleType name="IdRefSegment">
  <restriction base="IDREF">
    <annotation><appinfo>
      <mpeg7:refType value="mpeg7:SegmentType" />
    </appinfo></annotation>
  </restriction>
</simpleType>
<element name="SegmentRef" type="mpeg7:IdRefSegment" />
```

3.2.4 Exemplo de uso da DDL

Os elementos estruturais do XML *Schema* são amplamente usados em definições do MPEG-7. Como exemplo, será usada uma definição de tempo, ou seja, como especificar um instante de tempo (*TimePoint*) em um vídeo ou a duração de um segmento de vídeo. A descrição de tempo representa o tempo no mundo real (tipo de dado *Time*) assim como o tempo usado nos dados áudios-visuais (tipo de dado *mediaTime*). Em ambos os casos, tanto as instâncias de tempo como os intervalos podem ser descritos.

A figura 10(a) mostra o modo mais simples de se especificar um instante e um intervalo temporal. O instante t_1 pode ser definido pelo *TimePoint* e o intervalo $[t_1, t_2]$ pelo ponto de partida t_1 e a duração $t_2 - t_1$. Derivado da figura 10(a), o tipo de dado *TimeType* é expresso na DDL:

```

<complexType name="TimeType">
  <sequence>
    <choice>
      <element name="TimePoint" type="mpeg7:TimePointType"/>
      <element name="RelTimePoint" type="mpeg7:RelTimePointType"/>
      <element name="RelIncrTimePoint" type="mpeg7:RelIncrTimePointType"/>
    </choice>
    <choice minOccurs="0">
      <element name="Duration" type="mpeg7:durationType"/>
      <element name="IncrDuration" type="mpeg7:IncrDurationType"/>
    </choice>
  </sequence>
</complexType>

```

Para a especificação do tipo de dado *TimeType*, um tipo complexo é usado, sendo composto de dois elementos: o ponto de partida (*TimePoint*) e a duração. Se apenas o *TimePoint* for especificado, a duração que aparece em seguida pode ser omitida, o que é expresso pelo elemento *sequence* que indica que o ponto de partida deve ser especificado antes da duração.

Para especificar o ponto de partida, três alternativas podem ser usadas (especificado pelo elemento *choice*): *TimePoint*, que especifica um instante temporal concreto; *RelTimePoint* (figura 10(b)), cujo tempo de início é definido como um deslocamento temporal em relação a um ponto de referência; e *RelIncrTimePoint* (figura 10(c)), cujo tempo de início também é definido como um deslocamento temporal em relação a uma referência, mas nesse caso em respeito a unidades de tempo.

O elemento de duração é opcional, indicado pela faceta *minOccurs* = "0". Há duas alternativas para expressar a duração: *Duration*, que especifica a duração de um período de tempo em dias, horas, e assim por diante, e *IncrDurationType*, que define a duração de um período de tempo em relação a unidades de tempo.

3.3 O FORMATO BiM

Descrições MPEG-7 podem ser geradas automaticamente em muitos casos, principalmente quando se trata de descritores de baixo nível, que geralmente são prolixos. Por exemplo, um vídeo completo pode ser automaticamente decomposto em segmentos, e para cada segmento diferente, descritores de baixo nível podem estar associados. O documento XML resultante pode ser muito extenso, não sendo adequado ao consumo em um ambiente de

streaming e de banda restrita. Para superar a falta de eficiência nessa representação textual, o MPEG-7 define um *framework* genérico para facilitar a entrega e processamento de descrições MPEG-7: o formato BiM (*binary format for MPEG-7*). Ele possibilita a compressão e *streaming* de qualquer documento XML.

Descrições textuais não são aptas ao uso em ambientes de *streaming*, pois esse tipo de ambiente requer um alto nível de flexibilidade com relação à ordem de transmissão dos elementos. Além disso, acesso aleatório deve ser provido para que não seja necessário que o *parser* analise toda cadeia de dados. Acesso aleatório requer uma estrutura baseada em pacotes, nas quais os cabeçalhos (*headers*) são geralmente representados em formato binário. Essas descrições textuais também não são aptas ao uso em ambientes de banda restrita, pois ocupam mais espaço de armazenamento e assim mais recursos de banda, ao contrário do formato binário que tem altas taxas de compressão.

Descrições MPEG-7 podem ser representadas tanto em formato textual, em XML, como em formato binário (BiM), ou a mistura dos dois dependendo do requisito da aplicação. Na camada de compressão, o fluxo de unidades de acesso (tanto textuais como codificadas em binário) é analisado por um *parser*, e assim a descrição é reconstruída. O fluxo binário pode ser analisado pelo *parser* do formato BiM e então transformado em formato textual para ser processado e reconstruído. Uma vez reconstruída a descrição, a aplicação está apta a consumi-la.

A reconstrução de unidades de acesso como passo intermediário do processo de decodificação não é requerido, pois uma determinada parte da descrição pode ser decodificada isoladamente. Assim, o terminal que recebe as unidades de acesso já pode processá-las sem que seja preciso esperar pelas outras. Isso é útil em casos que apresentem congestionamentos de rede, pois as informações assim que recebidas já podem ser vistas no terminal, melhorando a interação com o usuário. O processamento parcial das descrições é possível, pois tanto o codificador quanto o decodificador são cientes das informações de *schema*, sendo essa uma das principais vantagens técnicas do formato BiM.

As unidades de acesso são estruturadas como comandos que encapsulam as descrições ou o *schema*. Os comandos provêm aspectos dinâmicos das descrições MPEG-7: permitem que elas sejam enviadas num único pedaço ou em fragmentos, conforme ilustrado na figura 13. A descrição ou *schema* em formato textual são estruturas de árvore. A parte superior da figura 13 mostra como essa árvore pode ser encapsulada em uma única unidade de acesso que é transmitida

ao terminal, e a parte inferior mostra a descrição sendo fragmentada em três pedaços que são encapsulados em diferentes unidades de acesso. A descrição final é reconstruída adicionando-se o conteúdo das unidades de acesso 2 e 3 aos nós apropriados da unidade de acesso 1.

Além da funcionalidade de adição de unidades de acesso, também há operações como atualizar valores de descritores, apagar parte da descrição ou adicionar um novo *schema*. A fase de reconstrução da camada de compressão atualiza as informações da descrição e informações associadas de *schema* pela leitura dos comandos. Em síntese, as unidades de acesso definem: o tipo do comando a ser executado no terminal (*add*, *delete*, *update*, etc); a descrição MPEG-7 ou *schema* a ser usado na modificação; e o nó da árvore onde a modificação deve ser feita. Assim, a descrição não precisa ser transmitida na íntegra se apenas um valor do descritor for modificado.

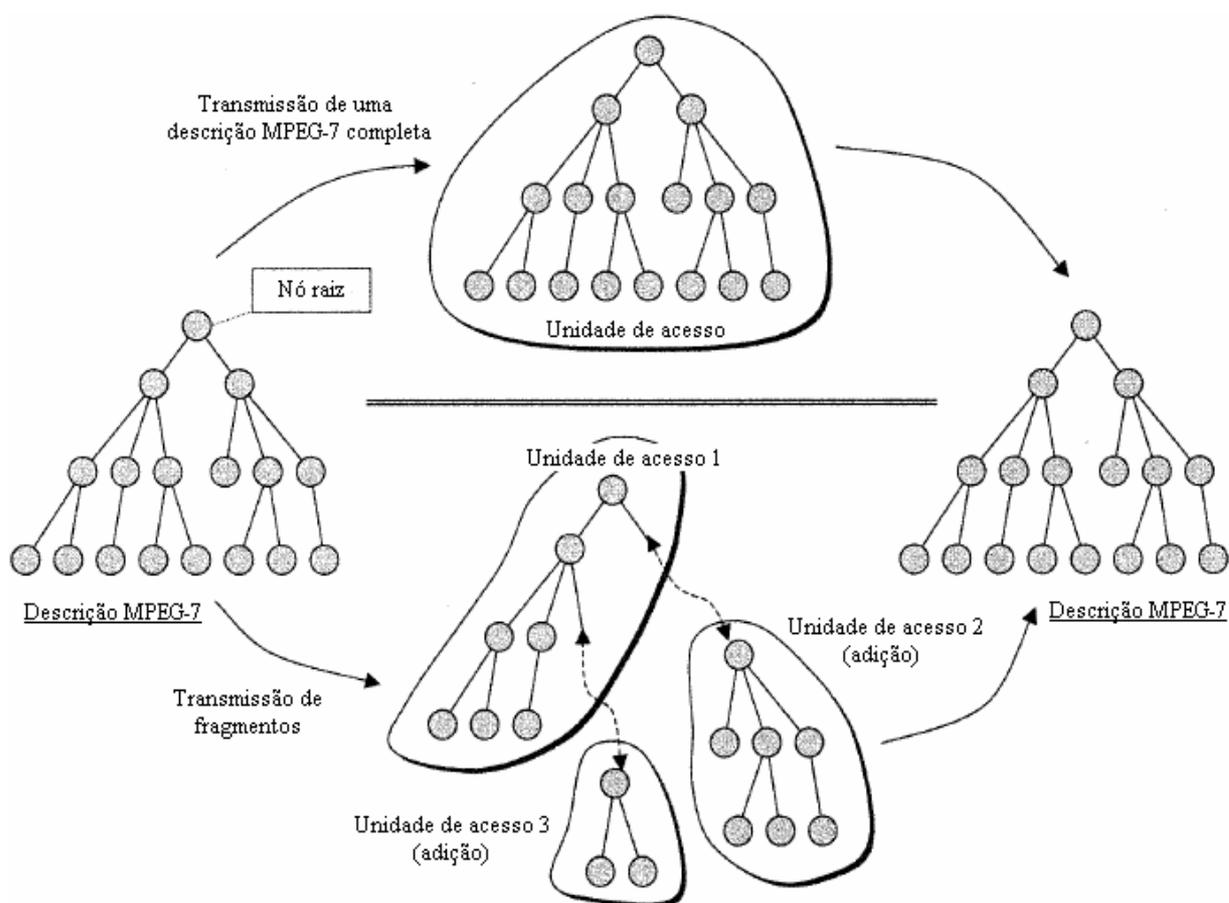


Figura 13 – Descrição MPEG-7 e unidades de acesso [AVARO; SALEMBIER, 2001]

3.4 CONSIDERAÇÕES FINAIS

O padrão MPEG-7 que foi detalhado nesse capítulo serve como base para a implementação da camada proposta. Ele provê a forma de como os metadados são estruturados e a camada se utiliza dessa informação para gerar as instâncias XML e criar a sintaxe de consulta, detalhada na subseção 5.1.1. Como é um padrão, as instâncias XML regidas por ele podem ser trocadas com outros sistemas que também empreguem o uso desse padrão.

4 O AMBIENTE DE APRENDIZADO ELETRÔNICO TIDIA-AE

O projeto TIDIA-Ae (Tecnologia da Informação para o Desenvolvimento da Internet Avançada – Aprendizado Eletrônico) é uma das três frentes do projeto TIDIA da FAPESP, junto com a incubadora virtual e o KyaTera. Conforme a descrição do projeto [FAPESP, 2005], buscase tanto a pesquisa quanto o desenvolvimento na área de aprendizado eletrônico, com os alicerces em redes de alta velocidade (Internet Avançada).

A especificação, projeto e implementação de um conjunto de ferramentas síncronas e assíncronas para ensino e aprendizado eletrônico estão entre os objetivos principais desse projeto, o que deve ter um impacto social profundo, pois serão construídas sob a filosofia do *software* livre. Isso significa baixo custo e flexibilidade, pois essas ferramentas podem ser combinadas e estendidas conforme a necessidade. Essas ferramentas devem cobrir três grandes grupos de ferramentas genéricas de aprendizado eletrônico: gerenciamento, coordenação e comunicação. Devem também, por requisito do projeto TIDIA, ser independentes de plataforma.

O desenvolvimento é baseado numa arquitetura de componentes de *software* que facilita a reutilização e a extensão da infra-estrutura resultante, direcionando a um desenvolvimento colaborativo que auxilia a pesquisa no projeto, gerenciamento e teste de novas ferramentas que podem ser incorporadas na infra-estrutura geral. Graças ao uso de componentes de *software*, há ganhos de tempo e de esforços de pesquisa.

O desenvolvimento da infra-estrutura do TIDIA-Ae representa um avanço no desenvolvimento do aprendizado eletrônico no estado de São Paulo, integrando vários esforços dispersos. O projeto investiga e emprega conhecimentos da teoria de aprendizado eletrônico, e também aplica e desenvolve técnicas de áreas que incluem a engenharia de software, interação humano-computador, inteligência artificial, multimídia, dentre outras.

As ferramentas construídas na primeira fase, finalizada em fevereiro de 2006, compreendem o *chat* [ELEUTÉRIO et al., 2005], correio eletrônico, *portfolio* [BEDER et al., 2005], editor colaborativo [APPEL; ALMEIDA; SCHIEL, 2005], comunicador instantâneo [LOBATO et al., 2005], *whiteboard* [KUDO et al., 2005] e laboratório remoto [PRAZERES et al., 2005], juntamente com o gerenciador de usuários [ZAINA et al., 2005], gerenciador de contextos, gerenciador de ferramentas [BRESSAN; FRANCO; RUGGIERO, 2005], gerenciador de atividades e gerenciador de conteúdo.

As ferramentas permitem tanto a interação presencial como remota, como, por exemplo, a ferramenta *whiteboard*, uma lousa branca virtual que tanto pode ser usada em uma aula presencial com um quadro branco eletrônico como em uma aula remota realizada com o suporte de redes de alta velocidade.

O núcleo do projeto TIDIA-Ae é o LMS – *Learning Management System*. Seu modelo conceitual é apresentado na figura 14. Ele provê ferramentas que suportam conceitos como atividades de aprendizado, participantes com papéis e conjuntos de permissões, contextos, objetos de aprendizagem assim como comunicação síncrona e assíncrona. Como apresentado na figura 14, a ferramenta se integra no modelo do LMS através de relacionamentos, como é o caso de *ToolManager*.

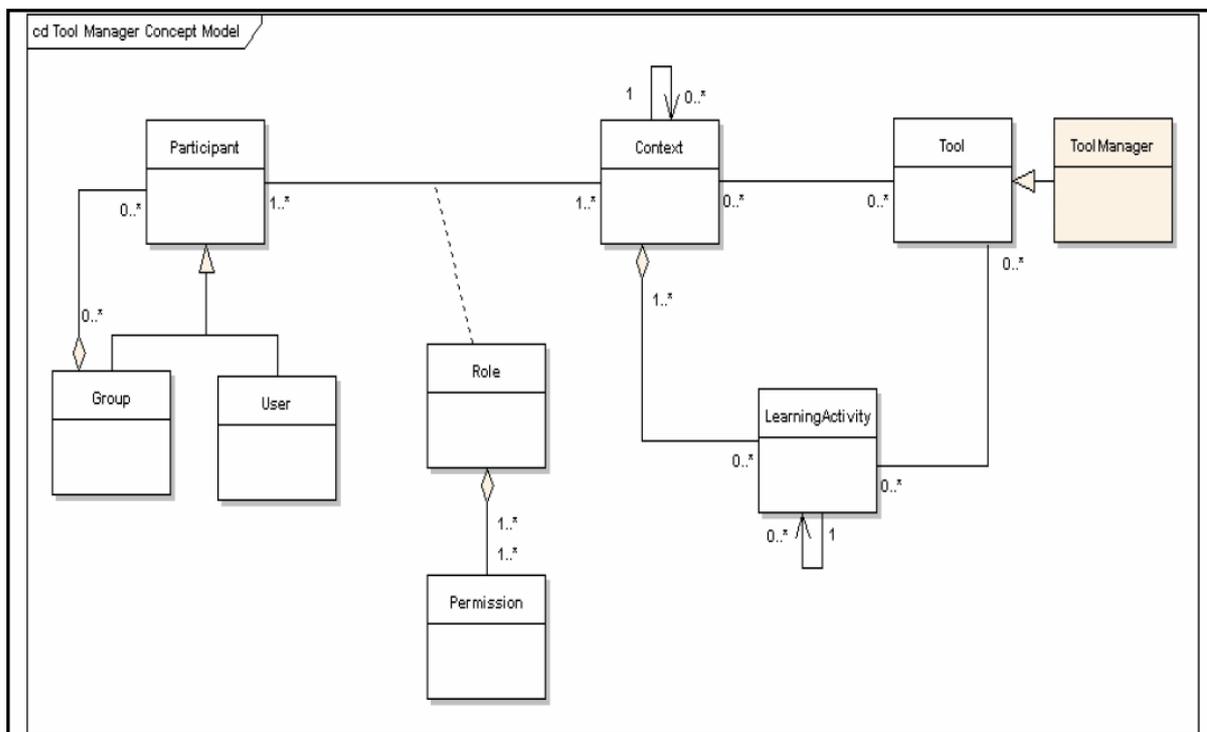


Figura 14 – Modelo conceitual do LMS [BRESSAN; FRANCO; RUGGIERO, 2005]

As ferramentas síncronas compreendem o *chat*, *whiteboard*, comunicador instantâneo e laboratório de acesso remoto. Para essas ferramentas, foi definido um esquema conceitual de banco de dados unificado, apresentado na figura 15. Esse banco de dados contempla os requisitos

de armazenamento das mídias e de seus metadados para todas as ferramentas síncronas. No caso das outras ferramentas (assíncronas), cada uma possui seu próprio esquema conceitual.

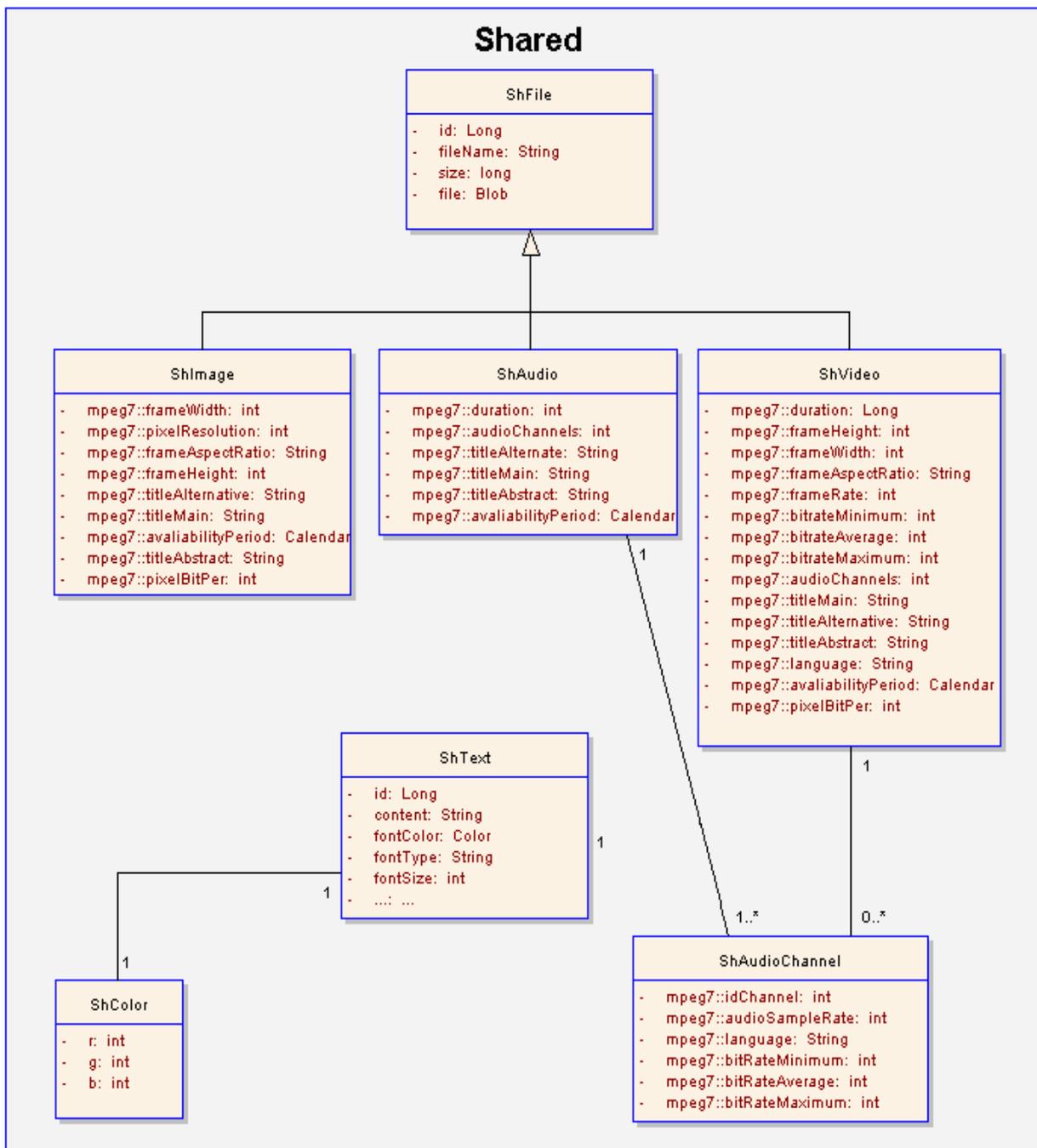


Figura 15 – Esquema conceitual de banco de dados das ferramentas síncronas

As classes *ShText* e *ShColor* armazenam os *strings* gerados pelas ferramentas *chat*, *whiteboard* e comunicador instantâneo. As demais classes armazenam os objetos multimídia, sendo eles: imagem (*ShImage*), áudio (*ShAudio*) e vídeo (*ShVideo*). Nesse modelo, há uma tentativa preliminar de se efetuar a modelagem multimídia de acordo com o padrão de descrição multimídia MPEG-7.

Porém, como apresentado no capítulo três, o padrão MPEG-7 tem estruturas e hierarquias diferentes das apresentadas nesse modelo, apenas alguns atributos se mantendo nele. Ele é resultado da primeira fase do projeto, e como os objetivos principais da primeira fase não englobavam aspectos multimídia das ferramentas, a modelagem apresentada é apenas um protótipo. Na segunda fase, é indispensável que haja a padronização e a unificação dos esquemas conceituais de bancos de dados para o armazenamento de conteúdo multimídia. Essa unificação pode ser feita na camada proposta nesse trabalho e para a padronização multimídia foi utilizado o padrão MPEG-7.

O ambiente TIDIA-Ae, por ser composto de várias ferramentas que fazem o uso intensivo de material multimídia, gera requisitos para a manipulação do mesmo e dessa forma a centralização da manipulação dos dados multimídia é desejável. A camada proposta vem então com o objetivo de unificar as necessidades de gerenciamento desses dados pelas ferramentas, integrando-as e servindo como mediadora. Cabe ao autor dessas ferramentas fazer o uso devido da interface para efetuar a persistência da mídia junto com seus metadados semânticos. A ferramenta deve identificar a mídia submetida por um usuário e então repassar os dados necessários à interface de persistência.

Há vários grupos que desenvolvem as ferramentas do TIDIA-Ae dispersos pelo estado de São Paulo, sendo que cada um faz seu próprio repositório de dados. Essa camada vem com o propósito de unificar o repositório de dados multimídia, para que desse modo as ferramentas produzidas por esses grupos possam persistir e consultar esses dados de maneira uniforme e integrada, utilizando um modelo de dados multimídia padronizado. Isso simplifica o trabalho de manipulação de dados multimídia por essas ferramentas e dá o poder de intercâmbio de dados entre elas próprias e entre outros ambientes/sistemas.

4.1 CONSIDERAÇÕES FINAIS

O ambiente de aprendizado eletrônico TIDIA-Ae, sendo um massivo utilizador de material multimídia devido à Internet com banda irrestrita, é um projeto adequado para utilizar a camada em questão, pois ela supre algumas das necessidades geradas pelo uso de dados multimídia do mesmo. Justifica-se o emprego da camada pela ausência de uma modelagem multimídia adequada às necessidades do projeto e pela ausência de interfaces que unifiquem a manipulação de dados multimídia por todas as ferramentas, de maneira uniforme e integrada.

5 A CAMADA DE MANIPULAÇÃO DE MÍDIAS

A proposta desse trabalho norteou a construção de componentes (relacionados aos módulos da figura 16) cujas interfaces provenham uma forma simples de armazenamento e consulta de mídias seguindo o padrão MPEG-7, que é a base desses componentes, sendo proposto seu uso como modelo de dados multimídia. O MPEG-7 é muito vasto, e por isso não é usado em sua totalidade neste trabalho. As partes desse padrão que são usadas estão localizadas no gerenciamento e descrição de conteúdo dos MDS: mídia, aspectos estruturais e conceituais. Essas partes em conjunto formam os metadados utilizados para efetuar as inserções com dados semânticos e as consultas sobre as mídias.

Essas interfaces unificam as necessidades de armazenamento e consulta de mídias do projeto TIDIA-Ae, agindo como uma central de manipulação de mídias onde quaisquer aplicações do projeto podem fazer uso das mesmas. Esses componentes foram feitos para serem genéricos, ou seja, o TIDIA-Ae é apenas um exemplo de uso dos mesmos. A generalidade desses componentes é reforçada pelo fato de estar apoiada sobre o padrão MPEG-7, que rege como as instâncias XML devem ser formadas.

Os trabalhos apresentados na subseção 2.3.2 e na seção 2.4 são sistemas de banco de dados multimídia, e cada um tem suas particularidades. Diferindo desses trabalhos, uma nova infra-estrutura multimídia genérica e reutilizável é proposta, não se igualando aos trabalhos descritos, mas oferecendo a infra-estrutura para a construção de tais sistemas. Essa infra-estrutura provê interfaces de inserção de mídias e consulta por metadados e oferece um meio de intercâmbio de dados com outros sistemas através do padrão MPEG-7. Além de disponibilizar a infra-estrutura para a construção de sistemas mais complexos, também pode ser reutilizado por aplicações mais simples, como as próprias ferramentas do TIDIA-Ae. A arquitetura da camada proposta é apresentada na figura 16, sendo composta dos seguintes módulos:

- Módulo de anotação: esse módulo é responsável pela inserção das referências das mídias no banco de dados (fisicamente elas estão localizadas em sistema de arquivo), juntamente com seus metadados. Alguns metadados podem ser obtidos automaticamente, como formato de codificação e duração, e outros devem ser obtidos manualmente, como a descrição e informações de eventos.

- Módulo de consulta: esse é o módulo que possibilita uma aplicação realizar uma consulta por metadados, sendo a resposta uma mídia ou um conjunto delas.
- Módulo de exportação: trabalhando em conjunto com o módulo de consulta, esse módulo fornece o(s) documento(s) XML referente(s) à(s) mídia(s) consultada(s) se a aplicação assim o(s) requisitar.

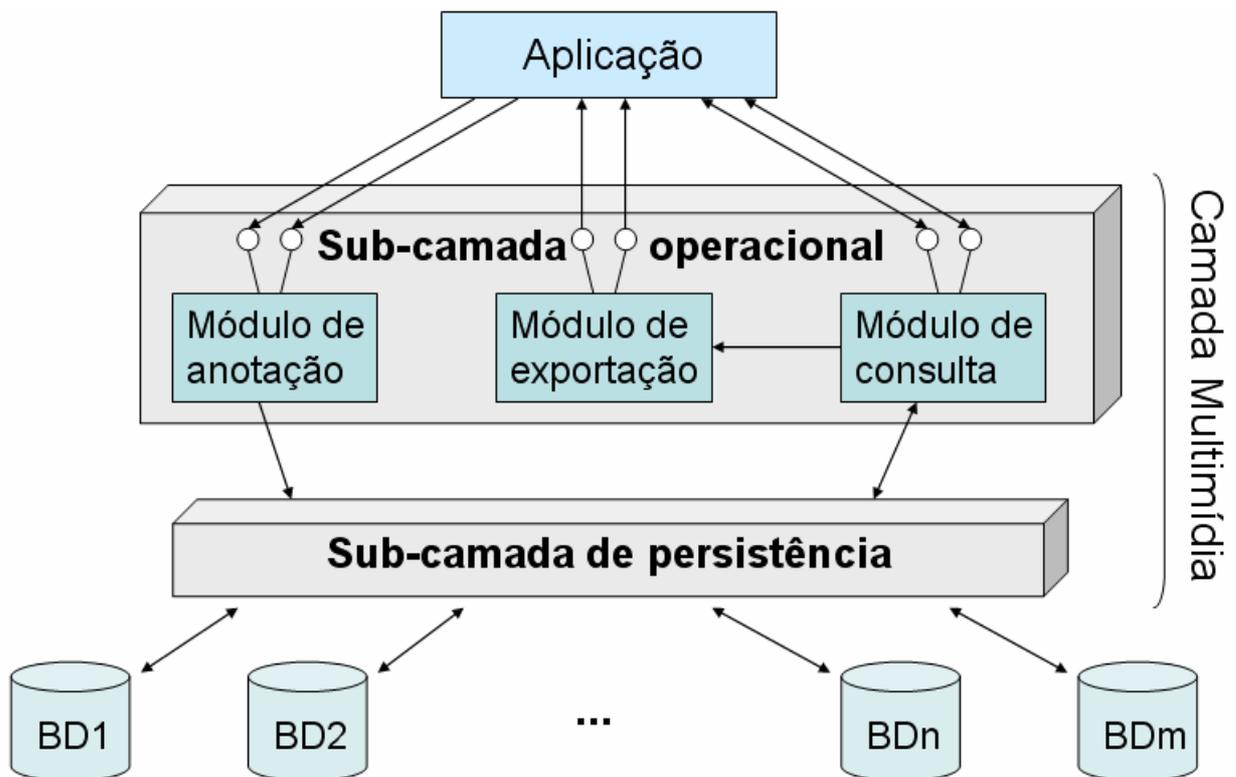


Figura 16 – Arquitetura do sistema

Essa camada foi implementada com base na linguagem Java da *Sun Microsystems*, junto com baterias de testes para verificação do correto funcionamento da mesma. O SGBD utilizado para esses testes foi o PostgreSQL (<http://www.postgresql.org>), o mesmo utilizado no projeto TIDIA-Ae, por ser de código livre. Porém, é importante ressaltar que a camada não está atrelada a um SGBD em particular, pois o propósito é que seja a mais genérica possível. Em se tratando do SGBD, o repositório será logicamente único e integrará os dados (isso não impede, porém, que os dados estejam fisicamente distribuídos caso haja suporte). Os componentes foram implementados com base na tecnologia de *Enterprise JavaBeans* (EJB) do J2EE (*Java 2 Enterprise Edition*), da *Sun Microsystems*.

Usou-se o *framework* EMF (*Eclipse Modeling Framework*) [ECLIPSE, 2007] para efetuar a modelagem da DDL do padrão em classes *Java*, a partir do modelo *ecore* que o EMF cria para guiar a construção dessas classes. Com isso, é possível manipular objetos complexos usando essas classes, dando poder para utilizar na linguagem de programação *Java* objetos que representem um *XML Schema*, no caso, a DDL do MPEG-7.

A partir daí, o uso desses objetos pode seguir vários caminhos, como criar uma instância XML ou efetuar a persistência deles em banco de dados. Nesse trabalho adotou-se a persistência desses objetos em banco de dados relacional pelo *framework* *hibernate* [BAUER; KING, 2004], através do *plug-in* *teneo* [ELVER, 2007] (sub-camada de persistência da arquitetura) do EMF. Esse *framework* possibilita a modelagem dos dados no paradigma orientado a objetos e a persistência no paradigma relacional (através do *teneo*), abstraindo os detalhes de como as tabelas são formadas nesse último paradigma. Isso possibilita uma plena modelagem do padrão MPEG-7, que representa os dados de uma forma rica (relacionamentos, heranças, polimorfismo, etc).

O *plug-in* *teneo* gera o modelo do *hibernate* a partir do modelo *ecore* criado pelo EMF. Com esse modelo, cria-se um banco de dados relacional que obedece ao *XML Schema* que foi fornecido, e dessa forma, persiste-se nesse banco de dados objetos que representam esse *XML Schema*. A figura 17 ilustra o papel do EMF e do *teneo* para “transportar” um *XML Schema* até um banco de dados relacional:

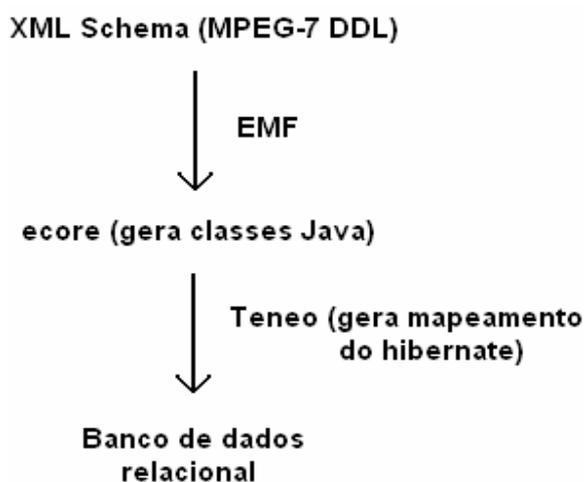


Figura 17 – Processo de conversão do *XML Schema* ao banco de dados relacional

Nessas interfaces, é possível efetuar a inserção de mídias num banco de dados através do módulo de anotação, tendo essa inserção as seguintes características:

- Todos os tipos de mídias, sejam áudio, vídeo ou imagem, têm metadados inerentes às mesmas. Esses metadados são detectados automaticamente e persistidos num banco de dados, respeitando o formato que o padrão MPEG-7 rege. São descritos a seguir:
 - Áudio, vídeo e imagem: duração, tamanho de arquivo, data da última atualização da mídia e sistema de codificação.
 - Áudio: taxa de amostragem (*sample rate – hz*), número de canais e quantidade de *bits* da amostra.
 - Vídeo: quadros por segundo (*frame rate – fps*), resolução (altura e largura), proporção da tela (*aspect ratio*) e taxa média de *bits* por segundo (*average bit rate*).
 - Imagem: resolução (altura e largura) e quantidade de *bits* de precisão do *pixel*.
- Na inserção, a cadeia de *bytes* da mídia é copiada em sistema de arquivo, havendo uma referência para ela nos metadados. Ou seja, no banco de dados são persistidos os metadados da mídia e não ela em si. Deve-se observar que essa decisão se deve ao fato de se buscar generalidade em termos de bancos de dados, ou seja, como não se sabe a priori o SGBD que será usado, há possibilidade de haver problemas se a cadeia de *bytes* da mídia for persistida no mesmo.
- Os dados semânticos são inseridos manualmente. O modo de como esses dados devem ser inseridos está detalhado na subseção 5.1.1.

Também é possível, via essas interfaces, consultar e obter as mídias ou seus documentos XML pelos metadados semânticos, do mesmo modo que essas mídias são anotadas. Isso é explicado com detalhes na subseção 5.1.1. A consulta da mídia ou de seus metadados é feita através dos módulos de consulta e exportação, respectivamente.

5.1 DADOS SEMÂNTICOS, INSERÇÕES E CONSULTAS

Foi feita a modelagem de alguns elementos semânticos do padrão MPEG-7 (Ds e DSs) nessas interfaces, considerando o que o projeto TIDIA-Ae poderá utilizar. Esses elementos semânticos são usados como metadados tanto na inserção/anotação quanto na consulta da mídia.

Um cenário típico de inserção/anotação da mídia é ilustrado na figura 18. Primeiramente, há um usuário de uma aplicação que submete uma mídia à mesma, juntamente com os metadados semânticos fornecidos através de uma interface gráfica; a aplicação, de posse da mídia e dos metadados, invoca a interface apropriada para efetuar a inserção/anotação da mesma, interface esta localizada no módulo de anotação; esse módulo então repassa a mídia e seus metadados para a sub-camada de persistência e esta persiste os metadados em um banco de dados relacional e a mídia em si no disco.



Figura 18 – Cenário típico de inserção/anotação da mídia

Um cenário típico de consulta da mídia é ilustrado na figura 19. Nesse cenário, o usuário da aplicação informa os metadados semânticos que serão usados na busca da mídia; a aplicação, de posse desses metadados, invoca a interface apropriada localizada no módulo de consulta, passando esses metadados a ela; assim, o módulo de consulta reconstrói os objetos do EMF que

representam a DDL do MPEG-7 a partir do banco de dados, e a aplicação recebe como resposta a mídia em si ou seu documento XML (metadados), dependendo da escolha feita pela mesma. Se a escolha for obter a mídia, esta é retornada pelo próprio módulo de consulta, mas se a escolha for obter o documento XML, este é retornado pelo módulo de exportação.

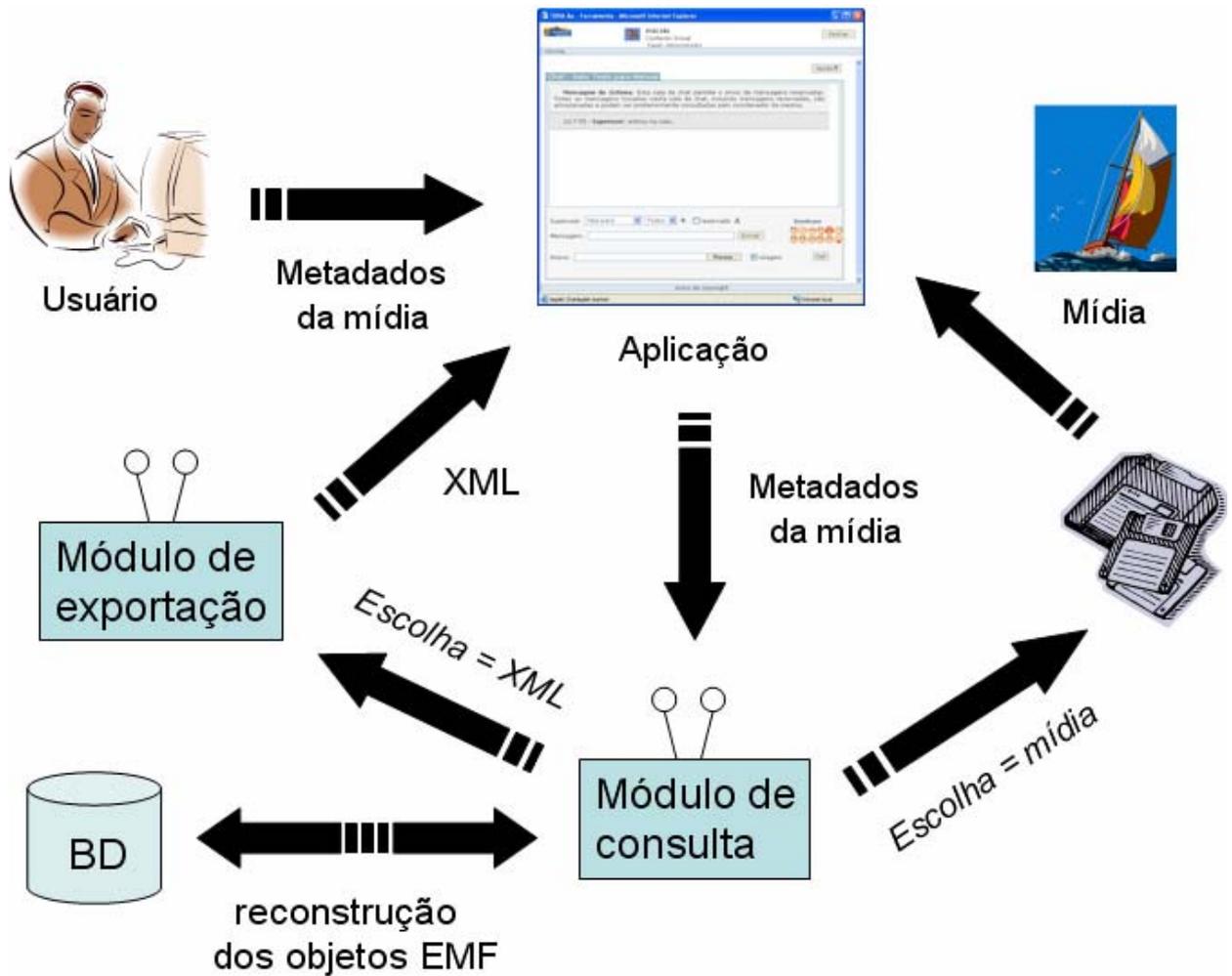


Figura 19 – Cenário típico de consulta da mídia

Para ilustrar os elementos semânticos modelados nessas interfaces, foi feita abaixo uma representação simplificada do XML Schema (MPEG-7 DDL), sendo que tudo é elemento, exceto o símbolo “@” que representa um atributo e “#” que representa um elemento abstrato. Os colchetes representam o elemento raiz, que pode ser referenciado. Novos elementos semânticos podem ser adicionados conforme necessidade, porém nesta versão do trabalho foram modelados os seguintes elementos, acompanhados de suas respectivas cardinalidades e descrições, este último na coluna da direita:

[AgentObject] (0..*) –	Agente semântico
@ID (1..1)	Identificador
Label (0..1)	Rótulo
Definition (0..1)	Descrição
Relation (0..*) – [Relation]	Relações com outras entidades semânticas Referência ao elemento <i>Relation</i>
ObjectRef (0..*) – [Reference]	Referências a outros objetos Referência ao elemento <i>Reference</i>
#Agent (0..1) – [Person] ou [PersonGroup] ou [Organization]	O que este agente semântico representa: Uma pessoa – referência ao elemento <i>Person</i> ou um grupo de pessoas – referência ao elemento <i>PersonGroup</i> ou uma organização – referência ao elemento <i>Organization</i>

[SemanticTime] (0..*) –	Tempo semântico
@ID (1..1)	Identificador
Label (0..1)	Rótulo
Definition (0..1)	Descrição
Relation (0..*) – [Relation]	Relações com outras entidades semânticas Referência ao elemento <i>Relation</i>

[SemanticPlace] (0..*) –	Local semântico
@ID (1..1)	Identificador
Label (0..1)	Rótulo
Definition (0..1)	Descrição
Relation (0..*) – [Relation]	Relações com outras entidades semânticas Referência ao elemento <i>Relation</i>
Place (0..1) – [Place]	Endereço Referência ao elemento <i>Place</i>

[Event] (0..*) –	Evento semântico
@ID (1..1)	Identificador
Label (0..1)	Rótulo
Definition (0..1)	Descrição
Relation (0..*) [Relation]	Relações com outras entidades semânticas Referência ao elemento <i>Relation</i>

[Object] (0..*) –	Objeto semântico
@ID (1..1)	Identificador
Label (0..1)	Rótulo
Definition (0..1)	Descrição
Relation (0..*) – [Relation]	Relações com outras entidades semânticas Referência ao elemento <i>Relation</i>

[VideoSegment] (0..*) –	Segmento de vídeo
@ID (1..1)	Identificador
TextAnnotation (0..1)	Descrição
TimePoint (1..1)	Instante do início
Duration (0..1)	Duração

Seguem abaixo as definições básicas de elementos que são usados pelos acima descritos, igualmente com suas cardinalidades e descrições:

[Person] –	Representação de uma pessoa
Name(1..1) – GivenName (1..1) FamilyName (0..1)	Nome completo Primeiro Nome Sobrenome
Affiliation (0..2) – Organization (0..1) – [Organization] PersonGroup (0..1) – [PersonGroup]	Afiliação A uma organização Referência ao elemento <i>Organization</i> A um grupo de pessoas Referência ao elemento <i>PersonGroup</i>
Address (0..1) – [Place]	Endereço Referência ao elemento <i>Place</i>
ElectronicAddress (0..1) – [ElectronicAddress]	Endereço eletrônico Referência ao elemento <i>ElectronicAddress</i>

[PersonGroup] –	Representação de um grupo de pessoas
Name (0..1)	Nome do grupo
Kind (0..1)	Tipo do grupo
Member (0..*) – [Person] ou MemberRef (0..*) – [Reference]	Membros que formam esse grupo Referência ao elemento <i>Person</i> Ou referências a membros Referência ao elemento <i>Reference</i>
Address (0..1) – [Place]	Endereço desse grupo Referência ao elemento <i>Place</i>
ElectronicAddress (0..1) – [ElectronicAddress]	Endereço eletrônico desse grupo Referência ao elemento <i>ElectronicAddress</i>

[Organization] –	Representação de uma organização
Name (0..1)	Nome da organização
Kind (0..1)	Tipo da organização
Address (0..1) – [Place]	Endereço da organização Referência ao elemento <i>Place</i>
ElectronicAddress (0..1) – [ElectronicAddress]	Endereço eletrônico da organização Referência ao elemento <i>ElectronicAddress</i>

[Place] –	Representação de local - endereço
Name (0..1)	Nome do local
PostalAddress (0..1) – AddressLine (0..1) PostingIdentifier (0..1)	Endereço postal Endereço por extenso Identificador postal, ou CEP

[ElectronicAddress] –	Representação de endereço eletrônico
Telephone (0..*)	Telefone
Fax (0..*)	Fax
Email (0..*)	Endereço de correio eletrônico
Url (0..*)	Endereço de sítio eletrônico

[Relation] –	Relação
@Type (0..1)	Tipo da relação
@Source (0..*)	Identificador da entidade semântica origem
@Target (1..*)	Identificador da entidade semântica destino
@Strength (0..1)	Grau de pertinência da relação

[Reference] –	Referência
@IDRef (0..1)	Identificador do referido
@XPath (0..1)	Caminho <i>XPath</i> do referido
@Href (0..1)	Endereço de sítio eletrônico

5.1.1 Modelo de representação semântica em interfaces e detalhes técnicos

A modelagem semântica feita nessas interfaces permite que metadados semânticos sejam inseridos através de um objeto *HashMap* do *Java*. A inserção desses metadados é denominada anotação semântica e é feita da seguinte maneira:

```
HashMap SemanticData = new HashMap();
SemanticData.put("string1", "string2");
```

Onde *string1* é o elemento semântico e *string2* o valor desse elemento. A construção da *string1* é o que denota esse modelo de representação semântica em interfaces proposto nesse trabalho. O começo da *string* é a entidade semântica, por exemplo, se for *AgentObject*, então a *string* começa com “*AgentObjectX*”, onde X é um número seqüencial de 1 até 99 (o limite superior pode ser modificado) devido à cardinalidade de *AgentObject* ser (0..*). A *string* continua com “.” seguido de elementos de *AgentObject*, que são *ID*, *Label*, *Definition*, *Relation*, *ObjectRef* e *Agent*. Para especificar uma descrição para o *AgentObject* em questão, usa-se o comando em Java:

```
SemanticData.put("AgentObject1.Definition", "Uma descrição textual");
```

Ainda no exemplo do *AgentObject*, pode-se especificar o que ele representa. Caso seja uma pessoa, pode-se especificar o nome dela com o seguinte comando:

```
SemanticData.put("AgentObject1.Agent_Person-GivenName", "Josué");
```

Note que o símbolo “_” foi usado com a intenção de representar um polimorfismo, já que *Agent* é um elemento abstrato. A *substring* “*Person-GivenName*” é da própria modelagem de *Person*. Caso esse *AgentObject* tenha afiliação com um grupo de pessoas, pode-se especificar o nome desse grupo e o nome de um membro com os comandos:

```
SemanticData.put("AgentObject1.Agent_Person-AffPersonGroup-Name", "G1");
```

```
SemanticData.put("AgentObject1.Agent_Person-AffPersonGroup-Member1.Person-GivenName", "Ricardo");
```

Após um número, sempre segue um “.” (exceto quando se trata de um elemento ou atributo folha), sendo o separador padrão o símbolo “-“. A composição da *string* do elemento semântico se dá por concatenação de *substrings* conforme se percorre a árvore de modelagem. As possíveis concatenações estão descritas abaixo na coluna da direita, seguido de um exemplo de construção da *string* do elemento semântico:

[AgentObject] (0..*) –	AgentObjectX.
@ID (1..1)	ID
Label (0..1)	Label
Definition (0..1)	Definition
Relation (0..*) – [Relation]	RelationX.
ObjectRef (0..*) – [Reference]	ObjectRefX.
#Agent (0..1) – [Person] ou [PersonGroup] ou [Organization]	Agent_

[SemanticTime] (0..*) –	SemanticTimeX.
@ID (1..1)	ID
Label (0..1)	Label
Definition (0..1)	Definition
Relation (0..*) – [Relation]	RelationX.

[SemanticPlace] (0..*) –	SemanticPlaceX.
@ID (1..1)	ID
Label (0..1)	Label
Definition (0..1)	Definition
Relation (0..*) – [Relation]	RelationX.
Place (0..1) – [Place]	

[Event] (0..*) –	EventX.
@ID (1..1)	ID
Label (0..1)	Label
Definition (0..1)	Definition
Relation (0..*) – [Relation]	RelationX.

[Object] (0..*) –	ObjectX.
@ID (1..1)	ID
Label (0..1)	Label
Definition (0..1)	Definition
Relation (0..*) – [Relation]	RelationX.

[VideoSegment] (0..*) –	VideoSegmentX.
@ID (1..1)	ID
TextAnnotation (0..1)	TextAnnotation
TimePoint (1..1)	TimePoint
Duration (0..1)	Duration

[Person] –	Person-
Name(1..1) – GivenName (1..1) FamilyName (0..1)	GivenName FamilyName
Affiliation (0..2) – Organization (0..1) – [Organization] PersonGroup (0..1) – [PersonGroup]	Aff
Address (0..1) – [Place]	
ElectronicAddress (0..1) – [ElectronicAddress]	

[PersonGroup] –	PersonGroup-
Name (0..1)	Name
Kind (0..1)	Kind
Member (0..*) – [Person] ou MemberRef (0..*) – [Reference]	MemberX. MemberRefX.
Address (0..1) – [Place]	
ElectronicAddress (0..1) – [ElectronicAddress]	

[Organization] –	Organization-
Name (0..1)	Name
Kind (0..1)	Kind
Address (0..1) – [Place]	
ElectronicAddress (0..1) – [ElectronicAddress]	

[Place] –	Place-
Name (0..1)	Name
PostalAddress (0..1) – AddressLine (0..1) PostingIdentifier (0..1)	AddressLine PostingIdentifier

[ElectronicAddress] –	
Telephone (0..*)	TelephoneX
Fax (0..*)	FaxX
Email (0..*)	EmailX
Url (0..*)	UrlX

[Relation] –	
@Type (0..1)	Type
@Source (0..*)	SourceX
@Target (1..*)	TargetX
@Strength (0..1)	Strength

[Reference] –	
@IDRef (0..1)	IDRef
@XPath (0..1)	XPath
@Href (0..1)	Href

Com isso, não há necessidade em digitar o conteúdo dos arquivos XML com toda modelagem semântica das mídias, o que seria muito custoso. Basta preencher as *strings* conforme detalhado nas tabelas anteriores que os documentos XML já estão formados. Alguns elementos intermediários foram omitidos da concatenação a fim de simplificar a *string* do elemento semântico, que de outra forma ficaria muito extensa e de difícil leitura. Segue um exemplo interativo na figura 20 de como ocorrem as concatenações para a formação da *string* “AgentObjectX.Agent_Person-AffPersonGroup-MemberX.Person-GivenName” (lembrando que a letra “X” representa um número):

Árvore	Valor de string1
[AgentObject]	string1 = AgentObjectX.
# Agent	string1 = AgentObjectX.Agent_
[Person]	string1 = AgentObjectX.Agent_Person-
Affiliation	string1 = AgentObjectX.Agent_Person-Aff
[PersonGroup]	string1 = AgentObjectX.Agent_Person-AffPersonGroup-
Member	string1 = AgentObjectX.Agent_Person-AffPersonGroup-MemberX.
[Person]	string1 = AgentObjectX.Agent_Person-AffPersonGroup-MemberX.Person-
GivenName	string1 = AgentObjectX.Agent_Person-AffPersonGroup-MemberX.Person-GivenName

Figura 20 – Exemplo de construção da *string* do dado semântico

Quanto às cardinalidades: algumas, sob avaliação, foram alteradas devido ao uso real do elemento. Por exemplo, *ElectronicAddress* no padrão possui cardinalidade (0..*) e foi-se usada (0..1) pois seus elementos (*Telephone*, *Email*, *Fax*, *Url*) já possuem cardinalidade (0..*). Outro exemplo é o atributo (@) *ID*, que muitas vezes deriva do tipo *DSType*. Nesse tipo ele é opcional, mas em alguns casos em que elementos mais próximos das folhas herdaram o *ID* de *DSType*, não pode ser, pois as referências para esses elementos são feitas em cima desse atributo.

Os elementos de tempo, como *TimePoint* e *Duration* devem ser preenchidos com valores seguindo esta forma: HH:MM:SS:NNN, onde H representa a hora, M minuto, S segundo e N milissegundo. O tipo da relação (*[relation]*) pode assumir os seguintes valores, conforme definidos pelo padrão MPEG-7: [MANJUNATH; SALEMBIER; SIKORA, 2002]

- Relações normativas semânticas: *Agent*, *agentOf*, *patient*, *patientOf*, *experiencer*, *experiencerOf*, *stimulus*, *stimulusOf*, *causer*, *causerOf*, *goal*, *goalOf*, *beneficiary*, *beneficiaryOf*, *them*, *themOf*, *result*, *resultOf*, *instrument*, *instrumentOf*, *accompanier*, *accompanierOf*, *summarizes*, *summarizedBy*, *state*, *stateOf*, *combination*, *specializes*, *generalizes*, *similar*, *opposite*, *exemplifies*, *exemplifiedBy*, *interchangeable*, *identifier*, *part*, *partOf*, *contrasts*, *property*, *entailment*, *entailmentOf*, *manner*, *mannerOf*, *influences*, *dependsOn*, *membershipFunction*, *key*, *keyFor*, *annotates*, *annotatedBy*, *shows*, *appearsIn*, *reference*, *referenceOf*, *quality*, *qualityOf*, *symbolizes*,

symbolizedBy, location, locationOf, source, sourceOf, destination, destinationOf, path, pathOf, time, timeOf, depicts, depictedBy, represents, representedBy, context, contextFor, interprets e interpretedBy.

- Relações normativas espaciais: *south, north, west, east, northwest, northeast, southwest, southeast, left, right, below, above, over, under.*
- Relações normativas temporais: *Precedes, follows, meets, metBy, overlaps, overlappedBy, contains, during, strictContains, strictDuring, starts, startedBy, finishes, finishedBy, coOccurs, contiguous, sequential, coBegin, coEnd, parallel, overlapping.*

Como um exemplo, pode-se definir uma relação em um *AgentObject* que diz que ele é simbolizado por um segmento de vídeo:

```
SemanticData.put("AgentObjectI.RelationI.Type", "symbolizedBy");
```

```
SemanticData.put("AgentObjectI.RelationI.Target", "SegmentoI_id");
```

A consulta é feita no mesmo formato que a inserção do metadado semântico, ou seja, a *string* é a mesma, apenas diferindo no fato que os números têm que ser omitidos. A omissão é necessária para melhorar a clareza da *string* de consulta. Por exemplo, considerando o seguinte comando para preencher o *HashMap* "*SemanticData*" da anotação da mídia:

```
SemanticData.put("EventI.Definition", "Churrasco da pós-graduação");
```

A consulta da descrição de um evento é feita criando-se um objeto *HashMap* com a mesma *string*, apenas omitindo o número do evento:

```
queryObjects.put("Event.Definition", "Churrasco");
```

Pode-se usar no campo de valor a *string* toda ou apenas parte dela. No caso de se usar apenas parte da *string* para fazer as consultas, todas as ocorrências que contêm essa *string* são colocadas no resultado. No exemplo acima, consultou-se a descrição do evento apenas por "Churrasco", e no caso, todas as mídias cujas descrições de evento contiverem essa *string* farão parte do resultado. Podem-se também colocar vários elementos semânticos nesse *HashMap*, que dessa forma todas as mídias correspondentes a esses elementos serão retornadas. Porém, deve-se considerar que cada elemento semântico representa uma consulta diferente, salvo quando se usa dois ou mais valores para o mesmo elemento semântico, o que resulta em uma consulta. Por exemplo:

```
queryObjects.put("Object.Label", "Trem || Vagão");
```

Como é requisitada a consulta com dois valores para o elemento semântico “*Object.Label*”, é gerada uma consulta onde o valor do elemento é “Trem” ou “Vagão” (“||” é o operador lógico “ou”). Consultas sobre o identificador (“ID”) não foram implementadas, pois não faz sentido consultar o identificador como metadado semântico.

5.2 DESCRIÇÃO DAS INTERFACES

As interfaces propriamente ditas consistem nos seguintes métodos:

```
public static Vector persistMedia (String mediaStr, HashMap semantics);
```

```
public static Vector retrieveMedia (HashMap queryObjects, boolean returnXML);
```

O método *persistMedia* tem dois argumentos: *mediaStr* e *semantics*. O primeiro se refere à URL da mídia, que deve ser no formato *file:/<LetraDoDrive>:/diretório/arquivo.ext*, *file:/directory/arquivo.ext* ou *http://www.url.com/arquivo.ext*. O segundo é o *HashMap* explicado na subseção 5.1.1, que contém os dados semânticos. O retorno na forma de *Vector* é apenas para fins de depuração. Este método captura todos os metadados inerentes às mídias automaticamente e processa as informações semânticas fornecidas, colocando tudo isso em um banco de dados relacional. Ele representa o módulo de anotação da arquitetura apresentada na figura 16.

O método *retrieveMedia* tem dois argumentos: *queryObjects* e *returnXML*. O primeiro é idêntico ao segundo argumento do método *persistMedia*, apenas diferindo no fato que os números têm que ser ocultados (por exemplo, ao invés de *AgentObject2.*, deve ser *AgentObject.*). O segundo diz se o retorno do método deve ser o XML dos metadados da mídia (valor *true*) ou ela em si (valor *false*). O objeto *Vector* de retorno contém objetos *ByteArrayOutputStream* em seus elementos, que são as cadeias de caracteres dos documentos XML ou das mídias. Esse método representa os módulos de consulta e exportação da arquitetura apresentada na figura 16.

Destaca-se que realmente a intenção foi com que as interfaces fossem as mais enxutas e simples possíveis para facilitar a interação do autor da aplicação com as mesmas. A maior “complexidade” é em preencher os dados semânticos, cujo modo é o mesmo tanto pra inserção quanto para a consulta das mídias. É pré-requisito o conhecimento do padrão MPEG-7 para lidar com esses dados semânticos.

5.2.1 Exemplo de utilização das interfaces

Com a aplicação de posse da mídia e de seus metadados, conforme ilustrado na figura 18, a mesma pode invocar o módulo de anotação e persisti-los. Sendo o método *persistMedia* o representante desse módulo, é ele quem deverá ser invocado. Considerando um vídeo de nome “P1000172.MOV” (criado por uma câmera digital), pode-se criar um objeto *HashMap* denominado “hm” para efetuar a anotação semântica e fornecê-lo ao método *persistMedia*:

```
HashMap hm = new HashMap();
hm.put("Description", "Trem passando pela ponte");
hm.put("SemanticPlace1.ID", "sp1_id");
hm.put("SemanticPlace1.Place-Name", "Lago grande");
hm.put("SemanticPlace1.Place-AddressLine", "São Carlos-SP, Brasil");
hm.put("SemanticPlace1.Relation1.Target1", "obj1_id");
hm.put("SemanticPlace1.Relation1.Type", "locationOf");
hm.put("Object1.ID", "obj1_id");
hm.put("Object1.Label", "Ponte ferroviária");
hm.put("Object1.Relation1.Target1", "obj2_id");
hm.put("Object1.Relation1.Type", "pathOf");
hm.put("Object2.ID", "obj2_id");
hm.put("Object2.Label", "Trem");
hm.put("Object2.Relation1.Target1", "obj1_id");
hm.put("Object2.Relation1.Type", "above");
hm.put("Event1.ID", "ev1_id");
hm.put("Event1.Label", "Passagem do trem na ponte");
hm.put("Event1.Definition", "Passagem do trem de passageiros na ponte ferroviária de São Carlos.");
hm.put("Event1.Relation1.Target1", "vs1_id");
hm.put("Event1.Relation1.Type", "during");
hm.put("VideoSegment1.ID", "vs1_id");
hm.put("VideoSegment1.TimePoint", "00:00:01:000");
hm.put("VideoSegment1.Duration", "00:00:07:000");
```

Essa descrição semântica relata o mundo narrativo localizado no Lago grande (*sp1_id*) onde um trem (*obj2_id*) passa por (*Object2.Relation1*) uma ponte (*obj1_id*) que serve de caminho para ele (*Object1.Relation1*). Essa ponte se encontra no próprio Lago grande (*SemanticPlace1.Relation1.Type*), e o evento da passagem do trem (*ev1_id*) ocorre durante (*Event1.Relation1*) o segmento de vídeo “*vs1_id*”. Feito isso, basta invocar a interface de inserção de mídia, o método *persistMedia*:

```
MediaHandler.persistMedia("file:/dev/medias/video/P1000172.MOV", hm);
```

Todo o processo de inserção da mídia com os metadados é ilustrado na figura 21, sendo baseada na figura 18:

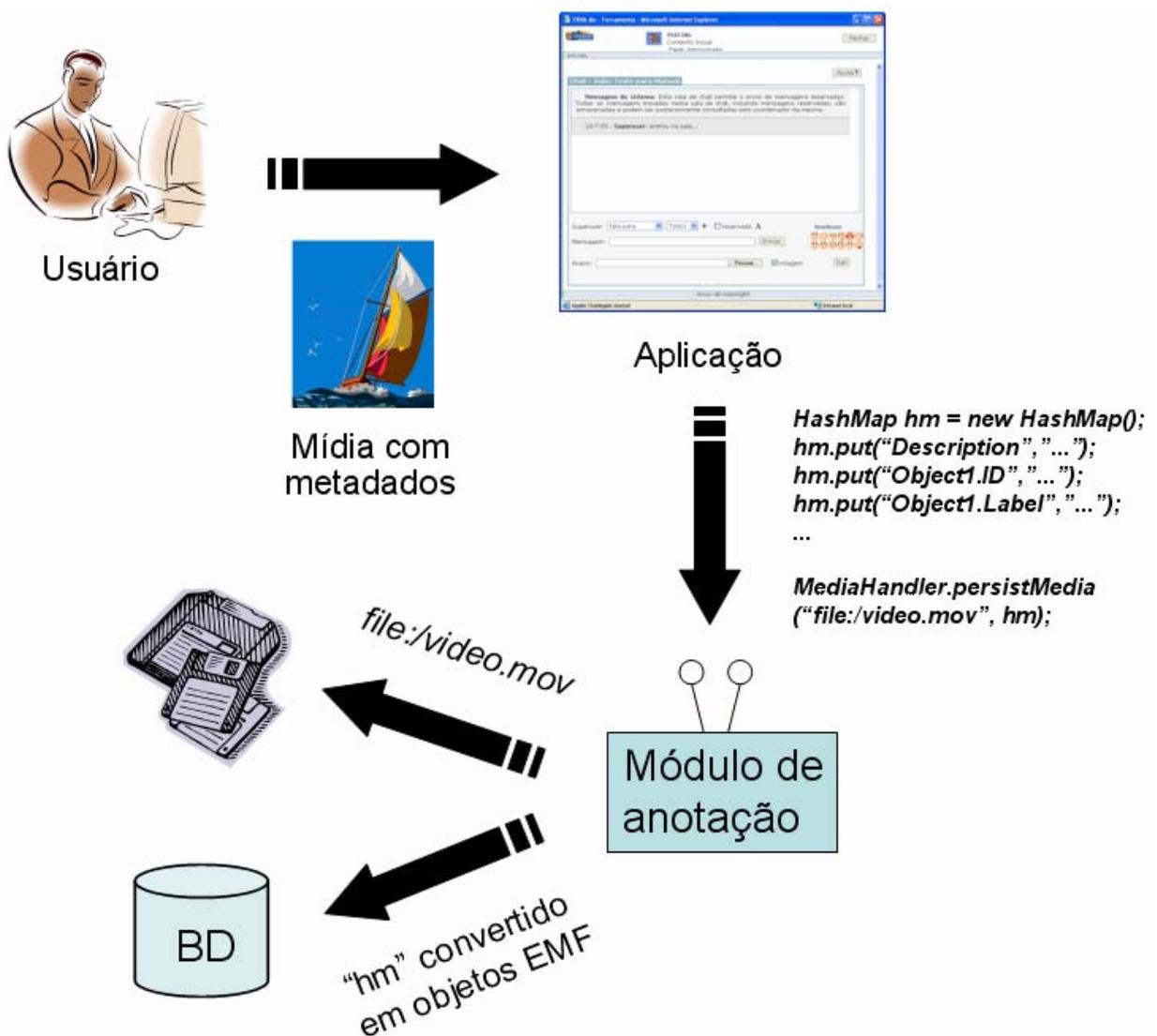


Figura 21 – Processo de inserção/anotação da mídia

Feita a invocação de *persistMedia*, os metadados já estão inseridos no banco de dados, prontos para serem consultados, por exemplo, pela construção do objeto *HashMap* denominado “*queryObjects*” e invocação da interface de consulta (método *retrieveMedia* que representa os módulos de consulta e exportação):

```
HashMap queryObjects = new HashMap();
queryObjects.put("Event.Definition", "passageiros");
Vector result = MediaHandler.retrieveMedia(queryObjects, true);
```

Um cenário típico de consulta foi apresentado na figura 19, sendo o processo dessa consulta exemplificado nessa subseção e ilustrado pela figura 22 (baseada na figura 19). Como o segundo argumento, *returnXML*, tem o valor true, o documento XML dos metadados da mídia é retornado como resultado dessa consulta (Escolha = XML da figura 21). Esse documento XML está contido no primeiro elemento desse *Vector* e é exibido nas figuras 23 e 24, que mostram respectivamente a parte semântica e a parte automaticamente obtida da mídia:

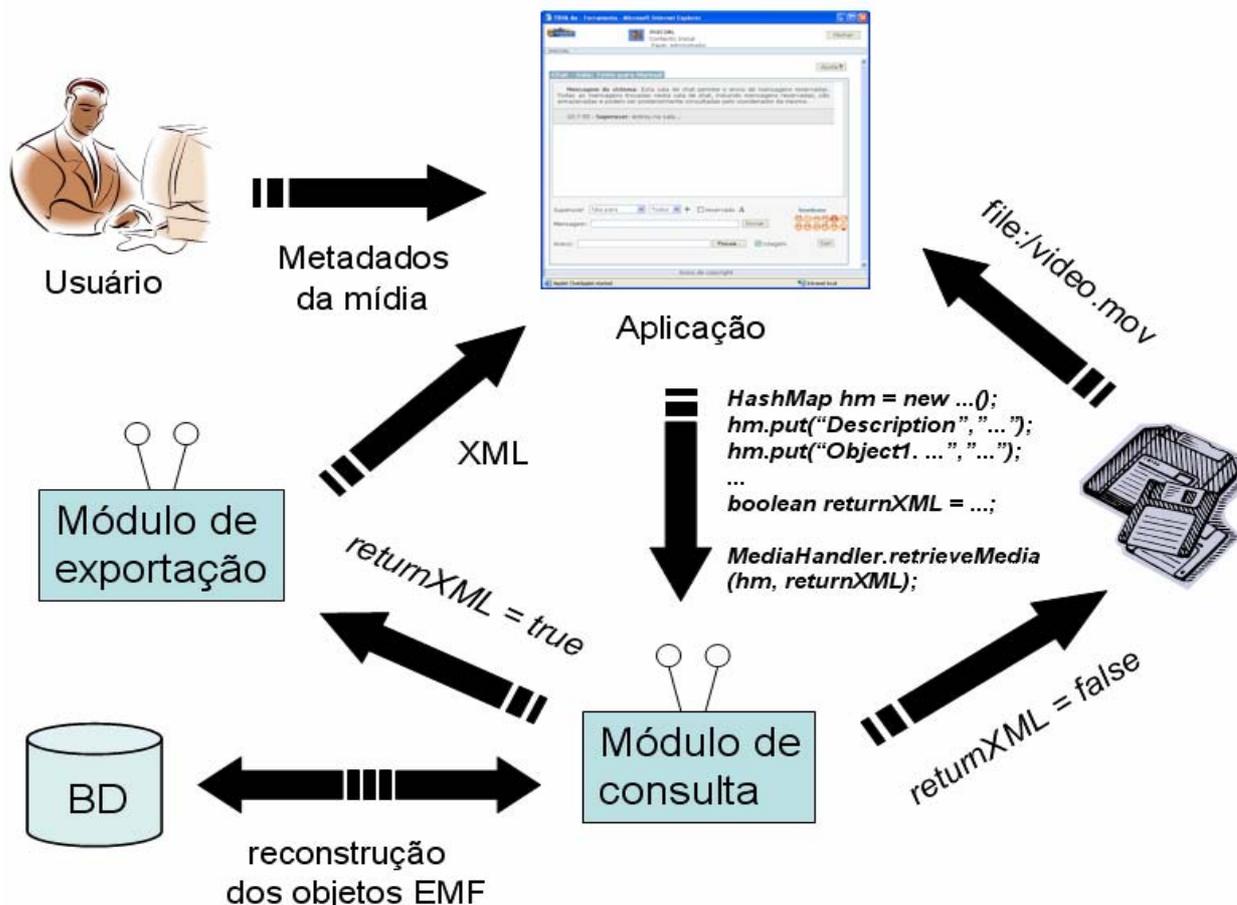


Figura 22 – Processo de consulta da mídia

```

<?xml version="1.0" encoding="ASCII"?>
<_2001:Mpeg7 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:_2001="urn:mpeg:mpeg7:schema:2001">
  <_2001:Description xsi:type="_2001:SemanticDescriptionType">
    <_2001:Semantics>
      <_2001:Definition>
        <_2001:FreeTextAnnotation>Trem passando pela ponte</_2001:FreeTextAnnotation>
      </_2001:Definition>
      <_2001:SemanticBase xsi:type="_2001:ObjectType" id="obj1_id">
        <_2001:Label>
          <_2001:Name>Ponte ferroviária</_2001:Name>
        </_2001:Label>
        <_2001:Relation target="#obj2_id" type="urn:...:pathOf"/>
      </_2001:SemanticBase>
      <_2001:SemanticBase xsi:type="_2001:ObjectType" id="obj2_id">
        <_2001:Label>
          <_2001:Name>Trem</_2001:Name>
        </_2001:Label>
        <_2001:Relation target="#obj1_id" type="urn:...:above"/>
      </_2001:SemanticBase>
      <_2001:SemanticBase xsi:type="_2001:EventType" id="ev1_id">
        <_2001:Label>
          <_2001:Name>Passagem do trem na ponte</_2001:Name>
        </_2001:Label>
        <_2001:Definition>
          <_2001:FreeTextAnnotation>Passagem do trem de passageiros na ponte ferroviária de São Carlos.
        </_2001:FreeTextAnnotation>
        </_2001:Definition>
        <_2001:Relation target="#vs1_id" type="urn:...:during"/>
      </_2001:SemanticBase>
      <_2001:SemanticBase xsi:type="_2001:SemanticPlaceType" id="sp1_id">
        <_2001:Relation target="#obj1_id" type="urn:...:locationOf"/>
        <_2001:Place>
          <_2001:Name>Lago grande</_2001:Name>
          <_2001:PostalAddress>
            <_2001:AddressLine>São Carlos-SP, Brasil</_2001:AddressLine>
          </_2001:PostalAddress>
        </_2001:Place>
      </_2001:SemanticBase>
    </_2001:Semantics>
  </_2001:Description>

```

Figura 23 – Parte semântica do XML exemplo

```

<_2001:Description xsi:type="_2001:ContentEntityType">
  <_2001:MultimediaContent xsi:type="_2001:VideoType">
    <_2001:Video id="P1000172.MOV.3.905993285757668">
      <_2001:MediaInformation>
        <_2001:Header xsi:type="_2001:DescriptionMetadataType">
          <_2001:LastUpdate>2007-4-25T19:14:23:375F1000-03:00</_2001:LastUpdate>
        </_2001:Header>
        <_2001:MediaProfile>
          <_2001:MediaFormat>
            <_2001:Content href="MPEG7ContentCS">
              <_2001:Name>audiovisual</_2001:Name>
            </_2001:Content>
            <_2001:FileSize>12074606</_2001:FileSize>
            <_2001:BitRate>11364335</_2001:BitRate>
            <_2001:VisualCoding>
              <_2001:Format>
                <_2001:Name>JPEG</_2001:Name>
              </_2001:Format>
              <_2001:Frame aspectRatio="1.3333" height="480" rate="30.0" width="640"/>
            </_2001:VisualCoding>
            <_2001:AudioCoding>
              <_2001:Format>
                <_2001:Name>LINEAR</_2001:Name>
              </_2001:Format>
              <_2001:AudioChannels>1</_2001:AudioChannels>
              <_2001:Sample bitsPer="8" rate="8000.0"/>
            </_2001:AudioCoding>
          </_2001:MediaFormat>
        </_2001:MediaProfile>
      </_2001:MediaInformation>
      <_2001:MediaLocator>
        <_2001:MediaUri>file://dev/medias/video/P1000172.MOV.3.905993285757668</_2001:MediaUri>
      </_2001:MediaLocator>
      <_2001:MediaTime>
        <_2001:MediaTimePoint>T00:00:00:000F1000</_2001:MediaTimePoint>
        <_2001:MediaDuration>PT8S500N1000F</_2001:MediaDuration>
      </_2001:MediaTime>
      <_2001:TemporalDecomposition>
        <_2001:VideoSegment id="vs1_id">
          <_2001:MediaTime>
            <_2001:MediaTimePoint>T00:00:01:000F1000</_2001:MediaTimePoint>
            <_2001:MediaDuration>PT00H00M07S000N1000F</_2001:MediaDuration>
          </_2001:MediaTime>
        </_2001:VideoSegment>
      </_2001:TemporalDecomposition>
    </_2001:Video>
  </_2001:MultimediaContent>
</_2001:Description>
</_2001:Mpeg7>

```

Figura 24 – Parte automaticamente obtida da mídia

5.3 CONSIDERAÇÕES FINAIS

Nesse trabalho foram definidos componentes genéricos com interfaces para a manipulação de mídias, sejam elas áudios, vídeos ou imagens. Esses componentes podem ser reutilizados por quaisquer aplicações que tenham requisitos de manipulação de dados multimídia. Por serem baseados em um padrão de descrição de dados multimídia, o MPEG-7, também permitem intercâmbio de dados com outros sistemas e entendimento unificado sobre os metadados das mídias, ou seja, quem conhece o padrão, conhece a representação desses metadados.

As contribuições obtidas nesse trabalho são as seguintes:

- Com a construção dos componentes, interfaces simples para o armazenamento e consulta de mídias foram obtidas. O uso dessas interfaces foi exemplificado na subseção 5.2.1;
- Construção “automática” do documento XML dos metadados da mídia baseado no padrão MPEG-7;
- Integração/uniformização das necessidades de manipulação de dados multimídia das ferramentas que forem utilizar essa camada, ou seja, interfaces padronizadas para realizar as operações (inserção e consulta) e repositório logicamente integrado;
- Um modelo, ou método para anotação e consulta semântica via construção de *strings* baseadas nos metadados do padrão MPEG-7;
- Intercâmbio de dados com outros sistemas/aplicações que façam uso de material multimídia, devido a utilização do padrão MPEG-7.

Esse trabalho é um primeiro passo de um processo que pode ter continuidade, abrindo a possibilidade de realização de diversos trabalhos futuros, sendo eles descritos a seguir:

- Implementação de outros Ds e DSs do padrão MPEG-7;
 - Desses Ds e DSs, podem-se empregar processamentos e algoritmos para preencher valores de Ds (características de baixo nível) e técnicas para o preenchimento semi-automático de valores semânticos e outros DSs.
- Construção de novas interfaces para acomodar novos requisitos de manipulação de mídias (por exemplo, versões);
- Utilização de mineração de texto sobre as cadeias de caracteres semânticas;

- Extensão de Ds ou DSs do padrão MPEG-7, porém sob a consequência de saírem do escopo dos metadados padronizados;
- Otimização do desempenho das consultas e extensão de suas capacidades (por exemplo, adição de outros operadores lógicos além do “||”);
- Proposta de um meio automático para a formação das *strings* de consulta a partir de qualquer XML *Schema*;

6 BIBLIOGRAFIA

ALLAMANCHE, E., *et al.* **Content-based Identification of Audio Material Using MPEG-7 Low Level Description**. Anais do International Symposium on Music Information Retrieval (ISMIR), 2001, p. 197 - 204.

APPEL, A. P., ALMEIDA, O. C. P. D. e SCHIEL, D. **Collaborative writing in distance education**. II Workshop Tidia FAPESP, 2005.

AVARO, O. e SALEMBIER, P. **MPEG-7 Systems: overview**. IEEE Transactions on Circuits and Systems for Video Technology, v.11, n. 6, p. 760 - 764 2001.

BACHLECHNER, A., *et al.* **The SMOOTH video DB - demonstration of an integrated generic indexing approach**. Anais do 8th ACM international conference on Multimedia, 2000, Marina del Rey, California, United States. p. 495 - 496.

BARAL, C., GONZALEZ, G. e NANDIGAM, A. **SQL+D: extended display capabilities for multimedia database queries**. Anais do 6th ACM international conference on Multimedia, 1998, Bristol, United Kingdom. p. 109 - 114.

BAUER, C. e KING, G. **Hibernate in Action**. Greenwich, CT, USA: Manning Publications Co., 2004.

BEDER, D., *et al.* **The TIDIA-Ae Portfolio Tool: a case study of its development following a component-based layered architecture**. II Workshop Tidia FAPESP, 2005.

BOBER, M. **MPEG-7 visual shape descriptors**. IEEE Transactions on Circuits and Systems for Video Technology, v.11, n. 6, p. 716 - 719, 2001.

BRESSAN, G., FRANCO, E. e RUGGIERO, W. **Tool Manager, Tool Interface and Metadata for Tidia-ae Learning Management System**. II Workshop Tidia FAPESP, 2005.

CASEY, M. **MPEG-7 sound-recognition tools**. IEEE Transactions on Circuits and Systems for Video Technology, v.11, n. 6, p. 737 - 747, 2001.

CHANG, S.-F., *et al.* **VideoQ: an automated content based video search system using visual cues**. Anais do 5th ACM international conference on Multimedia, 1997, Seattle, Washington, United States. p. 313 - 324.

CHANG, S.-F., SIKORA, T. e PURI, A. **Overview of the MPEG-7 standard**. IEEE Transactions on Circuits and Systems for Video Technology, v.11, n. 6, p. 688 - 695, 2001.

CHARLESWORTH, J. P. A. e GARNER, P. N. **SpokenContent Representation in MPEG-7**. IEEE Transactions on Circuits and Systems for Video Technology, v.11, n. 6, p. 730 - 736, 2001.

- CRYSANDT, H. e WELLHAUSEN, J. **Music classification with MPEG-7**. Anais do SPIE International Conference on Electronic Imaging - Storage and Retrieval for Media Databases, 2003, Santa Clara. p. 397 - 404.
- DAS, M., RISEMAN, E. M. e DRAPER, B. A. **FOCUS: searching for multi-colored objects in a diverse image database**. Anais do IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1997, San Juan. p. 756 - 761.
- ECLIPSE, F. **Eclipse Modeling Framework Project (EMF)**. 2007, <http://www.eclipse.org/emf> (Acessado em Abr/2007).
- EISENBERG, A. e MELTON, J. **SQL: 1999, formerly known as SQL3**. ACM SIGMOD Record, v.28, n. 1, p. 131 - 138, 1999.
- ELEUTÉRIO, P. M. S., *et al.* **A Communication for Tidia-Ae: the Chat Tool**. II Workshop Tidia FAPESP, 2005.
- ELVER, P. **Teneo Persistency**. 2007, <http://www.elver.org> (Acessado em Abr/2007).
- FAPESP **Description of the TIDIA-Ae Project - Definitions and objectives**. 2005, <http://tidia-ae.incubadora.fapesp.br/portal/desc/> (Acessado em Abr/2007).
- FLICKNER, M., *et al.* **Query by image and video content: the QBIC system**. IEEE Computer, v.28, n. 9, p. 23 - 32, 1995.
- GAVIOLI, A., BIAJIZ, M. e MOREIRA, J. **MIFLIR: A Metric Indexing and Fuzzy Logic-based Image Retrieval System**. Anais do IEEE International Workshop on Managing Data For Emerging Multimedia Applications (EMMA), 2005, Tokio, Japan. p. 1178.
- GHIAS, A., *et al.* **Query by humming: musical information retrieval in an audio database**. Anais do 3rd ACM international conference on Multimedia, 1995, San Francisco, California, United States. p. 231 - 236.
- GIBBS, S., BREITENEDER, C. e TSICHRITZIS, D. **Data modeling of time-based media**. ACM SIGMOD international conference on Management of data, ACM Press, 1994.
- GUDIVADA, V. N., RAGHAVAN, V. V. e VANAPIPAT, K. **A unified approach to data modeling and retrieval for a class of image database applications**. Em: Multimedia database systems: issues and research directions. Springer-Verlag, 1996, p. 37 - 78
- HUANG, T., MEHROTRA, S. e RAMCHANDRAN, K. **Multimedia Analysis and Retrieval System (MARS) Project**. 33rd Annual Clinic on Library Application of Data Processing - Digital Image Access and Retrieval, 1996.
- HUNTER, J. **An overview of the MPEG-7 description definition language (DDL)**. IEEE Transactions on Circuits and Systems for Video Technology, v.11, n. 6, p. 765 - 772, 2001.

- JEANNIN, S. e DIVAKARAN, A. **MPEG-7 visual motion descriptors**. IEEE Transactions on Circuits and Systems for Video Technology, v.11, n. 6, p. 720 - 724, 2001.
- JIANG, H. e ELMAGARMID, A. K. **Spatial and temporal content-based access to hypervideo databases**. The VLDB Journal — The International Journal on Very Large Data Bases, v.7, n. 4, p. 226 - 238, 1998.
- JIANG, H., MONTESI, D. e ELMAGARMID, A. K. **Integrated video and text for content-based access to video databases**. Multimedia Tools and Applications, v.9, n. 3, p. 227-249, 1999.
- KOSCH, H., *et al.* **SMOOTH - A Distributed Multimedia Database System**. Anais do 27th International Conference on Very Large Data Bases, 2001a, p. 713 - 714.
- KOSCH, H., *et al.* **SMOOTH - A Distributed Multimedia Database System - Versão estendida**. Anais do 27th International Conference on Very Large Data Bases, 2001b, p. 713 - 714.
- KUDO, T. N., *et al.* **Using UML Components for the specification of the Whiteboard tool**. II Workshop Tidia FAPESP, 2005.
- LOBATO, D. C., *et al.* **A Multimedia Instant Messenger for an e-Learning Environment**. II Workshop Tidia FAPESP, 2005.
- MANJUNATH, B.S., SALEMBIER, P. e SIKORA, T., **Introduction to MPEG-7: Multimedia Content Description Interface**. Wiley, 2002.
- MANJUNATH, B. S., *et al.* **Color and texture descriptors**. IEEE Transactions on Circuits and Systems for Video Technology, v.11, n. 6, p. 703 - 715, 2001.
- MARTÍNEZ, J. M. **MPEG-7 Overview (version 10)**. ISO/IEC JTC1/SC29/WG11N6828, 2004, <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm> (Acessado em Abr/2007).
- MEHROTRA, S., *et al.* **Multimedia Analysis and Retrieval System**. 3rd Int. Workshop on Information Retrieval Systems, 1997.
- MELTON, J. e EISENBERG, A. **SQL multimedia and application packages (SQL/MM)**. ACM SIGMOD Record, v.30, n. 4, p. 97 - 102, 2001.
- PRAZERES, C. V. S., *et al.* **Integrating “Tools for Accessing Resources Remotely” into an e-Learning Environment**. II Workshop Tidia FAPESP, 2005.
- QUACKENBUSH, S. e ADAM LINDSAY **Overview of MPEG-7 audio**. IEEE Transactions on Circuits and Systems for Video Technology, v.11, n. 6, p. 725 - 729, 2001.
- RUI, Y., HUANG, T. S. e CHANG, S.-F. **Image Retrieval: Past, Present, And Future**. International Symposium on Multimedia Information Processing, 1997.

SALEMBIER, P. e SMITH, J. R. **MPEG-7 multimedia description schemes**. IEEE Transactions on Circuits and Systems for Video Technology, v.11, n. 6, p. 748 - 759, 2001.

SANTOS, F. G. D., VIEIRA, M. T. P. e FIGUEIREDO, J. M. D. **MAE: uma ferramenta de autoria multimídia para a WEB**. Anais do V Symposium on Multimedia and Hypermedia Systems, 2002, Fortaleza, Ceará, Brazil. p. 289 - 297.

SIKORA, T. **The MPEG-7 visual standard for content description-an overview**. IEEE Transactions on Circuits and Systems for Video Technology, v.11, n. 6, p. 696 - 702, 2001.

SMITH, J. R. e CHANG, S.-F. **VisualSEEK: a fully automated content-based image query system**. Anais do 4th ACM international conference on Multimedia, 1997, Boston, Massachusetts, United States. p. 87 - 98.

SUBRAMANYA, S. R. **Multimedia Databases**. IEEE Potentials, v.18, n. 5, p. 16-18, 2000.

TUSCH, R., KOSCH, H. e BÖSZÖRMÉNYI, L. **VIDEX: an integrated generic video indexing approach**. Anais do 8th ACM international conference on Multimedia, 2000, Marina del Rey, California, United States. p. 448 - 451.

WEISS, R., DUDA, A. e GIFFORD, D. K. **Composition and search with a video algebra**. IEEE Multimedia, v.2, n. 1, p. 12-25, 1995.

ZAINA, L. A. M., *et al.* **Managing users of educational system based on role controls**. II Workshop Tidia FAPESP, 2005.

ZHONG, D. e CHANG, S.-F. **Video object model and segmentation for content-based video indexing**. Anais do IEEE International Symposium on Circuits and Systems, 1997, p. 1492-1495.

ZHOU, T. T., GEDEON, T. e JIN, J. S. **Automatic generating detail-on-demand hypervideo using MPEG-7 and SMIL**. Anais do 13th annual ACM international conference on Multimedia, 2005, Hilton, Singapore. p. 379 - 382.